2021-24-0076

A random forest algorithmic approach to predicting particulate emissions from a highly boosted GDI engine

Nick Papaioannou[1], Xiaohang Fang[1], Felix CP Leach[1], Andrew GJ Lewis[2], Sam Akehurst[2], James WG Turner[3]

1. Department of Engineering Science, University of Oxford, UK
2. University of Bath, UK
3. KAUST: King Abdullah University of Science and Technology, Saudi Arabia

SAE Technical Paper – Author's Accepted Manuscript

Abstract

Particulate emissions from gasoline direct injection (GDI) engines continue to be a topic of substantial research interest. Forthcoming regulation both in the USA and the EU will further reduce their emission and drive innovation. Substantial research effort is spent undertaking experiments to understand, characterize, and research particle number (PN) emissions from engines and vehicles. Recent advances in computing power, data storage, and understanding of artificial intelligence algorithms now mean that these are becoming an important tool in engine research. In this work a random forest (RF) algorithm is used for the prediction of PN emissions from a highly boosted (up to 32 bar BMEP) GDI engine. Particle size, concentration, and the accumulation mode geometric standard deviation (GSD) are all predicted by the model. The results are analysed and an in depth study on parameter importance is carried out. The Random Forest algorithm is used as an estimator and the various engine parameters are ranked with a permutation feature importance technique using mean squared error as a performance metric. The results showed that from 82 model parameters only 17 are important for predicting the above PN emission parameters. Moreover, the permutation importance algorithm showed that when the parameters are reduced to 9 the model accuracy is improved due to a reduction in model variance. Overall, the model shows excellent predictive performance for all three parameters even when an independent dataset is used.

Introduction

Particle number (PN) emissions from light duty vehicles have been regulated in Europe since 2011 (diesel) and 2014 (petrol) – currently at a level of 6×10^{11} #/km. PN emissions from Gasoline Direct Injection (GDI) engines have been a topic of research interest for some time [1] particularly because the reduced time for mixture formation compared to Port Fuel Injected (PFI) engines can lead to higher engine-out particulate emissions [2]. However, due to their lower CO_2 emissions than previous engine technologies, GDI engines are widely used in the market [3]. Particulate emissions from GDI engines are typically bilognormal in distribution, with two clear modes – the nucleation mode from 10-40 nm (typically consisting of volatile and semi-volatile particles) and the accumulation mode from 40-200 nm (typically consisting of solid particles with adsorbed material) [4].

Downsized engines are used to increase fuel economy and reduce CO_2 emissions from internal combustion engines and are available in the market today [5], and many further engines are in development [6, 7]. Downsizing moves the engine operating point to a more efficient region by reducing engine capacity; full load capability is maintained by pressurising the inlet air, generally with a turbocharger.

Particle Mass (PM) and PN emissions from downsized engines have been widely studied in recent years [8-12], with such engines showing a smaller and more diverse set of particulate emissions compared to diesel engines. However, both engine-out techniques and gasoline particulate filters have been shown to effectively remove such PN from these engines including in the sub-23nm region, and reduce these emissions to very low levels [13, 14].

Machine Learning (ML) techniques have been increasingly applied to internal combustion engines in recent years. Notably, artificial neural networks (ANN)s have been used predictively for a number of engine design and combustion parameters. This includes the design of engine controllers [15] and systems [16]. Combustion parameters that have been predicted by ANNs by researchers include indicated mean effective pressure (IMEP) and its coefficient of variation (CoV of IMEP) [17], BSFC and BTE [18] and exhaust temperature [19]. So far the studies mentioned have focused on diesel engines, in part due to their lower cycle-to-cycle variation compared to spark ignition (SI) engines; however more recent studies have used ANNs to study cycle-to-cycle variation in SI engines as well [17].

Emissions predictions using ANNs are less common, and most studies have focused on diesel engines. Desantes *et al.* as early as 2002, predicted NO_x and soot from a Euro IV heavy duty diesel engine [20], showing strong correlations. Subsequent studies from different diesel engines of all sizes have shown that ANNs can be used predictively, particularly for NO_x emissions [18, 21-25]. Studies using ML techniques for the prediction of emissions formation from gasoline engines, particularly GDI engines are sparse in the literature, and the authors are not aware of any studies which predict PM or PN emissions from GDI engines using ML techniques in the literature.

While ANNs have been used for the prediction of a number of ICE performance and emissions parameters, they are computationally expensive, and require large quantities of data. Other ML approaches used include genetic algorithms [26, 27] and the K-Nearest Neighbors Algorithm [28]. A Random Forest (RF) algorithm is a less computationally expensive approach that does not need as many data points for a good solution. Another advantage of the RF algorithm is that given the use of a greedy algorithm during training (refer to the *Methodology* section), Random Forests allow for the identification of important parameters to the model thus independent algorithms used to evaluate parameters importance are not required.

RF algorithm is an ensemble of Decision Trees that can be used for classification and regression problems with its output being the average value of all the Decision trees outputs. A simple two layer tree for a two variable regression problem is presented in Figure 2 for illustrative purposes. The two model parameters are engine power and engine blowby and the model is used to predict particle matter concentration (Both parameters are normalized so their magnitude is irrelevant).

