Predicting the Emissive Characteristics of an IC Engine Using DNN

Pravin M C¹, Mukilan M², Vishnu Prakash G³, Nithish P⁴, Monish Kanna B⁵, Logesh E⁶.

¹ Assistant Professor, Bannari amman Institute of Technology, Department of Mechatronics, sathyamangalam, Erode, Tamil Nadu, India.
², ³, ⁴, ⁵, ⁶ Students, Bannari amman Institute of Technology, Department of Mechatronics, sathyamangalam, Erode, Tamil Nadu, India.

Abstract. Biodiesel is the new form of automotive fuel that the world is now concerning and several researches are going on for the production of an efficient form of Bio-Diesel because of the fact that Diesel and Petrol are going to be exhaustible in nearly 60 years. In order to produce an efficient fuel, it is inevitable to calculate the emission characteristics concerning the fuel. This project deals with the efficient and intelligent way of analyzing and calculating the engine emission characteristics of Bio-diesel operated IC engines. A Machine Learning based model using TensorFlow library has been developed using python programming for the calculation of emission characteristics such as Carbon-monoxide (CO) and Carbon-dioxide (CO₂) of an IC engine upon injection of Bio-diesel as fuel in different proportions. These investigations and data-sets are considered for a four stroke internal combustion engine. In this Machine Learning model TensorFlow library has been used for the better visualization of the results and error rectification. The results of the developed TensorFlow model are then compared with an existing Fuzzy model for the same application. The results predicted by this model clearly are in good correlation with the actual values which depicts that this method is effective and the total error of the developed model was found to be ±0.02 which is comparatively lower than that of the existing Fuzzy model. Conclusively, the Machine learning model using TensorFlow was found to be the best model for the calculation of engine emission characteristics of Bio-diesel operated IC engines as it offers more visualization tools and better predictive analysis.

Keywords: Biodiesel, Carbon-monoxide, Carbon-dioxide, TensorFlow, TensorFlow.

1. Introduction

Diesel is our primary mode of fuel for almost all automotive vehicles along with Petrol. But the fact is that Diesel and Petrol are going to be extinct in future. Many alternative fuels have been developed to replace fossil fuels such as Diesel and Petrol in order to make the automotive sector keep moving. Out of those alternative fuels developed, in this project Biodiesel is mainly addressed as an efficient and eco-friendly alternative for Diesel. Bio- diesel (Pongamia pinnata) is a form of Diesel obtained from plants and animals and it is typically made by chemically reacting lipids such as soybean oil with alcohol. For concluding Biodiesel as a better alternative to Diesel, we should calculate the IC Engine performance characteristics such as Brake Thermal Efficiency (BTE) produced for different ratios of hydrogen injection and Brake specific energy consumption by considering the emission characteristics such as carbon monoxide and carbon dioxide. It is prepared from edible and non-edible vegetable oils which considerably reduce the CO2 emission when compared to diesel and also it acts as promising fuels in the upcoming decades.
R.S Karthick et al [1] developed a dataset by analyzing IC engines using biodiesel as a fuel. These engine parameters of the Bio-diesel engine should be effective so that it may serve as a better alternative for the Diesel. From the dataset obtained loads and bends are considered as input meanwhile emission of CO and CO2 emission are to be considered as output for this prediction model.

In this project, an intelligent model has been developed from machine learning using TensorFlow for the calculation of the above-mentioned engine parameters and engine efficiency for the usage of Bio-diesel as a fuel. A Machine Learning based model using TensorFlow library has been developed using python programming for the calculation of emission characteristics such as Carbon-monoxide (CO) and Carbon-dioxide (CO2) of an IC engine upon injection of Bio-diesel as fuel in different proportions. This investigation and data-sets are considered for a four-stroke internal combustion engine. In this Machine Learning model TensorFlow library has been used for the better visualization of the results and error rectification.

Visualization of a dataset is one of the important aspects for this proposed intelligent model. Visualization of a dataset using tensor flow is referred from [2] as a base for this prediction model. Fuzzy logic model for the same dataset has been developed in [1]. Result of this intelligent model is compared with a fuzzy intelligent model and the best intelligent model is discovered.

2. Proposed Machine Learning Model

2.1. TensorFlow

TensorFlow is an open source library for machine learning and deep learning developed using python programming, it uses differential calculus and dataflow programming as a base. TensorFlow is developed by Google. The proposed model uses TensorFlow version 2.x to build up the intelligent system for the calculation of efficiencies.

