SCIENTIFIC OASIS

Decision Making: Applications in Management and Engineering

ISSN: 2560-6018, eISSN: 2620-0104

THE CREATION OF A DIESEL ENGINE USING GENETIC ALGORITHM PREDICTION MODELS AND MACHINE LEARNING TO VERIFY FUEL CONSUMPTION, EMISSIONS, AND HEAT TRANSFER

Kokku Jayakrishna[1], Muttaiah Modugu[2]

[1] PG Student, Mechanical Engineering, Sree Vahini College of Engineering & Technology, Tiruvuru, A.P, India

[2] Associate Professor, Mechanical Engineering, Sree Vahini College of Engineering & Technology, Tiruvuru, A.P, India

ABSTRACT

The advancement of diesel engine design has become increasingly reliant on computational techniques to optimize performance while minimizing fuel consumption, emissions, and heat transfer losses. This research explores the integration of genetic algorithm (GA) prediction models and machine learning (ML) techniques to enhance diesel engine efficiency. The study employs GA to generate optimal engine configurations by simulating various design parameters, selecting the most efficient solutions based on performance criteria. Machine learning models are then trained on experimental and simulated data to validate the accuracy of fuel consumption, emission levels, and thermal characteristics. The combined approach enables real-time optimization by iteratively refining engine parameters through adaptive learning. By leveraging data-driven methodologies, this study provides a novel framework for designing high-efficiency diesel engines with lower environmental impact. The results demonstrate that ML-assisted GA optimization significantly improves fuel economy and emission compliance while enhancing thermal performance. This research contributes to the on-going efforts in sustainable engine development, offering a predictive and adaptable design methodology for future diesel engines. Traditional methods rely on empirical correlations or computational fluid dynamics (CFD) simulations, which are computationally expensive and time-consuming. This thesis explores the application of machine learning (ML) techniques to predict and analyze heat transfer in IC engines. By leveraging historical engine performance data and modern ML algorithms, we aim to provide an efficient, accurate, and scalable solution for IC engine heat transfer analysis. The findings demonstrate the potential of ML approaches to enhance the efficiency of engine development cycles.

Key Words: Design, Analysis, Machine Learning, Genetic Algorithm, Computational Fluid Dynamics, efficiency, IC engines, Heat Transfer.

I.INTRODUCTION

The advancement of diesel engine technology is crucial for improving fuel efficiency, reducing emissions, and enhancing thermal performance. Traditional design and optimization methods often rely on experimental testing and computational simulations, which can be time-consuming and costly [1-4]. However, with the emergence of artificial intelligence (AI) techniques, particularly genetic algorithms (GAs) and machine learning (ML), a more efficient and intelligent approach to diesel engine design has become possible. [4,6]Genetic algorithms, inspired by the principles of natural selection, are widely used for solving complex optimization problems. By applying GAs, diesel engine designs can be iteratively improved to achieve optimal performance in terms of fuel consumption, emissions, and heat transfer characteristics. These algorithms allow engineers to explore a vast design space, identifying configurations that balance efficiency and environmental impact. Furthermore, machine learning models can be integrated to predict engine behaviour based on historical data, simulations, and experimental results [7-9]. By training ML models with extensive datasets, engineers can accurately estimate fuel consumption, emission levels, and heat dissipation, enabling faster and more precise design iterations. The synergy between GAs and ML enhances the optimization process by identifying patterns, refining parameters, and ensuring robust verification of engine performance. This study explores the application of genetic algorithm prediction models and machine learning techniques in the creation of a next-generation diesel engine [10]. By leveraging AI-driven optimization and data-driven validation, this approach aims to revolutionize diesel engine design, paving the way for more sustainable and energy-efficient transportation solutions [11, 12].

1.1 Role of Machine Learning in IC Engine Heat Transfer Analysis

a) Surrogate Modelling

- ML serves as a surrogate model to approximate the results of computationally expensive CFD simulations.
- Surrogate models predict heat transfer characteristics like temperature distribution, heat flux, and thermal stresses based on input parameters (e.g., engine speed, combustion temperature, coolant flow rate).

b) Pattern Recognition

- ML identifies patterns and trends in heat transfer data, such as correlations between combustion parameters and heat losses.
- Insights from ML help in diagnosing inefficiencies and thermal anomalies in engine systems.

c) Optimization

- ML algorithms optimize engine design by iteratively learning the impact of design changes on heat transfer performance.
- Examples include optimizing cooling channel geometries or combustion chamber designs.

d) Real-Time Prediction

- ML models predict heat transfer behavior in real-time, enabling adaptive thermal management systems.
- These models are valuable for applications requiring fast responses, such as engine control units (ECUs).

1.2. Machine Learning Techniques Used

a) Regression Models

- Linear Regression, Polynomial Regression: Model simple relationships between input parameters and heat transfer outputs.
- Support Vector Regression (SVR): Handles non-linear relationships in heat transfer phenomena.
- Gaussian Process Regression: Provides uncertainty quantification, useful for surrogate modeling.

b) Neural Networks

- Deep Neural Networks (DNNs): Capture complex, non-linear relationships in heat transfer data.
- Convolutional Neural Networks (CNNs): Used for analyzing spatial heat transfer patterns (e.g., temperature maps).
- Recurrent Neural Networks (RNNs): Handle temporal data in transient heat transfer scenarios.

c) Ensemble Methods

- Random Forests, Gradient Boosting (e.g., XGBoost): Offer high accuracy and robustness in predicting heat transfer metrics.
- Useful for feature importance analysis to identify dominant factors affecting heat transfer.

d) Unsupervised Learning

- Clustering (e.g., k-Means): Identifies distinct regimes of heat transfer behavior.
- Principal Component Analysis (PCA): Reduces the dimensionality of complex datasets while retaining essential features.

e) Reinforcement Learning

- Optimizes engine operation and thermal management systems by learning optimal strategies through interaction with the environment.