The decision tree splits the parameter space into partitions (branches), according to a threshold which changes at every split (first line of each box in Figure 1). The parameter as well as the threshold to be used for splitting follows some specified performance criterion (indicated by *mse* in Figure 1), and the splitting of the tree is continued until a terminal node is reached (leaf) where a single prediction is given (indicated by *value* in Figure 1). The splitting of the tree as well as the number of terminal nodes can be controlled using various hyperparameters and it is discussed further in the Permutation feature importance section.

The model's output (prediction) is the mean of the target parameter's samples within the partition. This process is better understood by a 2D representation of the decision tree as can be seen in Figure 2. Figure 2 shows the corresponding values that the two model parameters take. The horizontal lines correspond to a splitting criterion for engine power while the vertical lines correspond to a splitting criterion for engine blowby. Given the simplicity of this example, engine power has two splitting criteria and engine blowby only one, hence the two horizontal and one vertical line respectively. These splitting criteria are indicated in the first line of the respective text box in Figure 1. As mentioned before the values presented here are normalized, so the reader should not focus on their physical interpretation rather on how they are used to form the resulting partitions. Once the tree is fully grown (i.e. when the model is trained), each resulting partition will include a given subset of the target variable and the average value of the subset will be the model's output for that partition. Clearly, the bigger the tree the more the resulting partitions, which allow for a higher fidelity model.

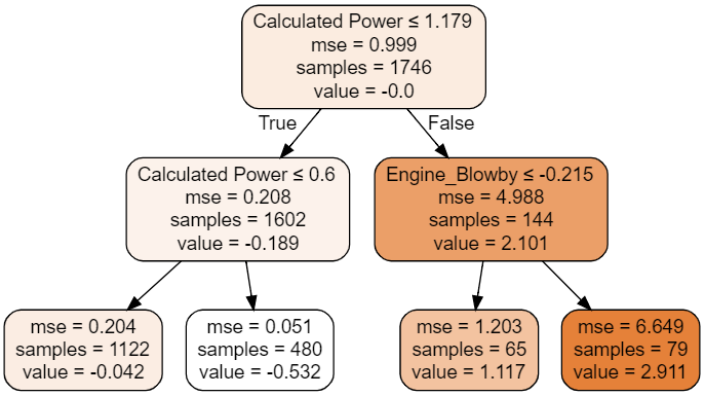

Figure 1: Typical decision tree structure for a two variable regression problem

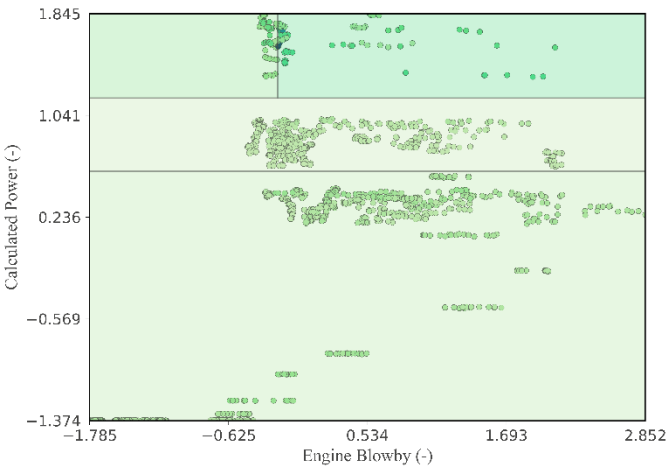

Figure 2: The equivalent 2D representation of Figure 1 in its parameter space

In decision trees, all the sample parameters and samples are considered during the creation of each node. This runs the risk of overfitting the model to the data – the model will not be able to generalize well outside the data it has been trained on – especially if the size of that data grows significantly. This is due to the fact that a small change in the dataset can result in a completely different tree structure. This is where the use of Random Forests becomes advantageous.

Random Forests gives the average value of the underlying decision trees – a procedure known as bagging. Sampling for each individual tree is commonly done using a technique called bootstrapping where sampling with *replacement* is carried out for each tree resulting in inherently different trees. More specifically, if the original sample size is N, each decision tree is using X samples for training ($X < N$), with the sample size X being randomly generated from the original sample. Before the next randomly generated sample is created (for the next decision tree) the samples used to train the first tree are replaced back into the original sample. Bootstrapping allows for the creation of independent samples when the original sample size is quite small and when obtaining new samples is either impractical or costly.

Further to this, at each node, a subset m of the total p parameters is selected at random, and only these m variables are considered for the partition at the node. This allows for the creation of even more independent trees since it is possible that some samples will be represented in more than one bootstrapped samples. Consequently, even if two trees were created from the same bootstrapped sample the resulting trees will likely be different [29].

Random Forest algorithms have been used to assess atmospheric particulate matter in the past – focusing on particle mass [30, 31]. In addition RF approaches have been used when assessing cycle-to-cycle variations in combustion quality and knock in internal combustion engines [32, 33]. To the authors' knowledge no studies have been undertaken using an RF algorithm to predict particulate matter emissions from any engine.

