Project Report

Digitalization of a Dual Fuel Engine and Creation of a Digital Twin

By

Billriz Condor

Hafeez Bashir

Kien van Ho

PM3E/ME3+

IMT Atlantique Nantes

June 2025

Table of contents

Table of Figures

1. Introduction

With the demand for internal combustion engines with higher efficiencies and lower emissions, investigations on dual-fuel combustion engines have experienced a recent increase. This type of engine can operate using both liquid and gaseous fuels. In most cases, the gaseous fuels such as biogas, compressed natural gas, producer gas, and hydrogen are used as primary fuel while diesel and other liquid fuels as the pilot fuel. (Pham et al., 2022) The technology promises a cleaner pathway for burning fuel; however, further optimization and development are essential to ensure best performance of the engine during operation.

Digital twin technology has gradually proliferated in most major sectors. From agriculture to renewable energy systems, digital twins have been deployed to increase the benefits and lower the negative impacts of modern systems. The technology involves creating a virtual replica of a physical system, process, or object that is continuously updated with real-time data from its physical counterpart. This digital model simulates, monitors, and analyzes the behavior and performance of the physical asset to optimize operations, predict failures, and support decision-making (Khan et al., 2025).

The integration of digital twin technology with dual-fuel combustion engines offers significant advantages in performance optimization, emissions control, and predictive maintenance. As energy systems shift toward greater efficiency and lower emissions, combining these two technologies becomes strategically important in both research and industry (Khawale et al., 2024).

In the present project, a digital twin is created for a dual fuel engine (18 kW, 4 cylinders) designed to be installed in a biogas power plant. First, a theoretical model was created based on the key parameters conventionally monitored in measuring engine performance was established and coded through Python. Next, a 3D model of a dual fuel engine that will be used as a visual tool while using the digital twin was created using Blender. Lastly, the python code and 3D model will be deployed on DataGrowb's GreenTwin platform wherein the users can access the current status of the dual fuel engine through its digital twin. The digital twin will allow the simulation, analysis, and improvement of the engine and process performance through real-time data collection potential identification of inconsistencies in its operation.

2. Background and Literature Review

The development of digital twins in different applications has been proposed to improve the overall efficiency of existing modern systems. The application of this technology in power generation has already been deployed in various ways. Here, we discuss the existing innovations and its possible application to a more niche problem which is for the improvement of dual-fuel engines.

2.1. Dual-Fuel Engine

A dual-fuel engine operates on a combination of biogas gas (primary fuel) and diesel (pilot fuel). This configuration is widely used in stationary power generation due to its fuel flexibility, reduced emissions, and improved efficiency. Thus, the main fuel that will be used is treated biogas and a pilot-fuel of diesel. Conventionally, in a dual-fuel engine: biogas, containing methane and carbon dioxide, is introduced into the intake air stream and pre-mixed before entering the cylinder. Diesel fuel is directly injected into the cylinder near the end of the compression stroke. The diesel acts as a pilot ignition source, initiating combustion of the natural gas–air mixture. This approach combines the high energy density and autoignition properties of diesel with the clean-burning characteristics of biogas (Sahoo et al., 2009). For this study, the engine used was the DE18E3 model from Cat®. See Figure 1.

Figure 1: Original DE18E3 model

The engine used in the study was originally a diesel-powered engine that was then customized to run on both biogas and diesel, as seen in Figure 2. Originally, the DE18E3 model was a diesel-powered generator set built around the Cat® C2.2 engine, engineered for mobile power applications. It complies with EU Stage IIIA emissions standards, making it suitable for environmentally regulated regions. The following are the key features of the engine can be found in the Appendix.

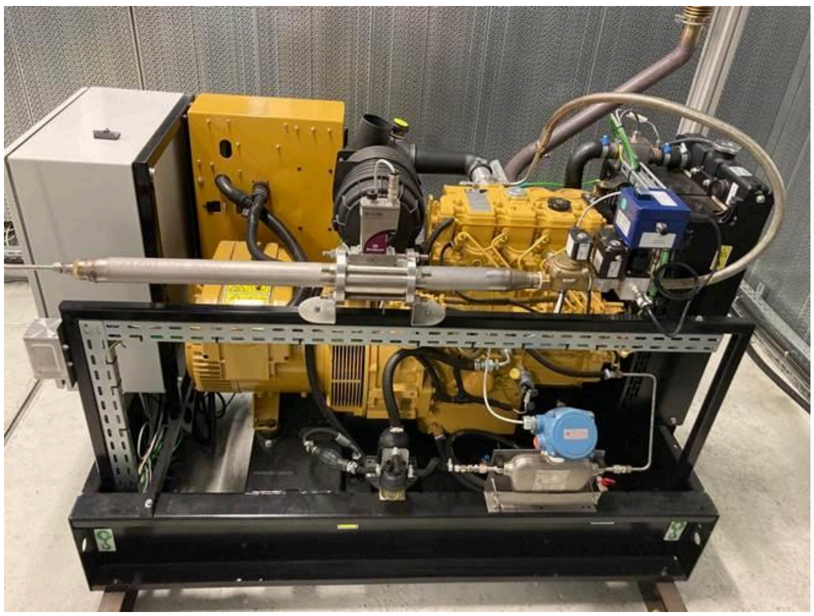


Figure 2: Modified dual-fuel engine used in the study

2.2. Digital Twin

Digital twin technology has rapidly evolved from manufacturing and aerospace into the domains of energy systems and mechanical engineering, where it plays a critical role in optimizing performance, improving reliability, and enabling predictive maintenance. Some exciting fields that are investigating the addition of digital twin technology include: power generation and energy grids, industrial mechanical systems, automotive and transportation, dual-fuel and hybrid systems. Digital twins are revolutionizing how energy systems are monitored, controlled, and improved. They reduce operational risks, enhance energy efficiency, and support sustainability targets. Their application in dual-fuel engine systems represents a cutting-edge integration of mechanical modeling and real-time analytics, paving the way for intelligent, low-emission energy systems (Nguyen et al., 2024). The project is co-supervised by DataGrowb and will be using the company's existing digital twin platform and tools for the deployment of the created digital twin.

The use of digital twins paired with other emerging technologies such as artificial intelligence is expected to accelerate the world's transition to robust, cheap, and clean energy technology. This has already been observed in other renewable energy fields such as wind and solar. The improvement of an energy system's efficiency, lifespan, and reliability can be achieved through using real-time data for optimization of its operation and maintenance. A typical digital twin framework can be seen in Figure 3 (Abdessadak et al., 2025).