1.3. Applications of ML in IC Engine Heat Transfer Analysis

a) Surrogate Models for CFD

- ML models trained on CFD or experimental data predict temperature fields and heat flux distributions.
- Surrogate models significantly reduce the time required for iterative simulations.

b) Thermal Management Systems

- Real-time prediction of engine component temperatures for adaptive control of cooling and lubrication systems.
- Minimizes overheating and enhances efficiency.

c) Combustion Optimization

- ML analyzes the impact of fuel properties, injection timing, and air-fuel ratios on heat transfer.
- Aids in designing combustion systems with minimal heat losses.

d) Material Performance Analysis

- ML predicts thermal stresses and deformation in engine components under varying operating conditions.
- Enables material selection and design improvements.

e) e. Emission Control

- ML links heat transfer characteristics to emission levels, guiding strategies to reduce NOx and particulate matter.

II. METHODOLOGY

2.1 Data Collection

The dataset used in this study includes:

- Engine operating parameters (e.g., RPM, load, air-fuel ratio).
- Geometrical parameters (e.g., cylinder bore, stroke length).
- Heat transfer measurements obtained from experiments and simulations.

2.2 Data Pre-processing

- Handling missing values through interpolation.
- Feature scaling using standardization.
- Feature selection to identify key parameters influencing heat transfer.

2.3 Model Development

The following ML algorithms were implemented:

- Linear Regression
- Random Forest Regression
- Gradient Boosting Machines (GBM)
- Artificial Neural Networks (ANN)

2.4 ML Methodologies

This subsection describes the pre-processing of diesel engine data sets. Firstly, a variety of datasets including engine speed, torque EGR rate and EGR gas temperature were acquired for different operating conditions, thus, equalizing the inputs and outputs data for modelling. Then, each dataset was scaled by using defining normalization formulation below. Where, x; l and r denote the input vector, average and standard deviation of input datasets. In this study, four different ML regression models (the decision tree regression, support vector machine regression, Gaussian processes regression and ensemble ML regression were implemented and tested to predict the NOx and BSFC of the diesel engine based on engine speed, torque, EGR rate, exhaust gas temperature entering the intake manifold. ML regression models are very sensitive to hyper-parameter changes of the models. After the model is chosen to define relationship between input and output of the EGR cooling system, it is necessary to tune its parameters to obtain optimal prediction performance [6]. There are different types and numbers of parameters for each ML model. Different combinations of hyper parameters can be defined by using an optimization technique that minimizes the error between the model output and the measurement results for a considered model type. Therefore, obtaining successful results from the designed ML models depends on the correct determination of hyper parameters. In this study, the grid search algorithm was used for the purpose of determining hyper parameters of ML models to design the optimal model.

Decision tree regression (DTR): Decision tree ML algorithms used for like classification or regression application were developed by dividing the dataset into smaller and smaller subsets. This ML method, in which both numerical and categorical data can be processed, consists of decision and leaf nodes to predict the system response. The hyper parameter of DTR is minimum leaf size.

Support vector regression (SVR): SVM regression was developed by Cortes and Vapnik. 24 The inputs of the proposed system are denoted as x = ½N Eng $_{Spd}$; S Eng $_{Trq}$; h EGR $_{Rate}$; TExh $_{Temp}$ while the outputs are y = NOx; BSFC Š. The transformation function y = **φ (x)** is defined between y and x. This estimated function represented as follows:

$$\psi(x) = \omega \cdot \varphi(x) + b$$

The C coefficient is defined as the regularization parameter. The nm and nmare slack variables. Equation can be defined with Lagrange multipliers considering the optimization constraints below in the explicit form.

$$\psi(x) = \sum_{m} (\alpha_m - \alpha_m^*)\kappa(x, \hat{x}) + b$$

Gaussian process regression (GPR): In this research study, GPR was used as another regression method that determines the relationship between diesel engine parameters. GPR can be defined for a given training dataset that contains input and output data. y indicates the Gaussian distribution of the input x. A GP is a collection of stochastic variables that is determined with mean and covariance matrix as follows

$$\bar{y}^* = \mu(x^*) + \kappa(x^*, x)\kappa_y^{-1}(y - \mu(x))$$

Ensemble ML methods: Bootstrap aggregation (bagging type) and boosting are known as the ensemble ML method. This technique was used as the ML analysis in this study. The predictors are randomly created and selected according to their accuracy success rate in the ensemble-based random forest (RF) algorithm. Using ensemble learning methods to model complex dynamic systems such as diesel engines are preferred as they perform better than global machine learning models which create a single model from data set. Random decision trees are created in the RF technique and are then selected and combined according to their success rate. However, although the success rate of the boosting ensemble method is high, the bagging ensemble method overcomes the over-fitting problem and reduces the variance of signal in data-based modelling.

GA and its integration with ML model: The GA is an evolutionary algorithm and widely used in many engineering optimization problems. The advantage of GA optimization is parallel operating capacity to solve complex problems at the cost of low computational efficiency. The

algorithm works iteratively to minimize cost function. An appropriate selection of fitness functions depends on the determination of GA parameters such as population size and genetic operator rates. The population of individuals are candidate solutions for a problem that is demonstrated by linear binary string states known as chromosomes. In GA, the chromosomes start randomly and will be evaluated with the fitness function. Thereafter, the next generation of chromosomes is created by implementing genetic operators such as mutation and crossover on the chromosomes. This GA process is iteratively repeated until the end criterion is fulfilled. In this study, the goal is to find an appropriate values combination for the proposed EGR cooling system variables so that they could be utilized as an objective function expressed by the ML model for achieving optimal EGR exhaust gas temperature and EGR rate values. The objective function obtained by the ML model for the proposed electromechanical EGR cooling system design is expressed as follows:

$$\text{minimize } f_{NOx,\ BSFC} = \sum_{i=1}^{N} ES_i, ET_i,\ EGRRate_i,\ EGRTemp_i$$

Here, ES; ET; EGR_{Rate} and EGR_{Temp} represent engine speed, torque, EGR gas rate and exhaust gas temperature entering the intake manifold. In addition, this optimization model is subjected to the following constraints:

Min ES < ES < max ES

Min ET < ET < max ET

0 < EGR Rate < 15

55 ^{0}C < EGR Temp < 130 ^{0}C

The above formula as a multi-objective optimization problem for the block diagram of the designed system with GA is shown in Figure --. As presented in this figure, after the ML model characterization was realized, as much data on the proposed EGR cooling system as possible to cover majority of different engine operating conditions was collected. Then, the GA structure was utilized to estimate the optimal EGR gas temperature and EGR rate by the obtained ML model for NEDC and WLTP drive cycles.