In this work, a Random Forest algorithm has been used to predict particulate matter emissions. More specifically the authors focused on the prediction of particle size, concentration and the accumulation mode geometric standard deviation (GSD) from a highly boosted GDI engine. Initially a multi-output regression Random Forest model is fitted to the training data and the trained model is used to highlight the most important parameters in the prediction of particulate emissions using a permutation feature importance technique. An extended discussion on the engine parameter importance and PM emissions then presented.

Methodology

A Random Forest model was trained and validated on a previously published set of experimental particulate emissions data [8]. This section will first detail the model configuration and then a brief overview of the experimental data.

Random Forests

In this work the splitting criterion used for each branch of the decision tree was mean squared error as shown by the following equation:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - f(X_i))^2 \quad (1)$$

where n is the number of predictor values within a given partition, Y_i are the target values in the partition and $f(X_i)$ is the corresponding predictor value which is the mean value at the leaf node (terminal node).

In order to find the optimum parameter and value to perform a split, a greedy algorithm was used. The algorithm initialises by considering all the available data, by selecting a splitting parameter j and a split value s. This creates two partitions $R_1(j,s)$ and $R_2(j,s)$ with the optimum values given by minimising the following equation.

$$\min(j,s)\left[\min\left(\frac{1}{n}\sum_{i=1}^{n}\left(Y_{R1,i} - f(X_{R1,i})\right)^2\right) + \min\left(\frac{1}{m}\sum_{i=1}^{m}\left(Y_{R2,i} - f(X_{R2,i})\right)^2\right)\right] \quad (2)$$

As it can be seen from Equation 2, for any given value of j and s, the internal minimisation value is given by the average of the values in that partition.

This process is repeated for all input variables until the optimal combination of j and s is found for a given node. Finally, the output of the Random Forest algorithm is given by the following Equation:

$$f_{RF}(X) = \frac{1}{B}\sum_{b=1}^{B} T_b(X) \quad (3)$$

Where B is the total number of trees used and $T_b(X)$ is the prediction for each tree.

The score of the model was assessed using the coefficient of determination R^2 of the prediction. The coefficient of determination is defined in Equation 4.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(Y_i - f(X_i))^2}{\sum_{i=1}^{n}(Y_i - \bar{Y}))^2} \quad (4)$$

Where $f(X_i)$ is the predictor value for Y_i and $\bar{Y}$ is the mean of the target values.

Permutation feature importance

Permutation feature importance is a model inspection technique that can be used for any trained model. It is defined as the reduction in a model score when an input parameter is randomly shuffled [29]. This method breaks the link between a parameter (or feature) and the target variable, hence the resulting drop in model score indicates how important the feature is to the model. Permutation feature importance is most commonly applied to a dataset that has not been used for training in order to highlight the importance of a feature in the generalization performance of the model.

To implement the permutation feature importance method, a model is trained and an initial, reference score is computed (e.g. R^2). Then each feature (or input parameter) is randomly shuffled and a new model score is calculated. Each feature can be shuffled several times and the average of the scores can be used as a more robust indicator. The importance of each feature is calculated according to the equation below:

$$i_x = s_{ref} - \frac{1}{K}\sum_{k=1}^{K} s_{kj} \quad (5)$$

where s_{ref} is the reference score for the model and s_{kj} is the new score for the kth shuffle of feature j.

Model configuration

A trained Random Forest model was used to highlight the importance of the input parameters using a technique called permutation importance. This technique removes parameters from the trained model and records the respective increase in error thus highlighting their importance. The higher the increase in model error the more important the parameter is. However in order for this method to be successful, correlation between model parameters must be very low. A highly correlated parameter might misleadingly show to have little importance since the model can take the same information from the correlated parameters thus resulting in small error (MSE) [34].

Therefore, a Pearson correlation test on the input parameters was performed prior to model training. The correlation test allowed for groups of highly correlated parameters ($R > 0.75$ and $R < -0.75$) to be identified and then a single parameter, representing that group, to be selected. All other parameters were then removed from the dataset. This process was carried out until no strongly correlated parameters were left and reduced the model parameters from 82 down to 21. Figure 3 shows a Pearson correlation heat-map of the remaining parameters after the correlation test was completed.

