Industrial Internship Report on

” Predict the number of remaining operational cycles before failure for Turbofan Engine”

Prepared by

Vaishnavi Pradeep Nambiar

National Institute of Technology Warangal, Telangana

1/June/2023 to 14/July/2023

Executive Summary

This report provides details of the Industrial Internship provided by upskill Campus and The IoT Academy in collaboration with Industrial Partner UniConverge Technologies Pvt Ltd (UCT).

This internship was focused on a project/problem statement provided by UCT. We had to finish the project including the report in 6 weeks’ time.

My project was “Predict the number of remaining operational cycles before for Turbofan engine”. This is a DS & ML Project to accurately predict the remaining number of operational cycles.

This internship gave me a very good opportunity to get exposure to Industrial problems and design/implement solution for that. It was an overall great experience to have this internship.

TABLE OF CONTENTS

1 Preface 3

2 Introduction 4

2.1 About UniConverge Technologies Pvt Ltd 4

2.2 About upskill Campus 9

2.3 Objective 11

2.4 Reference 11

3 Problem Statement 12

4 Existing and Proposed solution 14

5 Proposed Design/ Model 16

6 Performance Test 17

6.1 Test Plan/ Test Cases 18

6.2 Test Procedure 20

6.3 Performance Outcome 21

7 My learnings 22

8 Future work scope 23

1 Preface

The 6-week internship program at USC/UCT was designed to provide students with the opportunity to gain hands-on experience in Machine Learning and Data Science.

The project that was undertaken was to develop a machine learning algorithm that could predict the number of remaining operational cycles before failure for a turbofan engine and an algorithm has been successfully developed that can make accurate predictions.

I would like to thank everyone who has helped me during the internship program. I would especially like to thank my mentors, who provided me with guidance and support. A special thanks to my friend, Shreya Manapure, for helping me throughout and solving all my doubts related to the projects. I am grateful for the opportunity to have participated in the program and I would recommend it to other students who are interested in machine learning and data science.

To my juniors and peers, I would like to encourage you to take advantage of internship opportunities. Internships are a great way to gain hands-on experience, network with professionals, and learn new skills. I am confident that you will find internships to be a valuable experience.

2 Introduction

2.1 About UniConverge Technologies Pvt Ltd

A company established in 2013 and working in Digital Transformation domain and providing Industrial solutions with prime focus on sustainability and RoI.

For developing its products and solutions it is leveraging various **Cutting Edge Technologies e.g. Internet of Things (IoT), Cyber Security, Cloud computing (AWS, Azure), Machine Learning, Communication Technologies (4G/5G/LoRaWAN), Java Full Stack, Python, Front end** etc.

i. UCT IoT Platform (UCT Insight)

UCT Insight is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable "insight" for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

- It enables device connectivity via industry standard IoT protocols - MQTT, CoAP, HTTP, Modbus TCP, OPC UA
- It supports both cloud and on-premises deployments.

