

REGULAR ARTICLE

OPEN  ACCESS

# Prediction of engine NOx for virtual sensor using deep neural network and genetic algorithm

Jongmyung Kim<sup>1</sup>, Jihwan Park<sup>2</sup>, Seunghyup Shim<sup>2</sup> , Yongjoo Lee<sup>2</sup>, Kyoungdoug Min<sup>2</sup>, Sangyul Lee<sup>3,\*</sup>, and Minjae Kim<sup>1,\*</sup> 

<sup>1</sup> Department of Mechanical Engineering, Myongji University, 17058 Yong-in, Republic of Korea

<sup>2</sup> Department of Mechanical and Aerospace Engineering, Seoul National University, 08826 Seoul, Republic of Korea

<sup>3</sup> Division of Mechanical and Electronics Engineering, Hansung University, 02876 Seoul, Republic of Korea

Received: 21 June 2020 / Accepted: 27 September 2021

**Abstract.** The Nitrogen Oxides (NOx) from engines aggravate natural environment and human health. Institutional regulations have attempted to protect the human body from them, while car manufacturers have tried to make NOx free vehicles. The formation of NOx emissions is highly dependent on the engine operating conditions and being able to predict NOx emissions would significantly help in enabling their reduction. This study investigates advanced method of predicting vehicle NOx emissions in pursuit of the sensorless engine. Sensors inside the engine are required to measure the operating condition. However, they can be removed or reduced if the sensing object such as the engine NOx emissions can be accurately predicted with a virtual model. This would result in cost reductions and overcome the sensor durability problem. To achieve such a goal, researchers have studied numerical analysis for the relationship between emissions and engine operating conditions. Also, a Deep Neural Network (DNN) is applied recently as a solution. However, the prediction accuracies were often not satisfactory where hyperparameter optimization was either overlooked or conducted manually. Therefore, this study proposes a virtual NOx sensor model based on the hyperparameter optimization. A Genetic Algorithm (GA) was adopted to establish a global optimum with DNN. Epoch size and learning rate are employed as the design variables, and *R*-squared based user defined function is adopted as the object function of GA. As a result, a more accurate and reliable virtual NOx sensor with the possibility of a sensorless engine could be developed and verified.

## 1 Introduction

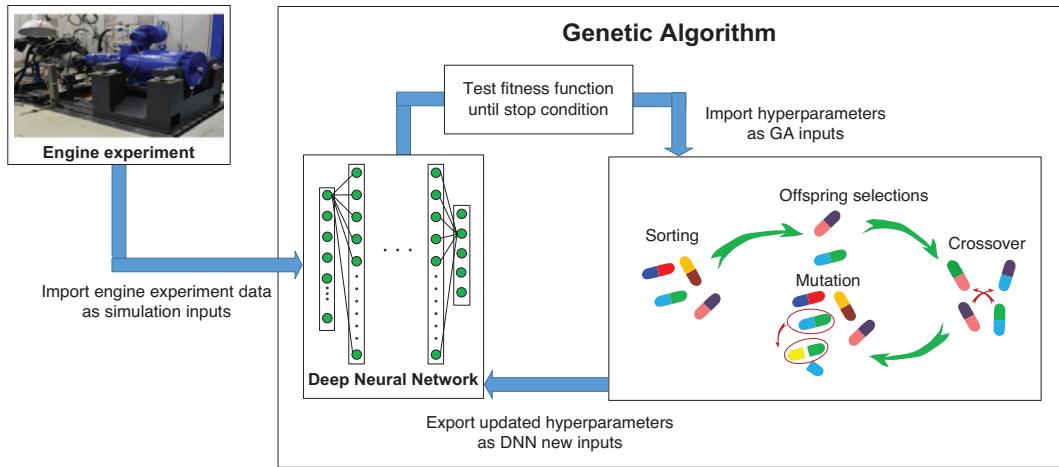
Air pollution caused by greenhouse gases and automobile emissions has created a worldwide need for strict regulation [1, 2]. An Internal Combustion Engine (ICE) equipped vehicle has difficulty in complying with the regulations and even the eco-friendly car such as a Hybrid Electric Vehicle (HEV) using ICE also couldn't escape from the problems. Among the emission gases, NOx causes environmental pollution such as acid rain, ground ozone, birth of the fine particles, smog, as well as critical diseases in humans [3]. Recently, the *California Air Resources Board (CARB)* regulations assert the amount of "Non-Methane Organic Gases (NMOG) + NOx" on all vehicles produced in 2019 must be below 0.090 g/mi. Moreover, the allowance amount would be "NMOG + NOx" below 0.050 g/mi after 2025. The European Union also corresponds to this trend

and car manufacturers are doing their best to minimize NOx emissions from the internal combustion engine [4–6].

Except for improvement by materials such as insulation coating [7], general NOx emission reduction techniques include engine-out and tailpipe NOx reduction techniques. The former techniques are, for example, injection strategy adjustment (pressure and timing) or Exhaust Gas Recirculation (EGR) applied to the combustion chamber of diesel engines [8, 9]. The latter techniques are such as Selective Catalytic Reduction (SCR) or Lean NOx Trap (LNT), and minimize airborne pollutants using filters, catalysts and other post-processing methodologies [10–12]. To achieve the desired performance of the NOx reduction techniques mentioned above, the accurate sensing of NOx emissions is highly recommended.

