High Level Design(HLD)

Jet Engine Remaining Useful Life(RUL) Prediction

Revision Number: 1.0
Last date of revision: 27/03/2023
Written By: Abhijeet Srivastav

Document Version Control

Date Issued	Version	Description	Author
27/03/2023	1.0	Initialized HLD	Abhijeet Srivastav

Contents

Abstract

Jet engines are one of the most sophisticated pieces of tech that humankind has known in its history. They are key components of any modern jet either commercial or military and few missiles and rockets also.

Their failures are catastrophic and lead to loss of millions of dollars and precious life. But their maintenance is too costly and time consuming but needs to be done routinely hence costs a fortune to maintain and check them for any type of damage propagation due mechanical tear.

To reduce this cost of maintenance by calculating Remaining Utility Life of jet engines based on various sensor data we have designed a robust machine learning solution.

1. Introduction

1.1. Why this High Level Design Document ?

The purpose of this High Level Design Document (HLD) is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at high level.

The HLD will:

- Present all of the design aspects and define them in detail
- Describe the user interface being implemented
- Describe the hardware and software interfaces
- Describe the performance requirements
- Include design features and the architecture of the project
- List and describe the non-functional attributes like:
    - Security
    - Reliability
    - Maintainability
    - Portability
    - Reusability
    - Application compatibility
    - Resource utilization
    - Serviceability

1.2. Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

1.3. Definitions

Term	Description
RUL	Remaining Useful Life
Database	Collection of all the information monitored by the system
IDE	Integrated Development Environment

2. General Description

2.1. Product Perspective

The Jet Engine RUL prediction solution is a hyper tuned machine learning model which predicts the Remaining Useful life of a jet engine based on data collected by various on board sensors during the flight monitoring various attributes of the engine.

2.2. Problem Statement

To create an machine learning solution to predict the Remaining Useful Life (RUL) of a jet engine with very high confidence interval to:

- Minimize the servicing cost of jet engines without compromising the life of product and consumers.
- Avoiding any catastrophic accident costing money and lives

2.3. Proposed Solution

The solution proposed is a machine learning model that can be implemented to perform above mentioned use cases. It will predict the RUL of the engine giving the servicing team an estimate of the health of the engine based on which they can decide how much priority they should give to the engine. If an engine has very low RUL they can give more time to check deploy whether this engine should be used further or retired, or if an engine has very high RUL then they should perform a routine checkup.

This will reduce the time and money invested in maintenance as well as it will divert attention to only those engines who require it more leading to less accidents.

2.4. Further Improvements

This machine learning solution performance can further be improved by increasing the number of more onboard sensors. Onboard sensor data can be transmitted and analyzed in real time for engine health monitoring.

2.5. Technical Requirements

This document addresses the requirements for detecting the anomalies in the engine at the early stages and recommending the necessary and rapid action to avoid any accident.

- Onboard sensor array to monitor the engine stats
- Connectivity system for sensor to broadcast the data to base station
- Cloud server to host the solution

2.6. Data Requirements

Data requirement to implement this machine learning solution are:

- Sensor data from various sensors (at least 20) with 10000 records
- Sensor data should be paired setting configurations data
- We need to calculate RUL based on various sensor data and add it to the record
- Irrelevant or constant sensor should be dropped
- Missing values will be dealt with different imputation methods to find best suited ones
- Sensor data pattern should be analyzed

2.7. Tools Used

- VS Code used as IDE.
- For visualization matplotlib, seaborn and plotly has been used.
- Pandas, Numpy are used to handle the data processing.
- Sklearn is the main library used to build the models.
- Railway and vercel has been used for visualization.
- MongoDB is used as the main database.

- PowerBI has been used as a visualization tool.
- Flask has been utilized for UI and backend.
- Github is the version control system utilized.
- Docker is the packaging system used to sandbox the project.

2.8. Constraints

This RUL predicts machine learning solutions must have high accuracy with very high confidence intervals and UI should be user friendly, as automated as possible and user should not be required to know any of the workings.