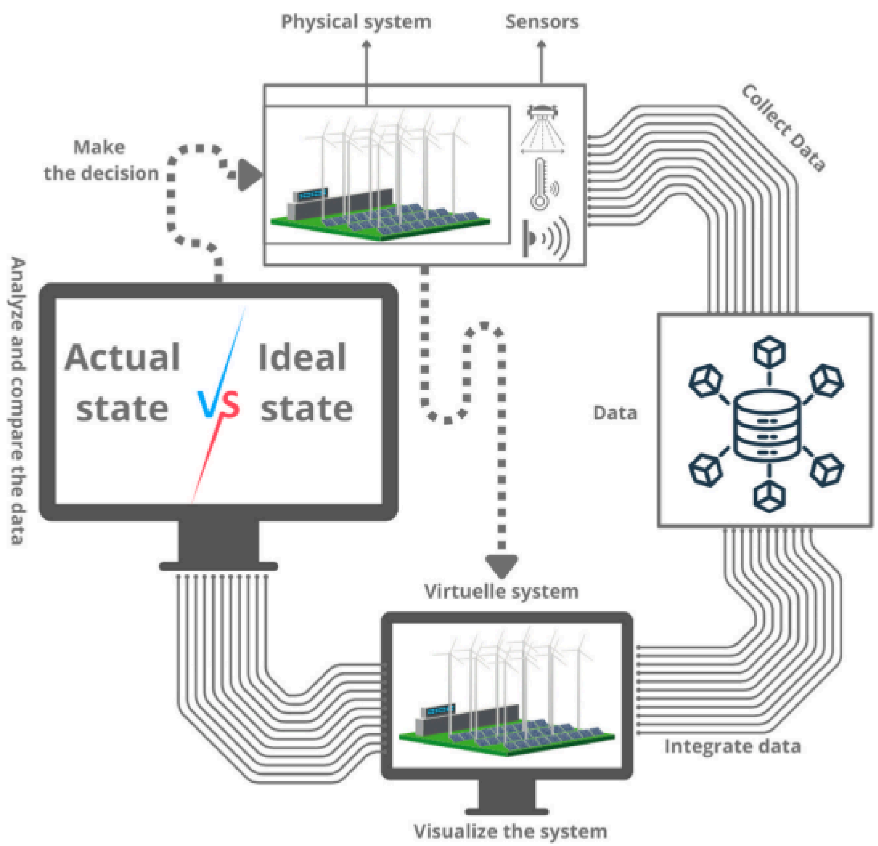


Figure 3: The ideal proposition for a complete digital twin from Abdessadak 2025

In this proposed project, we intend to lay down the foundations for the future development of a full-scale digital twin for a biogas combustion engine. The development of such a system would allow this niche sector to benefit from the advantages of these new technologies.

3. Methodology

Figure 4 shows a general overview of this study's digital twin architecture. To achieve this, the following general steps were conducted:

- Data Inventory
- Modeling of the dual fuel engine
- Development of 3D model
- Integration to DataGrowb's Platform

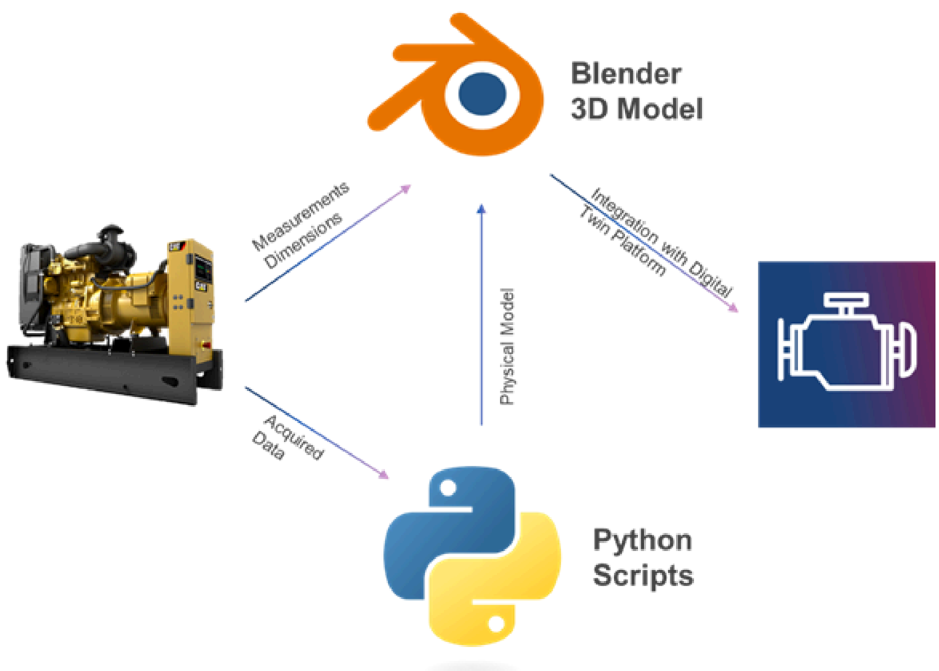


Figure 4: Overall Digital twin architecture

3.1. Data Inventory

Data and previous investigations on the dual fuel engine studied in this project were collected. Previously, a heat mapping study on the dual fuel engine was conducted to measure and optimize its performance. The heat mapping data, as seen in Figure 4, was used to develop models in this study. Important parameters monitored by the laboratory set-up that were used for the modeling were selected.

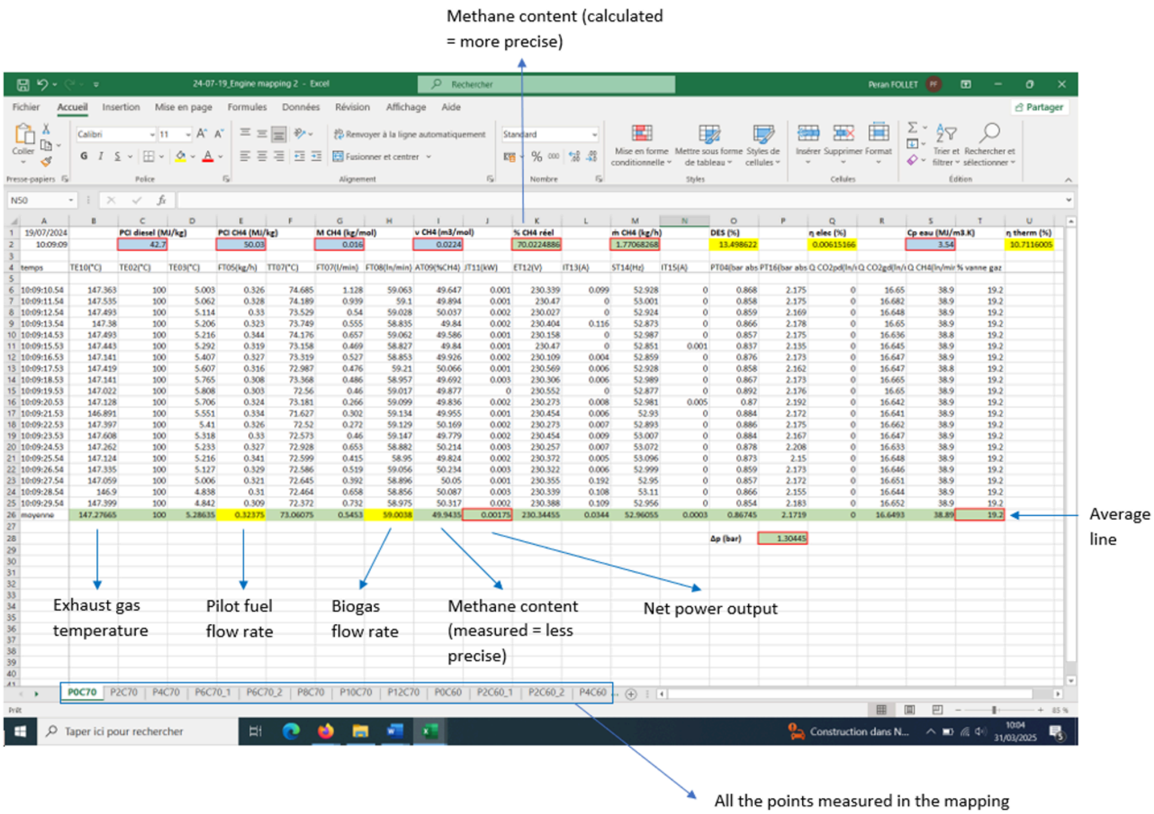


Figure 5: Real time data inventory

Data preprocessing in a data inventory improves data quality by eliminating errors and inconsistencies, enhances model performance through normalization and standardization, minimizes noise by filtering out irrelevant data, guarantees consistency and comparability by standardizing formats, accelerates processing by removing duplicates, and reduces bias by dealing with outliers, ultimately readying the data for more efficient and insightful analysis or modeling.

