Centre for Maintenance Optimization & Reliability Engineering

Director

Chi-Guhn Lee

Semi-annual report

December 2021

Mechanical & Industrial Engineering
UNIVERSITY OF TORONTO

cmore.mie.utoronto.ca | 416·946·7867

Table of Contents

Technical Reports

Appendices

C-MORE progress meeting agenda

Opening remarks, executive summary 9:00-9:40		Chi-Guhn Lee
Student research project		
9:40-10:00	Predictions of emergency workloads of the oil rigs in the North Sea by using ARIMA-GARCH and Facebook Prophet models	Ziyi Lyu
10:00-10:15	15-minute break	
Collaboration with Technical University of Denmark		
10:15-10:35	Maintenance Clustering Principles	Julie Krogh Agergaard
Tutorial		
10:35-10:55	Deep learning and failure detection	Chi-Guhn Lee
10:55-11:10	15-minute break	
Collaborations with industry partners		
11:10-11:30	KPI for Mining: Eliminate the Communication Gaps within the Organization	Blair Cui Gabriel Merisanu
11:30-11:50	UKMOD: Vehicle Performance Analysis using Machine Learning	Siddharth Behal
11:50-12:35	45-minute break	
12:35-12:55	UKMOD: Carrier aircrafts spares optimisation problem	Varun C Ananda Rao
12:55-1:15	Kinross: A Machine learning model for sensor-based maintenance of mining trucks	Vahid Najafi Zoha Sherkat-Masoumi
1:15-1:30	15-minute break	
1:30-1:50	TTC: Data processing and preparation for science	Janet Lam
1:50-2:10	Fault Detection for Wind Turbine Bearings Under Varying Rotational Speed	Mohamed Hassan Eileen Mendoza Dhavalkumar Patel Hazel Shi
2:10-2:25	15-minute break	
2:25-2:45	Criticality Analysis and Asset Management of a Power Generation Facility	Daisy Wang
Student research projects		
2:45-3:05	Time Series Classification for Combustion Signature Analysis in Freight Fires	Sophie Tian
3:05-3:25	Dataset for 3D Printer Process Modelling	Katie Xu
Closing remarks		Chi-Guhn Lee

Executive summary

Chi-Guhn Lee, C-MORE Director

Introduction

Although pandemic restrictions have eased somewhat over the past six months, we have continued to operate in the virtual world – including course instruction, formal meetings, informal get-togethers, and conference presentations. In fact, as I review our accomplishments in preparing this report, I am amazed at how balanced our portfolio is. We continue to pursue cutting-edge research while working closely with collaborating industries in the real world. We seek (and acquire) funding for our efforts; for example, we submitted a proposal to NSERC's Mission Alliance program in partnership with WSP. And our educational efforts have never slowed down. We held two PAM courses online, one in June and the other in August, and we have a full complement of graduate students working with us. Beyond our continuous work with consortium members, we have engaged with other industry partners on an ad-hoc basis, including Titan Technologies and Capital Power. The following report summarizes work undertaken since the meeting in June 2021.

The C-MORE team

Chi-Guhn Lee, Director

Chi-Guhn has continued to leverage the all-virtual environment to give invited talks and seminars at conferences around the world. A detailed list of his talks is reserved for the Activities section of the report. The resource extraction working group initiative has continued to grow, becoming a Capstone project for senior industrial engineering students and a research project for two Master of Engineering students. These projects will result in a working prototype of dashboard displaying KPIs based on a well thought-out design. Outside this particular sector, Chi-Guhn is increasing the presence of C-MORE in the asset management and maintenance optimization arena. In early September, he gave a talk on advanced machine learning methods applied to physical asset management with scarce data, and participated in a round-table discussion on how machine learning can be done with incomplete or insufficient data. In this six-month term, he submitted two project proposals: an application to the DSI Catalyst program on a probabilistic map for possible survivors after a natural disaster using flying sensors, and an application to IC-IMPACTS for a project on an intelligent irrigation system under weather uncertainty. C-MORE also signed a project contract on predict and control supply chain with Unilever, with the support of MITACS.