III. RESULTS AND DISCUSSION

3.1 Machine Learning: Importing and Analysing the Data

Finding and loading the dataset into the Python interpreter is the process of data import. All further investigation is predicated on this first step. Excel formats were the source of the datasets. Making the data accessible for analysis and preparation is the aim. Importing the required Python libraries, including Pandas, Numbly, Seaborne, Matplotlib, and Warnings, is the first and most important step in order to analyze the dataset. For ignoring any warnings that might appear during execution, the Warnings library is especially helpful. We import the experimental dataset using the Pandas library. We can tell that the dataset has 13 characteristics and 361 rows by looking at its shape.

Because manually recorded experimental data may contain errors, pre-processing the complete dataset is necessary to guarantee correct data for modelling purposes.

The purpose of this pre-processing phase is to align the data and prepare them for processing by machine learning algorithms. We provide the groundwork for additional analysis and modelling activities by importing the necessary Python modules and preparing the dataset, which includes cleaning and converting the data.

3.2 Modelling of system using machine learning

Microsoft Excel is used to store data gathered during physical experiments. Carbon monoxide, hydrocarbon, smoke, nitrogen oxide, and brake thermal efficiency are thought of as reaction factors, while load, biogas flow rate, and temperature are thought of as predictor variables. Approximately 325 independent experiments are conducted with different input parameters, and the outcomes are documented. A preliminary examination of the data is shown. After pre-processing and random forest training, the data is used to determine the ideal input feature values. The methods taken to arrive at the ideal value are shown in Figure-. Since all experimentation values are manually recorded, they are subject to error. The complete dataset is pre-processed to align the values appropriate for machine learning algorithms' processing before modelling begins.

Table 1. Analysis of dataset.

	Load	**Intake Temperature**	**Nox**	**BT (%)**
Mean	12.5	68.4	432	26.4
Std	5.8	24.5	340	7.2
Min	4.2	35	10	8.4
25%	8.8	35	134	20
50%	12.6	60	335	30
75%	16.3	80	712	32.2
Max	20	100	1260	35

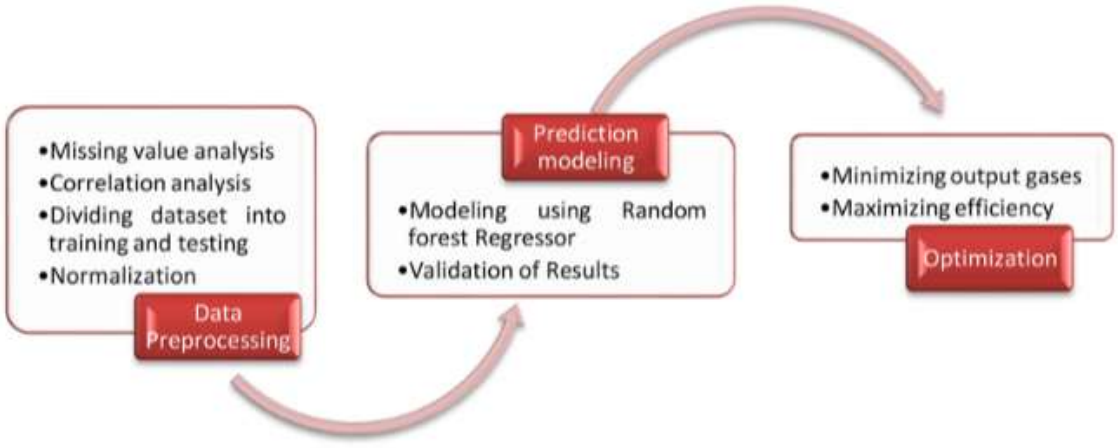

Figure. 1. Modelling methodology.

Each aspect may have a varied range of numbers, and the values are reported on various scales. Differences in the scales among the input variables could make it more difficult to comprehend the problem being modelled are outcomes. Since algorithms typically deal with numerical values, every piece of data is analyzed quantitatively, which will have an effect on the modelling. There will have a significant effect on the outcomes when each feature's range varies. Therefore, min–max normalization is used to normalize the data to a range between 0 and 1 in order to prevent such an anomaly. To lessen anomaly brought on by different numeric ranges, the training dataset is standardized. The input parameter-corresponding snapshot of normalized values.

The following activities were performed:

Pre-processing tasks like data cleaning, normalization, and feature engineering. Data contain outliers and duplicated values. These issues are addressed through techniques such as imputation and outlier removal. Normalization ensures that features are on similar scales [4].

Check for missing values: We looked for any missing values in the dataset. One missing value, found at the zeroth index, was found to be shared by all the features. A snippet of code is displayed in Figure 13 as a result of the complete row containing the null value being removed. As a result, the dataset's form shifted to 13 columns and 360 rows.

import pandas as pd
# Sample DataFrame with NaN values
data = {'A': [1, 2, None, 4], 'B': [5, None, 7, 8], 'C': [None, 11, 12, 13]}
df = pd.DataFrame(data)
# Drop rows with any NaN values
df_cleaned = df.dropna()
print(df_cleaned)4.1 Model Performance

```python
import pandas as pd

# Sample DataFrame with null values
data = {'A': [1, 2, None, 4], 'B': [None, 2, 3, 4]}
df = pd.DataFrame(data)

# Drop rows with any null values
df_cleaned = df.dropna()

print(df_cleaned)
```

Figure.2. Snippet of code for dropping the null value.

- ANN achieved the highest accuracy, with an R^2 score of 0.95.

- Random Forest showed robust performance with minimal overfitting.
- Linear models underperformed due to the non-linear nature of heat transfer phenomena.
- dropna() removes any row containing at least one NaN value.
- If you want to **drop columns** instead of rows, use df.dropna(axis=1).
- To **fill missing values instead of dropping**, use df.fillna(value).