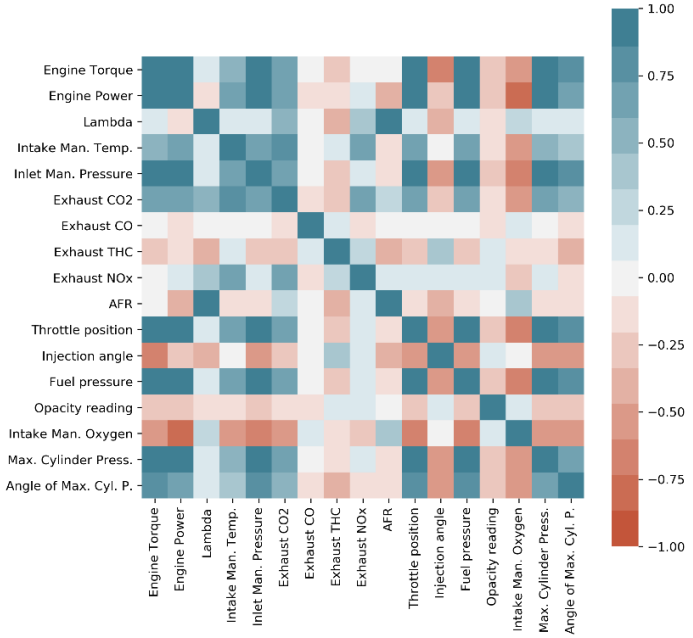

Figure 3: Pearson correlation heat-map for the model parameters

The next step involved training the Random Forests model. A common approach in training a machine learning model is to split the available data into a training set, on which the model will be trained on, and a validation set, on which the performance of the model will be tested. For the training of the Random forest models the training and validation sets were 75% and 25% of the total dataset respectively. Finally, all input and target parameters were standardized according to Equation 1, where x is the non-standardized parameter, $\bar{x}$ is the average of the values that x takes, and $\sigma_x$is the standard deviation of parameter x from the mean. This ensured that discrepancies in the range and magnitude of the various parameters will not have an effect on the training of the model.

$$x_{std} = \frac{(x-\bar{x})}{\sigma_x} \qquad (6)$$

The characterization of the underlying trees and the resulting forest is carried out using several parameters (also known as hyperparameters). These parameters can take a wide range of values depending on the problem at hand, which means that to find an optimum model configuration, the number of possible combinations to be tested becomes very large very quickly. To avoid the computational cost and time of testing all available combinations, a randomized parameter optimization method was used (*RandomizedSearchCV)*. This method randomly samples a predefined number of combinations and trains a user-defined model (in this case a Random Forests regressor), with the optimal configuration being the one with the highest score(R^2). During this process 100 combinations were tested and the range of values for each hyperparameter was based on a best-practice approach.

The hyperparameters tested were; the number of trees in the forest, the maximum number of layers in the tree (i.e. how deep the tree will be – for reference Figure 1 shows a tree with 2 layers), the number of features considered at each split (i.e. how many input parameters will be considered at each split), the minimum number of samples at each split (i.e. how many samples will be considered in Equation 1; in reference to Figure 2 this is the number of samples within a partition) and the minimum number of samples at the leaf (terminal) nodes (i.e. the minimum number of samples that will produce the final tree prediction; marked as *samples* at the final layer of the tree shown in Figure 1). The range of the hyperparameters tested as well as their final values are presented in Table 1.

Table 1: Tested and optimal hyperparameter values for the Random Forests model

Hyperparameter	Range	Optimal value
# of trees in the forest	(2, 12, 23, 34, 45, 56, 67, 78, 89, 100)	45
Max. tree layers	(1, 3, 6, 9, 11, 14, 17, 19, 22, 25)	19
# of features considered at each split	(all parameters, $\sqrt{all\ parameters}$)	$\sqrt{all\ parameters}$
Min. samples at each split	(2,5,10)	5
Min. samples at leaf nodes*	(4 6, 8, 10)	8

*See Fig 1.

Engine

The experimental data that was used in this work was obtained from the UB100 Ultraboost engine. The UB100 engine is a 'proof-of-concept' prototype engine and an introduction to this engine is given in Turner *et al.* [35]. This engine was a lab-based prototype, and as such the functions of the turbocharger and supercharger were provided using an external charging system (which controlled the inlet air temperature and pressure) in conjunction with an exhaust back pressure valve (which controlled the exhaust pressure). The engine was not fitted with any aftertreatment – of particular note in this instance is that there was no TWC, which has been shown to remove volatile particles and no particle filter. The specifications of the engine are shown in Table 2.

Table 2: Specifications of the test engine

Engine	UB100
Type	In-line 4 cylinder
Bore × Stroke	83 × 92 mm
Displacement	1991 cm^3
Valves per cylinder	2 intake, 2 exhaust
Compression ratio	9:1
Maximum fuel pressure	200 bar
Aspiration	External boost

Full details of the experimental set-up are given in [12, 22], but of particular note to this work is that the PN emissions were measured using a Cambustion DMS500 which gave the particle concentration, mode diameter (CMD), and GSDs used in this work.

Experimental test conditions

The experimental test results have been reported widely in other works [8, 10-12, 35], and are not the focus of this paper. Nevertheless, the test

conditions were selected to mimic typical highly-boosted GDI engine operation (within a homologation drive cycle envelope, in this case a point of high residence over a New European Drive Cycle), as well as to test the operating envelope of the system. The test conditions used in this work are explained fully in [8], however for convenience a summary is shown in Table 3. At all of the four test conditions a number of spark timings were tested, and another one (or in one case two) parameters was varied only at a single test condition, as detailed in Table 3.

Table 3: Engine test conditions

Test Condition	1	2	3	4
Engine speed (rpm)	1250	2000	3000	4000
Load	3.77 bar BMEP	Boosted WOT, KLSA*	Boosted WOT, KLSA*	Boosted WOT, KLSA*
Number of spark timings tested	8	10	10	3
Parameter varied	EGR (0-10%)	Inlet air temperature (20-40°C) and fuel injection timing (3 timings)	Exhaust Back Pressure (Super-turbo charged transition)	λ (1.0-0.875)