Janet Lam, Assistant Director
Janet has been working with graduate and undergraduate students to push on with various industry-sponsored research projects. In particular, she worked with Pooyan and Daisy on the Capital Power criticality project, and with Gabe and Blair on the Mining KPIs project. She worked directly with Jennifer Lu from TTC on extensions and updates to the TTC track inspection projects; this project will be presented in today's meeting. Janet met virtually with several potential collaborators to discuss ways to combine forces and made renewed contact with current members to keep projects fresh. In addition to these responsibilities, she taught one day of the five-day Physical Asset Management (PAM2) course, with Chi-Guhn, Ali Zuashkiani, and Jim Reyes Picknell in June and August. She also made a presentation at MainTrain, the annual PEMAC conference.

Andrew K. S. Jardine, Professor Emeritus
Andrew's main activity during the past six months was to finalize the 3rd edition of the book *Maintenance, Replacement and Reliability*. It was published in September, just in time to be one of the textbooks used in the University's five-day annual Physical Asset Management program designed for practicing engineers. In addition, he taught a graduate class at the University of Toronto, titled "Engineering Asset Management," reviewed a MITACS research proposal, and was the external examiner for a doctoral thesis at the University of the West Indies. During this COVID period, he presented a three-hour webinar, "Optimizing Preventive Maintenance, Inspection, and Predictive Maintenance Decisions," through International Business Conferences, India. He also continued his participation as a member of PEMAC's Awards Committee.

Dragan Banjevic, C-MORE Consultant
Dragan continued to collaborate with C-MORE on projects with consortium members, mostly with Kinross Gold, TTC, and UKMOD. He worked directly with Varun on the implementation of the UKMOD aircraft carrier spares management project. He also provided help in other projects with C-MORE students, as well as in their research.

Sharareh Taghipour, Ryerson, External Collaborator
Since the meeting in June 2021, Sharareh published a book chapter, "The Role of Emerging Technologies in Physical Asset Management," in *Maintenance, Replacement, and Reliability: Theory and Applications* (3rd ed) by Andrew Jardine and Albert Tsang. She lectured two sessions of the course "Engineering Asset Management" (graduate course, with Andrew Jardine) at University of Toronto. She also taught one session of the five-day PAM program in November. The session was about the role of emerging technologies in physical asset management. In addition, she taught one undergraduate course, "Experimental Design and Quality Assurance," and one graduate course "Advanced Reliability Modeling" at Ryerson University in fall 2021. She is working on a research program, "Decentralized Data Analytics and Optimization methods for Physical Asset Management" (NSERC Discovery grant), as well as two collaborative projects with industry partners: "Real-Time Optimization of Production Scheduling" and "Failure Modelling of Repairable Systems with Data Subject to Left Truncation and Right Censoring, and Improving the Models' Accuracy using Bayesian Techniques" with Axiom Group (NSERC Alliance) and Enbridge Gas (Consulting contract), respectively. In addition, she is using the "Industry 4.0 Smart Factory System" to develop predictive maintenance models and real-time optimization of production scheduling (funding from Ministry of Economic Development, Job Creation and Trade and John R. Evans Leaders Fund). Sharareh presented "The Role of Emerging Technologies in Physical Asset management" at the 15th International Physical Asset Management Conference, online, March 9-10, 2021, Tehran, Iran. She was a keynoter speaker at IEEE International Conference on

"Sensing, Diagnostics, Prognostics, and Control" (SDPC 2021), online, August 13-15, 2021, Harbin Institute of Technology, Weihai, China, and made a presentation on "Predictive Maintenance in the Industry 4.0 Context."

Scott Sanner, University of Toronto
Scott's group continues work on a range of applied projects covering data-driven control systems for residential HVAC (journal article published in *Energy and Buildings*) and anomaly detection for electrical systems (journal article on Smart Grid published in *IEEE Transactions*). Scott's group also continues fundamental AI research on the topic of continual learning for deep learning (journal article published in *Neurocomputing*) as well as conversational recommender systems (conference papers at UMAP-21 and SIGIR-21), explainability in AI (NeurIPS-21), and reinforcement learning (UAI-21, NeurIPS-21, IJCAI-21).

