Aero Engine Remaining Useful Life Prediction and Defect Detection using Deep learning and Machine Learning

MSc Research Project
Data Analytics

Akhil Anthony
Student ID: x22209395

School of Computing
National College of Ireland

Supervisor: Mr.Hicham Rifai

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Akhil Anthony
Student ID:	x22209395
Programme:	Data Analytics
Year:	2024
Module:	MSc Research Project
Supervisor:	Mr.Hicham Rifai
Submission Due Date:	15/09/2024
Project Title:	Aero Engine Remaining Useful Life Prediction and Defect Detection using Deep learning and Machine Learning
Word Count:	9862
Page Count:	24

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	16th September 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Aero Engine Remaining Useful Life Prediction and Defect Detection using Deep learning and Machine Learning

Akhil Anthony
x22209395

Abstract

In the rapidly evolving aviation industry, ensuring safety and efficiency of aircraft's is essential. This research investigates whether the integration of image processing algorithms and predictive maintenance algorithms can enhance the efficiency and reliability of aircraft engine maintenance operations. This paper addresses vulnerabilities in the traditional maintenance practices by focusing on real-life incidents which highlight the need for better maintenance systems which can be used during manufacturing and during the flights. This paper proposes a comprehensive system using hybrid models CNN-BiLSTM, which is a combination of deep learning models for RUL prediction and a hybrid model VGG16-SVM, which is a combination of deep learning and machine learning model for defect detection. The evaluation of the models showed CNN-BiLSTM model obtained an MAE of 48.03 and the VGG-SVM model achieved an overall accuracy of 90.79% The developed web application, designed for maintenance personnel demonstrates the systems ability to transform the aerospace maintenance. The findings indicate that integration of advanced image processing and predictive maintenance algorithms can improve the safety and efficiency of the aircraft engines. Thus the study proves, the combination of image processing and predictive maintenance techniques can enhance the aerospace maintenance operations.

1 Introduction

In the aerospace industry, the maintenance processes have significant impact on safety and reliability of aircraft's. Recently, a JetBlue Airbus A321neo which is fitted with a Pratt & Whitney PW110G engine encountered an issue and had to be brought down through an emergency landing in Shannon, Ireland. Even though the aircraft was delivered recently, this incident put-forward the critical need for advanced maintenance and monitoring systems for aircraft engines which can be used during manufacturing process and throughout their operational life. It highlights weakness of today's maintenance strategies and stresses the importance of automatic defect detection during the production, in addition to the continuous monitoring during the flight to help in early detection of issues as well as precise estimation of the life expectancy of the engine components. Developing systems that can easily identify flaws and accurately predict the remaining life of the engine is crucial to ensure safety and reliability in aviation. Frequent aircraft engine maintenance is a fundamental requirement in maintaining the reliability of aircraft's.

Figure 1: Turbo Engine Maintenance

Turbofan engines are the core component of aircraft and they operate under very harsh environments which result in high mechanical stress. These engines need to be regularly maintained and monitored to guarantee their reliability as they are crucial for the safe functioning of aircraft((Yang et al.; 2022)). Nonetheless, there are several obstacles in maintaining these engines, ranging from accurately calculating the Remaining Useful Life (RUL) to identifying flows in engine parts, especially the engine blades. The combination of these challenges calls for a novel strategy that can enhance maintenance operations reliability and efficiency.The complexity of the engines and the challenging operating conditions they face cause wear and tear, and these are the root causes of the maintenance issues occurring with turbofan engines. The conventional methods of maintenance, which often rely on planned examination and reactive fixes, have proven inadequate in preventing unexpected failures.
Unexpected engine failure can be expensive due to repair costs, airport charges, and potential catastrophic consequences. The aviation industries try to mitigate these risks while upholding strict requirements for efficiency and reliability. The integration of innovative technologies into maintenance practices offers a chance to address these challenges more effectively. Should the issue be resolved or even somewhat alleviated, there would be significant advantages. Predictive maintenance can enhance safety by proactively detecting possible failures. In addition to safety , predictive maintenance can reduce operational cost by minimizing pointless inspections and extending the lifespan of engine components. Furthermore, the mitigation of unforeseen downtime would improve airline operations overall efficiency, which contribute to higher profitability and customer satisfaction.
Despite the possible advantages, there is still work to be done in identifying fan blade flaws and accurately forecasting the remaining useful life of the turbofan engines. The biggest challenge is the intricate and nonlinear process of engine deterioration, which is influenced by various factors such as material properties, operating conditions and environmental impact((Abdulrahman et al.; 2023)). Furthermore, the detection of defects especially in engine fan blades require high precision due to the small size of the damage. The technical side of the issue contains several sophisticated and interrelated processes. The predictive maintenance relies on advanced data analytics and on various machine learning models to process and understand vast amounts of sensor data collected from the turbofan engines. Hybrid models like CNN-BiLSTM, which combine the strength of CNN and BiLSTM have proven to be very effective in handling time-series data and providing accurate Remaining Useful Life (RUL)predictions.
The application of image processing and computer vision technology is essential for de-

fect detection in engine blades. Traditional methods are useful and still have limitations in terms of efficiency and accuracy. The deep learning and machine learning models have developed significantly in detecting defects in images and classifying them. These models are trained on an extensive collection of images, they can detect various patterns and even the smallest flows that a human inspector might miss((Abdulrahman et al.; 2023)). Among these developments the VG16-SVM hybrid model which combines the advantages of CNN and traditional machine learning language has become a powerful technique. The combination of high accuracy of deep learning in feature extraction along with robust classification capabilities of SVMs, result in enhanced performance in image classification tasks. These points show that theVG16-SVM hybrid model will be more suitable for defect detection.

The goal of this research project is to develop a comprehensive system that combines image processing techniques like VG16-SVM hybrid for defect detection in turbofan blades and a predictive algorithm CNN-BiLSTM hybrid models for forecasting the remaining useful life of the engine. The combination of these innovative technologies reduces the gap in ongoing maintenance practices dramatically along with enhancing both accuracy and promptness of the maintenance measures. Taking preventive measures before conditions worsen and result in catastrophic failures have become easier with the use of image processing techniques which help in describing the defects in engine blades. A user-friendly application will be developed using Streamlit for making this system practical and accessible. This application will provide a interface where user can input sensor data and predict the remaining useful life and also detect the defect in the turbofan engines by uploading the images, making the system accessible to maintenance personals and decision makers

A thorough analysis of prior studies reveals the fundamental concepts on which this work is built. Predictive maintenance technology is becoming more and more necessary in the aerospace industry. Concurrently, improvements in image processing technology have demonstrated considerable potential for defect identifications. However combining these disparate methods into a single framework for turbofan engine maintenance is a novel field of study that could provide revolutionary results. In light of the points raised, the main research question is : "Can the efficiency and reliability of aerospace maintenance operations be enhanced by integrating advanced image processing algorithms for defect detection in turbofan blades with predictive maintenance techniques for assessing the remaining useful life of turbofan engines?" The objective of this study is to develop a web based application system which combines image classification and predictive maintenance for achieving advancements in aerospace maintenance procedures.

The detailed flow of the research is as follows. Section 2 covers the review of the related research done in this field. Section 3 explains the methodology for RUL prediction, Defect Detection and Web Application.The design specification is covered in section 4. The implementation and Evaluation is depicted in section 5 and 6. Finally, the section 7 shows the conclusion and future work.

2 Related Work

Pronostic and Health Management (PHM) is a crucial element in maintaining the safety and reliability of an aircraft((Ensarioğlu et al.; 2023)). Remaining Useful Life estimation and Defect Detection are the two important elements of PHM. PHM makes sure the systems work properly and prevent failures and reduce maintenance cost((Ramasso; 2014)).Prognostics focus on obtaining the health details of the system and the remaining life of the components inside the system. RUL prediction focuses on the remaining life of the aircraft engine, meanwhile defect detection looks for different damages present on the engine blade. Both RUL and defect detection methods can be classified under Model-based, Data-Driven, and Hybrid Models((Ensarioğlu et al.; 2023)).This literature review explores recent developments in RUL estimation techniques and defect detection methodologies.