Engine operating conditions are closely related to NOx emissions and these conditions can be identified by checking the status (state variables) of the engine [3]. This indicates if the engine operating conditions are adjusted through speed, torque, fuel injection, etc., the birth amount of NOx emissions changes; the optimization of NOx emission

\* Corresponding author: [minjk@mju.ac.kr](mailto:minjk@mju.ac.kr);  
[sangyul.lee@hansung.ac.kr](mailto:sangyul.lee@hansung.ac.kr)



**Fig. 1.** GA based hyperparameter optimization of DNN.

formation by engine control is possible and highly requested for a healthy driving [13, 14]. In the previous studies, understanding the formation process was required to optimize the NOx emission, and researches based on the thermodynamics or Computational Fluid Dynamics (CFD) have been conducted such as the NOx emission prediction using chemical kinetics, skeletal mechanisms, 2D/3D models, and two-zone thermodynamic model [15–19]. They made it possible to aware of what factors influence the NOx formation and reduction. However, understanding the relationship between the engine operating condition and emissions is not straightforward and thus the relationship has been indirectly analyzed using statistical model, control-oriented model, or Artificial Neural Networks (ANN) [20–25]. However, the relationships are still complex and input parameters used in the equations were too ideal. Currently, Deep Neural Networks (DNN) are considered in various engine analyses and adjustments. DNN is an advanced ANN with multiple layers for increased accuracy. This makes it a very suitable tool for engine research since unrealistic parameters for real applications, understanding the complex physical meaning, and developing highly accurate equations are not necessarily required [26–28]. Besides, the various engine operating conditions swept during the engine experiment can be used as the inputs of the DNN model, unlike the map-based model. Accordingly, prediction accuracy increase from the various engine operating conditions can be guaranteed after the appropriate DNN training. In DNN, data quantity for training and input/output definitions are well known for their impacts for problem-solving, and there are some studies to predict NOx emissions using DNN. However, despite the importance of the hyperparameter definitions, these values for DNN designs have often been overlooked or found manually (intuition or trial & error) in previous researches [29–33].

To address such problems, this study develops DNN based virtual NOx sensor model which adopts a Genetic Algorithm (GA) to determine the optimal hyperparameters: epoch size and learning rate. In the virtual sensor model, a user-defined object function using  $R$ -squared is applied for

**Table 1.** Engine specifications.

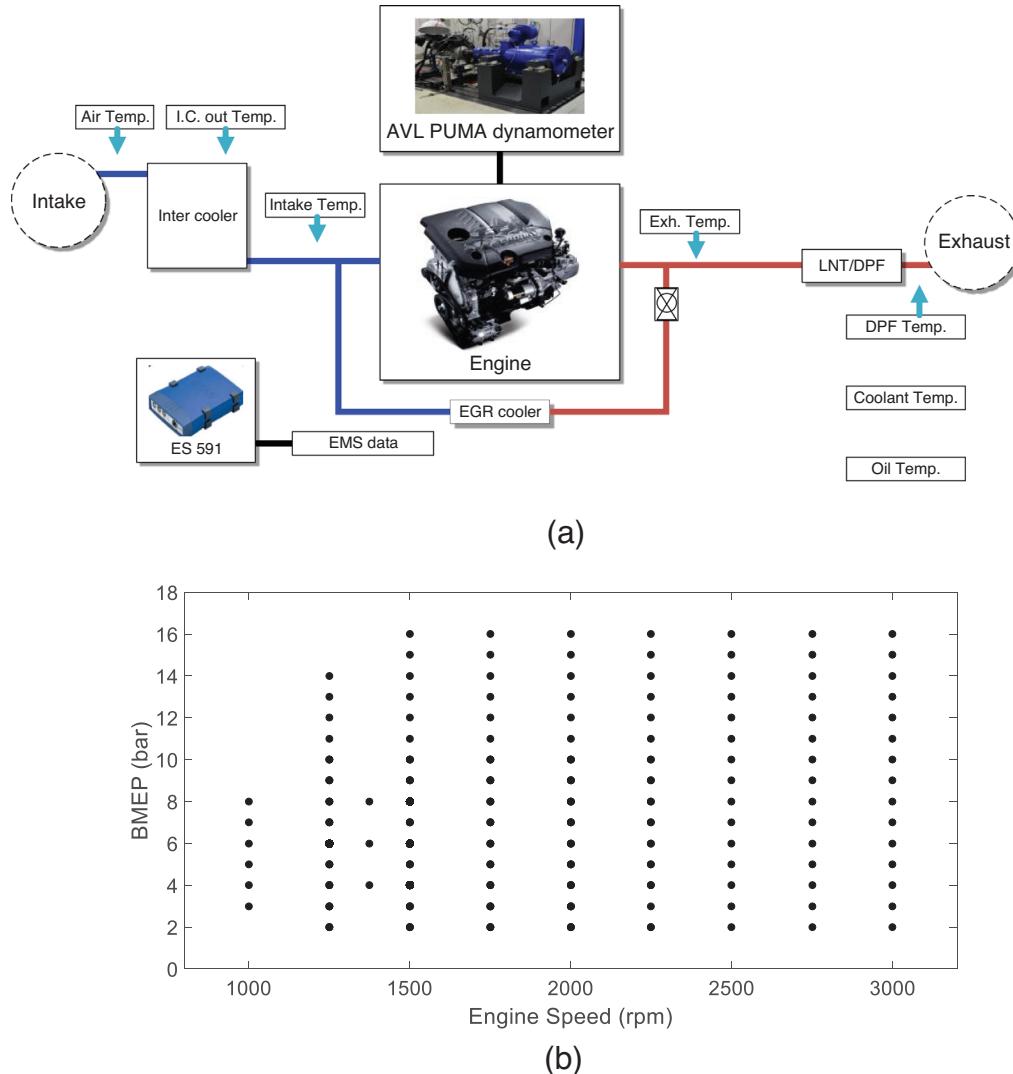
Engine type	1.6-L 4-cylinder CI engine
Displaced volume	1592 (cc)
Bore × Stroke	77 (mm) × 85.8 (mm)
Connecting rod	142 (mm)
Compression ratio	15.9:1
Turbocharger	Single-stage VGT
Fuel injection system	CRDI (Solenoid)

the global optimization and the necessary inputs for the virtual sensor model are presented. With the powerful tool of DNN combined with hyperparameter optimization, prediction performance increases as much as the real sensors in the engine such that the removal of physical sensors would be feasible soon. This leads to lower costs, prevents performance decreases over time, and improves sensor durability. Also, the effectiveness of this study could be proved with the accuracy comparison between the virtual and real sensors. Figure 1 shows the overall process of this study to develop a virtual sensor model of engine NOx emissions using DNN and GA, where the fundamental data for DNN training was acquired by real engine experiments.

The rest of this study is as follows. In Section 2, the experimental setup and environment for the data acquisition is introduced. Section 3 defines the virtual sensor model and calculation methods based on DNN. Section 4 explains the DNN hyperparameter optimization using GA, and Section 5 presents the simulation results and discussion for the novel virtual sensor. Finally, Section 6 concludes the study.