Fae Azhari, University of Toronto
Fae's research group now consists of four doctoral students, four MASc students, and two undergraduate students. Her projects include: naval asset management using sensor data, smart composites for distributed sensing, fibre optic sensors for vibration monitoring, wind turbine monitoring, condition-based maintenance of bridges, and swimming pool freshwater usage optimization. Her group members submitted a number of journal articles and presented at virtual conferences this past year. Fae has initiated a number of new collaborations with partners in wind energy, monitoring industries, electrical engineering, and medicine.

Jue Wang, Affiliate Professor
Jue Wang's recent research on joint monitoring and learning of degradation system was presented at both the CORS and the INFORMS annual meetings. Jue submitted a paper to *Management Science* in October. Roozbeh Yousefi, a PhD candidate under his supervision, has completed a PhD thesis and will defend in January. In the past few months, Jue has been collaborating with researchers in the US and China to study the value of information in sequential learning and to develop new approximate solution algorithms.

Ali Zuashkiani, Director of Educational Programs
Ali has been providing consulting services to various industries. such as oil and gas, power generation and distribution, mining, and petrochemical. He has been also working with Aramco in Saudi to train their engineers to get CMRP designation. Ali has been active with the Institute of Asset Management, where he is helping to develop a Subject Specific Guideline (SSG) on Management of Change and is working with General Forum on Maintenance and Asset Management (GFMAM) on the third edition of *Asset Management Landscape*. Ali started working with Society for Maintenance and Reliability Professionals (SMRP) to represent them at GFMAM meetings and to work on new body of knowledge of SMRP. He is also active on ISO 55000 TC 251 committee representing Canada on WG9 working on an Asset Management Data standard. Ali has continued his teaching efforts as well. In June, October, and November, he conducted a five-day CMRP preparation training program online to a group of asset managers from different industries. In September, October, and November he delivered five-day training on CAMA designation preparation**.** In September**,** he delivered a two-day Life Cycle Costing program, and two three-day Spare Parts Management programs to engineers from Arcelormital Dofasco, and Maple Leaf Canada. In November, he delivered a five-day Reliability Management course to Socar Chemical in Azerbaijan.

C-MORE graduate students and postdoctoral fellows

Postdoctoral fellows

Danial Khorasanian started his postdoctoral fellowship in MIE at the University of Toronto in September 2021. His main research is on dynamic routing in hazardous materials transportation. He is also co-leading two groups of MEng students in multiple projects in cooperation with Nestle and Unilever companies. These projects are in the areas of forecasting, supply chain management, and image processing.

Doctoral students

Mohamed Abubakr is a first-year PhD student. His research interest is domain adaptation for continuously evolving environments, with a focus on failure detection under varying operating conditions.

Kuilin Chen is a fourth-year PhD student. His current research interest is few-shot learning. A paper on incremental few-shot learning was published in ICLR 2021. He submitted a paper on few-shot learning with stochastic weight averaging to ICLR 2022.

Michael Gimelfarb has continued his work on knowledge transfer in reinforcement learning and has been a postgraduate affiliate of the Vector Institute since April 2020. His research focuses on transferring skills robustly and safely in a risk-aware setting. His current work leverages robust and risk sensitive MDPs, representation learning and planning under uncertainty.

Yunhan Kim is a doctoral candidate at the Department of Mechanical Engineering, Seoul National University, South Korea. He is a visiting doctoral student at the Department of Mechanical and Industrial Engineering, University of Toronto, in 2021. His research topics are prognostics and health management for mechanical and manufacturing systems using vibration signal analysis and deep learning.

Scott Koshman continued his research on equipment health monitoring (EHM) for Halifax Class Frigates, under the supervision of Professor Fae Azhari. His recent focus has been data conditioning, the optimum application of parallel assets to the analysis of largish data sets (40+ billion transactions), and the fusion of data across databases. He works with data from diverse sources including EHM systems, an ERP, internal reporting, and external public environmental data. This research will inform approaches for the development of maintenance optimization models given certain types of imperfect data inputs. Scott is a senior member of American Society of Quality and recently recertified both his Certified Reliability Engineer and Certified Manager of Quality / Organizational Excellence credentials. As a member of the International Council of Systems Engineering (INCOSE), he recently participated in the 1st Virtual INCOSE International Workshop (IW2021). In his workplace, he has also been the lead for the development of graduate courses and the transition to a virtual delivery model in the subject area of global strategic security studies.