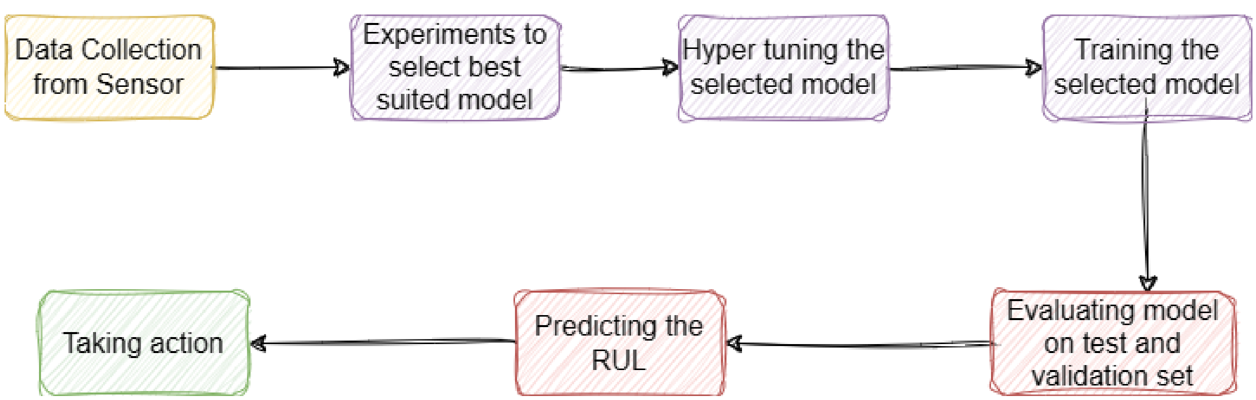
2.9. Assumptions

The main objective of the project is to implement the use cases as previously mentioned (2.2 Problem Statement) for new dataset that comes through flight onboard sensors. Machine Learning pipeline is used to predict the above mentioned se cases based on input data. It is also assumed that all aspects of this project have the ability to work together in the way the designer is expecting.

3. Design Details

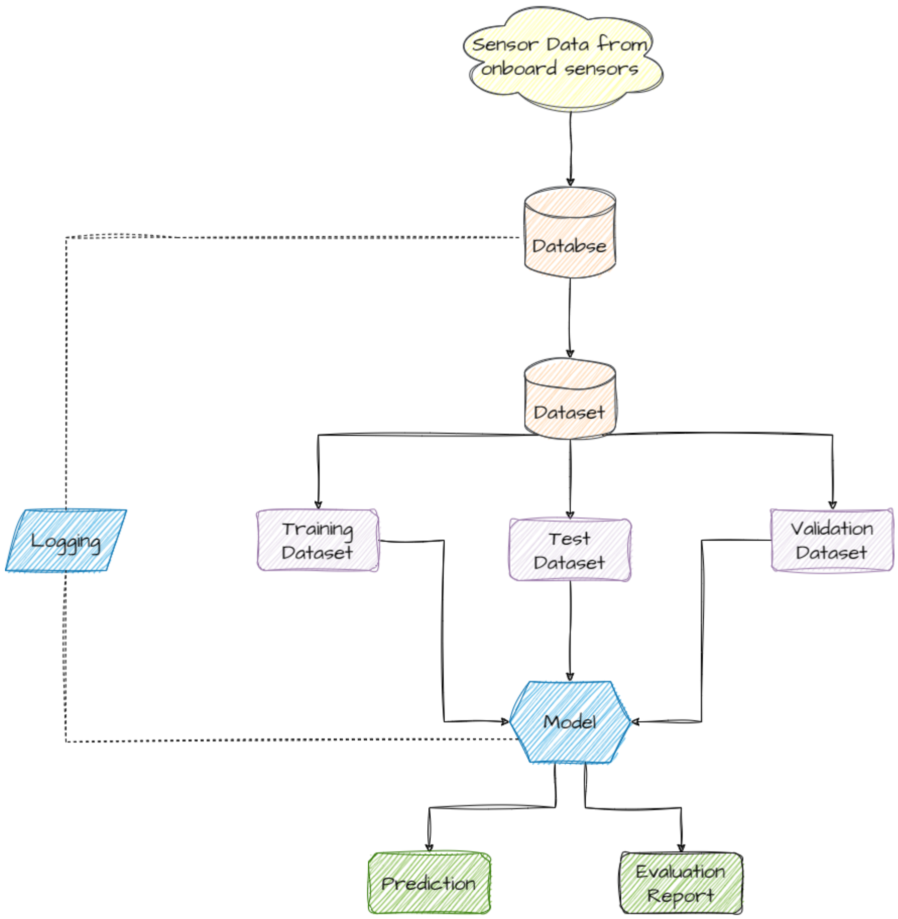
3.1. Process Flow

For identifying the different types of anomalies we will use a machine learning based model. Below is the process flow diagram.