3.3 Here's a Python snippet for removing outliers using the Interquartile Range (IQR) method:

```python
import pandas as pd
import numpy as np

# Sample DataFrame
data = {'A': [10, 12, 14, 100, 15, 16, 18, 110, 19, 21],
        'B': [5, 7, 9, 50, 10, 11, 13, 55, 14, 16]}

df = pd.DataFrame(data)

# Function to remove outliers using IQR
def remove_outliers_iqr(df):
    Q1 = df.quantile(0.25)  # First quartile (25%)
    Q3 = df.quantile(0.75)  # Third quartile (75%)
    IQR = Q3 - Q1  # Interquartile range
    # Filtering values within the IQR range
    df_filtered = df[~((df < (Q1 - 1.5 * IQR)) | (df > (Q3 + 1.5 * IQR))).any(axis=1)]
    return df_filtered

# Remove outliers
df_cleaned = remove_outliers_iqr(df)

print(df_cleaned)
```

Figure.3. Snippet of code for removing the outliers by using IQR method.

3.4 Results of the ML models

As a result of the research conducted in this study, it can be seen that the increase in EGR gas temperature is closely related to engine load and speed. The effect of the ratio of recirculating gases and temperature values is even more pronounced at low engine speeds. Low engine speed and exhaust gas temperature cause less thermal throttling. This situation causes more O2 addition

in EGR gases. Thus, the improvement in volumetric efficiency allows high EGR flow at low engine speeds. However, the increase in the EGR gas temperature at high engine speeds causes the O2 concentration and in-cylinder pressure to decrease. As a result, high EGR rates at low engine speeds have positive effects on NOx and BSFC. At high engine speeds, it can be said that lower EGR rates are effective in terms of NOx and BSFC. In the modelling process of this study, testing of all models was carried out by using k-fold cross-validation to ensure performance predictive performance and stability of the designed ML models with 10 folds. One tenth of the dataset was used in the testing process of models to ensure a statistically more accurate estimation in the model stability. As a result, error rates were calculated by the mean of each iteration. The initial ML method implemented was the decision tree algorithm. In this algorithm, the tuning parameter was minimum leaf size where it was varied in 1–35 range. The optimal leaf size parameters were determined to be 5 and 7 by using MSE for NOx and BSFC, respectively.

Table.2. Error analysis of the designed ML models

	ML models	RMSE	R^2	MSE	MAE
	DTR	33.443	0.77	1118.5	24.28
	SVR	29.674	0.82	880.52	23.353
	GPR	**18.602**	**0.93**	**346.05**	**11.921**
NOx	ENSEMBLE	31.032	0.81	962.98	22.12
	DTR	39.601	0.96	1568.2	21.162
	SVR	159.03	0.36	25290	101.56
	GPR	**6.643**	**0.99**	**44.129**	**4.2645**
BSFC	ENSEMBLE	40.7	0.96	1656.5	18.762

The SVR model was designed as the second ML method. To express the effect of hyper-parameters on the prediction performance of the SVR model, the grid search algorithm was employed to tune the optimizable hyper parameters. The Gaussian, linear, quadratic parameters were varied as kernel functions. However, many SVR models depending on the ε; C, r and g parameters in equations were also created. Here, the C coefficient was varied from 0.001 to 1000 and ε from 0.08 to 5 for both NOx and BSFC models. After different SVR models were created, the one with the lowest MSE value was chosen as the optimized model. The optimal SVR parameters were determined as C_{NOx} = 4:022 and ε_{BSFC} = 0; 128, ε_{NOx} = 10 and CBSFC = 46:41 quadratic and linear as kernel function for NOx and BSFC, respectively. The Gaussian processes method was the third attempt of this study. Similarly, the hyper parameter estimation of GPR significantly affected predictive performance of the model. Specifically, the chosen kernel hyper parameter was crucial because the Gaussian process consists of random variables. Therefore, the kernels in equations played a crucial role in the variance matrix mean. In the designing of the GPR model, squared exponential, exponential and rational quadratic were used as the kernel function for the mean and covariance functions were specified by hyper parameters. The variance function contains hyper-parameters which are kernel scale, basis function, and r. The optimal GPR parameters were determined as kernel scale 14:606 and 677:97 for NOx and BSFC models and σ_{NOx} = 0:19 and σ_{BSFC} = 1:1249; respectively. Finally, ensemble learning methods were applied by using bagging in this study [6]. The random forest method ensured that each decision tree in the ensemble was selected randomly, and then bagging was applied. Random forests are a combination of independently sampled random vectors and improve accuracy of bagged trees. The different ML models only the GPR model has higher prediction performance of both NOx and BSFC than

other ML models. The error evaluation criteria for MSE, RMSE, R^2 and MAE are shown as follows:

$$MAE\left(y_{Nox,BSFC}, \hat{y}_{Nox,BSFC}\right) = \frac{\sum_{k=1}^{N} y_{kNOx,BSFC} - \hat{y}_{kNOx,BSFC}}{N}$$

$$MSE\left(y_{Nox,BSFC}, \hat{y}_{Nox,BSFC}\right) = \frac{1}{N}\sum_{k=1}^{N}\left(y_{kNOx,BSFC} - \hat{y}_{kNOx,BSFC}\right)^2$$

$$RMSE\left(y_{Nox,\ BSFC}, \hat{y}_{Nox,\ BSFC}\right) = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left(y_{kNOx,BSFC} - \hat{y}_{kNOx,BSFC}\right)^2}$$

$$R^2\left(y_{Nox,\ BSFC}, \hat{y}_{Nox,\ BSFC}\right) = 1 - \left(\frac{\sum_{k=1}^{N}\left(y_{kNOx,BSFC} - \hat{y}_{kNOx,BSFC}\right)^2}{\sum_{k=1}^{N}\left(y_{kNOx,BSFC} - \bar{y}_{NOx,BSFC}\right)^2}\right)$$

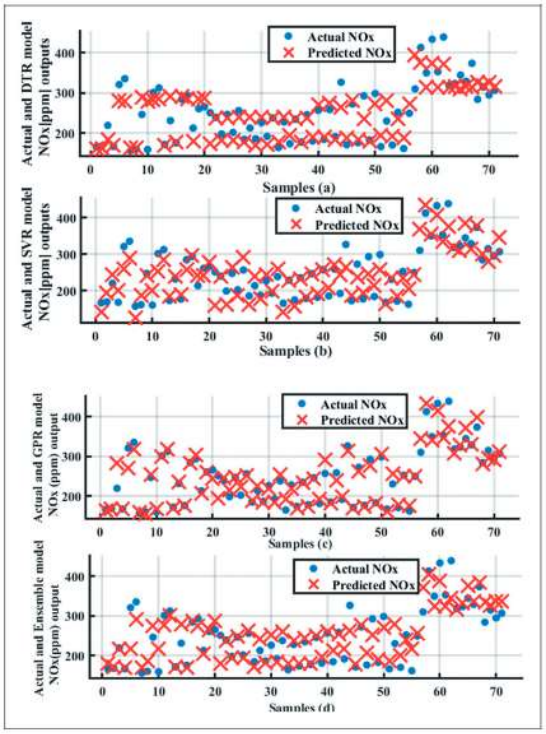

Figure.4. Comparison of results of the actual and ML model NOx outputs.