Yang Li is currently working toward the PhD degree at the school of Mechanical Engineering, Southeast University, Nanjing, China. He is a visiting PhD Student in the Department of Mechanical and Industrial Engineering, University of Toronto. His main research interests include acoustic emission signal processing and electromechanical equipment damage detection and identification.

Seyedvahid (Vahid) Najafi is a full-time PhD student working on the maintenance modelling and optimization of multi-unit series systems. He has developed condition-based maintenance strategies with general repair to minimize the operation and maintenance cost of multi-unit systems using deep reinforcement learning and dynamic programming. He recently passed his second committee meeting (November 2021).

Avi Sokol is a flex-time PhD student and a full-time employee. He continues to research integration of reinforcement learning and inventory control to reduce waste in supply chains and the benefits of reward decomposition. Avi applied reward decomposition in Q-learning and SARSA models to solve inventory control problems. Both models produced the expected results and converged to the near-optimal solution. Avi is currently exploring the application of reward decomposition in deep Q-learning models. In the past six months, Avi passed his qualifying exam and is expected to have first annual meeting in the beginning of 2021.

Master's students

Varun C Ananda Rao (MEng) is a first-year MEng student in the Mechanical and Industrial Engineering program focused on supply chain management and optimization under the ELITE emphasis. He is working on carrier optimization project working on building a model that provides an optimized solution for maintaining inventory on the carrier. His work is focused on the implementation of the optimization model, thereby creating an interface for users to easily obtain the solution.

Siddharth Behal (MEng) is currently working towards his MEng degree in Aerospace Engineering with dual emphasis on ELITE and advanced manufacturing. His background is in space engineering and rocketry. Currently, he is working on a project dealing with RUL prediction for military vehicles and how their parameters could be optimized for increased reliability, maintainability and availability.

Blair Cui (MEng) is a first-year MEng student in Mechanical and Industrial Engineering with an emphasis on data analytics and healthcare engineering. She is currently working on the development of KPIs (key performance indicators) for a mining sector project. Her work is focused on evaluating the factors affecting maintenance effectiveness and verifying the indicators' appropriateness for mining industries

Ziyi Lyu (MEng) is a second-year MEng student in Mechanical and Industrial Engineering. She works on time series forecasting of the oil rig maintenance worker's daily workloads. She is focusing on developing ARIMA-GARCH, Prophet, and regression models to improve forecasting accuracy.

Eileen Mendoza (MEng) is a part-time MEng student in Mechanical and Industrial Engineering with an emphasis on sustainable energy and ELITE. She is currently working with a team of MEng students to develop a technique for predictive maintenance of wind turbine bearings using vibration analysis and machine learning. Her work is focused on optimizing signal processing techniques to aid in premature fault detection of bearings.

Gabriel Merisanu (MEng) is a first-year MEng student in Mechanical and Industrial Engineering with an emphasis on data science and ELITE. With three years of professional experience in consulting and project management, Gabriel is passionate about leveraging this past industry knowledge while exploring greater meaning within his career and the overall positive impact engineering can have on society at large. Gabriel's research is primarily focused on developing a new KPI framework and toolset to bridge the communication gaps between maintenance teams and strategic leadership within the mining industry. Given his experience in consulting and passion for management, Gabriel is focused on delivering end-to-end solutions that integrate technology and human factors.

Dhavalkumar (Dhaval) Patel (MEng) started his MEng program in Mechanical and Industrial Engineering in September 2021. His work is focused on the exploration and application of machine learning algorithms for predictive maintenance. Currently, he is working on the development of a GAN, Generative Adversarial Network, inspired fault classification model for Wind Turbine bearings based on the data provided by the industry partner.

Jeong Cheol Seok (MEng) graduated in summer 2021 and is currently working as a machine learning engineer at one of the largest retail companies in Canada. During his time at C-MORE, he developed a MLP-LSTM-MLP architecture for prediction of engine failure in a mining site at Chile, achieving 87% recall and 95% precision. In addition, he took on a project that used a machine learning algorithm to determine the actual wind speed and direction using the LiDAR measurement, floating on sea surface. The machine learning model result showed great improvement from the benchmarked analytical model.