## 2 Experimental setup and environment

Engine experiment was conducted to obtain the engine state variables. The target engine was a 1.6-L four-cylinder engine and the displaced volume was 1592 cc. The bore and



**Fig. 2.** Engine test configuration and experiment area. (a) Experimental setup. (b) Engine operating points.

stroke of the engine were 77 mm and 85.8 mm, respectively, and the length of the connecting rod was 142 mm. The engine was equipped with a single-stage Variable Geometry Turbocharger (VGT) which contributes to combustion inside the cylinder to keep the flow constant. The specific engine specifications are listed in [Table 1](#).

The configuration of the engine test system and engine operating points are shown in [Figure 2](#) where the engine was controlled by AVL PUMA dynamometer [34]. The sample experimental min/max limitations for the engine dataset is listed in [Table 2](#).

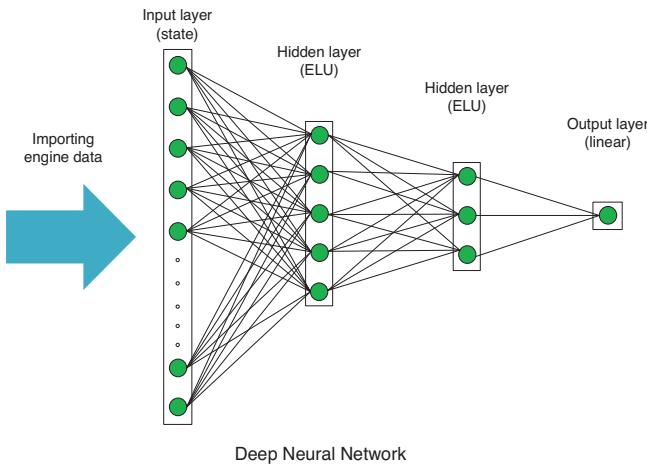
### 3 Virtual sensor model for NO<sub>x</sub> prediction

To develop a virtual NO<sub>x</sub> sensor model, supervised DNN was adopted. The data from the engine experiment were applied to the DNN training input/output and Python with TensorFlow was used for developing a NO<sub>x</sub> prediction model. To reduce the calculation burden of the virtual

**Table 2.** Experimental environment.

Input variable	Unit	Min	Max
Fuel injection/stroke	(mg/stk)	5.5	40
Acceleration pedal	(%)	20	35
Ambient temperature	(°C)	20	35
Humidity	(%)	20	45
Exhaust pressure	(kPa)	97	202
Inlet pressure	(kPa)	103	211
Exhaust temperature	(°C)	200	650
Inlet temperature	(°C)	35	42
EGR rate	(%)	20	44

NO<sub>x</sub> sensor in the embedded system, simple 2 hidden layers were adopted. Also, node number reduction (5 to 3) for preventing complexity and sudden information loss while passing through the hidden layers was used as in [Figure 3](#).



**Fig. 3.** DNN model structure.

### 3.1 DNN model

The output of the sensor model was defined as the engine NOx emission. The inputs were the engine operating conditions measured by the experiment in [Section 2](#). These engine state variables were grouped into the dataset comprising 696 experimental cases where 60% are used for training, 20% are used for validation, and the other 20% are used for testing. Each case provided engine rpm, fuel injection quantity (main, pilot, total), EGR rate, boost pressure, fuel injection timing (main, pilot), injection pressure, acceleration pedal, ambient temperature, humidity, ambient pressure, exhaust gas pressure, oil pressure, DPF temperature, coolant temperature, intake manifold temperature.

Dealing with many kinds of state variables can cause an overfitting problem in DNN. To prevent such the problem, drop out, batch normal, validation dataset and L2 regularization were used. Among them, batch normal helps to solve the DNN problem with both accuracy and speed, in spite of the large hyperparameters [35]. L2 regulation simplifies the model based on penalizing the loss function [36] and is expressed as follows;

$$L = \sum_{i=1}^n (y_i - h_\theta(x_i))^2 + \lambda \sum_{i=1}^n \theta_i^2, \quad (1)$$

where  $y_i$ ,  $h_\theta$ ,  $x_i$ ,  $\theta$ , and  $\lambda$  are the true value, activation function, input value, weight, and regularization rate of 0.01 [36].

### 3.2 Hyperparameter in DNN

Hyperparameters in [Table 3](#) have a great influence on DNN performance. They are different from input and output variables and, without a careful definition of them, the output results may diverge or be caught in a poor local optimum [34]. The representative DNN hyperparameters are learning rate, epoch number, node number, and hidden layer number. These hyperparameters in the model are tested and the greatly influential learning rates and epoch

**Table 3.** Hyperparameters in DNN.

Hyperparameters	Value
Learning rate	0.0001 – 0.01 (optimization variable)
Epoch size	1000 – 100 000 (optimization variable)
Mini batch size	10
Input node	21
Output node	1
Hidden node	5, 3
Activation function	ELU
Optimizer	Adam
Batch normalization	Included

size were selected for the optimization design variables in [Section 4](#). The learning rate is a hyperparameter that determines how far the next point will move, in contrast, epoch size refers to how much learning the entire dataset will accomplish. The inappropriate learning rate of the DNN model causes overshooting or local minima problems, whereas wrong selection of epoch size causes underfitting or overfitting problems. Therefore, the optimized selections of both design variables are very important for prediction accuracy. Also, each optimization needs up to 10 DNN implementation results for each GA generation, so it takes too much time if too large epoch size is set up.