2.2. Implementation

All the coding and the implementation of the proposed machine learning model is done by using Google Collaboratory. Google colaboratory uses an online cloud based Graphical Processing Unit(GPU) and Tensor Processing Unit(TPU) for developing intelligent machine learning and deep learning models. The proposed machine learning model uses a Tensor Processing Unit(TPU) to predict the emission characteristics of various proportions of Biodiesel operated IC engines upon hydrogen injection. Experiments were carried out using four-stroke, single cylinder and stationary compression ignition engines at a constant speed of 1500 rpm with the rated power of 5.2 KW. The emission characteristics such as Carbon monoxide(CO), Carbon dioxide(CO2) are measured with varying blends and loads using AVL gas analyser. Therefore for developing the model, the parameters such as Loads and Blends of Diesel and Biodiesel are considered as inputs and the emission parameters such as CO and CO2 are considered as outputs. The source of data sets were taken experimentally using a four-stroke, single cylinder ignition engine at a constant speed of 1500 rpm with the rated power of 5.2 KW. The data sets were considered as different input and output parameters for developing the machine learning model. The implementation of the algorithm of the model is shown as follows.
2.2.1. *Python Libraries Used*

There are many libraries available in python for machine learning. In order to train the model for prediction of unseen values, structuring of the data, mathematical modeling and visualization of the output are inevitable. These are achieved by the following libraries.

| S.No. | Library used | Function               |
|-------|--------------|------------------------|
| 1.    | tensorflow   | faster numerical computing |
| 2.    | Pathlib      | creating new path      |
| 3.    | matplotlib   | plotting 2D graphs     |
| 4.    | Numpy        | computation            |
| 5.    | Pandas       | data manipulation and analysis |
| 6.    | Seaborn      | statistical data visualization |

The data-sets required for the model are imported from the Google drive location from which it is already stored. The parameters such as Loads in the range of 0 - 100 and blends of Diesel and Biodiesel in the range of 0 - 100 are considered as inputs to the model and emission characteristics of CO and CO\textsubscript{2} are taken as output from the model.

2.2.2. *Visualization Of Inputs*

The input data-sets are visualized using a seaborn python library in order to get an overall idea on how to design the model as shown in the figure 2.1.

In the figure “e\textsubscript{co}” and “e\textsubscript{co2}” represents the emission characteristics of CO and CO\textsubscript{2} respectively.

![Figure 1: Visualization of input data-set](image)
2.2.3. **Splitting The Data**

Sometimes due to the over training of the model leads to over-fit of the training data (over-fit - performing very good on training data set and performing very poor on new unseen data set). To avoid this we are splitting the available data into two sections, one is "training data set" which is then divided into "training data set" and "validation data set" and another one is "test data set". The "validation data set" is a minimal subset of "training data set" in which values are taken randomly from the "training data set" automatically for the verification of the model and error rectification. First we are training the model with the training data set and the trained model is verified and error is rectified by using the "validation data set". Then the performance of the model is tested with unseen "test data set". Finally, from the predicted output we will find the error and correlation of the model with the actual values (true value - predicted value). Normally the train : test data set ratio is 8:2.

2.2.4. **Calculating Statistical Data**

Statistical data such as mean and standard deviation of the datasets provided is found by using the "describe()" function. These statistical data is helpful in normalizing the data in order to achieve the results effectively.

| Inputs    | Mean      | Standard deviation |
|-----------|-----------|--------------------|
| Load      | 50.833333 | 33.868886          |
| Diesel    | 65.416667 | 35.628416          |
| Bio-diesel| 34.583333 | 35.628416          |

2.2.5. **Extracting The Labels (Output Values)**

Emission of CO and CO2 is the output or predictable value in this model. So it is necessary to extract it from the training set and label it separately. Both the CO data and CO2 data are labeled separately and this label is later used to train the model as well as predicting and differentiating the outputs generated from the model.

2.2.6. **Normalizing The Data**

Data handled by the model can vary in range; this can cause huge error while predicting the output. So it is necessary to scale down those data by using normalization techniques. Even Though the input data does not vary, normalization will give better output. Here, we are normalizing the input parameters using their statistical parameters (Mean, standard deviation) in the range of -1 to 1. Normalization is done by using,

\[
\text{Normalized data} = \frac{\text{Data} - \text{Mean}}{\text{Standard deviation}}
\]

2.2.7. **Building The Model**

Here a function is defined for model building. The deep neural network model built here consists of three layers.

- First layer - Input layer
- Second layer - Hidden layer
- Third layer - Output layer
In the first layer there are 64 units which means 64 neurons, the hidden layer also has 64 neurons, on the third layer there is only one neuron because of the model is a single output model. In every layer we are using “relu” activation which means if the input value is less than zero it replaces that value with 0 and the rest will remain the same. For building the model we need to define the input shape by the “input_shape” parameter. In tensorflow there are a lot of optimizers which contain the inbuilt algorithms to reduce the error automatically. Here, we use the RMS prop optimizer (Root Mean Square) which will reduce the Mean Square Error (MSE) and Mean Attribute Error (MAE). For this optimizer the initial learning rate is given as 0.001. Then after setting all the instructions, the model is compiled using “model.compile”. The built model will work commonly for calculating both CO and CO2 emission by calling the model by using different data-sets and parameters for CO and CO2 respectively.