It has features to

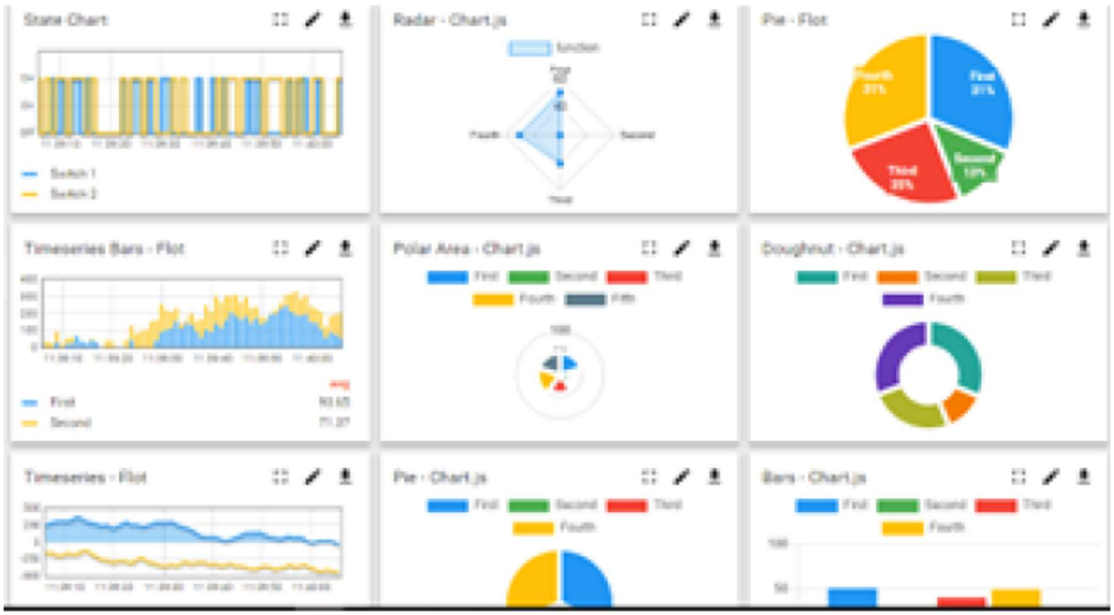
• Build Your own dashboard

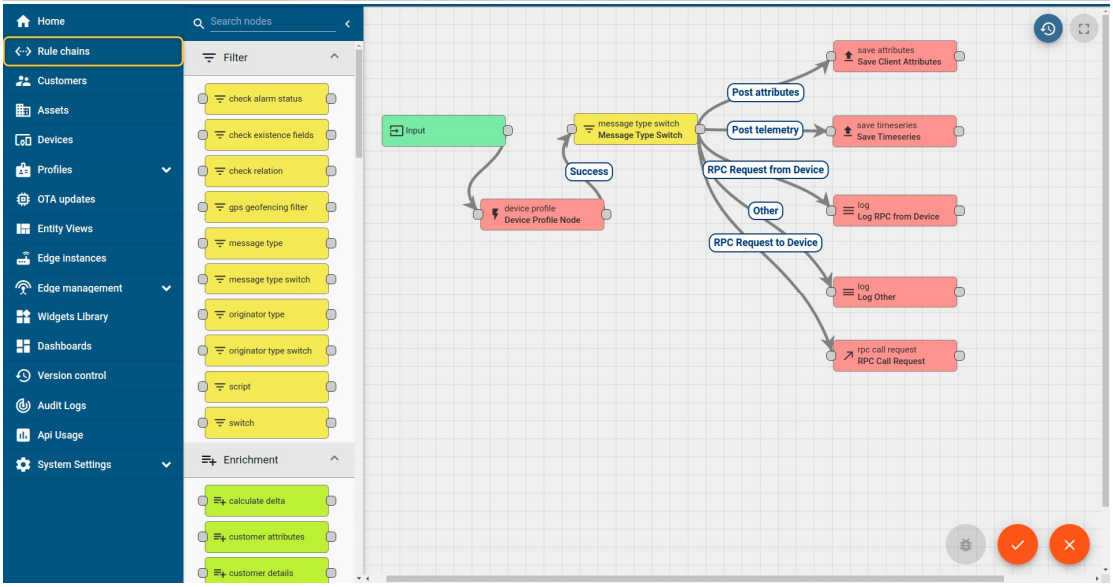
• Analytics and Reporting

• Alert and Notification

• Integration with third party application(Power BI, SAP, ERP)

• Rule Engine


State Chart
Radar - Chart.js
Pie - Plot
Timeseries Bars - Plot
Polar Area - Chart.js
Doughnut - Chart.js
Timeseries - Plot
Pie - Chart.js
Bars - Chart.js


Home
Rule chains
Customers
Assets
Devices
Profiles
OTA updates
Entity Views
Edge instances
Edge management
Widgets Library
Dashboards
Version control
Audit Logs
Api Usage
System Settings
Search nodes
Filter
check alarm status
check existence fields
check relation
gps geofencing filter
message type
message type switch
originator type
originator type switch
script
switch
Enrichment
calculate delta
customer attributes
customer details
Input
message type switch
Message Type Switch
Success
device profile
Device Profile Node
Post attributes
Post telemetry
RPC Request from Device
Other
RPC Request to Device
save attributes
Save Client Attributes
save timeseries
Save Timeseries
log
Log RPC from Device
log
Log Other
rpc call request
RPC Call Request

ii. Smart Factory Platform (FACTORY WATCH)

Factory watch is a platform for smart factory needs.