### 3.3 Activation function

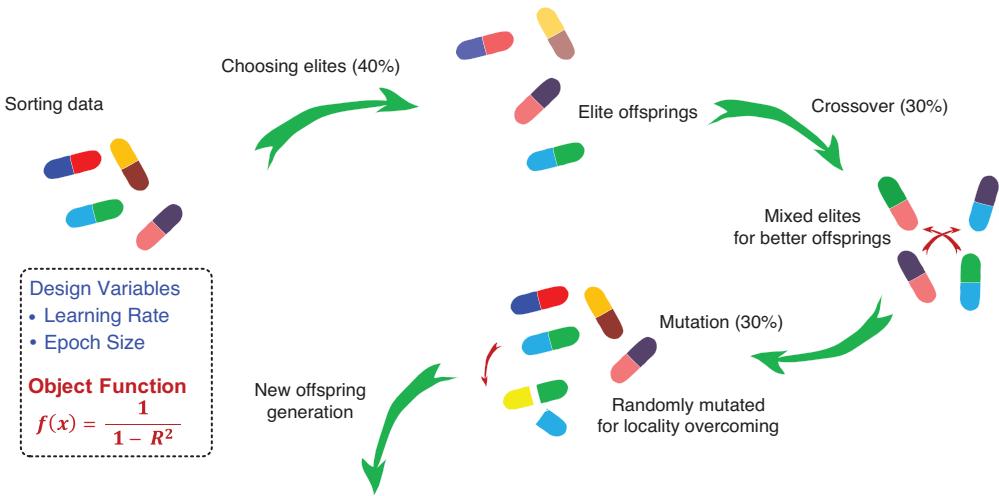
An Activation function converts the input signals of individual neurons into output signals. It adds non-linearities to neural networks by deciding whether weighted sum of neurons should be activated or not. In this study, a nonlinear function of Exponential Linear Unit (ELU) is used as the activation function. The ELU is an improved version of Rectified Linear Unit (ReLU). In addition to possessing all the advantages of ReLU, the ELU does not cause a dying ReLU problem. The basic equation of ELU is as follows:

$$h(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha(e^x - 1) & \text{if } x \leq 0 \end{cases}, \quad (2)$$

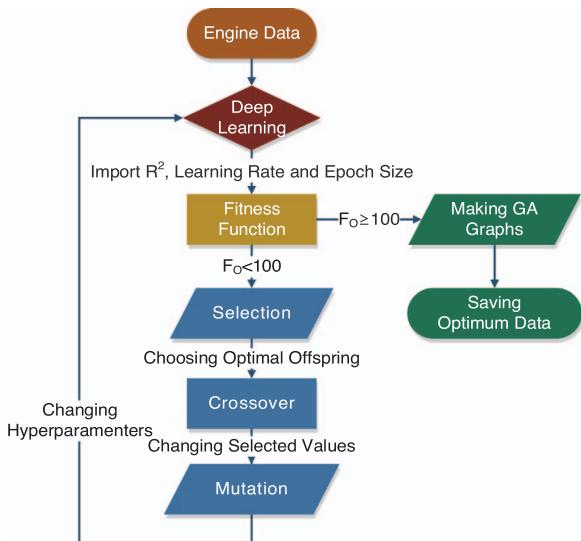
where  $\alpha$  is an ELU parameter. ELU has the advantage of learning considerably faster than the other sigmoid and  $\tanh$  functions. Moreover, it uses the exponential function. Therefore, negative input values can also be handled without a problem because the derivative value is non-zero [37].

### 3.4 Adam optimizer

Adaptive moment estimation (Adam) is a one-dimensional gradient-based optimizer, which demonstrated good performance in DNN studies [38–40]. This optimizer combines



**Fig. 4.** Genetic algorithm application.



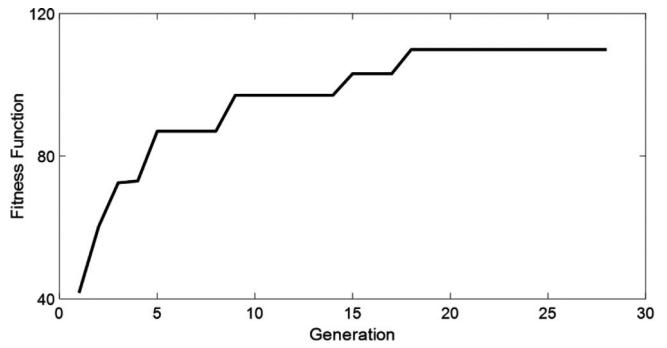
**Fig. 5.** Algorithm sequence using DNN and GA.

the advantages of the AdaGrad and RMSProp – it not only stores the exponential mean of the slope but also the exponential mean of the square value of the slope. The main advantage of the Adam optimizer is that the step size is not affected by gradient rescaling. Therefore, it is possible to stably descend for optimization, and the step size can be adapted by referring to the past gradient size. Equations (3) and (4) are the basic equations of the Adam optimizer [41]:

$$m_{t+1} = \beta_1 m_t + (1 - \beta_1) \cdot \nabla J(x_t), \quad (3)$$

$$v_{t+1} = \beta_2 v_t + (1 - \beta_2) \cdot (\nabla J(x_t))^2, \quad (4)$$

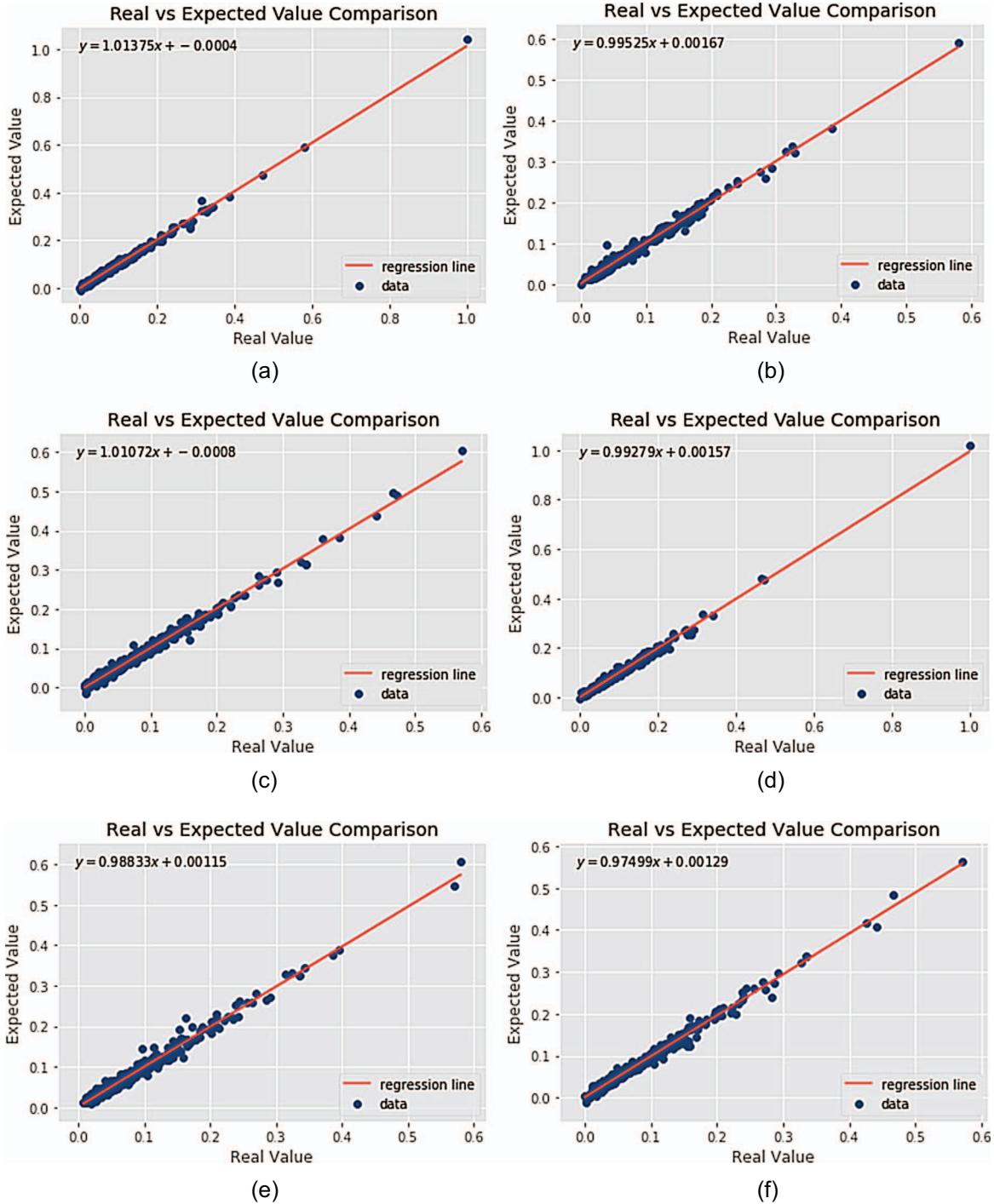
where  $m_t$ ,  $v_t$ ,  $\beta_1$ ,  $\beta_2$ , and  $J$  are the momentum, adaptive term, momentum decay rate of 0.9, adaptive term decay rate of 0.999, and cost function to be minimized, respectively.



**Fig. 6.** Fitness function of GA.

#### 4 Hyperparameter optimization using genetic algorithm

Hyperparameter optimizations were conducted manually in the previous researches. However, the quality of DNN supervised learning highly depends on the definition of hyperparameters and this study proposes using GA for the high performance. As shown in Figure 4, GA emulates the evolution of living things to obtain an optimal solution. Basically, the goal is to cross the data like genes, create and analyze mutations, and continue to produce offspring until they reach their optimum [42]. In this study, the optimization was started with the random selection of learning rates and epoch sizes where 10 offsprings (population) were made in the first generation. These 10 populations continue over the generations. The best 40% different elites for the next generation were chosen preferentially. Then, the elite offsprings crossed over each other and these were followed by random mutations where crossover and mutation rates were set to be 30% and 30% among 10 offsprings. In short, 4:3:3 of elites, crossover, mutation population ratio to make the weighting of elites, crossover, and mutation similar was used and continued during the GA simulation.