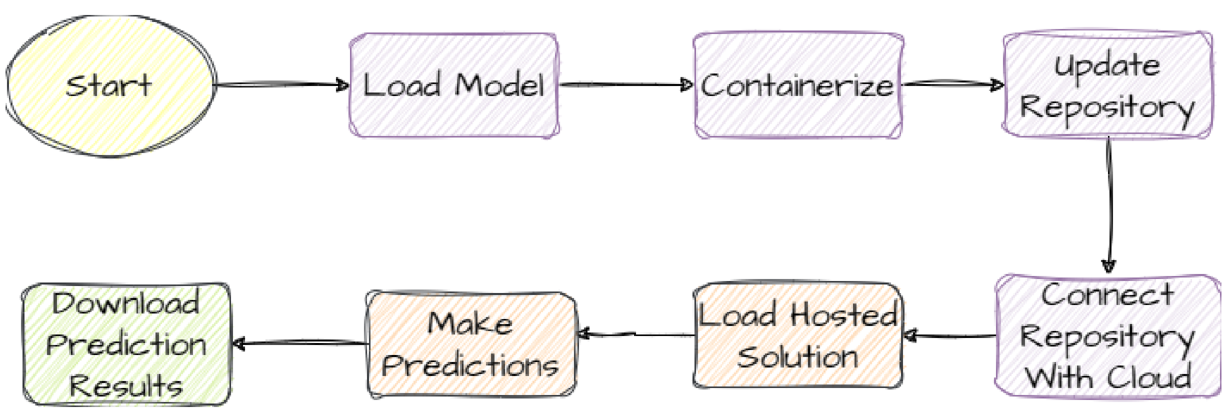


3.1.1. Model Training and Evaluation

An overview view of the deployable machine learning pipeline structure which has been finalized after several experimentations is depicted in the flow diagram below.



3.1.2. Deployment Process



3.2. Event Log

The system should love every event so that the user will know what process is running internally.

Initial Step-by-step description:

1. The system identifies at what step logging is required.
2. The system should be able to log each and every system flow.
3. Developers can choose logging methods. (i.e Database logging/File logging)
4. System should not hang even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

3.3. Error Handling

When errors are encountered, an explanation will be displayed as to what went wrong? An error will be defined as anything that falls outside the normal and intended usage of the system.

For handling of error a custom error handler will be developed to catch and display custom error messages.

4. Performance

The RUL prediction machine learning solution is used for detection of any anomaly and predict the remaining useful life of a jet engine which should be highly accurate as lives of people depend on it as well as high amount of money is consumed in servicing these engines.

Also the model retraining is very important to improve the performance.

4.1. Reusability

The code is written in modular fashion with various components specifically designed to perform a certain task and have the ability to be reused.

Various components of the RUL prediction system are:

- Data Ingestion
- Data Validation
- Data Transformation
- Model Trainer
- Model Evaluation
- Model Pusher

Input to each module are specified separately as **Configuration File** and output of each of the components are specified separately as **Artifact File**.

Except that we have various **Supporting Components** and **User Interface Components**.

4.2. Application Compatibility

The different components for this project will be using Python as an interface between each of them. Each component will have its own task to perform, and it is the job of Python to ensure proper transfer of information from component to component.

4.3. Resource Utilization

When any task is performed, it will likely use 60 percent of the processing power available until that function is finished, except the **Model Training** component which would take exponential time compared to other components.

UI Component is a completely web based solution so mainly consumes internet resources on the client side.

4.4. Deployment

We are deploying our solution on **Railway** and **Vercel** cloud platforms as of now.

5. Dashboards

Dashboards will be implemented to display the sensor data to monitor the change in the behavior of jet engines that could cause catastrophes of unimaginable impact.

As and when the system starts to capture the historical/ periodic data for a user, the dashboards will be included to display charts over time with progress on various factors.

6. Conclusion

The designed RUL prediction system will detect any anomaly in the jet engines and will predict their Remaining Useful life based on various sensor data coming from the flight onboard sensors which will be used to train our algorithm, so we can minimize the servicing cost of these jet engines and as well as prevent any accident that could cost lives.

7. References

1. https://ieeexplore.ieee.org/document/4711414/
2. Engines (nasa.gov)
3. scikit-learn: machine learning in Python — scikit-learn 1.2.2 documentation
4. Overview of ML Pipelines | Machine Learning | Google Developers