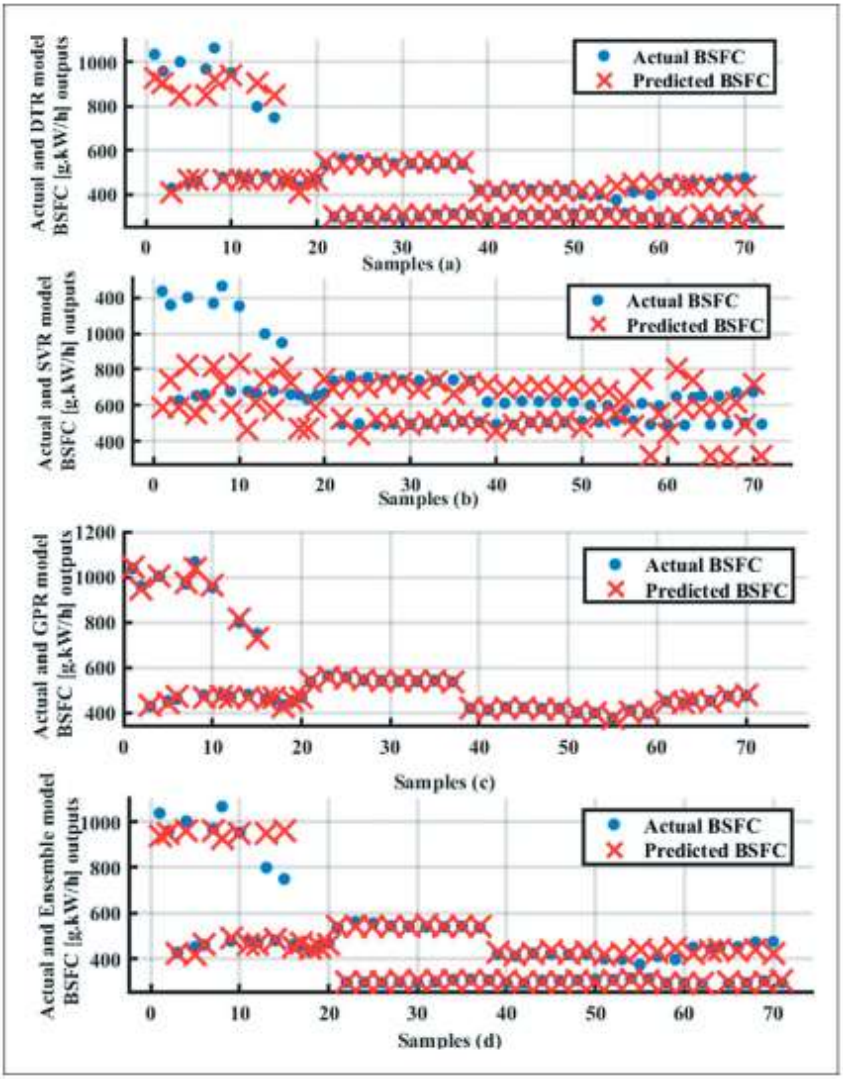

Figure.5. Comparison of results of the actual and ML model BSFC outputs.

Where the actual values are represented by y, and the mean and forecast values are shown by $\hat{Y}$ and $\tilde{Y}$, respectively. The results of the DTR, SVR, GPR, and ensemble model as well as the real NOx and BSFC data, respectively. In Figure 3, the red crosses represent the model results, while the blue circles represent the real NOx data. Because of the in-cylinder combustion instability that occurs when the engine runs at low speed and load settings, NOx and BSFC output variables may vary from other operating conditions on the above graphs.

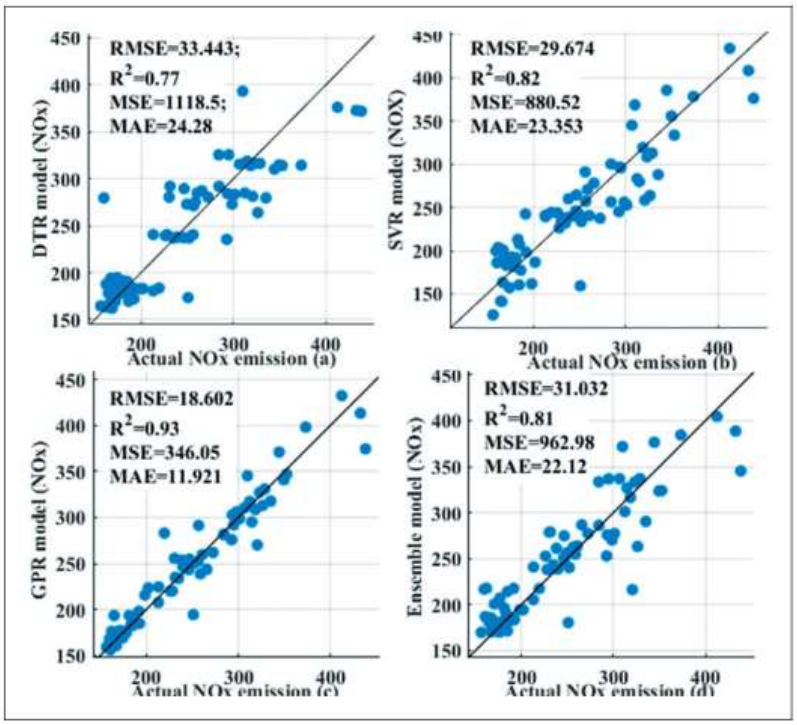

Figure.6. Actual values of NOx emission versus values predicted using developed ML models.