*Wide Open Throttle, Knock Limited Spark Advance – i.e. full load

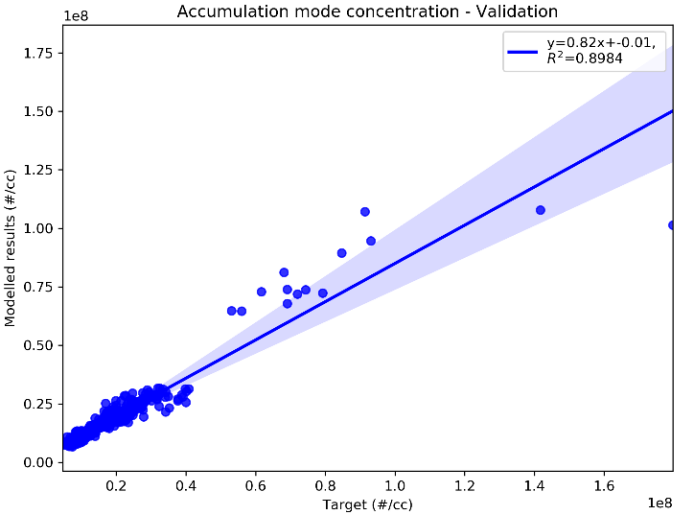

Figure 4: Random Forests prediction of particle concentration for the validation dataset

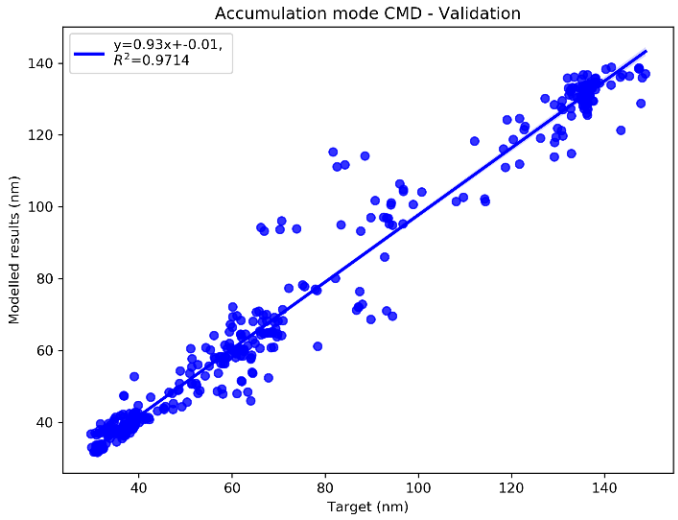

Figure 5: Random Forests prediction of CMD for the validation dataset

Results

Validation and parameter importance results

The Random Forests model was trained using the hyperparameters shown in Table 1 and the validation set was then fitted on the trained model. The performance criterion used to gauge model accuracy was the coefficient of determination, R^2 (Equation 4), and the model's error was assessed using the mean squared error (MSE) (Equation 1).

Figures 4-6 shows the concentration, CMD and GSD results for the validation set (as a reminder, the validation set are data points which are not included in the model training at all, but are equally representative of all the engine operating points) where a very good agreement between the prediction and the target values can be seen, indicating a good generalization performance. As may be expected, the training results showed an equally good performance and are thus not presented here.

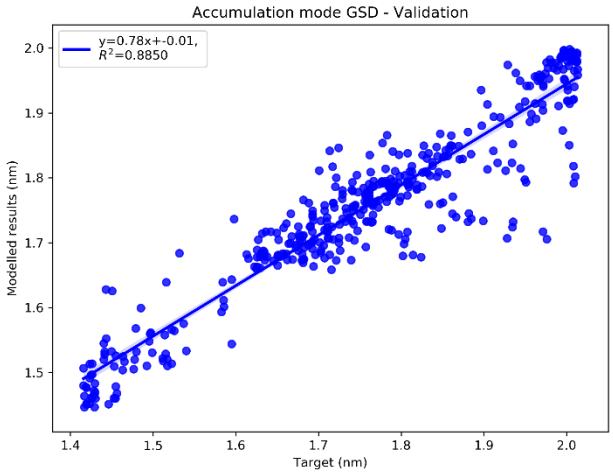

Figure 6: Random Forests prediction of GSD for the validation dataset

Permutation feature importance results

As already discussed, during permutation importance a model feature is shuffled randomly several times and the drop in score is recorded (i.e. its importance). Here, each parameter was permuted 30 times and the process was repeated 10 times to reduce the effects of randomness. Figure 7 shows the results of the permutation importance where the bars indicate the average importance value over the 10 trials and the error bars indicate the one standard deviation limit.