![Neural network representation](image)

**Figure 2: Neural network representation**

2.2.8. **Calling The Model**

Two models are created for CO and CO2 and they are called for execution. Finally the models are tracked for their route in different layers. By using the “summary()” function we can print the performance of every layer for the CO and CO2 model as shown in the table 2.3. It explains the input and output shapes of the layers.
### Table 3: Tracking the CO and CO2 model’s route

| Layers (type) | Output shape  | Parameters |
|---------------|---------------|------------|
| Layer 1 (Dense) | (None, 64)    | 256        |
| Layer 2 (Dense) | (None, 64)    | 4160       |
| Layer 2 (Dense) | (None, 1)     | 65         |
| **Total parameters** |            | **4,481** |
| **Trainable parameters** |            | **4,481** |
| **Non-Trainable parameters** |            | **0**      |

2.2.9. **Training The Model (For Co)**

Training of the model is done between normalized training datasets and training labels of CO and CO2. Here what happens is the training datasets is split into training and validation datasets and the training is done in an iterative manner by using epochs (iterations), then the validation loss of the model is calculated and monitored by using an “early stopping callback()” function which is an inbuilt function in the keras module of tensorflow. The early stopping callback function monitors the validation loss produced in the model and stops the training of the model where the validation loss becomes a constant value upon iterations. Hence overfitting of the model is prevented by using this function.

2.3. **Error Analysis**

2.3.1. **Error Analysis For CO And CO2**

Here we are plotting a graph between epochs (iterations) and MAE (Mean Attribute Error) and also between epochs and MSE (Mean Square Error) by using the values of validation loss and training loss obtained from the early stopping function in the of the training model.

![Figure 3: Epochs vs MAE for CO](image1)

![Figure 4: Epochs vs MSE for CO](image2)
From the plots shown, the straight line represents the training losses and the dotted line represents the validation losses of both the CO and the CO\textsubscript{2} models. It is clearly seen that the validation losses and the training losses of both the CO and CO\textsubscript{2} models nearly coincide and the error gets reduced up to iterations. The total error thus produced from the CO model is 0.034 and for the CO\textsubscript{2} model is 0.074. So, from these graphs it is concluded that the models developed show good coherence with the original values and the error is also minimal and acceptable.

3. Results

3.1. Tolerance Level

The tolerance level for CO\textsubscript{2} and CO emission characteristics are plotted below. The maximum tolerance for CO\textsubscript{2} and CO emission is more or less than ± 0.002 that means that this model is pretty good at predicting the unseen data values.
3.2. Prediction Model

Above mentioned dataset is divided as training dataset and test dataset. Hitherto we have used a training data set and executed our model. In order to depict our intelligent model as the best we have to prove that the original value should be more or less equal to the prediction value. For that a prediction graph is developed as it is shown in the figure 3.3 and 3.4.

![Figure 9: Prediction graph for CO2](image1)
![Figure 10: Prediction graph for CO](image2)

4. Conclusion

For an efficient model the prediction model should lie in the center line, if the deflection is more its efficiency is less and its error is more. In our model its deflection is very low so its efficiency is high. Its efficiency is approximately 99.5 and its error is 0.05.

Error of this intelligent model using TensorFlow is calculated by using a formula described below.

\[
\text{Error 1} = \text{CO prediction} - \text{CO label} \\
\text{Error 2} = \text{CO2 prediction} - \text{CO2 label}
\]

After calculating Error 1 and Error 2, overall error for this intelligent model is calculated. That overall error is approximately 0.05.

When compared to a fuzzy intelligent model its error is 0.91 [1]. By comparing these two values it is clear that the efficiency of intelligent models using TensorFlow is better than the efficiency of intelligent models using fuzzy logic.

From the above described results it is clear that, Comparing the results obtained from the two models, it is found that both the models seem to be equally effective and intelligent. Bio-diesel as an IC engine fuel offers better efficiency and produces good engine knocks upon hydrogen injection. Hence, concludingly Biodiesel can be used as a better alternative for Diesel.
REFERENCE

[1] Karrthik, R.S., Baskaran, S. and Raghunath M., (2019), “Development of intelligent model to predict the characteristics of biodiesel operated CI engine with hydrogen injection”, Advances in Computational Design, Vol. 4, No. 4 (2019) 367-379 DOI:https://doi.org/10.12969/acd.2019.4.4.367

[2] Wongsuphasawat, K., Smilkov, D., Wexler, J., Wilson, J., Mane, D., Fritz, D., ... Wattenberg, M. (2018). Visualizing Dataflow Graphs of Deep Learning Models in TensorFlow. IEEE Transactions on Visualization and Computer Graphics, 24(1), 1-12. doi:10.1109/tvcg.2017.2744878

[3] Berber, A., Tinkir, M., Gultekin, S. and Celikten, I. (2011), “Prediction of a diesel engine characteristics by using different modeling techniques”, J. Phys. Sci., 16, 3979-3992.