As a result, the prediction success rate of these models falls between 1 and 20. For both NOx emission and BSFC values of the diesel engine, the GPR model's performance accuracy results are superior to those of the other models. In particular, BSFC differs significantly from other ML models in terms of MSE and other evaluation metrics. The suggested connections between the chosen ML models and the measured NOx and BSFC respectively. Based on coefficient of determination (R^2), the findings of the created GPR model outputs are 0.93 and 0.98 for both NOx and BSFC, respectively. Consequently, the GPR model outperforms the DTR, SVR, and ensemble ML models in terms of NOx emission and BSFC.

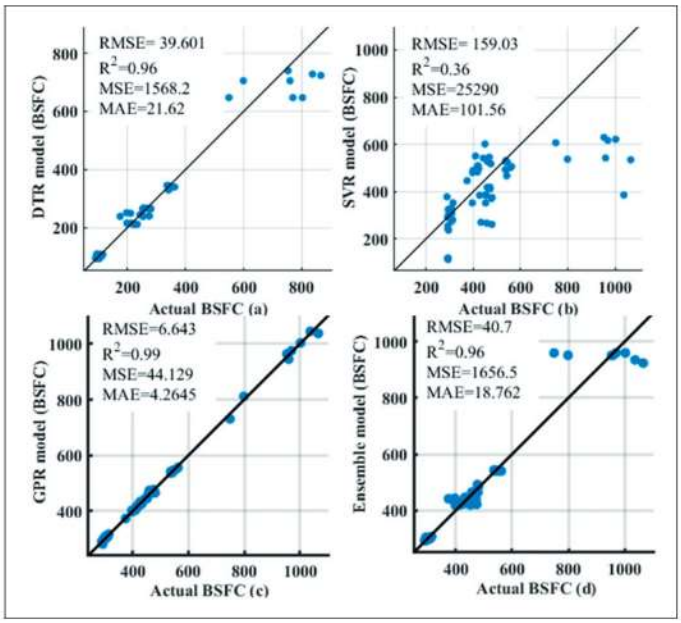

Figure.7. Actual values of BSFC values versus values predicted by developed ML models.

3.5 Python Code for Heat Transfer Prediction & Visualization:

Here's a **Python code snippet** that uses **machine learning to analyze and visualize heat transfer in IC engines**. This example demonstrates **heat transfer prediction using Random Forest Regression and graphical representations** (heatmaps, surface plots, and line graphs).

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from mpl_toolkits.mplot3d import Axes3D
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, r2_score

# 📌 Step 1: Load Sample Data (Simulated IC Engine Heat Transfer Dataset)
np.random.seed(42)
data_size = 200
engine_speed = np.random.uniform(1000, 4000, data_size)  # RPM
engine_load = np.random.uniform(10, 100, data_size)  # Percentage Load
coolant_temp = np.random.uniform(70, 120, data_size)  # Coolant Temperature (°C)
heat_flux = 50 + 0.02 * engine_speed + 0.5 * engine_load - 0.3 * coolant_temp + np.random.nor

df = pd.DataFrame({'Engine_Speed': engine_speed, 'Engine_Load': engine_load,
                   'Coolant_Temp': coolant_temp, 'Heat_Flux': heat_flux})
```

```python
# 📌 Step 2: Split Data into Train & Test Sets
X = df[['Engine_Speed', 'Engine_Load', 'Coolant_Temp']]
y = df['Heat_Flux']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 📌 Step 3: Train a Machine Learning Model (Random Forest Regression)
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# 📌 Step 4: Make Predictions and Evaluate Model
y_pred = model.predict(X_test)
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f"Model Performance: MAE = {mae:.2f}, R² = {r2:.2f}")

# 📌 Step 5: Heatmap Visualization of Correlations
plt.figure(figsize=(8, 6))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm', linewidths=0.5)
plt.title("Feature Correlation Heatmap")
plt.show()

# 📌 Step 6: 3D Surface Plot (Engine Speed vs Load vs Heat Flux)
fig = plt.figure(figsize=(10, 6))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(df['Engine_Speed'], df['Engine_Load'], df['Heat_Flux'], c=df['Heat_Flux'], cmap='
ax.set_xlabel("Engine Speed (RPM)")
ax.set_ylabel("Engine Load (%)")
ax.set_zlabel("Heat Flux (W/cm²)")
ax.set_title("3D Heat Transfer Analysis")
plt.show()