2.1 Remaining Useful Life (RUL) Estimation

2.1.1 Model - based

Estimating Remaining Useful Life (RUL) is essential in prognostic and health management to optimize maintenance plans and guarantee system dependability particularly in intricate fields like aeronautical engineering. Jonathan and Christian ((Garay and Diedrich; 2019)) proposed a study that tackles the challenges of applying stochastic methods for RUL estimation. The Bayesian prognosis, wiener processes and Monte Carlo simulation were adapted to predict the remaining cycles before failure. The method proved to be highly accurate, with results falling to underestimated confidence intervals in 80% of cases. Focusing on the remaining cycles, this approach improves overall efficiency and reduces downtime. Similarly, a web-based tool designed by Daniel, Bernadette, and Alberto ((Azevedo et al.; 2019)) for simulating a Prognostic Health Management (PHM) system focuses on the Remaining Useful Life prediction of specific components. Three AI inspired methodologies are implemented, a similarity-based method, a neural network and an extrapolation-based method. Despite the contributions there are limitations, as the model is tested on synthetic data, it may not capture the complexities and variabilities present in the real-world data also as the dataset increases the resources required also increase and it can impact the tools usability in real world scenarios.

2.1.2 Data-driven Approaches

Han, Niu and Wang ((Bingjie et al.; 2021)) introduced Similarity based methods which are useful while implementing global models due to system complexity. They used historical degradation data to predict the future performance. They used a system which incorporated operating condition clustering and information fusion from multiple sensors. They validated the model using the aero turbofan engine which demonstrated high accuracy and robustness, which had a Root Mean Square Error of 25.4, which reflected the performance and highlighted the method's effectiveness. The experiment conducted by Jianguo, Yujie, Xiao, Liying, Dong, Wenyou, Xiaochu, and Jinglin ((Cui et al.; 2022)) integrated the Sequenze- and-Excitation mechanism with a bidirectional long short-term memory network (BiLSTM), creating a robust SE-BiLSTM network. The model was trained with the NASA's engine performance degradation dataset and compared with the traditional models which include BILSTM, LSTMBS, and LSTM, and the SE-BiLSTM significantly reduced the root mean square error by 9.01%, 16.59%, and 23.05% respectively. These results show the model's superior predictive capability.
In a related approach, Shuan, Yucai, Jipeng, Jian ((Zhao et al.; 2022)) combined an Attention mechanism with s Long Short-Term Memory (LSTM) and this model optimized the Remaining useful life prediction by selecting the key features. The model

employed L2 regulation early stopping mechanism and dropout to prevent overfitting and the model achieved a lower RMSE of 13.25, 22.57, 12.98, 23.88. The integration of the Multi-head Attention system with LSTM increases the computational complexity and also the model's performance and scalability will be affected when applied to real time application data. Similarly, Hilal, Sevinc, and Kadir((Tekgöz et al.; 2022)) examined various Machine learning like Random Forest(RF), Support Vector Regression(SVR), Multi-Layer Perceptron (MLKP), AdaBoost, and Gradient Boosting Regression(GBR) and Deep learning model like Long Short Term Memory(LSTM), Gated Recurrent Unit(GRU), Convolutional Neural Network(CNN), CNN-LSTM, CNN-GRU and Temporal Convolutional Network. This study gives an understanding on selecting the appropriate model based on the characteristics of the data.
The study conducted in the year 2023 by Xiaojun, Yunpeng and Gong((Bai et al.; 2023)),focus on advance machine learning.Faced with the difficulty of inadequate feature extraction and inaccurate predictions inspired the researchers to create a fusion model that integrated features of attention mechanisms, stacked noise reduction, and Bidirectional Gated Recurrent Units (BiGRU). This model assigned weight to sensor data, extracted vital features, and predicted time series data with high accuracy. The model is optimized using the Bayesian algorithm and the model validated using the CMPASS dataset and achieved an RMSE of 13.51 and a score of 252.61. Moreover, Sijie, Nan, Jin, and Yafeng((Liu et al.; 2023)) introduced a Condition - Based Maintenance (CBM) system by using the Transformer-GRU network. By combining the Transformer encoding capability and GRU decoder the proposed model captures information from the sensor data. The proposed system having an RMSE of 12.38 shows the superior performance by the proposed model. The use of Transformer components can be computationally intensive, and this limits the practical application in environments with limited computational resources.
Finally, the study by Vaasudev, Riyansh, Sharanya((Sharma et al.; 2024)), Shows the importance of Artificial Intelligence and Machine Learning by using a XGBoost model for finding the remaining useful life of the turbofan engines, which had an RMSE of 15.44 and MAE of 8.97. These values prove that this model outperformed various traditional methods.The requirement of high computational requirements and the presence of noisy data make it difficult while applying the proposed model in settings with little resources.

2.1.3 Hybrid Model

A study conducted in the year 2023, by Kiymet, Tulin and Erdal((Ensarioğlu et al.; 2023)) introduces a hybrid model that combines One Dimensional Convolutional Neural Network and Long Short Term Memory(LSTM) for better performance. The 1D-CNN-LSTM model acquired and RMSE value of 16.1. The performance of the model relies on the techniques difference-based feature construction and change point detection based PWL labeling. VAriations or errors in these will have a huge impact on the performance of the model. All the reviewed studies contribute to my project, which is to develop a web application for remaining useful life (RUL) prediction and defect detection.

2.2 Defect Detection

Similar to the Remaining Useful Life estimation the defect detection can also be categorized under model-based, Data-Driven and Hybrid models.

2.2.1 Model-Based

Jing, Chang, Zhi-Yuan, Luo-Dan, and Ying-tao((Li et al.; 2020)) proposed a hybrid model combining Delaunay triangulation and a mesh growth algorithm to achieve the 3-D reconstruction of a defect point cloud from ultrasonic C-Scan data. The model had a few different limitations which include Error in Reconstruction mesh and measurement errors. But the study provided a better understanding of the hybrid model and gave the idea of using the hybrid model in my research.

2.2.2 Data-Driven Approaches

The paper proposed by kechen, Xiangkun, Shuai, and Yunhui ((Song et al.; 2023)) addresses the challenges in quality inspection of aero engine blades. They propose a solution which integrates Cross layer semantic guide network (CSGNet), YOLOv6 and Furthest Dynamic Copy Paste (FDCP) data augmentation method CSGNet achieved 2.3% more than compared to the YOLOv6. Even with these developments the detection speed requires further optimization while using real time data, and the complexity of the proposed model demands higher hardware adaptability. Compared to VGG16, the VGG16 layers can automatically extract the features in detail and easily improve the detection of small defects.
In another approach, a vision-based framework for aero engine blade surface defect detection proposed by Dawei, Yida, Qian, Yuxiang, Zhenghao, and Jun ((Li et al.; 2021)) acquired superior results when compared with traditional models it acquired 93.5% accuracy, precision of 94.8%, recall of 96.1% and an F1 score of 95.4%. Even with these outstanding results the chance for overfitting especially due to the lack of data. Chuhan and Haiyong ((Wang and Chen; 2023)) introduced an Efficient Edge Detection Network (EEDN) which addresses the key challenges such as micro sized defect features and varying defect scales using depth wise separable convolutions and a Multiscale Feature Enhanced Attention (MFEA) module. The proposed approach improved multiscale expressiveness and captured long range channel information and achieved superior performance with 0.859 ODS and 97 FPS. Despite these advancements these models can struggle with the precise classification of defects if the size and variability of defects are small, which lead to missed detection or false positives.
In the year 2023 Yusra, Mohammed, Abdulla, Brain, and Yahya ((Abdulrahman et al.; 2023)) conducted a systematic review which offers comprehensive insights into various deep learning models, a total of 13 primary studies were conducted. The CNN with Feature Point Extraction acquired an accuracy of 95.2%, the CNN framework acquired an accuracy of 0.9803, FWNet acquired 89.4%, Coarse to fine detection system had 93.5%, The Mask R-CNN had 0.82, ResNet CNN had a precision of 63%,enhance YOLOv5 and adapted YOLOv5 respectively had 98.3% and 93% accuracy, GPTNet had an accuracy of 84.9%, Enhanced Mask R-CNN had 60.4%, the ensemble learning model had f1 score of 0.77, DBFF-YOLOv4 had 96.7% of precision and finally the GAN-based Detection had an accuracy of 0.911. This paper provides a clear understanding of the traditional methods, and this review is instrumental while selecting appropriate models for various projects including my research which employs a VGG16-SVN model.