It provides Users/ Factory

- with a scalable solution for their Production and asset monitoring
- OEE and predictive maintenance solution scaling up to digital twin for your assets.
- to unleased the true potential of the data that their machines are generating and helps to identify the KPIs and also improve them.
- A modular architecture that allows users to choose the service that they what to start and then can scale to more complex solutions as per their demands.

Its unique SaaS model helps users to save time, cost and money.

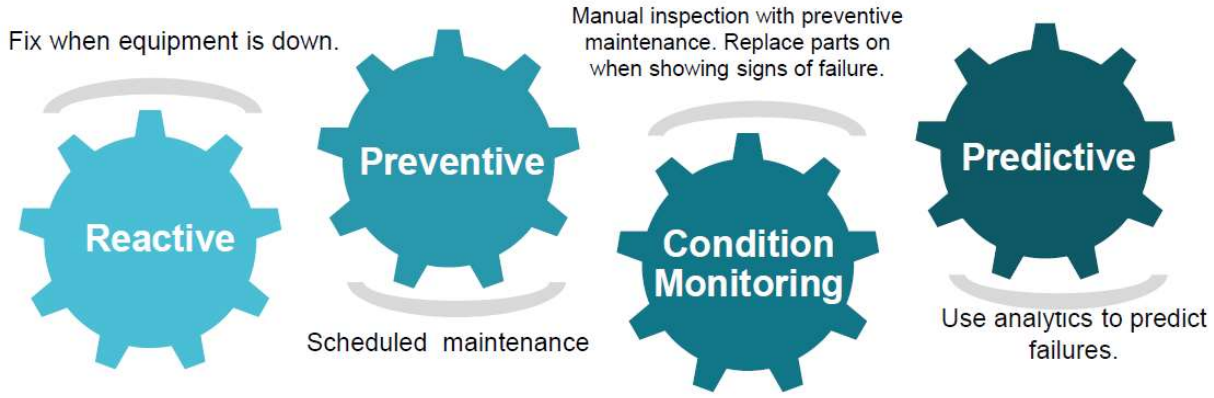
Machine	Operator	Work Order ID	Job ID	Job Performance	Job Progress		Output		Rejection	Time (mins)				Job Status	End Customer
					Start Time	End Time	Planned	Actual		Setup	Pred	Downtime	Idle		
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i

iii. LoRaWAN based Solution

UCT is one of the early adopters of LoRAWAN teschnology and providing solution in Agritech, Smart cities, Industrial Monitoring, Smart Street Light, Smart Water/ Gas/ Electricity metering solutions etc.