# 📌 Step 7: Line Graph of Actual vs Predicted Heat Flux
plt.figure(figsize=(8, 5))
plt.plot(y_test.values, label="Actual Heat Flux", marker='o', linestyle='dashed')
plt.plot(y_pred, label="Predicted Heat Flux", marker='x')
plt.xlabel("Test Sample")
plt.ylabel("Heat Flux (W/cm²)")
plt.title("Heat Transfer Prediction: Actual vs Predicted")
plt.legend()
plt.show()
```

Figure.8. Python Code for Heat Transfer Prediction.

4.6 Extension of Variable Domain:

A Machine Learning-Grid Gradient Ascent (ML-GGA) technique was created in this study to maximize internal combustion engine performance. ML provides a means of converting intricate physical processes found in combustion engines into condensed informational operations. A recently constructed Machine Learning-Genetic Algorithm (ML-GA) was compared to the developed ML-GGA model. To increase the precision and resilience of the optimization process, in-depth analyses of the optimization solver parameters and variable limit extension were carried out in the current ML-GGA model [3]. Here are thorough explanations of the various steps, optimization tools, and requirements that must be met for a successful output. When compared to the optimal result of a comprehensive computational fluid dynamics (CFD) guided system optimization, the ML-GGA approach produced improvements in the merit function of >2%. Engine CFD simulations were used to validate the predictions made by the ML-GGA technique. When compared to conventional methods, this study shows how ML-GGA may drastically cut down on the amount of time required to solve optimization issues without sacrificing accuracy. Some design parameters, including the number of nozzles (nNoz), start of injection (SOI), total nozzle area (TNA), nozzle angle (NozAngle), swirl ratio (SR), exhaust gas recirculation (EGR), injection pressure (Pinj), and swirl ratio (SR), can exhibit notable variations. The only variables that are comparable between runs are the temperature (Tivc) and the optimum intake valve closing pressure (Pivc). Using Rmalschains to find the best collection of optimum solutions can be a laborious process that calls for specialized knowledge. In order to address this, a novel approach is used in this work by employing a distinct GA, GGA, which is straightforward to install and lacks black-boxed information despite being traditional and forward.

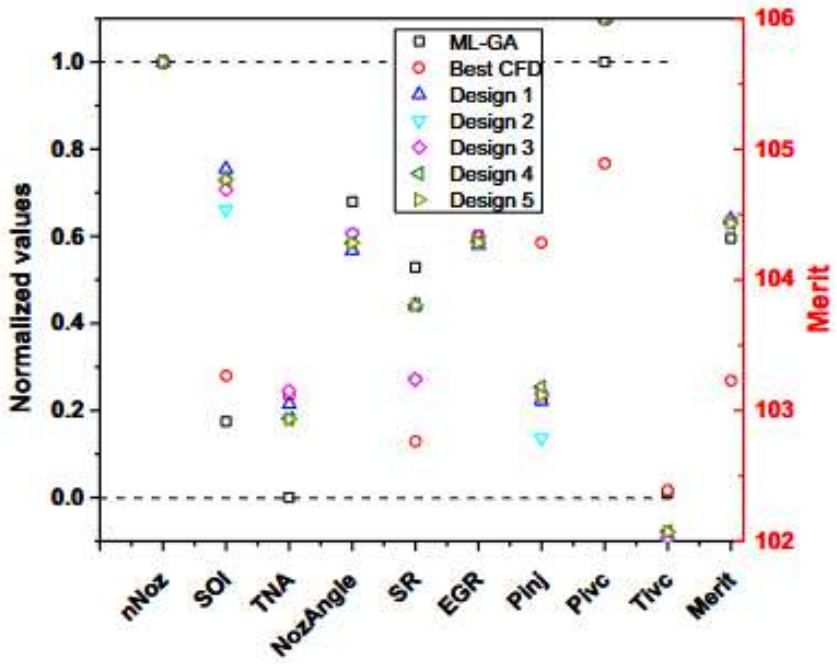

Figure.9. Optimum designs from the ML-GGA method with extended limits. [3]

Some of the design variables in the best designs found in this work and the optimum designs reported by Moiz et al. [40] are on or close to the boundaries established by the predefined constraints. These factors include TNA, the temperature during injection valve closure, and the number of nozzles. Better performance might result from expanding the design parameter range beyond its original preset bounds. This is illustrated by repeating the ML-GGA technique with Moiz et al.'s case study, extending the design parameters by 10% around their upper and lower bounds. In order to maintain sufficient ML predictability and stay inside the feasible design area, a mere 10% buffer is employed here. The five best optimum designs are displayed in absolute values, while Fig. - displays their normalized values. Included are the actual CFD simulation results for the top five designs found in this study. The same CFD model used by Moiz et al. [40] was used for this CFD verification exercise. The anticipated ISFC and merit value by the ML, if good predictability is attained, are also presented for reference. Moiz et al.'s best ML-GA and CFD-GA results [40]. Here, a code for the Machine Learning-Genetic Algorithm (ML-GGA) was created and verified using literary data. Two case studies were used to illustrate the possibilities of the ML-GA code, along with additional analysis and enhancements. Which concentrated on optimizing the operating parameters of a heavy-duty engine operating in the GCI mode, was replicated and examined in the first case study. According to our findings, improved ideal conditions could be obtained by doing a sensitivity study and expanding the parameter range beyond the training data boundaries. The ML-GGA code used here produced improved piston bowl geometries with up to 2.35% increases in merit value when compared to the top CFD designs.

IV. CONCLUSION

This paper presents a systematic strategy for applying machine learning to engine optimization challenges. There is discussion of important precautions, suggested algorithms, and appropriate optimization strategies. Because internal combustion engines are linked and highly non-linear, optimizing them is an extremely difficult task. Consequently, design failure in operation modes outside of the focused optimization requirements can be prevented using the multidimensional optimization approach. One crucial first step is to carefully define the optimization's objective function. To increase the prediction efficiency, training data for these machine learning algorithms must be carefully prepared. To steer clear of local optimum designs with lower merit value, global optimum search optimization techniques must be used. The optimization outputs must also be post-processed in order to use robustness and sensitivity analysis to assess the suggested design. Here, a code for the Machine Learning-Genetic Algorithm (ML-GGA) was created and verified using literary data. Two case studies were used to illustrate the possibilities of the ML-GA code, along with additional analysis and enhancements. This concentrated on optimizing the operating parameters of a heavy-duty engine operating in the GCI mode, was replicated and examined in the first case study. According to our findings, improved ideal conditions could be obtained by doing a sensitivity study and expanding the parameter range beyond the training data boundaries. The second case study focused on optimizing a heavy-duty GCI engine's piston bowl geometry under various operating circumstances. The ML-GGA code used here produced improved piston bowl geometries with up to 2.13% increases in merit value when compared to the top CFD designs.

This thesis demonstrates the potential of machine learning to revolutionize IC engine, combined approach enables real-time optimization by iteratively refining engine parameters through adaptive learning. By leveraging data-driven methodologies, this study provides a novel framework for designing high-efficiency diesel engines with lower environmental impact. The results demonstrate that ML-assisted GA optimization significantly improves fuel economy and emission compliance while enhancing thermal performance. This research contributes to the on-going efforts in sustainable engine development, offering a predictive and adaptable design methodology for future diesel engines.. By combining data-driven insights with traditional thermal science, ML models can significantly enhance the efficiency of engine development processes.