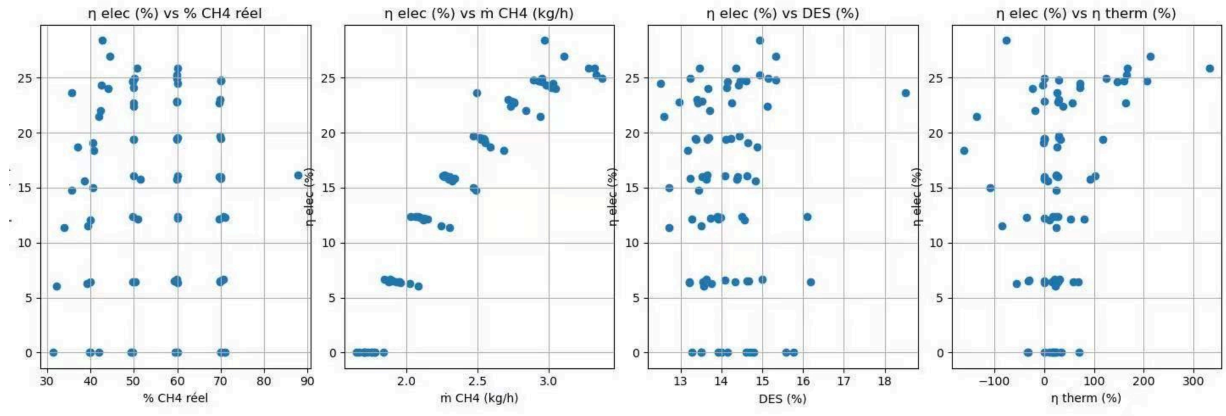


Figure 6: Data pre-processing

After the data from the heat engine mapping was cleaned and pre-processed, the selected key outputs were computed from the physical model formulas. In addition, linear regression and K-nearest neighbor models were also used to predict some exhaust gas temperature and efficiency, respectively.

3.2. Physical Modeling of the Dual Fuel Engine

The formulae for key inputs and outputs were established. Python was used to implement and solve these equations, providing initial simulation results.

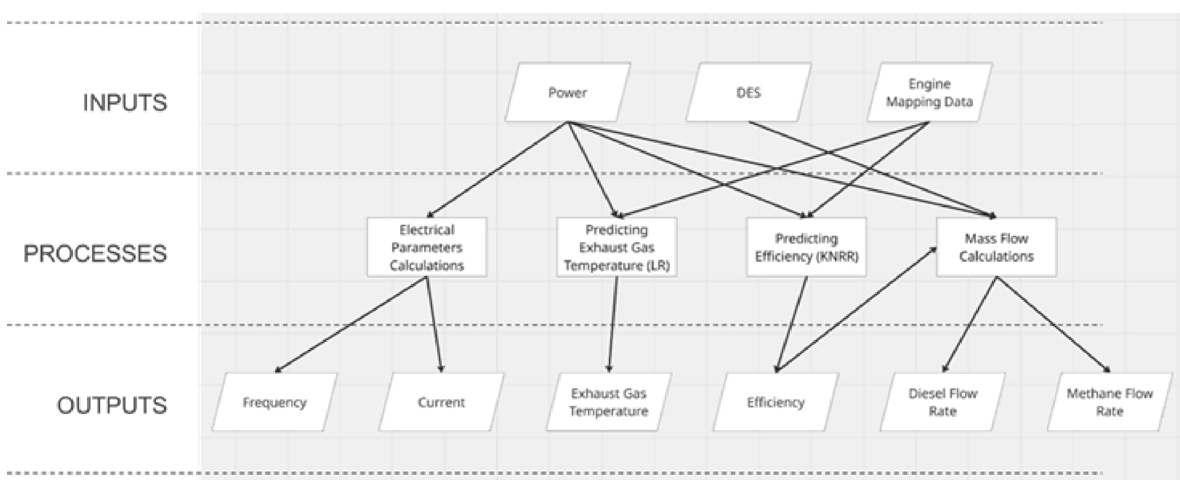


Figure 7: Identified inputs and outputs to our model

The following table summarizes the definition of key parameters measured by the engine's monitoring system:

Table 1. Key parameters used in the dual fuel engine model

Parameters	**Unit**	**Symbol**	**Definition**
Methane Content	%	C_{CH4}	This refers to the volume percentage of methane (CH_4) in the biogas fuel mixture. It directly influences the energy content of the biogas, as methane is the main combustible component.
Biogas Flow Rate	Ln/min	Q_{biogas}	This refers to the fuel supply rate from the renewable gas or the volume of biogas supplied to the engine per unit of time.
Net Power Output	kW	P_{net}	This represents the effective power available for external loads after subtracting internal losses such as friction, cooling fans, and auxiliary systems.
Pilot Fuel Flow Rate	kg/h	Q_{Diesel}	This is the amount of diesel fuel injected into the combustion chamber per unit of time, used to initiate ignition in a dual-fuel engine
Exhaust Gas Temperature	°C	$T_{exhaust}$	The temperature of the combustion gases as they exit the engine's exhaust system gives insight into combustion efficiency, thermal load, and potential engine issues like incomplete combustion or knocking.
Diesel Energy Share	%	DES	This refers to the percentage of total fuel energy provided by diesel in a dual-fuel engine. Lower share indicates higher reliance on the gas fuel and improved fuel cost savings and emissions reduction.

The following are the equations and formulae used in the modeling of the dual fuel combustion engine:

- **For electrical efficiency:**

$$\mu_{elec} = \frac{P_{net}}{(Q_{Diesel} \cdot LHV_{Diesel}) + (Q_{CH_4} \cdot LHV_{CH_4})}$$

Where:

μ_{elec}:	Electrical efficiency
P_{net}:	Net power output
Q_{Diesel}, Q_{CH_4} :	Flow rates of diesel, methane
LHV_{Diesel}, LHV_{CH_4}:	Lower heating value of diesel, methane at inlet

- **For diesel energy share:**

$$DES = 100 \cdot \frac{Q_{Diesel} \cdot LHV_{Diesel}}{(Q_{Diesel} \cdot LHV_{Diesel}) + (Q_{CH_4} \cdot LHV_{CH_4})}$$

Where:

DES:	Diesel energy share
Q_{Diesel}, Q_{CH_4}:	Flow rates of diesel, methane
LHV_{Diesel}, LHV_{CH_4}:	Lower heating value of diesel, methane at inlet

- **For mass flow rate of methane:**

$$F_{CH_4} = Q_{biogas} \cdot C_{CH_4} \cdot \frac{M_{CH_4}}{V_{M,CH_4}}$$

Where:

F_{CH_4}:	Mass flow rate of methane
Q_{biogas}:	Flow rates of biogas
C_{CH_4}:	Methane content of biogas

M_{CH_4}, V_{M,CH_4}: Molar mass, molar volume of methane

- **For methane content:**

$$C_{CH_4} = 100 \cdot \frac{Q_{CH_4}}{Q_{CH_4}+Q_{CO_2}}$$

Where:

C_{CH_4}: Methane content of biogas

Q_{Diesel}, Q_{CH_4}: Flow rates of methane, carbon dioxide

3.3. Development of 3D model

For the modeling component of this project, Blender was used. Blender is a free software for 3D modeling, animation and rendering that was first released in 1994 and is supported by the Blender Foundation. Since 2019, it has garnered increasing recognition from 3D animation companies such as Epic Games, Ubisoft, and NVIDIA. The program offers advanced modeling features, including 3D sculpting, texture mapping, UV editing, as well as animation capabilities like rigging and shape keys. It allows for rendering using both GPU and CPU and also provides video editing without a predetermined timeline, includes composition tools, and features a nodal material design. Additionally, Blender can manage a range of physical effects, such as fabric simulations, particle systems, solid body dynamics, collisions and smoke by using its cycle rendering engine.

3.4. Integration of the digital twin model to the platform

The integration of the digital twin model to the GreenTwins website involves several steps. First, the 3D model (insert file type of the 3D model) created in Blender was uploaded to GitHub. An URL of the 3D model from GitHub was automatically generated. This URL was used when calling the image of the model in the GreenTwins web application. Next, a cloud storage containing the code and data used for the development of the digital twin model was set up in

Microsoft Azure. After the data containers were set up, the receival and handling of data was done using a serverless compute service, Azure functions. Lastly, to display current state and parameter calculations of the digital twin to the GreenTwin web application's dashboards a Python web framework, FASTAPI, was used. Figure 8 shows the overall general process of how the integration of the developed digital twin to the GreenTwin platform works.

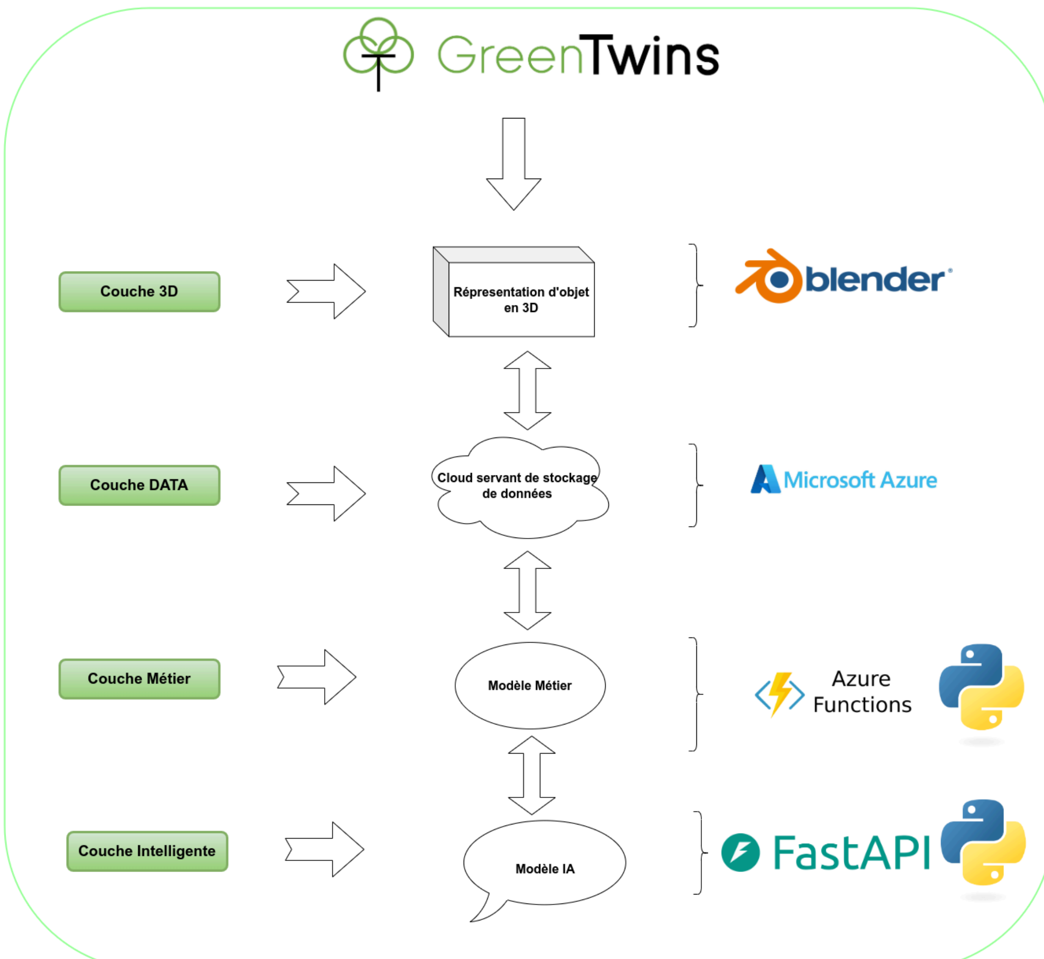


Figure 8: Workflow of digital twin in GreenTwin platform

3.5. Repository of codes and data used for the digital twin

The full python project is detailed in a github repository and can be found under the following URL: https://github.com/kivanho/dual-fuel-engine_twin. A docker image to run the app with an API for further integration can be pulled with this command: *docker pull lvanho/dual_fuel_app*.

4. Results and Discussion

The following sections describe the results of the modeling work of both the digital twins physical model and the 3D model. Also, the final look and features of the digital twin after its integration to the GreenTwin platform are explored.

4.1 Working principle of the digital twin

The digital twin of the dual fuel engine was implemented in Python and functions by taking user-defined inputs - desired power output, diesel energy share, and methane percentage in biogas. From these, it predicts target efficiency and target exhaust gas temperature with machine learning models, which gets detailed in the next section. Using physical relationships, the model calculates the required mass flows, which are then regulated via a simple PID controller. While mass flows get adjusted, the actual engine performance metrics - power output, current, and torque - are computed simultaneously, allowing the system to replicate real engine behavior under varying conditions.

Using the digital twin, simulations were run across a range of operating conditions. First a local dashboard for validating the model was built before deploying it on a Webapp for further integration (Figure 9). For instance, at a target power output of 11.75 kW with an 18.89 % diesel share and 72.5 % methane content, the predicted efficiency reached 23.52 %, and the exhaust temperature stabilized at 347 °C. The dataset used to train the model indicates 23.48 % (-0.04 % difference) efficiency and a temperature of 342.7 °C (-4.3 °C difference). The mass flows calculated from these calculations correspond perfectly with the dataset: 2.92 kg/h for methane and 0.80 kg/h for diesel. (See Figure 9)

The PID controller demonstrated stable regulation with a settling time under 5 seconds, still allowing a spiky swinging behavior like the real engine. These results validate the digital twin's ability to accurately model engine dynamics and fuel interactions, making it a reliable tool for performance prediction and control strategy testing.