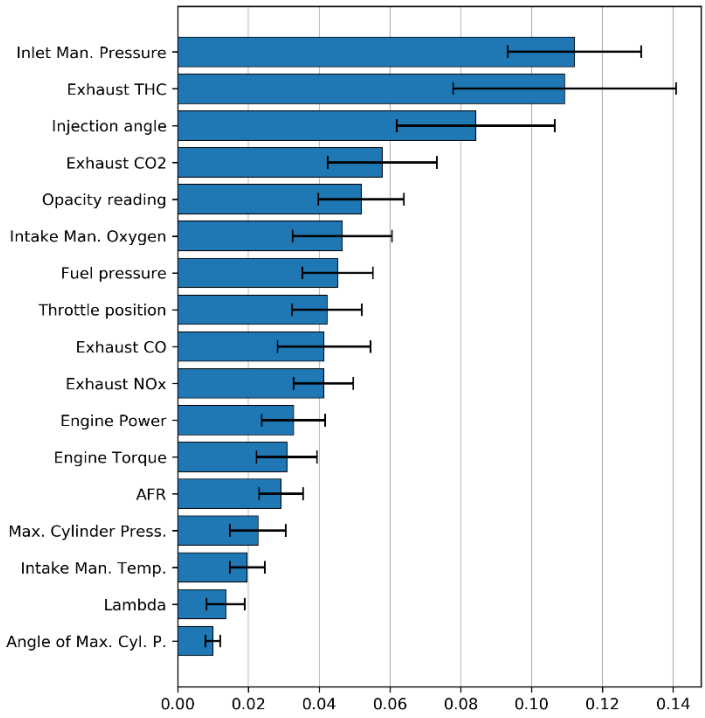

Figure 7: Permutation importance results. The bars indicate the average feature importance (i.e. the drop in model score, R^2) over 10 runs and the error bars indicate one standard deviation from the mean.

It is interesting to consider the parameters that are observed to be the most important in Figure 7 in the context of current understanding about particulate emission formation. Note that the presence of a parameter in Figure 7 does not imply a positive correlation, it could equally imply a negative correlation. Some parameters here are directly related to particulate formation such as the opacity reading. Others relate to engine load directly or indirectly (inlet manifold pressure, injection angle, exhaust CO_2, throttle position, power, torque, maximum cylinder pressure, and angle of maximum cylinder pressure) – and engine load is known to be closely correlated with particulate matter emissions[4]. Other parameters are also expected to have a direct relationship with particulate matter emissions such as AFR & lambda [36], THC & CO emissions (indicative of incomplete combustion and hence higher particulate matter emissions), fuel pressure [9], intake manifold oxygen (indicative of EGR), and intake manifold temperature [8]. By interpreting the RF parameter importance in this way it is possible to link the data-driven ML approach to the physical understanding of the problem, demonstrating the power of the RF model alongside the physical realities of the system.

The results in Figure 7 can also be used to remove low importance features from the model without compromising accuracy. With this in mind any parameter with an importance value below 0.05 was removed which reduced the model parameters from 17 to 9. Considering that the accuracy of the original model is R^2=0.9196 (refer to Table 4, average R^2 value) removing the parameters below the 0.05 importance threshold in Figure 7, resulted at a higher model score of R^2=0.9396 (refer to Table 4 – average R^2 value for the reduced model). This increase in model's accuracy makes sense as low importance features increase the model's variance. The reduced model was then trained using the hyperparameters shown in Table 1. Overall the reduced model performed equally well compared to the original indicating that the reduction in model parameters did not have an adverse effect on model performance. A comparison between the original and reduced models in terms of score (R^2) and error (MSE) can be seen in Tables 3 and 4 for the training and validation datasets respectively.

Table 4: Comparative error (MSE) and score (R^2) results for the original and reduced-size models for the training dataset.

	Training set					
	Accumulation mode concentration		**Accumulation mode CMD**		**Accumulation mode GSD**	
	Orig.	**Red.**	**Orig.**	**Red.**	**Orig.**	**Red.**
MSE	0.0792	0.0829	0.0231	0.0211	0.0824	0.0854
R^2	0.9260	0.9215	0.9776	0.9793	0.9263	0.9180

Table 5: Comparative error (MSE) and score (R^2) results for the original and reduced-size models for the validation dataset.

	Validation set					
	Accumulation mode concentration		**Accumulation mode CMD**		**Accumulation mode GSD**	
	Orig.	**Red.**	**Orig.**	**Red.**	**Orig.**	**Red.**

MSE	0.0983	0.1116	0.0308	0.0295	0.1376	0.1456
R^2	0.9010	0.8837	0.9716	0.9728	0.8862	0.8741

Summary / Conclusions

This work presented the use of a Random Forests algorithm in predicting, particle size, concentration, and the accumulation mode geometric standard deviation over a range of engine operating conditions for a highly boosted GDI engine. Further to this a permutation importance algorithm was also used to highlight the most important parameters to the model.

The performance of various model hyperparameters was tested and the optimum combination was selected based on a mean-squared error minimization criterion. Prior to model training, highly correlated parameters were removed to increase the performance of the permutation importance algorithm. This process resulted in a total of 17 input parameters out of the original 82.

The trained model showed a very good agreement for all target variables and a validation test on the model's performance on an independent dataset (i.e. have not been used for model training) showed equally good performance. This is a very important outcome as with a single model all three target variables are predicted accurately. To the authors' knowledge this is the first time that a multi-output, data-driven model yields such accurate results in predicting PM emissions, especially when using a relatively simple and computationally inexpensive method like the Random Forests algorithm.

Finally this work showed that by complementing the RF algorithm with a permutation importance method the user can get significant insight into the model's performance. The permutation importance method highlighted, in order of importance, the most important parameters to the model thus allowing the authors to remove low importance parameters. This resulted in a final model using a total of 9 parameters out of the original 82 and to also further improve the predictive ability of the model. Most importantly though, interpreting the importance of the various parameters alongside the physical realities of the problem can provide useful insights into the physical understanding of the problem.