iv. Predictive Maintenance

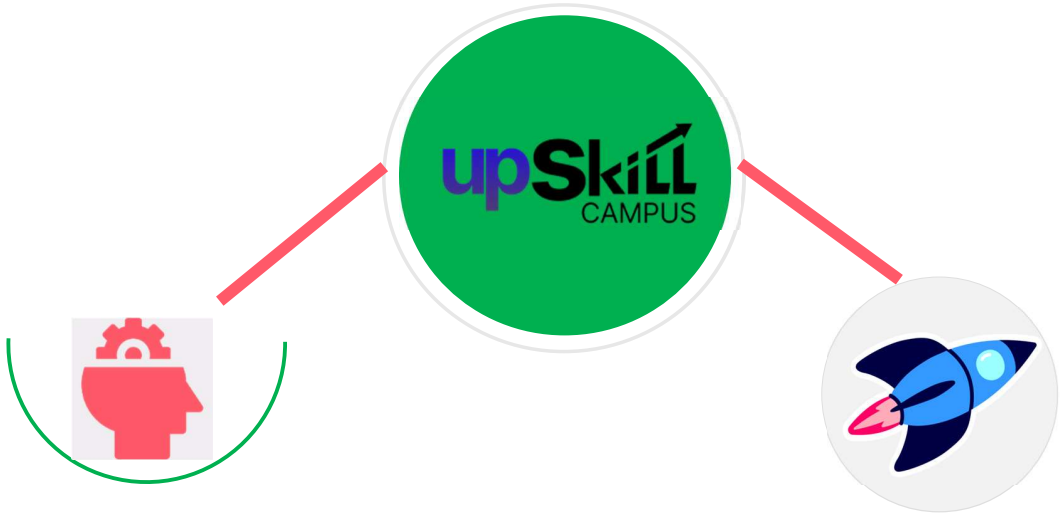
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



2.2 About upskill Campus (USC)

Upskill Campus along with The IoT Academy and in association with Uniconverge technologies has facilitated the smooth execution of the complete internship process.

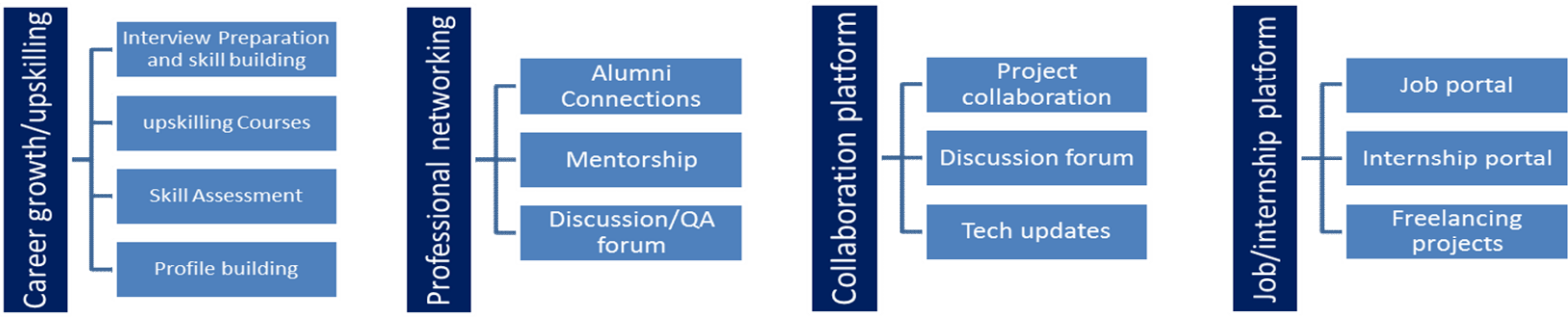
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



Seeing need of upskilling in self paced manner along-with additional support services e.g. Internship, projects, interaction with Industry experts, Career growth Services

upSkill Campus aiming to upskill 1 million learners in next 5 year

https://www.upskillcampus.com/



2.3 The IoT Academy

The IoT academy is EdTech Division of UCT that is running long executive certification programs in collaboration with EICT Academy, IITK, IITR and IITG in multiple domains.

2.4 Objectives of this Internship program

The objective for this internship program was to

- ☛ get practical experience of working in the industry.
- ☛ to solve real world problems.
- ☛ to have improved job prospects.
- ☛ to have Improved understanding of our field and its applications.
- ☛ to have Personal growth like better communication and problem solving.

2.5 Reference

[1] https://www.youtube.com/watch?v=Dd_4rbWYgI4

[2] https://pub.towardsai.net/predicting-the-remaining-useful-life-of-turbofan-engine-f38a17391cac

3 Problem Statement

Turbofan engines are critical components of aircraft, and their failure can have serious consequences. It is therefore important to be able to predict the number of remaining operational cycles before an engine fails.