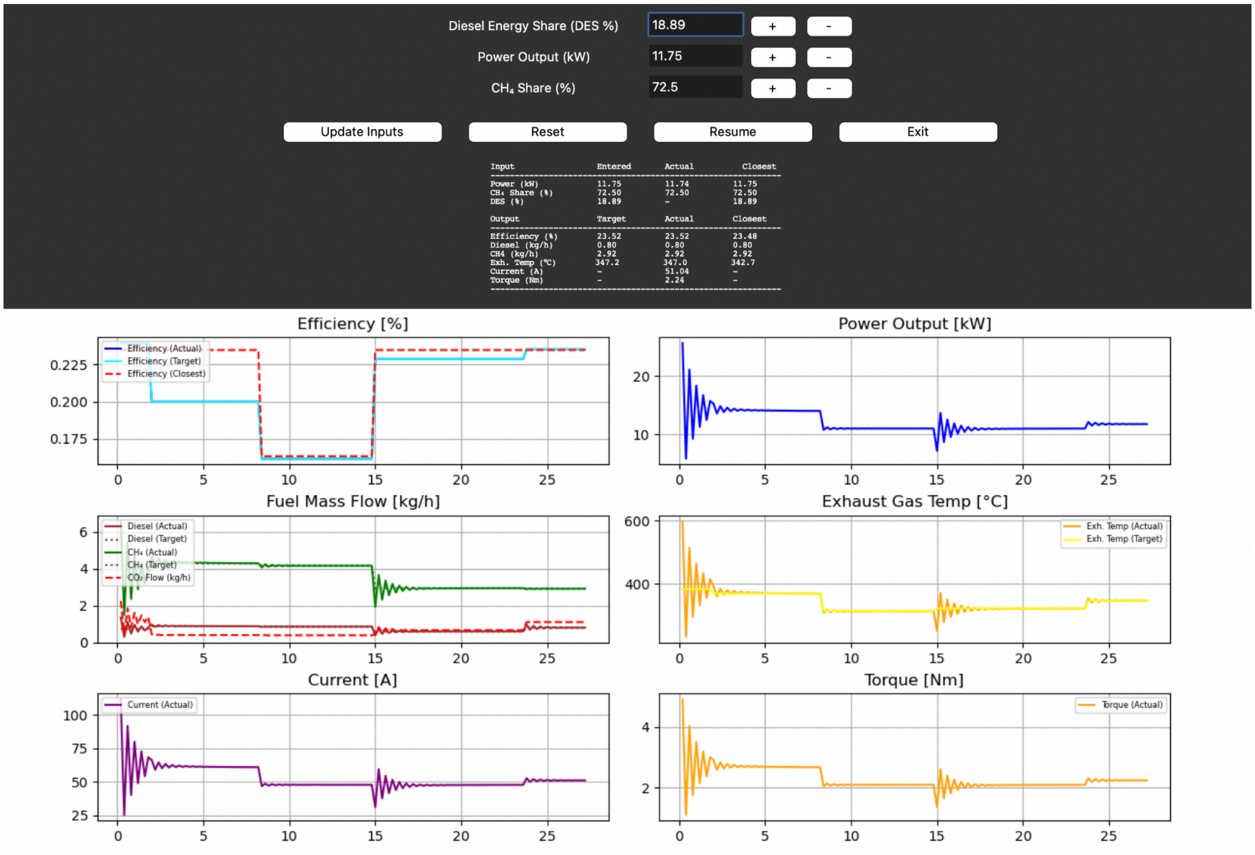


Figure 9: Local dashboard

4.1.1 Linear Regression for predicting Exhaust Gas Temperature

A simple linear regression was applied to predict the exhaust gas temperature using power output. The results of the regression showed a good fit between the two variables with a R^2 of 0.956 (See Figure 10). This means that 95.6% of the variance can be explained by the linear regression model based on the engine's power output. The DES and %CH4 were further used to precise the prediction. As shown in Figure 10, there is no significant outlier in the data and the model is very reliable in predicting exhaust gas temperatures within the operating range of 150 °C to 450 °C. Thus, this simple linear model is robust enough to predict real-time exhaust gas temperatures for the digital twin that is being developed for this project. The code for this regression can be found in Appendix XX.

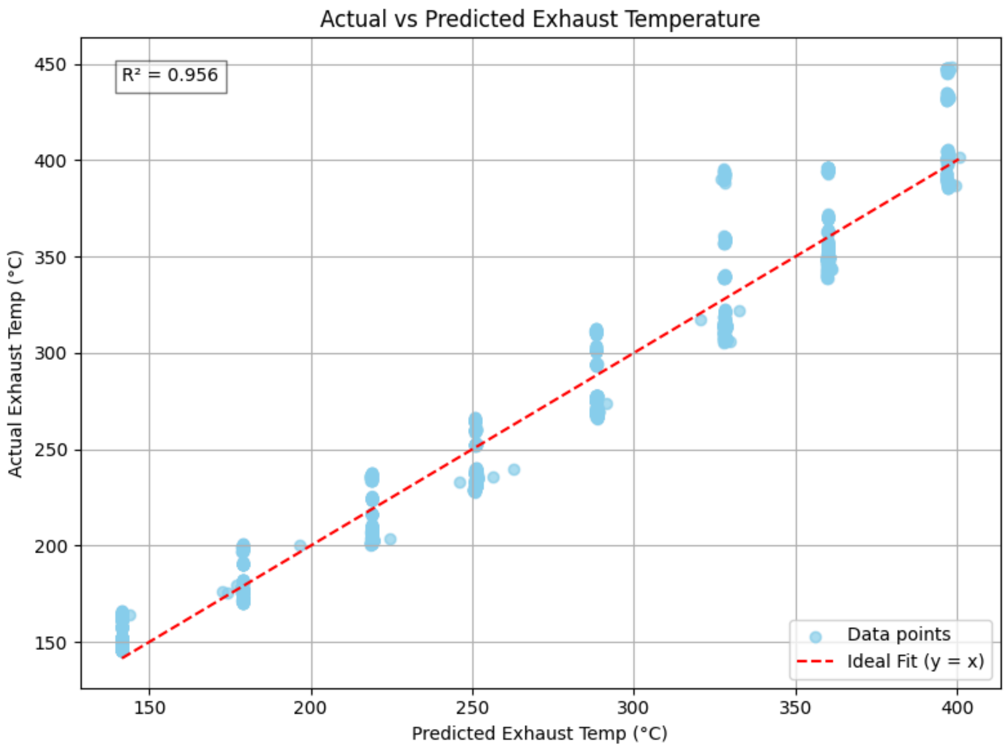


Figure 10: Plot for actual vs predicted exhaust temperatures from LR model

4.1.2 K-nearest neighbor for predicting Electrical Efficiency

For predicting the electrical efficiency of the engine, a k-nearest neighbor (KNN) model is employed. Because the relationship between the efficiency and power output can't be explained linearly, a non-parametric regression model was used for this parameter. The fine tuning of the parameters of the KNN model determined that the optimal conditions are: 10 for n neighbors, 1 for KNN p, and uniform KNN weights. As input features the power output and the %CH4 were used. The results of the modeling showed a very high R^2 of 0.929 (see Figure 11). The spread of the different data points for predicted vs actual values is uniform and does not deviate far from the ideal fit. Like the linear regression model for the exhaust gas temperature, the results from modeling shows that it can be included in creating the engine's digital twin. The predicted values were then used to calculate the target values for the other output values via the physical formulas presented in section 3.2..