2.2.3 Hybrid Models

The paper proposed by yunfeng, Min, Yiqiong, Xueping ((Ma et al.; 2024)) addresses the challenging problem of automated automated inspection of aero engine blades which focus on tiny and weak surfaces. They proposed a SPDP-Net (Semantic Prior Guided Defect Perception Network), the solution acquired a precision of 95.9 and a recall of 94.0. Due to the sophisticated modules in this model, it can lead to higher computational requirements, and it can limit its real time performance and as the generalizability of the model to entirely new or unseen data in different environments cannot be predicted. The VGG16-SVM model provides a more computational efficiency and a better alternative.
The limitations and challenges identified in various models for surface defect detection in aero engine blades such as complexity, data dependency, and hardware requirements as well as the need for robust real time detection highlighted by CSGNet((Li et al.; 2020)) and other papers inspired me to explore alternative solutions. These models required high computational resources and extensive data preprocessing. These reasons gave me the idea of using the VGG16-SVM model for my classification task. This approach is not only computationally efficient but also adaptive to various environments.
The article by Fatine, Raed, Niamat, Marc, Chad, Safae ((Elakramine et al.; 2022)) Put forward a fundamental approach for integrating MBSE in aircraft maintenance. The study highlights the potential of MBSE by showcasing the application of SysML in modeling and analyzing complex maintenance systems. An integrated aircraft maintenance training platform that combines simulation training with auxiliary operations to enhance maintenance efficiency and reduce errors was introduced by Wei, Dan and Yanfu((Zhang et al.; 2023)). The system also tries to prevent human errors which play a vital role in flight delays and cancellations.
From the studies by Wei and Fatine show that advanced modeling techniques can significantly improve overall maintenance efficiency. This idea, along with the insights from other papers in defect detection and remaining useful life prediction has inspired the development of an advanced web-based system. With the combination of both Remaining Useful Life estimation and Defect detection this system can provide a holistic solution for aircraft maintenance, which can help in reducing maintenance errors and improved operational efficiency.

3 Methodology

This research focuses on the development of a web based application system for forecasting Remaining Useful Life (RUL) and detecting various Defects present in the engine blades. This methodology includes, data collection, preprocessing, and model building of RUL prediction and data preprocessing and model building of Defect Detection, and Web Application.

3.1 Data Collection

Obtaining good quality datasets is crucial for the development of machine learning and deep learning models. This research employs two datasets from NASA's Open Data Portal and Kaggle to build the web based application. The dataset from NASA's open data portal data addresses the prediction of Remaining Useful Life (RUL) and the data

set from Kaggle addresses Defect Detection.
The first dataset, CMAPSS is secured from NASA's open data portal. In which the FD001 data is used for predicting Remaining Useful Life (RUL).Figure 2 shows the contents of data. The dataset includes various multivariate time series, which represent the operational data of different engines from a fleet of the same type. The dataset contain 26 columns which intricate the engine id, cycles, operational settings and sensor data. The second dataset acquired from Kaggle is designed for studying aeroengine defect de-

No.of sensors	Symbols	Description	Units
1	T2	Total temperature at fan inlet	°R
2	T24	Total temperature at LPC outlet	°R
3	T30	Total temperature at HPC outlet	°R
4	T50	Total temperature at LPT outlet	°R
5	P2	Pressure at fan inlet	psia
6	P15	Total pressure in bypass-duct	psia
7	P30	Total pressure at HPC outlet	psia
8	Nf	Physical fan speed	rpm
9	Nc	Physical core speed	rpm
10	Epr	Engine pressure ratio(P50/P2)	-
11	Ps30	Static Pressure at HPC outlet	psia
12	phi	Ratio of fuel flow to Ps30	pps/psi
13	NRf	Corrected fan speed	rpm
14	NRc	Corrected core speed	rpm
15	BPR	Bypass ratio	-
16	farB	Burner fuel-air ratio	-
17	htBleed	Bleed enthalpy	-
18	Nf-dmd	Demanded fan speed	rpm
19	PCNfr-dmd	Demanded corrected fan speed	rpm
20	W31	HPT coolant bleed	lbm/s
21	w32	LPT coolant bleed	lbm/s

Figure 2: Turbo Engine Dataset

tection. The dataset contains turbine blade images along with detailed defect annotation information. This dataset contains images of scratch, creese, dot, and damage.

3.2 Remaining Useful Life (RUL)

3.2.1 Data preprocessing

There are several crucial steps that should be considered for preprocessing sensor and label data for analysis and modeling. As the first step the train data, test data and the Rul data is loaded using a function that can read data into a DataFrame. A histogram is plotted for understanding the data. Next, the test data is merged with the RUL values, and this is the most crucial part of data preparation. The RUL is calculated using the formula given below This step is the most crucial step in Remaining Useful Life analysis. The RUL for the training data is calculated by finding the maximum time cycle and subtracting the current time cycle. The Remaining Useful Life (RUL) for each unit in the test data is calculated using the following equation 1 and for training data equation 2.

$$\mathrm{RUL} = (T_{\max} + \mathrm{RUL}_{\mathrm{given}}) - t \tag{1}$$

where:

- $T_{\max}$ is the maximum time cycle of the unit.
- $\mathrm{RUL}_{\mathrm{given}}$ is the provided RUL value for the unit.
- t is the current time cycle.

$$\text{RUL} = T_{\max} - t \tag{2}$$

For understanding the frequency of the RUL values in the data, a histogram is plotted, and this provides a clear depiction of the expected lifespan of the units based on the test data. The columns which are not useful for the prediction are dropped from the training data and with this the dimensionality and the noise in the data can be reduced and allows the model to focus on the sensor data that are used for predicting the Remaining Useful Life. Before the data is given into the model it is scaled using the MinMaxScaler to normalize it to a range between 0 and 1. Then the data is reshaped to sequence suitable for the LSTM models.

Different aspects such as row shape, data type and null values are printed for analyzing the training, test and RUL data. The summary statistics for each sensor is also printed. This analysis helped in understanding the Structure, content and quality of the data. For finding the relation between the sensors a correlation matrix is also printed.

Coming to the test data, it is prepared similarly to the train data, by merging it with the RUL values and dropping the unrelated columns. With this the test data is similar to the training data and facilitates accurate evaluation of the model performance. These steps transform the raw data into a structured and standardized format for training the CNN-BiLSTM model and predicting the Remaining Useful Life.

3.2.2 Exploratory Data Analysis

A series of Exploratory Data Analysis steps are performed to gain through understanding about the data. This section includes summarization of the main characteristics of the dataset and visualizations. Examining the distribution of time cycles and Remaining Useful Life along with identifying relationships between different features helped in gaining insights that guide data preprocessing. The EDA include Data overview, Data Cleaning, Descriptive Statistics, Data Visualization, and Data Transformation.

1. **Data Overview**

   Initially the dataset is loaded into the pandas dataframe, which makes it easier for analysis. Afterwards the first few rows of the data are displayed to get an understanding of the structure and content of all three datasets.

2. **Data Cleaning**

   As the first step in data cleaning, the presence of missing values in the train and test dataset are checked. Upon this examination there were no missing values present in both datasets. This ensures that the data is ready for analysis.The descriptive statistics was performed after for each unit in the training data. This provided a summary of the central tendency, dispersion, and shape of the training dataset. Figure 3 shows the summary statistics for the training data.