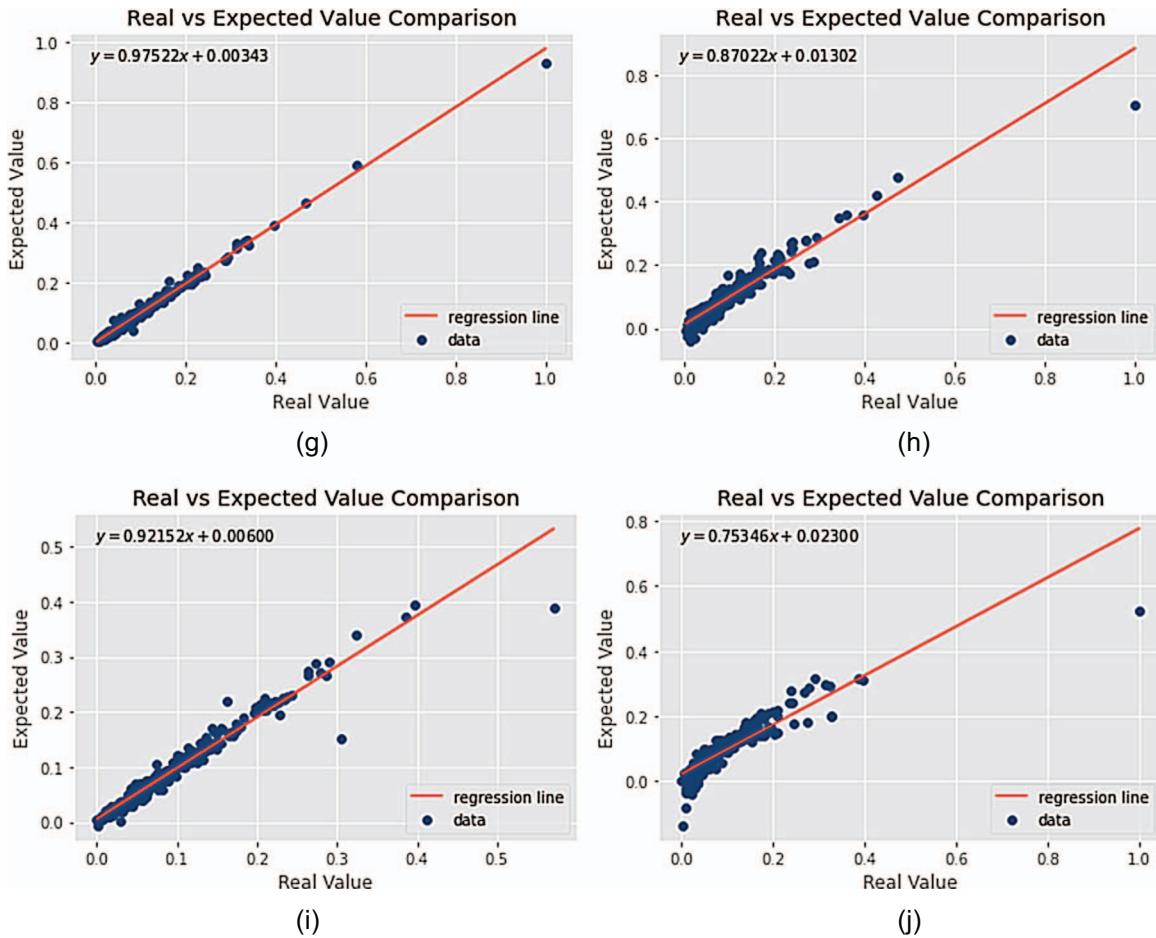


**Fig. 7.** Optimization results after the convergence. (a)  $R^2 = 0.9909$ , (b)  $R^2 = 0.9868$ , (c)  $R^2 = 0.9884$ , (d)  $R^2 = 0.9903$ , (e)  $R^2 = 0.9793$ , (f)  $R^2 = 0.9862$ , (g)  $R^2 = 0.9891$ , (h)  $R^2 = 0.9237$ , (i)  $R^2 = 0.9573$ , (j)  $R^2 = 0.7844$ .

To select the best offspring elites, design variables and fitness function definitions were necessary. The design variables were the learning rates and epoch sizes, as explained in Section 3.2. The fitness function  $F_O$  of GA for maximization is defined by,

$$F_O = \frac{1}{1 - R^2}, \quad (5)$$

where  $R^2$ ,  $R$ -squared, is called the coefficient of determination [43].  $R^2$  gives the information of how well the actual and predicted values are related because is the index for linear regression between the real and expected values. Accordingly, maximizing the fitness function,  $R^2$  becoming 1, during GA process (cf. Fig. 1) is suitable for the purpose of the model development. The closer  $R^2$  gets to 1, even though the subtle, small changes brought

**Fig. 7.** Continued.

about by  $R^2$ , the larger  $F_O$  becomes such that the fitness function increment is evident. The overall process of GA flow to optimize DNN hyperparameters is in [Figure 5](#).

## 5 Simulation results and discussion

The initial 10 populations of DNN were set to be random between 0.001 and 0.005 for learning rate, and between 10 000 and 40 000 for epoch size. The best score of the fitness function with the initial condition was given by 41.5 ( $R^2$  of 0.9759), however, prediction quality to replace real sensors demands better accuracy.

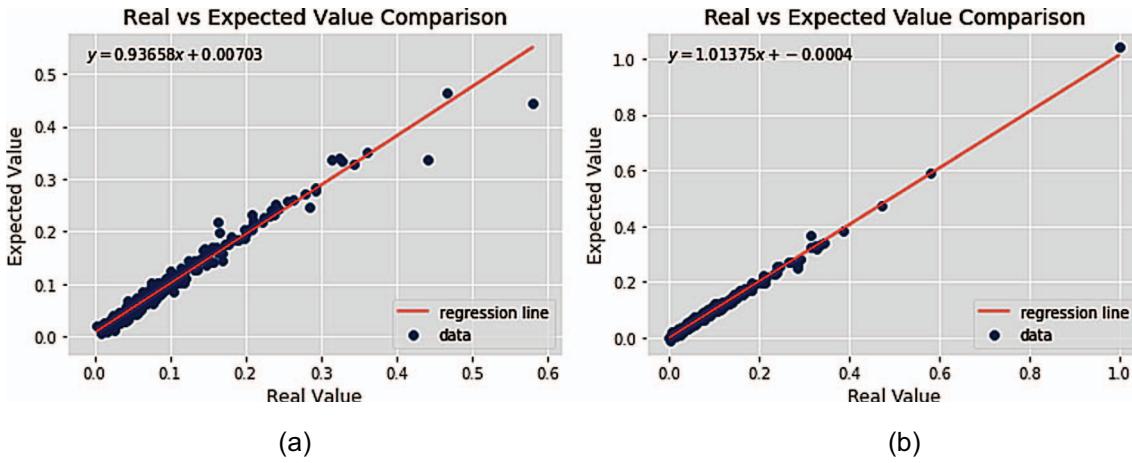
The learning rates and epoch sizes evolve as the generation continues in GA until the fitness function meets the convergence condition. GA is a global searching algorithm, however, it is not easy to judge whether the answer found by GA is global or not. However, it can be judged by checking the progress of fitness function. The convergence condition in this problem was defined as no more increase over 10 generations because if it is difficult to find a better optimum during such long generations, the found answer has a high possibility of the global optimum. [Figure 6](#) denotes  $F_O$  over GA generations. They show the best elite results until satisfying convergence condition. The score of the fitness function is lifted up to optimum in the 18th

**Table 4.** Learning rates and epoch sizes after convergence of GA.

Graph indication of <a href="#">Figure 7</a>	Learning rate	Epoch size
(a)	0.006926	72 321
(b)	0.007209	72 886
(c)	0.008894	72 321
(d)	0.007052	69 626
(e)	0.006052	69 321
(f)	0.007894	72 886
(g)	0.007209	72 626
(h)	0.005236	57 244
(i)	0.004612	13 809
(j)	0.003336	28 387

generation and this value continues until the 28th generation, where  $R^2$  is 0.9909 and very close to 1. Therefore, GA searching for optimum was successful.

In the DNN training process, the training cost for 28 generations of GA was about 7 h and 39 min. The training cost greatly depends on the learning rate and epoch size



**Fig. 8.** Performance comparison between the initial and optimal virtual NOx model. (a)  $R^2 = 0.9759$ , (b)  $R^2 = 0.9909$ .

such that the amount of time it takes for the single DNN training varies greatly from seconds to minutes.

Figure 7 shows the performance of 10 offsprings when GA meets the convergence condition, where the X-axis is real NOx emissions by experiment, and the Y-axis is the expected value by the virtual NOx sensor model. Basically, mse (mean square error) in DNN minimizes the error between the real and predicted values. In addition to this,  $R^2$  based object function makes the data points be lined up as much as the linear regression function. To be noted, a difference of 10% or more between the values occurs frequently even when  $R^2$  exceeds 0.97. Therefore, stricter  $R^2$  is requested for the real sensor replacement.