References

1. Raza, M., et al., *A Review of Particulate Number (PN) Emissions from Gasoline Direct Injection (GDI) Engines and Their Control Techniques.* Energies, 2018. **11**(6):1417. https://doi.org/10.3390/en11061417
2. Zhao, H., *Overview of Gasoline Direct Injection Engines*, in *Advanced direct injection combustion engine technologies and development: Gasoline and gas engines*. 2010, Woodhead Publishing Ltd.
3. Leach, F., et al., *The scope for improving the efficiency and environmental impact of internal combustion engines.* Transportation Engineering, 2020. **1**:100005. https://doi.org/10.1016/j.treng.2020.100005
4. Eastwood, P., *Particulate Emissions from Vehicles*. 2008: SAE International and John Wiley & Sons, Ltd.
5. Friedfeldt, R., et al., *Three-Cylinder Gasoline Engine with Direct Injection.* Auto Tech Review, 2013. **2**(2): p. 32-37.
6. Martin, S., C. Beidl, and R. Mueller, *Responsiveness of a 30 Bar BMEP 3-Cylinder Engine: Opportunities and Limits of Turbocharged Downsizing*. SAE Technical Paper 2014-01-1646, 2014, https://doi.org/10.4271/2014-01-1646
7. Hancock, D., Fraser, N., Jeremy, M., Sykes, R. et al., *A New 3 Cylinder 1.2l Advanced Downsizing Technology Demonstrator Engine*, SAE Technical Paper 2008-01-0611, 2008, https://doi.org/10.4271/2008-01-0611
8. Leach, F., et al., *Particulate emissions from a highly boosted Gasoline Direct Injection engine.* International Journal of Engine Research, 2018. **19**(3): p. 347–359. https://doi.org/10.1177/1468087417710583
9. Leach, F., Knorsch, T., Laidig, C., and Wiese, W., *A Review of the Requirements for Injection Systems and the Effects of Fuel Quality on Particulate Emissions from GDI Engines*, SAE Technical Paper 2018-01-1710, 2018, https://doi.org/10.4271/2018-01-1710
10. Leach, F.C.P., et al., *The effect of oxygenate fuels on PN emissions from a highly boosted GDI engine.* Fuel, 2018. **225**: p. 277-286. https://doi.org/10.1016/j.fuel.2018.03.148
11. Leach, F.C.P., et al., *The effect of fuel composition on particulate emissions from a highly boosted GDI engine – an evaluation of three particulate indices.* Fuel, 2019. **252**: p. 598-611. https://doi.org/10.1016/j.fuel.2019.04.115
12. Leach, F., Lewis, A., Akehurst, S., Turner, J. et al., *Sub-23 nm Particulate Emissions from a Highly Boosted GDI Engine*, SAE Technical Paper 2019-24-0153, 2019, https://doi.org/10.4271/2019-24-0153
13. Joshi, A. and T.V. Johnson, *Gasoline Particulate Filters—a Review.* Emission Control Science and Technology, 2018. **4**(4): p. 219-239. https://doi.org/10.1007/s40825-018-0101-y
14. Strzelec, A. and J. Kasab, *Automotive Emissions Regulations and Exhaust Aftertreatment Systems*. 2020: SAE International.
15. Hafner, M., M. Schüler, and R. Isermann, *Fast Neural Networks for Diesel Engine Control Design.* IFAC Proceedings Volumes, 1999. **32**(2): p. 8154-8159. https://doi.org/10.1016/S1474-6670(17)57391-7
16. Czarnigowski, J., *A neural network model-based observer for idle speed control of ignition in SI engine.* Engineering Applications of Artificial Intelligence, 2010. **23**(1): p. 1-7. https://doi.org/10.1016/j.engappai.2009.09.008
17. Di Mauro, A., H. Chen, and V. Sick, *Neural network prediction of cycle-to-cycle power variability in a spark-ignited internal combustion engine.* Proceedings of the Combustion Institute, 2019. **37**(4): p. 4937-4944. https://doi.org/10.1016/j.proci.2018.08.058
18. Roy, S., R. Banerjee, and P.K. Bose, *Performance and exhaust emissions prediction of a CRDI assisted single cylinder diesel engine coupled with EGR using artificial neural network.* Applied Energy, 2014. **119**: p. 330-340. https://doi.org/10.1016/j.apenergy.2014.01.044
19. Parlak, A., et al., *Application of artificial neural network to predict specific fuel consumption and exhaust temperature for a Diesel engine.* Applied Thermal Engineering, 2006. **26**(8): p. 824-828. https://doi.org/10.1016/j.applthermaleng.2005.10.006
20. Desantes, J., López, J., García, J., and Hernández, L., *Application of Neural Networks for Prediction and Optimization of Exhaust Emissions in a H.D. Diesel Engine*, SAE Technical Paper 2002-01-1144, 2002, https://doi.org/10.4271/2002-01-1144
21. Zhang, Q., Pennycott, A., Burke, R., Akehurst, S. et al., *Predicting the Nitrogen Oxides Emissions of a Diesel Engine using Neural Networks,* SAE Technical Paper 2015-01-1626, 2015, https://doi.org/10.4271/2015-01-1626
22. Fang, X., et al., *On the application of artificial neural networks for the prediction of NOx emissions from a high-speed direct injection diesel engine.* International Journal of Engine