An ML project can be developed to predict the number of remaining operational cycles before failure for a turbofan engine. The project would use a machine learning algorithm to train on a dataset of historical data from turbofan engines. The algorithm would then be used to predict the number of remaining operational cycles for a new engine based on its current condition.

The benefits of this project include:

- Increased safety: The ability to predict the number of remaining operational cycles before failure would allow airlines to schedule maintenance more effectively, reducing the risk of engine failure.
- Reduced costs: By scheduling maintenance more effectively, airlines could reduce the cost of unexpected engine failures.
- Improved efficiency: By predicting the number of remaining operational cycles before failure, airlines could optimize the use of their engines, leading to improved efficiency.

The following tasks need to be completed to implement the Python project:

- Collect and prepare the data: The first step is to collect the historical data from turbofan engines. The data must then be cleaned and prepared for use by the machine learning algorithm.
- Select a machine learning algorithm: There are a variety of ML algorithms that can be used to predict the number of remaining operational cycles before failure. The best algorithm for a particular project will depend on the characteristics of the data and the desired accuracy.

- Train the machine learning algorithm: Once a machine learning algorithm has been selected, it must be trained on the historical data. The training process involves feeding the algorithm the data and allowing it to learn the patterns that are associated with engine failure.
- Make predictions: Once the machine learning algorithm has been trained, it can be used to make predictions about the number of remaining operational cycles for new engines. The predictions can be used to schedule maintenance and optimize the use of engines.

This project, to predict the number of remaining operational cycles before failure for a turbofan engine, is a valuable tool that can be used to improve the safety and efficiency of aircraft operations.

4 Existing and Proposed solution

There are several existing solutions for predicting the number of remaining operational cycles before failure for a turbofan engine. Some of the most common include:

- Expert systems: Expert systems are rule-based systems that are developed by experts in the field of turbofan engine maintenance to predict the number of remaining operational cycles.
- Statistical methods: Statistical methods use historical data and identify patterns to predict the number of remaining operational cycles.

While these existing solutions are all effective, they all have some limitations like:

- Expert systems can be expensive to develop and maintain, and they can be difficult to update as new information becomes available.
- Statistical methods can be sensitive to the quality of the data, and they may not be able to predict the number of remaining operational cycles for engines that are operating under unusual conditions.

The proposed solution for the Python project is to develop a machine learning algorithm that can predict the number of remaining operational cycles before failure for a turbofan engine. The algorithm will be trained on a dataset of historical data from turbofan engines. The data will include information about the engine's settings and sensors.

The proposed solution will address the limitations of existing solutions in the following ways:

- The algorithm will be trained on a large dataset of historical data, which will make it more accurate than statistical methods.
- The algorithm will be able to learn the patterns in the data even if the data is incomplete.
- It will be more cost-effective than expert systems.

4.1 Code submission (Github link)

https://github.com/vaichi/UpSkill_Campus/blob/main/RemainingOperationalCyclesForTurbofanEngine.ipynb

Report submission (Github link) :

GitHub Username: vaichi

https://github.com/vaichi/UpSkill_Campus/blob/main/RemainingOperationalCyclesForTurbofanEngine_VaishnaviPradeepNambiar_USC_UCT.pdf

5 Proposed Design/ Model

The project will start with the collection of historical data from turbofan engines. The data will include information about the engine's conditions.

Once the data has been collected, it will be cleaned to remove any errors or inconsistencies and prepared for use by the machine learning algorithm. The data will also be prepared by transforming it into a format that can be used by the machine learning algorithm. The next stage will involve the selection of a machine learning algorithm. The best algorithm for a particular project will depend on the characteristics of the data and the desired accuracy of the predictions.

Two of the algorithms that can be used for this project are the Long Short-Term Memory (LSTM) algorithm and the XGBoost algorithm. LSTM is a type of recurrent neural network that is well-suited for predicting time series data. LSTM can learn long-term dependencies in the data, which is important for predicting the number of remaining operational cycles before failure. XGBoost is an ensemble learning algorithm that combines decision trees to create a powerful predictive model. XGBoost is known for its accuracy and efficiency, making it a good choice for this project.