   This statistics provides a proper overview of the datasets numerical features, central tendencies, variability and range. The mean and median values gives the idea on central tendency and this is calculated for each and every senors. For example the mean, minimum, maximum and standard deviation of the sensor s_1 is 518.67 and this suggest that sensor is providing a static measurement or it might not be relevant for the modeling task. Similarly the patterns for all the sensors are observed.

	unit_number	time_cycles	setting_1	setting_2	setting_3	s_1
count	20631.000000	20631.000000	20631.000000	20631.000000	20631.0	20631.00
mean	51.506568	108.807862	-0.000009	0.000002	100.0	518.67
std	29.227633	68.880990	0.002187	0.000293	0.0	0.00
min	1.000000	1.000000	-0.008700	-0.000600	100.0	518.67
25%	26.000000	52.000000	-0.001500	-0.000200	100.0	518.67
50%	52.000000	104.000000	0.000000	0.000000	100.0	518.67
75%	77.000000	156.000000	0.001500	0.000300	100.0	518.67
max	100.000000	362.000000	0.008700	0.000600	100.0	518.67

s_16	s_17	s_18	s_19	s_20	s_21
2.063100e+04	20631.000000	20631.0	20631.0	20631.000000	20631.000000
3.000000e-02	393.210654	2388.0	100.0	38.816271	23.289705
1.387812e-17	1.548763	0.0	0.0	0.180746	0.108251
3.000000e-02	388.000000	2388.0	100.0	38.140000	22.894200
3.000000e-02	392.000000	2388.0	100.0	38.700000	23.221800
3.000000e-02	393.000000	2388.0	100.0	38.830000	23.297900
3.000000e-02	394.000000	2388.0	100.0	38.950000	23.366800
3.000000e-02	400.000000	2388.0	100.0	39.430000	23.618400

Figure 3: Image of Descriptive Statistics

3. **Data Visualization**

(a) Distribution of Time cycles

The histogram illustrate the time cycle distribution in the training data. The x-axis represents the number of time cycles which ranges from 0 to 350 and the y axis represent the frequency of each time cycle which ranges between 0 to 800. Along with this a Kernel Density Estimate curve is also printed to show the probability density.

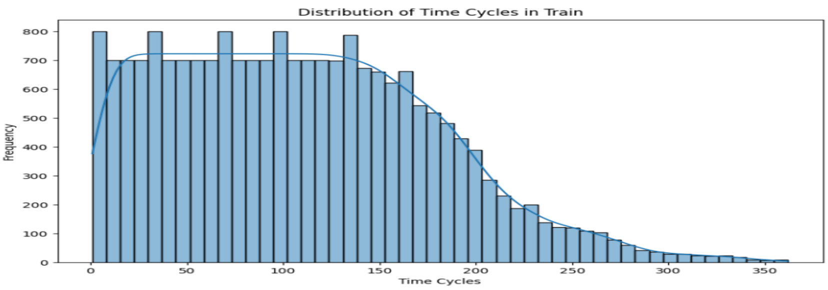

Figure 4: Distribution of Time Cycles in Train Data

The histogram have multiple peak frequency between 0 and 150 time cycles, and the frequency ranges between 600 and 800 units. This proves that many of the units operated with in this duration and this also suggest a uniform distribution of lifespan for many units in the data. After 150 the frequency gradually decrease, this continues till 250 and then the unit drops sharply. The KDE line which is printed along the histogram shows that small subset of units operate for longer period, and this can be understood from the high frequency for time cycles till 150. This also confirms that shorter lifespan are more common. The spread of the data shows it is having a right-skewed distribution and more number of sensors have a moderate lifespan. The higher frequency units suggest that maintenance strategies should prioritize these range as most failure occur here. The histogram gave insights on the operational lifespan of the units in the dataset which crucial for developing the model.

(b) Distribution of RUL in Test Data

The histogram Figure 5 gives insight to the distribution of RUL data in the test data. Similar to the histogram for Time cycle, the x axis of the histogram ranges from 0 to 350 and the y axis represent the frequency. The histogram shows a peak around 100 and 150 with the maximum frequency near 150 cycles and the frequency is close to 700 units. The distribution of data is symmetrical, as the frequency value rise steadily and then gradually decrease and this also suggest a normal distribution of RUL values. With this histogram valuable insights about the expected lifespan of the units in the test data is gained.

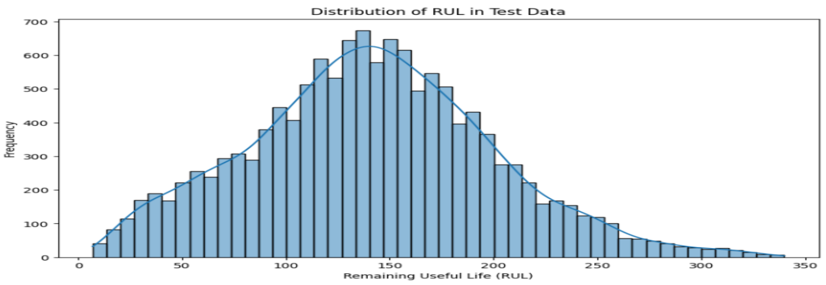

Figure 5: Distribution of RUL in Test Data

(c) Histogram for Sensor Readings

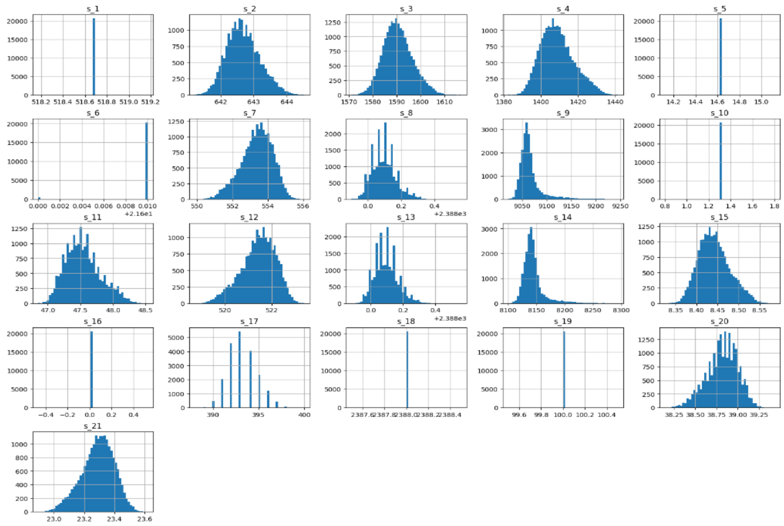

Figure 6: Histograms of Sensor Readings

Figure 6 represents the histogram for all the sensors, from s_1 to s_21 in the data.This give insight on the distribution of each sensor data in the dataset. For example, the histogram for s_1 that is the sensor 1 have a single peak at 518.67 and this indicates constant reading across all observations along with this the variability of the data is also checked and no variability is present in the data. Similarly the sensors s_2, s_3, s_4, s_7, s_8, s_11, s_12, s_14, s_15,s_20 and s_21 is having normal distribution. Meanwhile, for the sensors s_5, s_10, s_16, s_18, and s_19 the distribution is constant. Furthermore,the sensor 6 is having a narrow distribution and sensors s_9 is having a peak around 9065.24, s_13 is having a narrow peak at 521.5, and s_17 is having several discrete peaks. This histogram proves most of the sensors are having a normal distribution of data and this proves that the data is reliable.

4. **Data Transformation**

Data transformation is one of the most crucial step before implementing the model. This step involves scaling and reshaping of the data. For preparing the data for RUL prediction the MinMaxScalar is used for normalizing the data, this normalization is important for algorithms like LSTM. The LSTM models have a specific sequential format for input data, therefore the data is also transformed using a reshape data function and this transforms the data into overlapping sequence. By scaling and reshaping the data we can ensure that the data is appropriately formatted and normalized for training the model.

3.2.3 Model Building

One of the crucial parts in developing an Aero engine Management system is developing a reliable model for predicting Remaining Useful Life (RUL). This section outlines the building of the model. The RUL model was built using advanced deep learning methods, combining Convolutional Neural Network (CNN) and Bidirectional Long Short-Term Memory(BiLSTM) to extract the spatially and temporally related features from the sensor data.The model is compiled using the Adam optimizer and Mean Squared Error (MSE) as loss function to minimize prediction error and Mean Absolute Error (MAE) used to analyze the performance.
Furthermore, a function is developed to make the RUL prediction on new data. The function ensures the data is correctly shaped and processed before fitting into the model. Finally, the impact of each sensor on the RUL predictions is analyzed. In summary, my model is developed by combining CNN and BiLSTM and using this the spatial and temporal features from the sensor data is captured for predicting Remaining Useful Life (RUL). The Adam optimizer was used to train the model and MSE was used for validating the performance of the model. This approach ensures the development of a robust system.