Pooyan Sharifi (MEng) completed his MEng program in Industrial Engineering with an emphasis on data science and analytics in August 2021. During his time at the University of Toronto, he completed his MEng project with C-MORE. His project involved the asset management and criticality analysis of a medium sized power generation facility in the GTA. Pooyan completed his undergraduate degree at the University of Waterloo in Chemical Engineering with a specialization in Process Modelling, Optimization and Control. He has over two years of experience working full-time in power generation, as well as experience working in other industries including automotive manufacturing and the defence industry. The blend of these skills were useful for tackling his MEng project. Pooyan is currently looking for opportunities in data science and analytics in a process/industrial setting which best combine his skills and experience.

Zoha Sherkat Masoumi (MEng) completed her MEng student project on sensor-based maintenance for Kinross Gold Corporation. She was aiming to predict engine failures for trucks in a real mine operating environment with the help of sensor readings. She is now a PhD student in Mechanical and Industrial Engineering.

Hazel Shi (MEng) is a first-year MEng student in Mechanical and Industrial Engineering with an emphasis in data analytics. She is currently working on the toy signal generation for the machine learning model to detect the faulty state of wind turbine bearings.

Jahyun Shin (MEng) completed her deep learning research project that automatically detects and classifies restricted items for an airport's X-ray baggage scanner. She has completed her MEng program and has graduated.

Sophie Tian (MASc) is a second-year MASc student working on the combustion signature analysis project in collaboration with the NRC. She has been focused on proposing a time series classification model to identify fire hazard characteristics.

Katie Xu (MASc) is a second-year MASc student working on the use of machine learning for process monitoring and control in 3D printing. In recent months, she has been focused on creating a dataset for this purpose.

Zihan Zhang (MASc) finished her MASc thesis titled "Pattern-driven stochastic process for degradation forecasting with applications to rechargeable batteries." She attended the MIMAR conference and gave an oral presentation on her thesis work. The extended paper is under second review by *IEEE Transaction on Industrial Informatics*. She is now pursuing her PhD degree in Industrial Engineering at Georgia Tech.

C-MORE activities with consortium members

Defence Science and Technology Laboratory (DSTL)
The aircraft carrier spares optimization project continued under Dragan's supervision with MEng student Varun C. Ananda Rao. They are developing a software tool to optimally stock spare parts. A second project in interpreting and fusing data from disparate environmental conditions began with Siddharth Patel, another MEng student. Both projects will be presented at the December meeting.

Kinross Gold Corporation
Zoha Sherkat-Masoumi completed the machine learning business case project in August. The project demonstrated ways of predicting remaining useful life of haul trucks in time for maintenance to be scheduled, avoiding unexpected downtime. The results of this project will be presented by Vahid Najafi at the December meeting. We also began discussions on two potential projects. One is to develop a framework for selecting optimal components to be replaced together under a constrained budget at a company-wide level. The second is to develop a model that defines the optimal life of an asset, incorporating many different variables. These projects are planned to be launched in early 2022.

Toronto Transit Commission (TTC)
TTC set a distinct timeline of interest for the linetest and re-inspection projects for decision making. The process of cleaning and processing the data will be presented at the December meeting.

C-MORE educational programs

C-MORE's new five-day course, "**Machine Learning and AI Applications in Physical Asset Management" (**PAM2), was delivered virtually twice over the past six months, once in June and once in August. It was offered through the School of Continuing Studies. Instructors included Janet, Chi-Guhn, Ali Zuashkiani, and Jim Reyes Picknell. Other educational programs in Spare Parts Management, Life Cycle Costing, and Asset Management will also be held in December 2021, and during 2022.

In November 2021, as in previous years, Andrew Jardine led a five-day course in Physical Asset Management through the School of Continuing Studies. Other lead instructors were Don Barry and Sharareh Taghipour. Chi-Guhn Lee presented as well. As in November 2020, it was delivered virtually.

Conclusion

The past six months have been busy, with C-MORE staff, colleagues, and collaborators hard at work on both new and ongoing projects.