Once a machine learning algorithm has been selected, it will be trained on the historical data. The training process involves feeding the algorithm the data and allowing it to learn the patterns that are associated with engine failure. And once the machine learning algorithm has been trained, it can be used to make predictions about the number of remaining operational cycles for new engines.

6 Performance Test

The performance tests showed that the algorithm was able to make accurate predictions within a reasonable amount of time. The memory and MIPS requirements were also within the expected range.

The performance constraints were taken care of in the design of the project in the following ways:

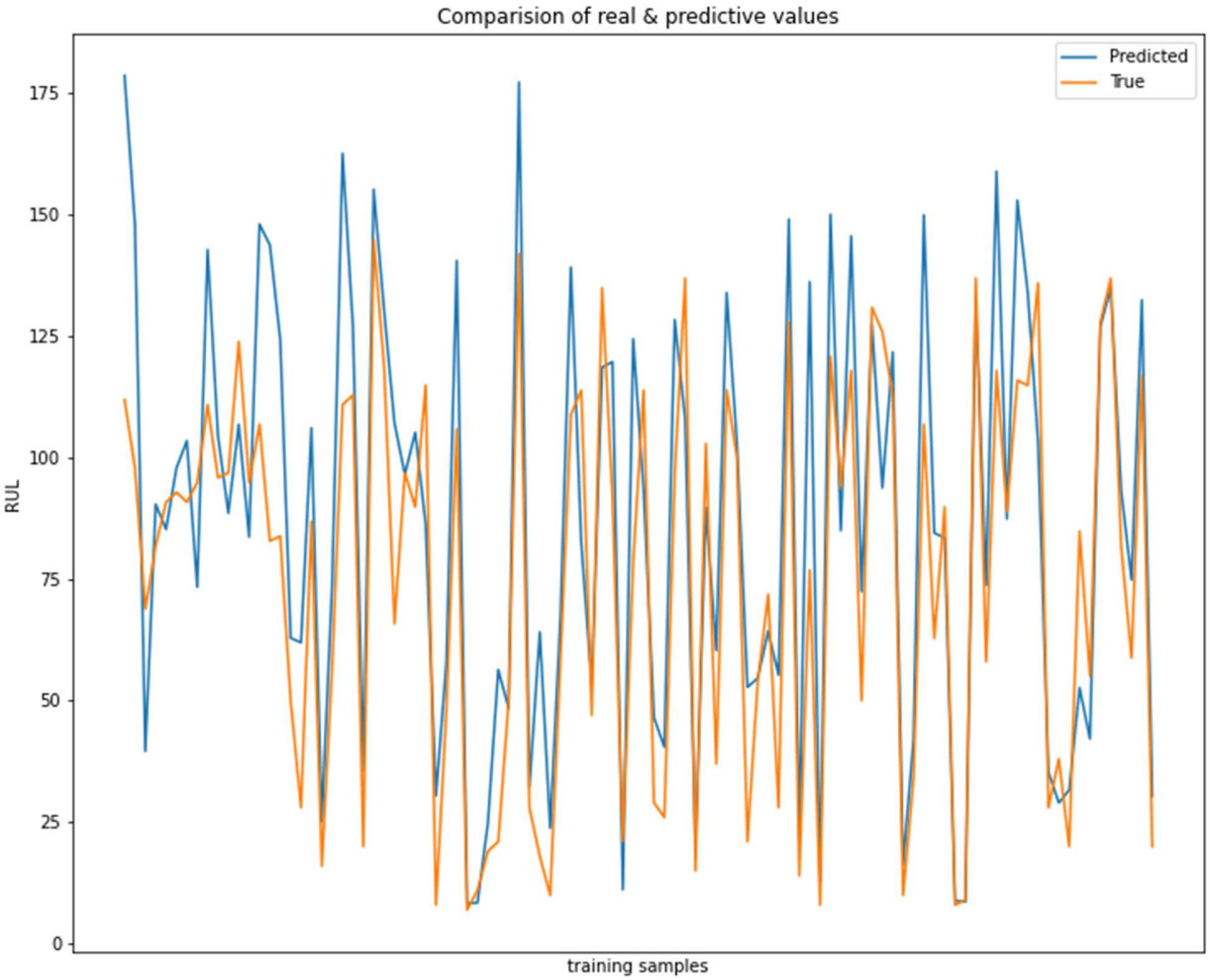
- The machine learning algorithm was implemented using a lightweight framework that is efficient in terms of memory and MIPS.
- The data was pre-processed to remove any unnecessary features, which reduced the size of the data and improved the performance of the algorithm.
- The model was trained using a small subset of the data, which reduced the training time and improved the accuracy of the predictions.