3.3 Defect Detection

3.3.1 Data preprocessing

Similar to RUL prediction there are several key steps that should be followed for preprocessing image and label data for analysis and modeling. Initially the images are resized and then the pixels are normalized. The dimensions are also expanded, to be used as the input of the VGG16 model. Additionally, thumbnails were generated for every photograph to facilitate visualization. A function was introduced which resizes the images and encodes them in base64 format. This format is useful for simplifying the sharing and presentation of results by embedding images into web pages without the need for separate image files.
Furthermore, the features were extracted, and thumbnails were automated. The tag which indicates the dataset split along with image filename, extracted feature vector, and the thumbnails are stored as a list. Afterwards a directory was created where keys represent the train data, validation data and the values which contain the file path of the corresponding image. After processing the images in each directory, the result was combined into a single DataFrame. The filenames were also modified for generating label names for merging with label data. Label data was stored in text files which contain separate directories for training and validation sets. The text files included annotations for every image which specify the type of defects. A function that can process each file in the specified directories is utilized for reading and combining the label data into a single

file. Each row in the Data Frame included the label, corresponding dataset slit tags and bounding box coordinates.
The final step in preprocessing is merging the image DataFrame with the label DataFrame. This merging was performed on the basis of filenames, which ensured that every image feature vector was correctly connected with its corresponding label data. Additionally, the numerical labels were mapped to their respective class names "Scratch, Dot, Crease, and Damage. These comprehensive preprocessing ensured that both the image data and label data were efficiently prepared for modeling and further analysis.

3.3.2 Model Building

In the development of systems for defect detection in engine blades, both traditional machine learning and deep learning approaches were meticulously explored. I have chosen a hybrid model combining VGG-16 and Support Vector Machine (SVM). Integration of these two models achieved high accuracy in defect detection. The data from Kaggle which contain the images of engine blade and label of defects are resized to 224*224 to match the VGG-16 model's input size. The pixel values are also normalized to a range between 0 and 1 and the data also undergoes dimensionality reduction using Uniform Manifold Approximation and Projection (UMAP). The VGG-16 convolutional neural network, which is pretrained on ImageNet dataset, is employed for feature extraction. The fully connected layers of the VGG-16 are removed and replaced with a pooling layer to obtain a compact feature representation of every image. This allows the model to effectively capture the high-level features extracted by VGG-16 for defect detection.
The core of the system is a hybrid VGG-16 and SVM model. The high-dimensional feature vectors from VGG-16 are imputed into the Support Vector Machine (SVM) classifier for performing the last step of classifying the defect in the images. The SVM was selected due to its robustness against overfitting and its effectiveness in high dimensional space. Training of the model involved using VGG16 model to transform the blade images into high dimensional feature vectors and the features are fed into the SVM classifier for detecting the defect. The model's performance was evaluated by comparing predicted classifications against ground truth annotations. Accuracy, Precision, Recall, and F1-Score are used for analyzing the model's effectiveness in detecting defects. For deployment, the hybrid model was integrated into the AeroEngine Health Management System which combines both Remaining Useful Life prediction and Defect Detection. This enables real time defect detection in engine blades.

3.4 Web Application

In the development of Aero Engine Health Management System, a web application is developed to make it easier for users to interact with Remaining Useful LIfe analysis and Defect Detection. The web application is developed using Streamlit, an powerful and user-friendly Python framework which can be used to develop interactive web interfaces. The main objective of the web application is to provide a platform for users to upload the sensor data for analyzing the Remaining Useful Life and also the images of engine blades for defect detection. The sidebar navigation allows users to switch between Defect Detection and RUL Analysis. This user interface optimizes the developed models accessibility and usability, allowing engineers and maintenance personnel to use the system effectively.

3.4.1 Features and Functionality

1.Remaining Useful Life (RUL)

The interface for RUL prediction consist of Sliders to input data, Prediction button for initiating prediction and finally the generated result will be displayed

2.Defect Detection Interface

The interface for Defect Detection contain an button to upload image and a Classification button to initiate the defect detection process.

4 Design Specification

This Section provides both the individual architecture of RUL prediction and Defect Detection along with the complete architecture of Aero Engine Maintenance System.

4.1 Remaining Useful Life (RUL)

The individual architecture of Remaining Useful Life (RUL) prediction is given in Figure 7. Initially the data is collected, then the data is preporcessed, after preprocessing the data is given to the CNN-BiLSTM model, after the training of the model it is evaluated and finally, the prediction is performed.

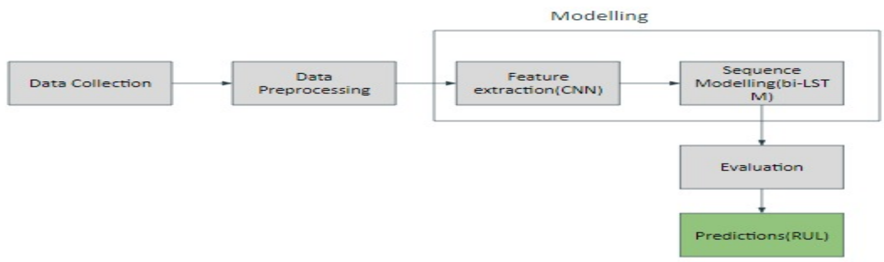

Figure 7: Architecture of Remaining Useful Life (RUL) Prediction

4.2 Defect Detection

The architecture of Defect Detection is given in the Figure 8. Initially the data is collected and fed in to a pretrained VGG16 model for feature extraction and the extracted data is given for dimentionality reduction using UMAP, the result from UMAP is fitted into SVM classifier after the model is trained it is evaluated and finally, the trained model is used for prediction

4.3 Web Development

The developed web page is the final end product of the research, Aero Engine Maintenance System. The entire architecture is given in the Figure 9 Both the Defect detection module and the RUL module is connect to an Application UI and the corresponding inputs, images for defect detection and the sensor data are given in to the application and the corresponding results are generated in the application.

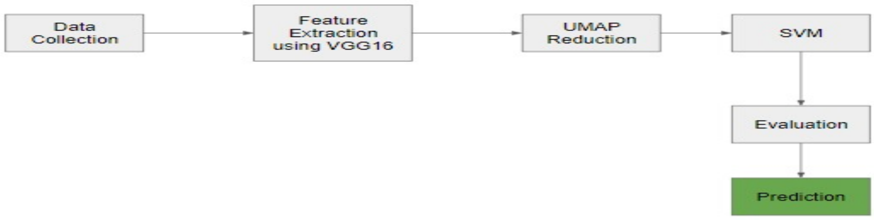

Figure 8: Architecture of Defect Detection

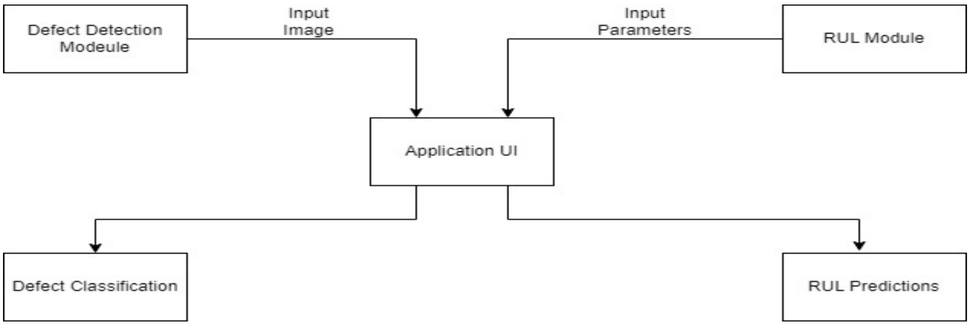

Figure 9: Architecture of AeroEngine Maintenance system

5 Implementation

5.1 Remaining Useful Life (RUL)

Remaining Useful Life (RUL) prediction model was developed in google colab using the python programming language.