Chi-Guhn Lee

December 2021

C-MORE leadership activities

Chi-Guhn Lee, Director

Chi-Guhn has leveraged the all-virtual environment to give invited talks and seminars at many conferences around the world. While the detailed list of talks is reserved for the Activities section of the report. The resource extraction working group initiative continues to grow to a capstone project for senior industrial engineering students as well as a research project for two master of engineering students. These projects will result in a working prototype of dashboard displaying KPIs based on well thought-out design. Outside of this particular sector, Chi-Guhn is increasing the presence of C-MORE in the asset management and maintenance optimization. In the early September, Chi-Guhn gave a talk on advanced machine learning methods applied to physical asset management with scarce data, and participated in round-table discussion to voice how machine learning can be done with incomplete or insufficient data. In this term, we submitted two project proposals: applications to DSI Catalyst program on probabilistic map for possible survivors after natural disaster using flying sensors, and IC-IMPACTS project on intelligent irrigation system under weather uncertainty. We also signed one project contract on predict and control supply chain with Unilever under the support of MITACS.

Janet Lam, Assistant Director

Janet has been working with graduate and undergraduate students to push on with various industry-sponsored research projects. In particular, she worked with Pooyan and Daisy on the Capital Power criticality project, and with Gabe and Blair on the Mining KPIs project. She worked directly with Jennifer Lu from TTC on extensions and updates to the TTC track inspection projects; this project will be presented in today's meeting. She played a critical role in the submission of the NSERC Mission Alliance proposal in partnership with WSP.

She met virtually with several potential collaborators to discuss ways to combine forces, as well as renewed contact with current members to keep projects fresh.

She taught one day of the 5-day Physical Asset Management 2 course, with Chi-Guhn, Ali and Jim Reyes Picknell in June and August. She presented a talk at MainTrain, the annual conference by PEMAC.

Andrew K. S. Jardine, Professor Emeritus

Andrew's main activities during the past six months have been finalizing the 3rd Edition of the book Maintenance, Replacement and Reliability that was published in September, just in time to be one of the textbooks used on the University's 5-day annual Physical Asset Management program that is designed for practicing engineers.

In addition, he taught a graduate class at the University titled Engineering Asset Management, reviewed a MITACS research proposal and was the external examiner for a doctoral thesis at the University of the West Indies. During this COVID period he presented a 3-hour webinar titled "Optimizing Preventive Maintenance, Inspection, and Predictive Maintenance Decisions" through International Business Conferences, India.

His participation continued as a member of PEMAC's Awards Committee.

Dragan Banjevic, C-MORE Consultant

Dragan continued to collaborate with C-MORE on projects with consortium members, mostly with Kinross Gold, TTC, and UKMOD. He worked directly with Varun on the implementation of UKMOD aircraft carrier spares management project. He also provided help in other projects with C-MORE students, as well as in their research.

Fae Azhari, University of Toronto

Fae's research group now consists of 4 doctoral students, 4 MASc students, and 2 undergraduate students. Her projects include: naval asset management using sensor data, smart composites for distributed sensing, fibre optic sensors for vibration monitoring, wind turbine monitoring, condition-based maintenance of bridges, and swimming pool freshwater usage optimization. Her group members submitted a number of journal articles and presented at virtual conferences this past year. Fae has initiated a number of new collaborations with partners in the wind energy and bring monitoring industries and several colleagues in electrical engineering and medicine.

Scott Sanner, University of Toronto

Scott's group continues work on a range of applied projects covering data-driven control systems for residential HVAC (journal article published in Energy and Buildings) and anomaly detection for electrical systems (journal article published in IEEE Transactions on Smart Grid). Scott's group also continues fundamental AI research on the topic of continual learning for deep learning (journal article published at Neurocomputing) as well as conversational recommender systems (with conference papers published at UMAP-21 and SIGIR-21), explainability in AI (NeurIPS-21), and reinforcement learning (UAI-21, NeurIPS-21, IJCAI-21)