FUTURE WORK

- Expanding the dataset to include diverse engine types and operating conditions.
- Developing hybrid models that integrate physics-based and ML approaches.
- Exploring real-time deployment of ML models in engine control systems.

REFERENCES

[1] Plotnikov, Leonid. "Preparation and analysis of experimental findings on the thermal and mechanical characteristics of pulsating gas flows in the intake system of a piston engine for modelling and machine learning." *Mathematics* 11.8 (2023): 1967.

[2] Bappy, Md Aliahsan, and Manam Ahmed. "Assessment of data collection techniques in manufacturing and mechanical engineering through machine learning models." *Global Mainstream Journal of Business, Economics, and Development & Project Management* 2.04 (2023): 15-26.

[3] Badra, Jihad A., et al. "Engine combustion system optimization using computational fluid dynamics and machine learning: a methodological approach." *Journal of Energy Resources Technology* 143.2 (2021): 022306.

[4] Sanjeevannavar, Mallesh B., et al. "Machine learning prediction and optimization of performance and emissions characteristics of IC engine." *Sustainability* 15.18 (2023): 13825.

[5] Probst, Daniel M., et al. "Evaluating optimization strategies for engine simulations using machine learning emulators." *Journal of Engineering for Gas Turbines and Power* 141.9 (2019): 091011.

[6] Kaleli, Alirıza, and Halil İbrahim Akolaş. "The design and development of a diesel engine electromechanical EGR cooling system based on machine learning-genetic algorithm prediction models to reduce emission and fuel consumption." *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 236.3 (2022): 1888-1902.

[7] Amar, Yehia, et al. "Machine learning and molecular descriptors enable rational solvent selection in asymmetric catalysis." *Chemical science* 10.27 (2019): 6697-6706.

[8] Amini, Mahyar, Koosha Sharifani, and Ali Rahmani. "Machine learning model towards evaluating data gathering methods in manufacturing and mechanical engineering." *International Journal of Applied Science and Engineering Research* 15.2023 (2023): 349-362.

[9] Kodavasal, Janardhan, et al. "Using machine learning to analyze factors determining cycle-to-cycle variation in a spark-ignited gasoline engine." *Journal of Energy Resources Technology* 140.10 (2018): 102204.

[10] Pei, Yuanjiang, et al. "CFD-guided heavy duty mixing-controlled combustion system optimization with a gasoline-like fuel." *SAE International Journal of Commercial Vehicles* 10.2017-01-0550 (2017): 532-546.

[11] Xu, Bin, et al. "Real-time realization of Dynamic Programming using machine learning methods for IC engine waste heat recovery system power optimization." *Applied Energy* 262 (2020): 114514.

[12] Karunamurthy, Krishnasamy, et al. "Prediction and optimization of performance and emission characteristics of a dual fuel engine using machine learning." *International Journal for Simulation and Multidisciplinary Design Optimization* 13 (2022): 13.

[13] Yang, Ruomiao, Tianfang Xie, and Zhentao Liu. "The application of machine learning methods to predict the power output of internal combustion engines." *Energies* 15.9 (2022): 3242.

[14] Lebedevas, Sergejus, and Tomas Čepaitis. "Parametric analysis of the combustion cycle of a diesel engine for operation on natural gas." *Sustainability* 13.5 (2021): 2773.

[15] de Souza, Gustavo Rodrigues, et al. "Study of intake manifolds of an internal combustion engine: A new geometry based on experimental results and numerical simulations." *Thermal Science and Engineering Progress* 9 (2019): 248-258.

[16] Reitz, R.D.; Ogawa, H.; Payri, R.; Fansler, T.; Kokjohn, S.; Moriyoshi, Y.; Agarwal, A.K.; Arcoumanis, D.; Assanis, D.; Bae, C.; et al. IJER editorial: The future of the internal combustion engine. Int. J. Engine Res. 2020, 21, 3–10. [CrossRef]

[17] Guzzella, L.; Onder, C. Introduction to Modeling and Control of Internal Combustion Engine Systems; Springer: Berlin/Heidelberg, Germany, 2010; 362p.

[18] Lumley, J.L. Engines, an Introduction; Cambridge University Press: Cambridge, UK, 1999; 272p.

[19] Plotnikov, L.V. Unsteady gas dynamics and local heat transfer of pulsating flows in profiled channels mainly to the intake system of a reciprocating engine. Int. J. Heat Mass Transf. 2022, 195, 123144. [CrossRef]

[20] Pranoto, S.; Ubaidillah, A.; Lenggana, B.W.; Budiana, E.P.; Wijayanta, A.T. Fluid Flow Analysis at Single and Dual Plenum Intake Manifolds to Reduce Pressure Drops Using Computational Approach. J. Adv. Res. Fluid Mech. Therm. Sci. 2022, 97, 1–12.

[21] Zhang, S.; Li, Y.; Wang, S.; Zeng, H.; Liu, J.; Duan, X.; Dong, H. Experimental and numerical study the effect of EGR strategies on in-cylinder flow, combustion and emissions characteristics in a heavy-duty higher CR lean-burn NGSI engine coupled with detail combustion mechanism. Fuel 2020, 276, 118082.

[22] Zhao, D.; An, Y.; Pei, Y.; Shi, H.; Wang, K. Numerical study on the asymmetrical jets formation from active pre-chamber under super-lean combustion conditions. Energy 2023, 262, 125446.

[23] Yuan, C.; Li, S.; Qin, Z.; Lu, J.; Peng, S. A multi-process coupling study of scavenging pressure effect on gas exchange of a linear engine. Appl. Therm. Eng. 2022, 217, 119254.

[24] Yin, S.; Ni, J.; Fan, H.; Shi, X.; Huang, R. Study on Correction Method of Internal Joint Operation Curve Based on Unsteady Flow. Appl. Sci. 2022, 12, 11943.

[25] Marelli, S.; Capobianco, M.; Zamboni, G. Pulsating flow performance of a turbocharger compressor for automotive application. Int. J. Heat Fluid Flow 2014, 45, 158–165.

[26] Leng, L.; Qiu, H.; Li, X.; Zhong, J.; Shi, L.; Deng, K. Effects on the transient energy distribution of turbocharging mode switching for marine diesel engines. Energy 2022, 249, 123746.

[27] Liu, Z.; Liu, J. Investigation of the Effect of Altitude on In-Cylinder Heat Transfer in Heavy-Duty Diesel Engines Based on an Empirical Model. J. Energy Resour. Technol. Trans. ASME 2022, 144, 112303.

[28] Nie, X.; Bi, Y.; Liu, S.; Shen, L.; Wan, M. Impacts of different exhaust thermal management methods on diesel engine and SCR performance at different altitude levels. Fuel 2022, 324, 124747.

[29] Ma, F.; Yang, W.; Wang, Y.; Xu, J.; Li, Y. Experimental research on scavenging process of opposed-piston two-stroke gasoline engine based on tracer gas method. Int. J. Engine Res. 2022, 23, 1969–1980.

[30] Mazuro, P.; Kozak, D. Experimental investigation on the performance of the prototype of aircraft Opposed-Piston engine with various values of intake pressure. Energy Convers. Manag. 2022, 269, 116075.