5.1.1 Data Preprocessing

1. **Data Loading**

   The initial step in building the model is loading the data. The training, test and the RUL data corresponding to different fleets are loaded using a 'load_data' function. This function reads the data from the text files, assigns appropriate column names for all three datasets.

2. **Data Analysis**

   The presence of missing values, shape of the data, the data type and the descriptive statistics are checked to ensure the data quality and proper formatting. From the output no missing values were present and the data type was float and int, which are the required data types for developing the model. Furthermore, the distribution of time cycles in the training data and the distribution of RUL in the test data is also analyzed using histogram to understand the variability and characteristics of the data. Figure 4 shows the distribution of time cycle and Figure 5 shows the distribution of RUL in test data.

3. **Data Preprocessing**

As part of data preprocessing, the RUL is calculated for each unit in the training dataset, which is done by subtracting the current cycles count from the maximum cycles count and this provides the target variable for our model. Furthermore, irrelevant columns are dropped, and the remaining sensor data is scaled to ensure it is ready to be used for model training. The dropped columns do not contribute to the model. After the scaling is performed the data is again transformed or reshaped into a sequence of specified length which is crucial for the LSTM layers in the model and the sequences represent a series of sensor readings.

4. **Model Training**

   A model is built by combining the CNN and BiLSTM, the model captures the spatial and temporal features for predicting the Remaining Useful Life. The Conv1D layers which are the convolutional layers help in learning the local features which are present within the sensor data, meanwhile the MaxPooling layers reduce the dimensionality of the features and also help in capturing the patterns. The dropout layers introduced after the pooling layer prevent overfitting of the model by setting some neurons off in the training phase. The bidirectional LSTM layer captures the temporal dependencies present in the data from both forward and backward directions. Finally, the dense layer is added to the model with a single neuron, this acts as the output layer to predict the Remaining Useful Layer. Furthermore, the compilation of the model is performed. The Adam is used as the optimizer, Mean Squared Error is used as the loss function and the Mean Absolute Error is used as another perspective to analyze the model's performance.
   After compilation the model is trained using the 'fit' method. The model is trained with 50 epochs which means the model will iterate over the training data 50 times. A batch size is specified as 64 which means the model can process 64 samples at same time before updating the weights. The validation split is given as 0.2. After the training, the prediction is performed using the test data. The samples containing NaN are filtered out from the predictions, to ensures the models accuracy of the evaluation. After the predictions the Mean Absolute Error (MAE) and Mean Squared Error (MSE) are then computed to measure the model's performance. The model is saved using the 'save' function. For making predictions with new data a function is developed which ensures the sensor data is correctly shaped and fed into the model. Finally, a function verifies the length of the input data to check the number of features and create a full sequence and finally reshape it to fit the model and the predictions are made. prediction.

5.2 Defect Detection

5.2.1 Data Preprocessing

Defect detection was implemented in google colab, using Python Programming language and various functions and libraries available in the language.

1. **Data Preparation**

   The first step in developing the model is loading the data. The path of the directory is specified using 'glob' and this allows the glob module to access all the images in the directory. Various key constants are used while reading the data which include 'size', 'stop', and 'thumbnail_size'. A function png is designed to convert the images into base64- encoded strings which is the thumbnail version of the image.The base64 format enables easy inclusion of thumbnails in web pages without the need for external files.

2. **Feature Extraction**

   A pre-trained VGG16 model with a flattening layer is used for feature extraction. A new model is created with the original VGG16 inputs and the flattened output and this makes the model more suitable for extracting features. For Preprocessing and utilizing the VGG16 model for extracting features a function 'embed_vgg16 is implemented. After the image is read the image is resized and normalized. The processed image is taken as the input to VGG16 for predicting and extracting features.

3. **Data Processing**

   A flatten function is developed which serves as a utility to simplify the nested structure, making it easier to handle the output in remaining stages. The individual elements are aggregated into a cohesive list by iteratively traversing the nested lists, this also eliminates the complexity associated with nested data formats. All the images for developing the model are stored in a separate directory, a function get_picture_from_glob is developed to access, extract features and generate thumbnails of the images from the directory. The results are gathered into a list of series objects, and each series contains the tag, filename, extracted features and thumbnails of the images.

4. **Data Preprocessing for Model**

   (a) Loading and Processing Images : A DataFrame is prepared which contains the extracted features and all other relevant information from the images. For this purpose, a directory is created by mapping directory names to their corresponding file path. With the use of previously defined functions, the processed images are flattened and stored to a pandas DataFrame. Finally, to ensure that each image feature is appropriately labeled a column is added to the DataFrame which includes the file names.

   (b) Merging Label Data and Splitting data into Training and Validation Sets Integration of label data with the prepared image DataFrame begins by reading label files from the training and validation subfolders. Label data along with other required information such as source folder, is then collected in a different DataFrame. The merging of the label and image data is done by matching file names and the numerical labels are mapped to respective class names and this helps in easier interpretation and analysis of the data. The Combined DataFrame is divided into Training and Validation sets. The subset of data which is properly labeled for subsequent model training and evaluation are stored in separate dataframes named 'train_df' and 'val_df.

5. **Visualization**

   A histogram Figure 10 is plotted using Seaborn and it plots the distribution of classes in the training dataset. The histogram shows that the class scratch is having a count of 200 which is followed by damage which is nearly 100 and crease having a count of 50 and finally dot with a count more than 25.

6. **Dimensionality Reduction using UMAP**

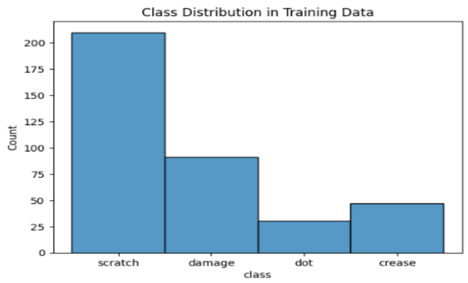

Figure 10: Visualization of Image Data

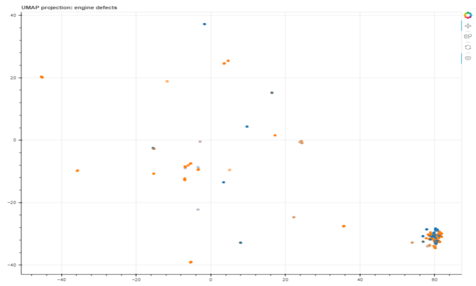

Figure 11: Visualization of UMAP

Reducing the dimensionality of the feature vectors is performed using the UMAP (Uniform Manifold Approximation and Projection). The process begins by initializing the UMAP and specifying parameters such as random_state, verbose and n_epochs. The feature vectors are transformed into 2D coordinates by fitting the UMAP algorithm. This compresses the high-dimensional feature space into a 2-dimensional space. Finally, the Bokeh is used to develop an interactive visualization of UMAP result

The output of this visualization displays the 2D UMAP coordinates of the feature vectors. The data points are represented as circles on the plot and colors indicate different classes. Moving over a data point shows the class label and thumbnail image. This plot make it easy to explore the distribution of the data and identify patterns and clusters.

7. **Model Training**

   Preparing the data and guaranteeing the models's optimal performance requires the scaling of the training data for the Support Vector Machine (SVM) classifier. In this section the features and target labels are separated from the training and validation data. The training data for the model is also extracted from the training DataFrame and the validation data is from the validation DataFrame. After separating the data, features are standardized and the SVM classifier is trained. After the standardization of the data an SVM classifier with a linear kernel is initialized and both the regularization strength and probability estimates are specified. 'joblib' library is used to save both the scaler and the trained SVM model after training.

   The trained model is evaluated using the validation set. The validation data is used for generating the predictions by the trained SVM model. After the predictions the accuracy and the classification of the predictions is generated. Accuracy helps in understanding overall performance of the model and the report provides detailed information on precision, recall and F1-score for each class. During prediction the input image is loaded and it is resized to 244*244 pixels, normalized and a batch dimension is added. Then the features are extracted using the VGG16 model. The extracted features are then scaled using the saved scaler and then fed into the pre-saved SVM classifier and the confidence score is obtained. The class with the highest confidence score is taken as the prediction, and if the confidence score is below the predefined threshold it will be considered as an "unknown class".