After simulation, the sweet spot was around 0.007 (learning rate) and 70 000 (epoch size), and the optimum was found in Figure 7a, where learning rate and epoch size were 0.006926 and 72 321, respectively ( $R^2$  of 0.9909). Figures 7a–7g also presented good performance and it was deduced that the selections of learning rate and epoch size for accurate prediction were not narrowly confined. However, too small epoch size always shows poor performance. Large values of epoch size and learning rate usually presented good performance, however, too large learning rate or epoch size would cause overshooting or overfitting problem, respectively.

Table 4 shows the learning rates and epoch sizes after GA optimization convergence condition. In the table, (a)–(d) show the four different elites from the previous generation, and (e)–(g) are the crossover results made by (a)–(d). Finally, (h)–(j) are the mutation results. They were found in a totally different area compared to the elites of (a)–(d)'s location. Mutation makes randomness in the chromosome. Therefore, the chromosome after the mutation can be very poor like (j), where the slope of the regression function is much less than 1. Although mutation often presents such poor performance, it has an important role in GA to overcome the local optimum problem.

Figure 8 shows the virtual NOx sensor performance comparisons between the initial and optimal models, where initial and optimal  $R^2$  were 0.9759 and 0.9909, respectively.

In Figure 8a the prediction accuracy is not guaranteed over the whole area and deviations of more than 10% are frequently found. However, overall uniform results are presented in Figure 8b. In particular, the points are attached to the regression function in an almost straight line indicates that the result of the prediction shows the performance enough to replace the actual sensor. As a result, it was found GA was able to derive the optimum hyperparameters in DNN resulting in performance improvements in the practical and straightforward handling of optimization for the virtual NOx sensor model. When we control the NOx formation in the engine, the engine operating strategy is based on the birth amount of NOx itself. The proposed virtual NOx sensor can perceive the effects of the various engine operating conditions with high sensing speed such that more exquisite engine control based on accurate prediction would be possible without real sensors.

## 6 Conclusion

In this study, a virtual NOx sensor model using DNN and a GA was developed using Python with TensorFlow. The training data was obtained from real engine experiment. The initial DNN-based NOx model produced a poor  $R^2$ . However, after GA optimization, the optimum  $R^2$  could be realized. To achieve such the accuracy, overlooked but influential hyperparameters, learning rates, and epoch sizes were selected and the fitness function  $1/(1 - R^2)$  was maximized until the convergence condition. As a result, the optimum value was found when the learning rate and epoch size were 0.006926 and 72 321 respectively, and the inaccurate and manually-tuned hyperparameters are defined evidently and automatically with GA such that the quality can be guaranteed to a practical level. It should be noted that once this sensor model has been developed, there is potential for the model and algorithm to be applied to other types of applications with some adjustments. Moreover, the reduction of real NOx sensors would lead to lower costs, prevent performance decreases over time,

and improve system durability. In the future works, virtual NOx sensor adaptation for the effective LNT/SCR operation would be also followed.

**Acknowledgments.** This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean Government (MSIT) (Nos. 2019R1A2C1090927 and 2021R1F1A1063048).