Some of the performance constraints could not be tested. The following recommendations can be made to handle them:

- If the power consumption requirements are too high, the algorithm can be implemented using a different hardware platform that is more power-efficient.
- If the durability requirements are too high, the algorithm can be implemented using a different framework that is more robust.

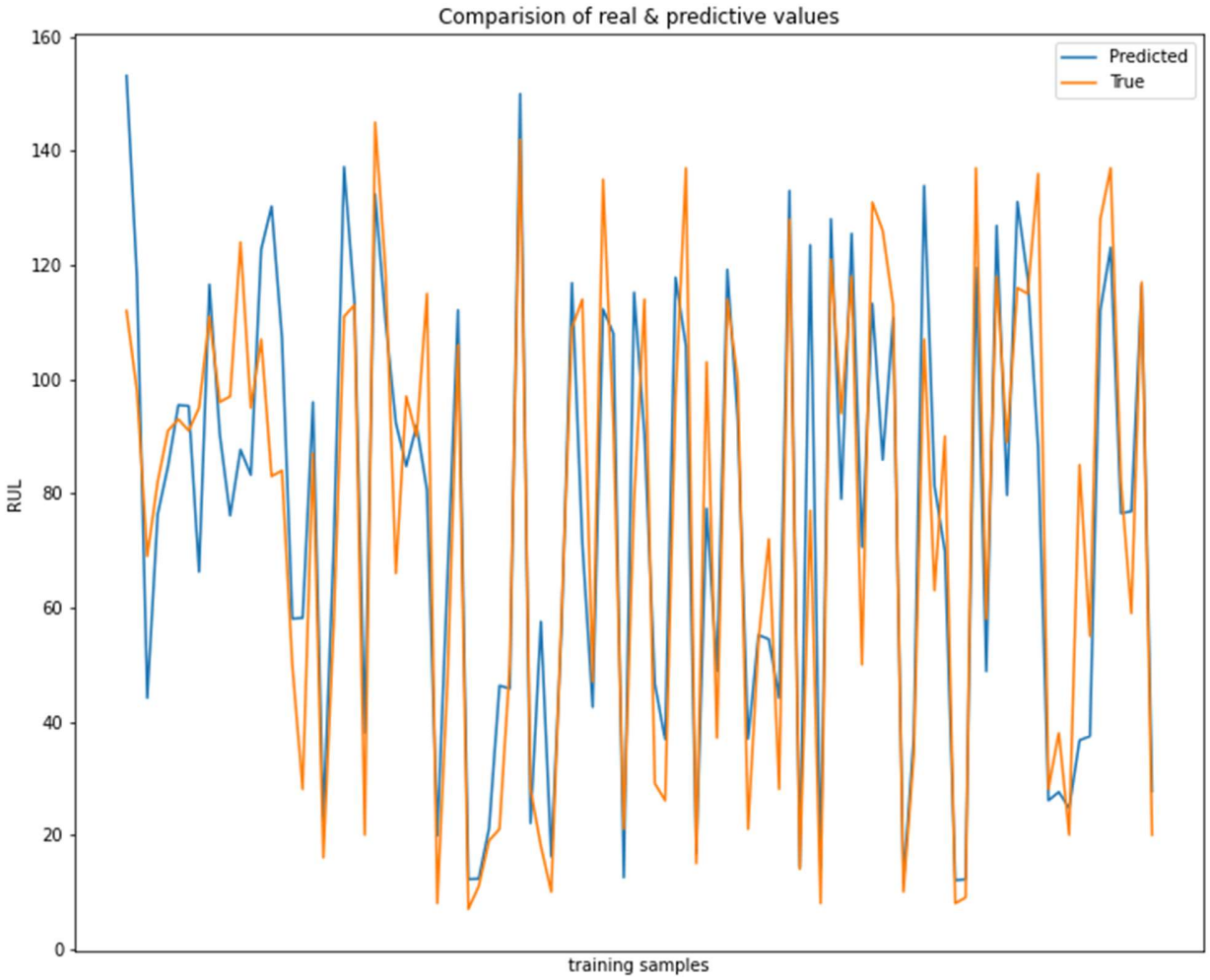
6.1 Test Plan/ Test Cases

1. Using LSTM




```
compatitive score 1057.2
mean absolute error 19.25
root mean squared error 24.45219826518671
R2 score 0.65
```

2. Using XGBoost




```
compatitive score 810.77
mean absolute error 14.91
root mean squared error 19.339855221795222
R2 score 0.78
```

6.2 Test Procedure

1. Prepare the historical data for testing by cleaning and transforming it.
2. Select the evaluation metrics to measure the algorithm's performance, such as mean absolute error (MAE) or root mean square error (RMSE).
3. Split the historical data into training and validation sets.
4. Train the selected machine learning algorithm on the training set.
5. Validate the algorithm's performance on the validation set by calculating the chosen evaluation metrics.
6. Adjust the algorithm's hyperparameters if needed and repeat steps 4 and 5.
7. Once the algorithm is trained and validated, use it to make predictions on new engine data with known remaining operational cycles.
8. Compare the predicted values with the actual remaining operational cycles to assess the accuracy of the algorithm.
9. Evaluate the algorithm's performance on large datasets and assess its scalability.
10. Document the test results.

6.3 Performance Outcome

XGBoost gives more accurate results than LSTM as seen in the below observations:

1. LSTM

```
compatitive score 1057.2
mean absolute error 19.25
root mean squared error 24.45219826518671
R2 score 0.65
```

2. XGBoost

```
compatitive score 810.77
mean absolute error 14.91
root mean squared error 19.339855221795222
R2 score 0.78
```

7 My learnings