5.3 Web Application

The Aero Engine Maintenance System application was developed in visual studio code by combining both CNN-BiLSTM and VGG16-SVM models.

(a) Environment Setup

The python libraries, Streamlit, NumPy, Pillow, scikit-learn, TensorFlow and Keras and joblib are used for setting up the environment for the application to run smoothly.

(b) File Organization

The entire implementation of the model depend on the pre-trained models and scaler objects. The models include, 'bilstm_model.h5' the trained CNN-BiLSTM model for RUL prediction, and this is loaded using the function 'predict_rul', "Svm_model.pkl" is the trained support vector machine model for defect detection and it is loaded using the function 'load_and_preprocess_image', and "scaler.pkl' is the trained scaler for feature scaling.

(c) Streamlit Interface

In the development of the application using the streamlit, the initial step involves giving the title using the st.title function, followed by sidebar with a dropdown menu which allows user to select the two functionalities "Defect Detection" and "RUL Analysis". Coming to the defect detection interface, initially a header is specified for indicating the section, followed by an file uploader to upload the images. Finally, classify button on clicking this button the classification for the uploaded image is performed. This displays the predicted class and associated confidence score of the image.

Following the Defect Detection section is the RUL analysis interface, here initially a header is provided which indicate the RUL analysis section and sliders are provided for 21 sensors using the st.slider function, these allow the users to input data values by moving the sliders. Finally, Predict RUL button is provided, by clicking the button RUL is predicted.

6 Evaluation

In this section the effectiveness and reliability of the CNN-BiLSTM and VGG16-SVM hybrid models which are used in the Aero Engine Maintenance System are evaluated. The evaluation assesses the system's performance in accurately predicting the remaining useful life and detecting defects.

6.1 Experiment on Remaining Useful Life (RUL)

During the development of the CNN-BiLSTM model for predicting Remaining Useful Life (RUL), various measures were used for evaluating the model. I have used Mean Absolute Error (MAE) and Mean Squared Error (MSE) and the corresponding visualization of MAE and the training and validation loss for analyzing the performance of the model. The training history of the model shows the loss and MAE for both training and validation sets over 50 epochs. Figure 12 shows the training and validation loss. In the plot the loss decreases steadily from 11,000 to under 1000, indicating effective learning of the model.

The initial decrease of the validation loss indicates improved performance on unseen data, and it stabilizes around epoch 20 which suggests that model reached an optimal learning point. Meanwhile the MAE plot in the Figure 12 shows the mean absolute error for both training and validation set. The MAE decreases consistently from 85 to under 20, this indicates that the model's predictions are more accurate as the training progresses. Both the loss and mae follows a similar trend and this means that the model maintains its predictive accuracy on new unseen data. The final evaluation metrics MAE of 48.03 and MSE of 3308.89 suggest that the model performs reasonably well. The average prediction error of 48.03 cycles and low MSE indicate the predictions are accurate and reliable.

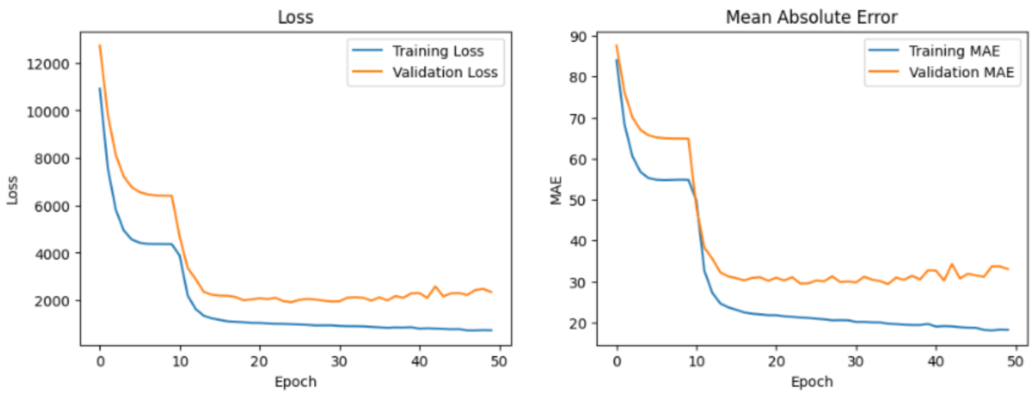

Figure 12: MAE and RMSE of RUL

6.2 Experiment on Defect Detection

The performance of the VGG16-SVM hybrid model was evaluated using several metrics, which include Precision, recall, F1-score and confusion matrix.

SVM accuracy: 0.9079

	precision	recall	f1-score	support
crease	0.67	0.80	0.73	5
damage	0.90	0.95	0.92	19
dot	0.67	1.00	0.80	2
scratch	0.96	0.90	0.93	50
accuracy			0.91	76
macro avg	0.80	0.91	0.84	76
weighted avg	0.92	0.91	0.91	76

Figure 13: Classification Metrix

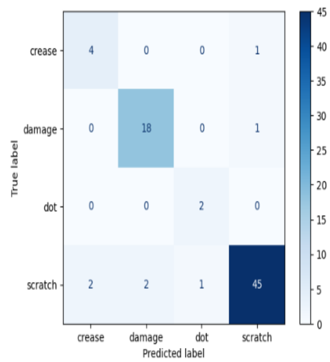

Figure 14: Confusion Matrix

The classification metrics in the Figure 13 gives the performance of the model in between different defect types. The dataset used for evaluation is the validation data which contains 76 images.

The model achieved an overall accuracy of 90.79% coming to the metics for each class, the class Crease is having a precision of 0.67 which indicates that 67% were correctly classified as crease, furthermore the recall of 0.80 indicates that 80% of all actual crease were correctly identified by the model and a F1-Score of 0.73 indicate the harmonic mean of precision and recall. Similarly, the class Damage has a precision of 0.90, recall of .0.95 and F1 score of 0.92. Meanwhile, the class dot has a precision of 0.67, recall of 1.00 and F1-score of .80 and finally, class scratch has a precision of 0.96, recall of 0.90 and F1-score of 0.93. While evaluating the Macro-Averaged metrics the average precision is 0.80, average recall is 0.91 and the average F1-Score is 0.84 and the Weighted Average metrics shows that it's having precision of 0.92, this reflects the overall precision when accounting for class imbalance, recall of 0.91 and F1-Score of 0.91.

The confusion matrix Figure 14developed for the VGG16-SVM model provides detailed insight on performance of the model across defect classes, this includes Crease, Damage, Dot, and Scratch. In the matrix the rows represent the True label, and the columns represent the predicted labels. For the class 'Crease' the model correctly identified 4 instances but 1 instance was misclassified as 'Scratch'. For class 'Damage', the model correctly predicts 18 instances but misclassifies 1 instance as 'Scratch'. Meanwhile, the class 'Dot' identifies all 2 instances correctly without any misclassifications. Lastly, the model classified 45 instances of class "Scratch" but it misclassified 2 instances as crease, 2 instances as damage and 1 instance as dot,this indicates the model is having overall good accuracy but there is room for improvements.

6.3 Web Application and Results

This section provides the overall front end of the AeroEngine Maintenance System Web page, along with its individual components and their outputs.

Figure 15 Give the front end with the drop down menu for selecting the Defect detection and RUL analysis

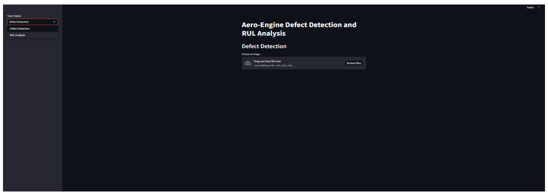

Figure 15: Aero Engine Maintenance System

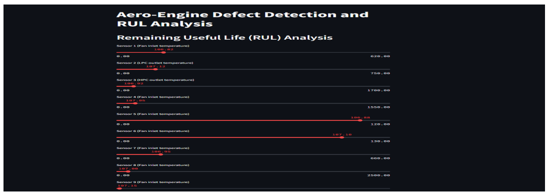

Figure 16: RUL Prediction

6.3.1 RUL Prediction

The Figure 16 and Figure 17 shows the sliders for all the sensors and the analysis button and the generated sample output.The maximum and minimum values of the sensors are taken from the histogram developed for evry sensors.