## References

- 1 Jiang J., Li D. (2016) Theoretical analysis and experimental confirmation of exhaust temperature control for diesel vehicle NOx emissions reduction, *Appl. Energy* **174**, 232–244.
- 2 Mera Z., Fonseca N., López J.-M., Casanova J. (2019) Analysis of the high instantaneous NOx emissions from Euro 6 diesel passenger cars under real driving conditions, *Appl. Energy* **242**, 1074–1089.
- 3 Ehsani M., Gao Y., Longo S., Ebrahimi K. (2018) *Modern electric, hybrid electric, and fuel cell vehicles*, CRC Press, Taylor & Francis Group, Boca Raton, FL.
- 4 The California Low-Emission Vehicle Regulations [<https://www.arb.ca.gov>].
- 5 Regulation (EC) No 715/2007 of the European Parliament and of the Council of 20 June 2007 on type approval of motor vehicles with respect to emissions from light passenger and commercial vehicles (Euro 5 and Euro 6) and on access to vehicle repair and maintenance information [<http://data.europa.eu/eli/reg/2007/715/oj>].
- 6 Lešnik L., Kegl B., Torres-Jiménez E., Cruz-Peragón F. (2020) Why we should invest further in the development of internal combustion engines for road applications, *Oil Gas Sci. Technol. – Rev. IFP Energies nouvelles* **75**, 56.
- 7 Chérel J., Zaccardi J.M., Bouteiller B., Allimant A. (2020) Experimental assessment of new insulation coatings for lean burn spark-ignited engines, *Oil Gas Sci. Technol. – Rev. IFP Energies nouvelles* **75**, 11.
- 8 Plee S., Ahmad T., Myers J.P., Faeth G.M. (1982) Diesel NOx emissions – A simple correlation technique for intake air effects, in: *Symposium (International) on Combustion*, Elsevier, pp. 1495–1502.
- 9 Tullis S., Greeves G. (1996) Improving NOx versus BSFC with EUI 200 using EGR and pilot injection for heavy-duty diesel engines, *SAE Trans.* 1222–1237. <https://doi.org/10.4271/960843>.
- 10 Lee K.H. (1997) Trends in technologies of exhaust gas in diesel engines, *Auto J.* **19**, 5, 9–19.
- 11 Smokers R., Vermeulen R., van Mieghem R., Gense R., Skinner I., Fergusson M., MacKay E., Brink P., Fontaras G., Samaras Z. (2006) Review and analysis of the reduction potential and costs of technological and other measures to reduce CO<sub>2</sub>-emissions from passenger cars, *TNO Rep.* **6**, 1.
- 12 Praveena V., Martin M.L.J. (2018) A review on various after treatment techniques to reduce NOx emissions in a CI engine, *J. Energy Inst.* **91**, 5, 704–720.
- 13 Yoo J.-H., Kim D.-W., Yoo Y.-S., Eum M.-D. (2009) Study on the characteristics of carbon dioxide emissions factors from passenger cars, *Trans. Korean Soc. Autom. Eng.* **17**, 4, 10–15.
- 14 Tsokolis D., Tsiakmakis S., Dimaratos A., Fontaras G., Pistikopoulos P., Ciuffo B., Samaras Z. (2016) Fuel consumption and CO<sub>2</sub> emissions of passenger cars over the New Worldwide Harmonized Test Protocol, *Appl. Energy* **179**, 1152–1165.
- 15 Schluckner C., Gaber C., Landfahrer M., Demuth M., Hochenauer C. (2020) Fast and accurate CFD-model for NOx emission prediction during oxy-fuel combustion of natural gas using detailed chemical kinetics, *Fuel* **264**, 116841.
- 16 Li T., Skreiber Ø., Løvås T., Glarborg P. (2019) Skeletal mechanisms for prediction of NOx emission in solid fuel combustion, *Fuel* **254**, 115569.
- 17 Ji J., Cheng L., Wei Y., Wang J., Gao X., Fangn M., Wang Q. (2019) Predictions of NOx/N<sub>2</sub>O emissions from an ultra-supercritical CFB boiler using a 2-D comprehensive CFD combustion model, *Particuology* **2**, 49.
- 18 Falcitelli M., Pasini S., Tognotti L. (2002) Modelling practical combustion systems and predicting NOx emissions with an integrated CFD based approach, *Comput. Chem. Eng.* **26**, 9, 1171–1183.
- 19 Vihar R., Bašković U.Ž., Katrašnik T. (2018) Real-time capable virtual NOx sensor for diesel engines based on a two-Zone thermodynamic model, *Oil Gas Sci. Technol. – Rev. IFP Energies nouvelles* **73**, 11.
- 20 Filippone A., Bojdo N. (2018) Statistical model for gas turbine engines exhaust emissions, *Transp. Res. Part D: Transp. Environ.* **59**, 451–463.
- 21 Chen S.K., Mandal A., Chien L.-C., Ortiz-Soto E. (2018) Machine learning for misfire detection in a dynamic skip fire engine, *SAE Int. J. Engines* **11**, 2018-01-1158, 965–976.
- 22 Li N., Lu G., Li X., Yan Y. (2016) Prediction of NOx emissions from a biomass fired combustion process based on flame radical imaging and deep learning techniques, *Combust. Sci. Technol.* **188**, 2, 233–246.
- 23 Li H., Butts K., Zaseck K., Liao-McPherson D., Kolmanovsky I. (2017) *Emissions modeling of a light-duty diesel engine for model-based control design using multi-layer perceptron neural networks*, SAE Technical Paper.
- 24 Oduro S., Ha Q.P., Duc H. (2016) Vehicular emissions prediction with CART-BMARS hybrid models, *Transp. Res. Part D: Transp. Environ.* **49**, 188–202.
- 25 Guardiola C., Pla B., Blanco-Rodriguez D., Calendini P.-O. (2015) ECU-oriented models for NOx prediction. Part 1: A mean value engine model for NOx prediction, *Proc. Inst. Mech. Eng. Part D: J. Automobile Eng.* **229**, 8, 992–1015.
- 26 Bertram A.M., Kong S.-C. (2017) *Augmentation of an Artificial Neural Network (ANN) model with expert knowledge of critical combustion features for optimizing a compression ignition engine using multiple injections*, SAE Technical Paper.
- 27 Ganesan V., Porai P.T. (2013) *Optimization of fuel injection timing of a gasoline engine using artificial neural network*, SAE Technical Paper.
- 28 Lucido M., Shibata J. (2018) *Learning gasoline direct injector dynamics using artificial neural networks*, SAE Technical Paper.
- 29 Wang G., Awad O.I., Liu S., Shuai S., Wang Z. (2020) NOx emissions prediction based on mutual information and back propagation neural network using correlation quantitative analysis, *Energy* **198**, 117286.
- 30 Arsie I., De Cesare M., Lazzarini F., Pianese C., Sorrentino M. (2017) Neural network models for virtual sensing of NOx emissions in automotive diesel engines with least square-based adaptation, *Cont. Eng. Pract.* **61**, 11–20.

- 31 Wang Y.-Y., He Y., Rajagopalan S. (2011) Design of engine-out virtual NO<sub>x</sub> sensor using neural networks and dynamic system identification, *SAE Int. J. Engines* **4**, 1, 828–836.
- 32 Yang G., Wang Y., Li X. (2020) Prediction of the NO<sub>x</sub> emissions from thermal power plant using long-short term memory neural network, *Energy* **192**, 116597.
- 33 Xie P., Gao M., Zhang H., Niu Y., Wang X. (2020) Dynamic modeling for NO<sub>x</sub> emission sequence prediction of SCR system outlet based on sequence to sequence long short-term memory network, *Energy* **190**, 116482.
- 34 Hoos H., Hutter F., Leyton-Brown K. (2014) Proc. An efficient approach for assessing hyperparameter importance, in: *International Conference on Machine Learning, June 21–June 26, 2014 and in Beijing, China*, pp. 754–762.
- 35 Ioffe S., Szegedy C. (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: *32nd International Conference on Machine Learning, Lille, France*.
- 36 van Laarhoven T. (2017) L2 regularization versus batch and weight normalization, in: *31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA*.
- 37 Haykin S. (1994) *Neural networks: A comprehensive foundation*, Prentice Hall PTR, Upper Saddle River, NJ, United States.
- 38 Bock S., Goppold J., Wei M. (2018) An improvement of the convergence proof of the ADAM-Optimizer, in: *Conference Paper At Oth Clusterkonferenz 2018, 13 April, 2018*.
- 39 Balles L., Hennig P. (2017) Dissecting Adam: The sign, magnitude and variance of stochastic gradients, in: *35th International Conference on Machine Learning, Stockholm, Sweden*.
- 40 Jacobson S., Reichman D., Bjornstad B., Leslie M., Collins L.M., Malof J.M. (2019) Proc. Reliable training of convolutional neural networks for GPR-based buried threat detection using the Adam optimizer and batch normalization, in: *Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XXIV*, International Society for Optics and Photonics, 1101206 p.
- 41 Kingma D.P., Ba J. (2014) Adam: A method for stochastic optimization, in: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA*.
- 42 Shopova E.G., Vaklieva-Bancheva N.G. (2006) BASIC – A genetic algorithm for engineering problems solution, *Comput. Chem. Eng.* **30**, 8, 1293–1309.
- 43 Gelman A., Goodrich B., Gabry J., Vehtari A. (2019) R-squared for Bayesian regression models, *Am. Stat.* **73**, 3, 307–309.