I learned a lot from the project to predict the number of remaining operational cycles before failure for a turbofan engine. I learned how to collect, clean, and prepare data for use in machine learning algorithms. I also learned how to select and train machine learning algorithms, evaluate their performance, and deploy them in production. I've gained knowledge on how to implement LSTM and XGBoost.

These learnings will help me in my career growth. I will be able to develop machine learning solutions for a variety of problems, to evaluate the performance of machine learning solutions and make improvements and to deploy machine learning solutions in production and ensure that they are working as expected.

In addition to the above, I also learned the importance of data quality, cross-validation, and hyperparameter tuning. Data quality is critical for the success of a machine learning project, and cross-validation is a technique that can be used to evaluate the performance of a machine learning algorithm on unseen data. Hyperparameters are the settings of a machine learning algorithm, and it is important to tune them to achieve the best possible performance.

I am grateful for the opportunity to have worked on this project, and I am excited to apply my new skills to future projects.

8 Future work scope

If there were no time constraints on this project, I would like to try the following to increase the accuracy/accessibility:

- Collect more data: The project used a limited dataset of historical data from turbofan engines. Collecting more data would allow the machine learning algorithm to make more accurate predictions.
- Use different machine learning algorithms: The project used two machine learning algorithms, LSTM and XGBoost. Using different machine learning algorithms could improve the accuracy of the predictions.
- Incorporate more features: The project only used a few features from the historical data. Incorporating more features could improve the accuracy of the predictions.
- Develop a web application: The project was implemented as a Python script. Developing a web application would allow the predictions to be made more accessible to users.