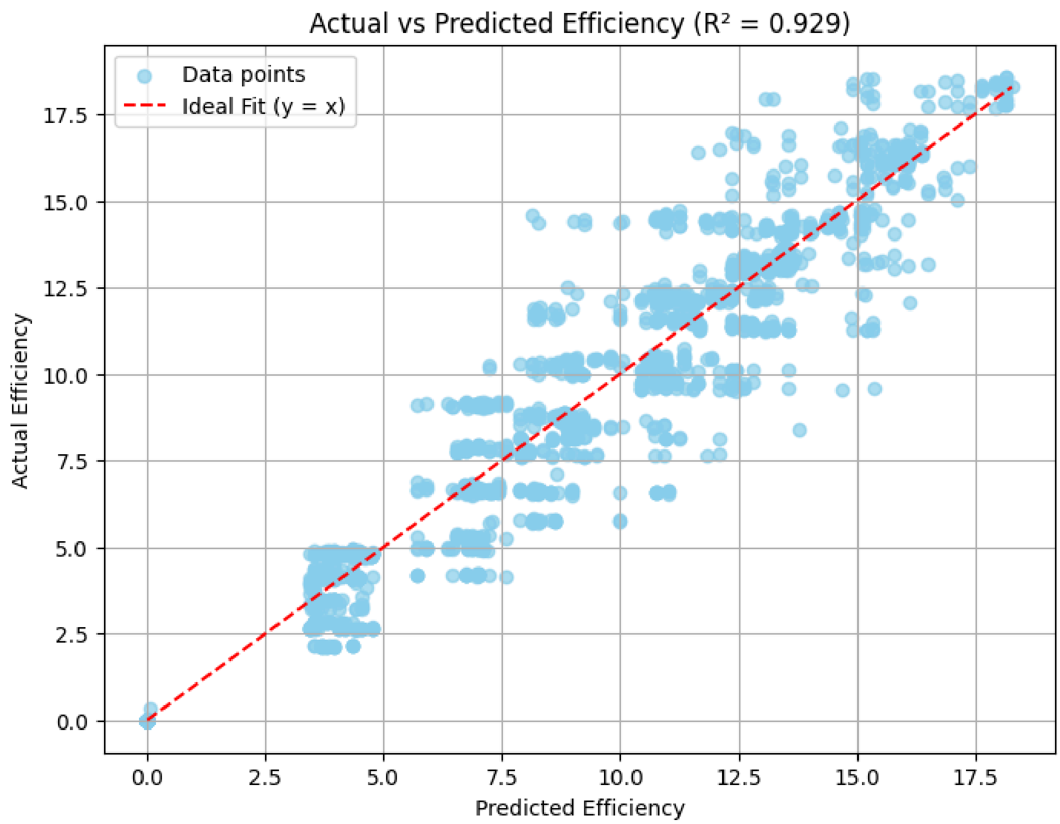


Figure 11: Plot for actual vs predicted efficiencies from KNN model based on power output

4.2 3D Modeling of the Dual Fuel Engine

The second phase involved the development of a 3D model of the engine using Blender, to create a visual representation that complements the numerical simulation and enhances system understanding.

Developing a digital twin largely depends on 3D modeling, wherein Blender, a computer-aided design software, is utilized to construct the four-cylinder dual-fuel engine. This model incorporates elements such as the crankshaft, engine block, electrical alternator, pistons, and the pipes for air, water, and fuel. This CAD facilitates a digital depiction of the real-time dual-fuel engine.

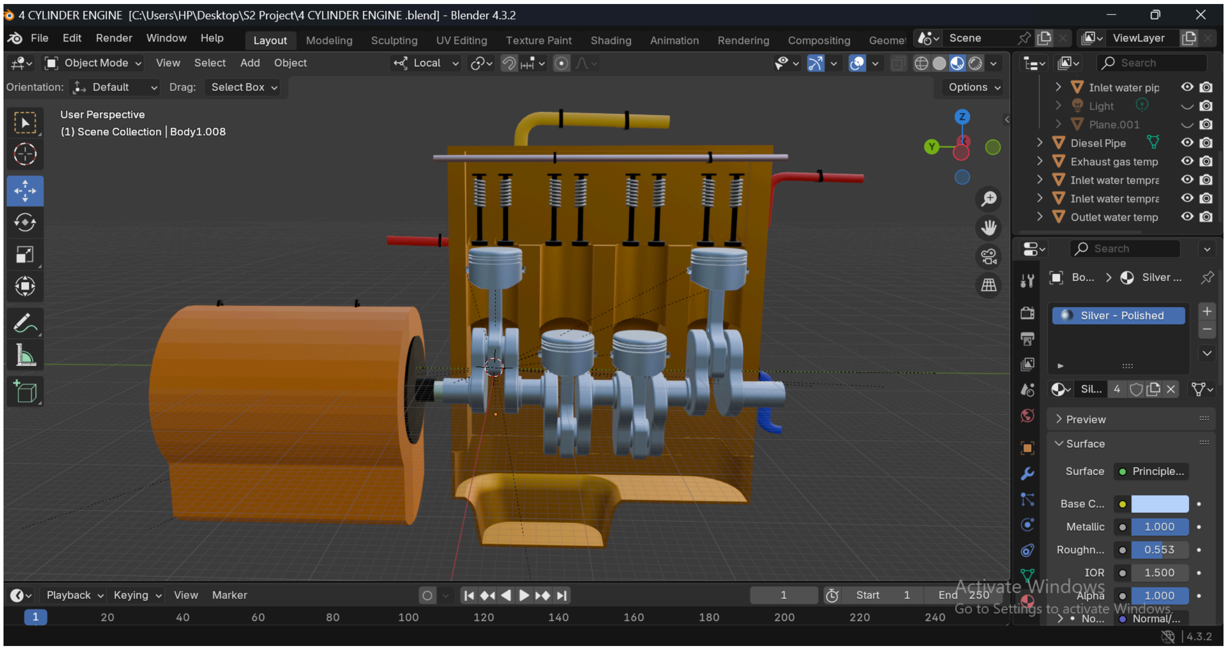


Figure 12 : Blender interface with 4 cylinder CAD

Following the model's creation, the intake and exhaust pipes were given distinct colors to simplify identification. In the model, the biogas intake pipe is marked in yellow, the diesel intake pipe in gray, and the exhaust gas pipe in red. Sensors were strategically placed on various components to monitor specific parameters of interest. For example, the yellow biogas pipe has two sensors displaying the flow rate and %CH_4 content, the exhaust gas pipe sensor shows the exhaust gas temperature, the diesel pipe sensor indicates the diesel energy share (DES%) and diesel flow rate, the electrical alternator sensor tracks electrical current and efficiency, and the crankshaft sensor measures torque.

Blender was selected due to its various advantages. It effortlessly integrates with Microsoft Azure's cloud computing platform, allowing for the concurrent development of numerous 3D designs. It also provides access to tailored storage options. This helps in seamlessly incorporating the 3D model to the following steps of this project which involves the integration of the models to the GreenTwin's web application. Additionally, its compatibility with Python scripting facilitates the automation of digital twin results, enhancing collaboration with other 3D design tools like SOLIDWORKS.