Research, 2021. **22**(6): p. 1808-1824. https://doi.org/10.1177/1468087420929768
23. Papaioannou, N., et al. *Prediction of NOx Emissions for a Range of Engine Hardware Configurations Using Artificial Neural Networks*. in *ASME 2020 Internal Combustion Engine Division Fall Technical Conference*. 2020. https://doi.org/10.1115/ICEF2020-2911
24. Taglialatela, F., Lavorgna, M., Di Iorio, S., Mancaruso, E. et al., *Real Time Prediction of Particle Sizing at the Exhaust of a Diesel Engine by Using a Neural Network Model*, SAE Int. J. Engines 10(4):2202-2208, 2017, https://doi.org/10.4271/2017-24-0051
25. Fang, X., et al., *Artificial Neural Network (ANN) Assisted Prediction of Transient NOx Emissions from a High-Speed Direct Injection (HSDI) Diesel Engine.* International Journal of Engine Research, 2021. **0**(0). https://doi.org/10.1177/14680874211013254
26. Botticelli, M., et al. *Application of Machine Learning to Gasoline Direct Injection Systems: Towards a Data-Driven Development*. in *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*. 2020. https://doi.org/10.1109/ICMLA51294.2020.00131
27. Badra, J., et al. *Engine Combustion System Optimization Using CFD and Machine Learning: A Methodological Approach*. in *ASME 2019 Internal Combustion Engine Division Fall Technical Conference*. 2019. https://doi.org/10.1115/1.4047978
28. Liu, J., C. Ulishney, and C.E. Dumitrescu. *Predicting the Combustion Phasing of a Natural Gas Spark Ignition Engine Using the K-Nearest Neighbors Algorithm*. in *ASME 2020 International Mechanical Engineering Congress and Exposition*. 2020. https://doi.org/10.1115/IMECE2020-23982
29. Breiman, L., *Random Forests.* Machine Learning, 2001. **45**(1): p. 5-32. https://doi.org/10.1023/A:1010933404324
30. Vu, T.V., et al., *Assessing the impact of clean air action on air quality trends in Beijing using a machine learning technique.* Atmos. Chem. Phys., 2019. **19**(17): p. 11303-11314. https://doi.org/10.5194/acp-19-11303-2019
31. Brokamp, C., et al., *Predicting Daily Urban Fine Particulate Matter Concentrations Using a Random Forest Model.* Environmental Science & Technology, 2018. **52**(7): p. 4173-4179. https://doi.org/10.1021/acs.est.7b05381
32. Liu, J., C. Ulishney, and C.E. Dumitrescu, *Random Forest Machine Learning Model for Predicting Combustion Feedback Information of a Natural Gas Spark Ignition Engine.* Journal of Energy Resources Technology, 2020. **143**(1). https://doi.org/10.1115/1.4047761
33. petrucci, L., Ricci, F., Mariani, F., Cruccolini, V. et al., *Engine Knock Evaluation Using a Machine Learning Approach,* SAE Technical Paper 2020-24-0005, 2020, https://doi.org/10.4271/2020-24-0005
34. Strobl, C., et al., *Conditional variable importance for random forests.* BMC Bioinformatics, 2008. **9**(1): p. 307. https://doi.org/10.1186/1471-2105-9-307
35. Turner, J., Popplewell, A., Patel, R., Johnson, T. et al., *Ultra Boost for Economy: Extending the Limits of Extreme Engine Downsizing*, SAE Int. J. Engines 7(1):387-417, 2014, https://doi.org/10.4271/2014-01-1185
36. Leach, F., et al., *Predicting the particulate matter emissions from spray-guided gasoline direct-injection spark ignition engines.* Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 2016. https://doi.org/10.1177/0954407016657453

Contact Information

Felix Leach
Dept of Engineering Science
University of Oxford
Parks Rd
Oxford
OX1 3PJ
UK

felix.leach@eng.ox.ac.uk

Acknowledgments

The authors acknowledge the Technology Strategy Board (now Innovate UK), the UK's innovation agency, for the partial funding of this work. Consortium members GE Precision Engineering, Jaguar Land Rover, Shell, Lotus Engineering, CD-Adapco, Imperial College London, University of Bath and the University of Leeds have all made various portions of this work possible.

Definitions/Abbreviations

AFR	Air:Fuel ratio
ANN	Artificial Neural Network
BMEP	Brake Mean Effective Pressure
BSFC	Brake Specific Fuel Consumption
BTE	Brake Thermal Efficiency
CoV	Coefficient of Variation
DMS	Differential Mobility Spectrometer
CMD	Count Mean Diameter
EGR	Exhaust Gas Recirculation
EU	European Union
GDI	Gasoline Direct Injection
GSD	Geometric Standard Deviation
ICE	Internal Combustion Engine
IMEP	Indicated Mean Effective Pressure
KLSA	Knock Limited Spark Advance
ML	Machine Learning
MSE	Mean Squared Error
PFI	Port Fuel Injection

PM	Particle Mass	**THC**	Total HydroCarbon
PN	Particle Number	**TWC**	Three Way Catalyst
RF	Random Forest	**WOT**	Wide Open Throttle
SI	Spark Ignition		