[4] Oliveira, A., Santos, E., Botelho, G., Valente, O. and Sodre, J. (2013), “Hydrogen electronic injection system for a diesel power generator”, J. Green Energy, 3(19), 7986-799.

[5] https://doi.org/10.1016/j.ijhydene.2013.04.118.

[6] Parlak, A., Islamoglu, Y., Yasar, H. and Egrisogut, A. (2006), “Application of artificial neural networks to predict specific fuel consumption and exhaust temperature for a diesel engine”, J. Appl. Thermal Eng., 26(8-9), 824-828. https://doi.org/10.1016/j.applthermaleng.2005.10.006.

[7] Rai, A., Satheesh Kumar, N., Pai, S. and Rao, S. (2012), “Fuzzy logic based prediction of performance and emission parameters of a LPG-Diesel dual fuel engine”, J. Appl. Thermal Eng., 38, 280-292.

[8] https://doi.org/10.1016/j.proeng.2012.06.036.

[9] Sakthivel, G., Ilangkumaran, M., Syamkumar, U., Vasudevan, M., Venkatesh, S. and Viswanathan, B. (2013), “Development of fuzzy logic model to predict the engine performance of fish oil biodiesel with diethyl ether”. J. Ambient Energy, 47, 1-33. https://doi.org/10.1080/01430750.2013.829786.

[10] Sakthivel, G. and Ilangkumaran, M. (2014), “Neural network (ANN) approach to predict the engine performance of fish oil biodiesel with diethyl ether using back propagation algorithm”, J. Ambient Energy, 78, 9-40.

[11] Saravanan, N. and Nagarajan, G. (2009), “Performance and emission study in manifold hydrogen injection with diesel as an ignition source for different start of injection”, J. Renewable Energy, 34(1), 328-334. https://doi.org/10.1016/j.renene.2008.04.023.

[12] https://doi.org/10.1016/j.ijhydene.2007.12.065.

[13] Saravanan, N. and Nagarajan, G. (2009), “Performance and emission study in manifold hydrogen injection with diesel as an ignition source for different start of injection”, J. Renewable Energy, 34(1), 328-334. https://doi.org/10.1016/j.renene.2008.04.023.

[14] Tasdemir, S., Saritas, I., Ciniviz, M. and Allahverdi, N. (2011), “Artificial neural network and fuzzy expert system comparison for prediction of performance and emission parameters on a gasoline engine”, J. Expert

[15] https://doi.org/10.1016/j.ijhydene.2007.12.065.

[16] http://dx.doi.org/10.1016/j.renene.2008.04.023.

[17] https://doi.org/10.1016/j.ijhydene.2007.12.065.

[18] https://doi.org/10.1080/01430750.2013.829786.

[19] https://doi.org/10.1016/j.proeng.2012.06.036.

[20] https://doi.org/10.1016/j.proeng.2012.06.036.

[21] https://doi.org/10.1016/j.proeng.2012.06.036.

[22] https://doi.org/10.1016/j.proeng.2012.06.036.

[23] https://doi.org/10.1016/j.proeng.2012.06.036.

[24] https://doi.org/10.1016/j.proeng.2012.06.036.

[25] https://doi.org/10.1080/01430750.2013.829786.

[26] https://doi.org/10.1016/j.proeng.2012.06.036.

[27] https://doi.org/10.1016/j.proeng.2012.06.036.

[28] https://doi.org/10.1016/j.proeng.2012.06.036.

[29] https://doi.org/10.1016/j.proeng.2012.06.036.

[30] https://doi.org/10.1016/j.proeng.2012.06.036.

[31] https://doi.org/10.1016/j.proeng.2012.06.036.

[32] https://doi.org/10.1016/j.proeng.2012.06.036.

[33] https://doi.org/10.1016/j.proeng.2012.06.036.

[34] https://doi.org/10.1016/j.proeng.2012.06.036.

[35] https://doi.org/10.1016/j.proeng.2012.06.036.

[36] https://doi.org/10.1016/j.proeng.2012.06.036.

[37] https://doi.org/10.1016/j.proeng.2012.06.036.

[38] https://doi.org/10.1016/j.proeng.2012.06.036.

[39] https://doi.org/10.1016/j.proeng.2012.06.036.

[40] https://doi.org/10.1016/j.proeng.2012.06.036.

[41] https://doi.org/10.1016/j.proeng.2012.06.036.