6.3.2 Defect Detection

The Figure 18 shows the option to upload the image and the uploaded image can be viewed before it is processed. Finally, the classify button gives the generated classification.

Figure 17: Rul Prediction

Figure 18: Defect Detection

6.4 Discussion

This study investigates the performance of both CNN-BiLSTM model used for Remaining Useful Life prediction and the VGG16-SVM model used for defect detection in the Aero Engine Maintenance System. With a MAE of 47.83 and MSE of 3284.00 along with a notable decrease in the training and validation loss, the CNN-BiLSTM model showed significant efficiency in predicting RUL. This result demonstrates how well the model can extract temporal patterns and predict the RUL. The sensor impact study reveals that the sensors 7, 8, 14, 6, and 19 have the most influence in Remaining Useful Life of the turbofan engines. While analyzing the VGG16-SVM model for defect classification, the model demonstrated an accuracy of 90.79%. The model demonstrated high precision and recall scores while classifying 'Damage' and 'Scratch', the class damage has a precision of 0.90 and recall of 0.95 mean while the class scratch has 0.96 for precision and 0.90 for recall. The development of the confusion matrix highlighted opportunities for improvements in the detection of 'Crease', 'Damage', and 'Scratch' defects. The findings of both the hybrid models to with the existing literature by showcasing the practical benefit of these models in predictive maintenance and defect detection. The ability of CNN-BiLSTM model to handle time series data and the performance of VGG16-SVM model indicates the notable advancements in these domains.

Overall, this research has presented a valuable contribution to developing the Aero Engine Maintenance System. The findings demonstrate significant improvements in handling time-series data in predictive maintenance and classification problems along with offering valuable contributions to the advancement in these fields.

7 Conclusion and Future Work

In this study I investigated if the maintenance practices in the aerospace field can be improved by combining image processing and predictive maintenance techniques. My findings indicate that the CNN-BiLSTM model achieved a MAE of 48.03 and MSE of 3308.89 and the VGG16-SVM model achieved an overall accuracy of 90.79% this shows that the proposed models outperforms all the traditional models and the web application developed by combining both the RUL prediction and Defect Detection can significantly impact the aerospace maintenance practices and revolutionize them. The research has successfully addressed the research question whether the integration of predictive maintenance and image processing can enhance the efficiency and reliability in aerospace maintenance operations. Overall, my research contributes to the aerospace maintenance practices significantly, by introducing a novel approach of integrating both image pro-

cessing and predictive maintenance. The proposed system along with all the results lay a strong foundation for future research and for developing a more comprehensive system by aerospace industries.
As for the future work integrating the proposed model with IOT devices can result in the development of technology which can monitor the engine in real time especially during the flights, by which the safety can be guaranteed. Furthermore, as this system focuses on turbofan engines, expanding this research to all other types of engines make this system practical for all aircrafts. Finally, developing systems which can predict the life of every component in the aircraft and incorporating it with the Aero Engine Maintenance System can revolutionize the entire aerospace industry.

References

Abdulrahman, Y., Eltoum, M. A. M., Ayyad, A., Moyo, B. and Zweiri, Y. (2023). Aero-engine blade defect detection: A systematic review of deep learning models, *IEEE Access* **11**: 53048–53061.

Azevedo, D., Ribeiro, B. and Cardoso, A. (2019). Web-based tool for predicting the remaining useful lifetime of aircraft components, *2019 5th Experiment International Conference (exp.at'19)*, pp. 231–232.

Bai, X., Sun, Y. and Meng, G. (2023). Remaining useful life prediction of aircraft engines based on fa-sdae-bigru, *2023 6th International Conference on Computer Network, Electronic and Automation (ICCNEA)*, pp. 376–381.

Bingjie, H., Wei, N. and Jichao, W. (2021). An improved similarity-based prognostics method for remaining useful life estimation of aero-engine, *2021 IEEE/ACIS 20th International Fall Conference on Computer and Information Science (ICIS Fall)*, pp. 38–41.

Cui, J., Wang, Y., Cui, X., Jiang, L., Liu, D., Du, W., Tang, X. and Wang, J. (2022). Prediction of aeroengine remaining useful life based on se-bilstm, *2022 34th Chinese Control and Decision Conference (CCDC)*, pp. 1781–1786.

Elakramine, F., Jaradat, R., Ullah Ibne Hossain, N., Banghart, M., Kerr, C. and El Amrani, S. (2022). Applying systems modeling language in an aviation maintenance system, *IEEE Transactions on Engineering Management* **69**(6): 4006–4018.

Ensarioğlu, K., İnkaya, T. and Emel, E. (2023). Remaining useful life estimation of turbofan engines with deep learning using change-point detection based labeling and feature engineering, *Applied Sciences* **13**(21): 11893.

Garay, J. M. and Diedrich, C. (2019). Remaining useful life estimation for production devices in the automotive industry based on health threshold and cyclical data streaming, *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 910–915.

Li, D., Li, Y., Xie, Q., Wu, Y., Yu, Z. and Wang, J. (2021). Tiny defect detection in high-resolution aero-engine blade images via a coarse-to-fine framework, *IEEE Transactions on Instrumentation and Measurement* **70**: 1–12.

Li, J.-J., Yan, C.-F., Rui, Z.-Y., Zhang, L.-D. and Wang, Y.-t. (2020). A quantitative evaluation method of aero-engine blade defects based on ultrasonic c-scan, *2020 IEEE Far East NDT New Technology Application Forum (FENDT)*, pp. 91–95.

Liu, S., Zhou, N., Li, J. and Wu, Y. (2023). A hybrid transformer-gru network for remain useful life prediction of aero-engine systems, *2023 IEEE 6th International Conference on Automation, Electronics and Electrical Engineering (AUTEEE)*, pp. 522–527.

Ma, Y., Liu, M., Zhang, Y., Wang, X. and Wang, Y. (2024). Spdp-net: A semantic prior guided defect perception network for automated aero-engine blades surface visual inspection, *IEEE Transactions on Automation Science and Engineering* pp. 1–10.

Ramasso, E. (2014). Investigating computational geometry for failure prognostics., *International Journal of prognostics and health management* **5**(1): 005.

Sharma, V., Dagar, R. and Sharanya, S. (2024). A xgboost optimized ensemble model for remaining useful life prediction of aircraft turbofan engines, *2024 International Conference on Emerging Smart Computing and Informatics (ESCI)*, pp. 1–5.

Song, K., Sun, X., Ma, S. and Yan, Y. (2023). Surface defect detection of aeroengine blades based on cross-layer semantic guidance, *IEEE Transactions on Instrumentation and Measurement* **72**: 1–11.

Tekgöz, H., Omurca, S. I. and Koc, K. Y. (2022). Estimation of remaining useful life based on time series analysis, *2022 7th International Conference on Computer Science and Engineering (UBMK)*, pp. 273–277.

Wang, C. and Chen, H. (2023). Eedn: An efficient edge detection network for aeroengine blade defect segmentation, *2023 42nd Chinese Control Conference (CCC)*, pp. 8594–8598.

Yang, C., Wang, T. and Li, Y. (2022). A prediction model of turbofan engine remaining life based on improved algorithm, *2022 IEEE 6th Information Technology and Mechatronics Engineering Conference (ITOEC)*, Vol. 6, pp. 309–314.

Zhang, W., Zhu, D. and Xie, Y. (2023). An integrated aircraft maintenance training platform based on "simulation training+auxiliary operation", *2023 International Conference on Computer Science and Automation Technology (CSAT)*, pp. 59–63.

Zhao, S., Pang, Y., Chen, J. and Liu, J. (2022). Predication of remaining useful life of aircraft engines based on multi-head attention and lstm, *2022 IEEE 6th Information Technology and Mechatronics Engineering Conference (ITOEC)*, Vol. 6, pp. 1530–1534.