4.3 The Digital Twin in GreenTwin Platform

The digital twin is now hosted in the GreenTwins Platform. Using the web application, users can interact with the models developed in this project. The following sections contain images and descriptions of the different features of the digital twin.

4.3.1 The 3D model in GreenTwin

The 3D model of the dual fuel engine is currently stored in a github repository. The GreenTwin website allows users to view and interact with the model. As seen in Figure 12 and Figure 13, the full 3D model developed containing the engine block, pistons, intake and outlet pipes, and generator can be viewed. By clicking on the different components, the users can see the status of the measured parameters. For example, clicking on the generator displays the efficiency and power output of the engine while clicking the exhaust pipe displays the exhaust gas measured by the temperature sensors.

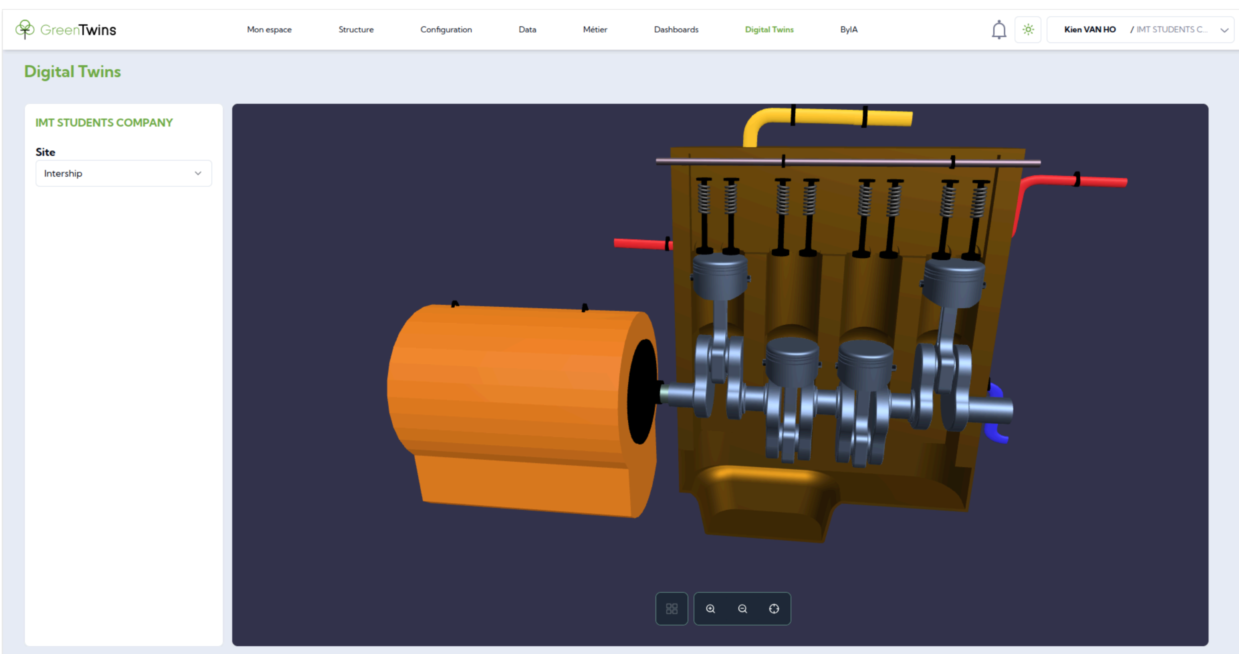


Figure 13: The 3D model viewed from GreenTwin web application

4.3.2 Digital Twin Features

The digital twin deployed in GreenTwin operates in the same principle as described in 4.1. It also employs the same models developed and trained in the previous subsections. In Figure 14, by clicking on the display dashboard button, it will display a pop-up window with a form where the user can input values for the model's main input parameters: diesel energy share, power output, and methane concentration. After inputting the values, the users can run the models and can see the computed values for efficiencies, exhaust gas temperature, and fuel flow rates.

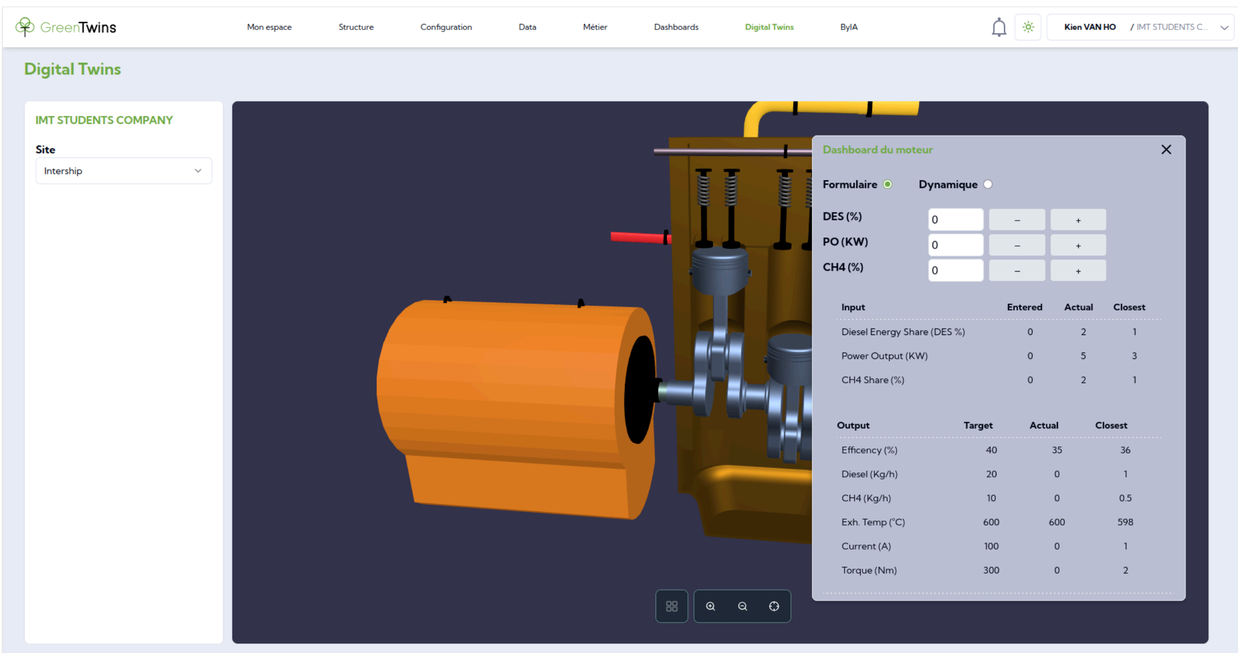


Figure 14: Input value forms and calculated parameters

Once the model runs all the calculations, the dashboard also displays a time-graph of how the digital twin is running (see Figure 15. Data visualizations for both the input and calculated performance indicators are shown. For example, users can see how the efficiency is affected when power output is increased. Moreover, once the inputs are changed again, users can see how the digital twins adjust to these changes. For this model, the process control features show how the dual-fuel engine adjusts to the changes in input parameters by swinging back and forth until it reaches the set values. In principle, the inputs should be automated and should come from real-time measurements of sensors installed on the real engine. While this may be a limit of what the project was able to achieve, the underlying theory and calculations are the same and have laid down strong foundations for further development down the road.

Figure 15: Time-series graphs of key parameters in the GreenTwin dashboard

5. Conclusion

The application of digital twin technology to dual-fuel engines intends to accelerate their development and adoption. In this study, the physical model of a dual-fuel engine powered by diesel and biogas was developed. Key parameters that measure its performance were selected, namely, the power output, methane content, diesel energy share, biogas flow rate, and exhaust gas temperature. Using the engine's heat mapping data, the model was validated and tuned to increase its accuracy and reliability. In parallel, a digital 3D model of the engine was constructed which was used to visualize the digital twin. Finally, both the 3D reconstruction and the physical model were connected to the GreenTwin platform that allows users to interact and observe with the digital twin.

6. Future Work and Recommendations

At the current state of the development, the digital twin was able to reliably compute and predict the engine's key performance indicators. Future work leans towards connecting the digital twin to the operating engine in real-time. By accessing real-time data, other features can be added to the digital twin such as predictive maintenance and operation optimization.

7. References

Abdessadak, A., Ghennioui, H., Thirion-Moreau, N., Elbhiri, B., Abraim, M., & Merzouk, S. (2025). Digital twin technology and artificial intelligence in energy transition: A comprehensive systematic review of applications. In *Energy Reports* (Vol. 13, pp. 5196–5218). Elsevier Ltd. https://doi.org/10.1016/j.egyr.2025.04.060

Khan, W. A., Pakseresht, A., Chua, C., & Yavari, A. (2025). Digital twin role for sustainable and resilient renewable power plants: A systematic literature review. In *Sustainable Energy Technologies and Assessments* (Vol. 75). Elsevier Ltd. https://doi.org/10.1016/j.seta.2025.104197

Khawale, R. P., Chowdhury, D., Goyal, R., Singh, S. K., Bhatt, A., Gainey, B., Lawler, B., Crawford, L., & Rai, R. (2024). Digital twin-enabled autonomous fault mitigation in diesel engines: An experimental validation. *Control Engineering Practice*, *152*. https://doi.org/10.1016/j.conengprac.2024.106045

Nguyen, V. N., Nayak, S. K., Le, H. S., Kowalski, J., Deepanraj, B., Duong, X. Q., Truong, T. H., Tran, V. D., Cao, D. N., & Nguyen, P. Q. P. (2024). Performance and emission characteristics of diesel engines running on gaseous fuels in dual-fuel mode. In *International Journal of Hydrogen Energy* (Vol. 49, pp. 868–909). Elsevier Ltd. https://doi.org/10.1016/j.ijhydene.2023.09.130

Pham, Q., Park, S., Agarwal, A. K., & Park, S. (2022). Review of dual-fuel combustion in the compression-ignition engine: Spray, combustion, and emission. In *Energy* (Vol. 250). Elsevier Ltd. https://doi.org/10.1016/j.energy.2022.123778

Piston Engine-Based Power Plants. (n.d.).

Sahoo, B. B., Sahoo, N., & Saha, U. K. (2009). Effect of engine parameters and type of gaseous fuel on the performance of dual-fuel gas diesel engines-A critical review. In *Renewable and*

Sustainable Energy Reviews (Vol. 13, Issues 6–7, pp. 1151–1184). https://doi.org/10.1016/j.rser.2008.08.003

8. Appendices *(if needed)*

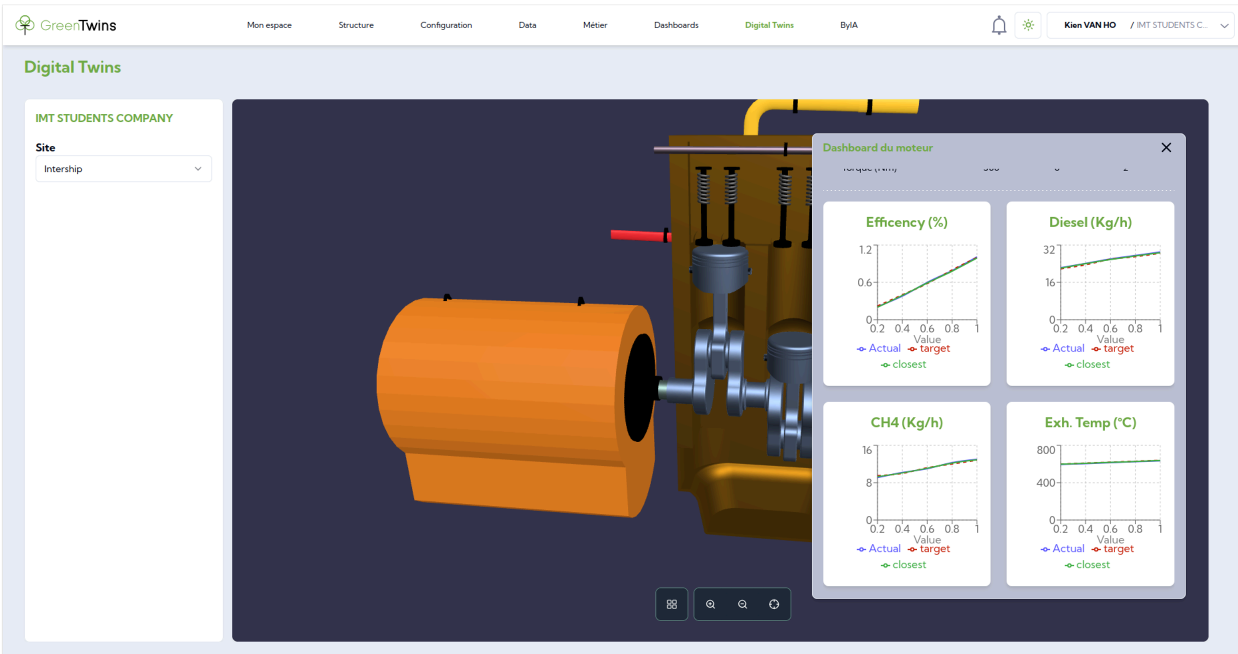
APPENDIX A. Engine Information

Displacement	2.2 liters
Compression Ratio	23.3:1
Rated Speed	1500 rpm (50 Hz) / 1800 rpm (60 Hz)
Fuel Type	Diesel (A2 category or BSEN590 standard)
Fuel Tank Capacity	66 L
Lubrication Oil Capacity	8.9–10.6 L
Cooling System	Water-cooled with centrifugal pump, designed for 50°C ambient

Electrical Specifications

Data	**Value**	**Unit**	**Symbol**
Voltage	230	V	V_{Gen}
Rotational Speed	1500	rpm	n
Number of Poles	2	-	p

APPENDIX B. 3D MODEL CONSTRUCTION



APPENDIX C. Code snippets

exhaust_temp_model.py > train_exhaust_temp_model

```python
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline

def train_exhaust_temp_model(df):
    """
    Trains a simple linear regression model to predict exhaust gas temperature
    based on power output, CO₂ volumetric flow, and CH₄ mass flow.
    """
    df_clean = df[['power_output', 'exhaust_temp', 'calculated_ch4_share_percent', 'des_percent']].dropna()
    X = df_clean[['power_output', 'calculated_ch4_share_percent', 'des_percent']].values
    y = df_clean['exhaust_temp'].values

    # Create a pipeline with scaling and linear regression
    model = Pipeline([
        ('scaler', StandardScaler()),  # Standardize features
        ('regressor', LinearRegression())
    ])

    model.fit(X, y)

    return model
```