Sharareh Taghipour, Ryerson University, External Collaborator

From June to December 2021, Sharareh published a book chapter entitled "The Role of Emerging Technologies in Physical Asset Management" In Maintenance, Replacement, and Reliability-Theory and Applications, Third Edition, authored by Jardine A.K.S. and Tsang A.H.C. (pp 227-263. ISBN 9780367076054. CRC Press). She lectured two sessions of the course MIE 1723H-Engineering Asset Management (graduate course, with Prof. Andrew Jardine) at University of Toronto. She also taught one session of the 5-day PAM program held from Nov 15-19, 2021. The session was about the role of emerging technologies in physical asset management. In addition, she taught one undergraduate course "Experimental Design and Quality Assurance" and one graduate course "Advanced Reliability Modeling" at Ryerson University in Fall 2021. She is working on a research program entitled "Decentralized data analytics and optimization methods for Physical Asset Management" (NSERC Discovery grant), as well as two collaborative projects with industry partners: "Real-time optimization of production scheduling" and "Failure modeling of repairable systems with data subject to left truncation and right censoring, and improving the models' accuracy using Bayesian techniques" with Axiom Group (NSERC Alliance) and Enbridge

Gas (Consulting contract), respectively. In addition, she is using the “Industry 4.0 Smart Factory System” to develop predictive maintenance models and real-time optimization of production scheduling (funding from the Ministry of Economic Development, Job Creation and Trade and John R. Evans Leaders Fund). Sharareh also presented "The Role of emerging technologies in physical asset management" at the 15th International Physical Asset Management Conference. Online, March 9 and 10, 2021, Tehran, Iran. She also was a keynoter speaker at IEEE International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC 2021). Online, August 13-15, 2021, Harbin Institute of Technology, Weihai, China, and presented the talk entitled “Predictive maintenance in the Industry 4.0 context”

Jue Wang, Affiliate Professor

Jue Wang's recent research on joint monitoring and learning of degradation system is presented in both CORS and INFORMS annual meeting. The paper is submitted to Management Science in October. Roozbeh Yousefi, a PhD candidate under his supervision, has completed the PhD thesis and will defend in January. In the past few month, Jue is collaborating with researchers in the US and China to study the value of information in sequential learning and to develop new approximate solution algorithms.

Ali Zuashkiani, Director of Educational Programs

Ali has been active in providing consulting services to various industries such as oil and gas, power generation and distribution, mining, and petrochemical. He has been also working with Aramco in Saudi to train their engineers to get CMRP designation.

Ali has been also active with Institute of Asset Management on developing a Subject Specific Guideline (SSG) on Management of Change and is working with General Forum on Maintenance and Asset Management (GFMAM) on the third edition of Asset Management Landscape. Ali has started working with Society for Maintenance and Reliability Professionals (SMRP) to represent them at GFMAM meetings and to work on new body of knowledge of SMRP. He is also active on ISO 55000 TC 251 committee representing Canada on WG9 working on Asset Management Data standard.

In June, October, and November and, Ali conducted a 5-day CMRP preparation training program online to a group of asset managers from different industries.

In September, October, and November he delivered 5-day training on CAMA designation preparation

In September Ali delivered 2-day Life Cycle Costing program, and two 3-day Spare Parts Management programs to engineers from Arcelormital Dofasco, and Maple Leaf Canada.

In November he delivered a 5-day Reliability Management course to Socar Chemical in Azerbaijan.

CMORE held AM4.0 program during the last week of August. The program will be held again in April 2022. Other educational programs in Spare Parts Management, Life Cycle Costing, and Asset Management will also be held during Dec 2021, and 2022

Overall project direction

Janet Lam, Assistant director

Goals and retrospectives

This section highlights the some of the main achievements in C-MORE for the period June 2021 – December 2021. Throughout the months of the pandemic, the C-MORE team continued to operate with work-from-home modifications, with all meetings happening virtually.

With the benefit of virtual conferences, C-MORE was very active at a wide range of meetings, as demonstrated in the activities below. C-MORE held two PAM2 courses online; once each in June and August of 2021. Another is planned for early 2022.

We continue to seek opportunities to leverage our work with external funding. We submitted a proposal to NSERC's Mission Alliance program in partnership with WSP.

Beyond our continuous work with consortium members, we have engaged with other industry partners on an ad-hoc basis, including Titan Technologies and Capital Power. Projects from both of these partners will be presented today.

Activities

Collaboration with companies and site visits

This section gives details on progress in research conducted with consortium members

Member	Collaborations
Defence Science and Technology Laboratory	The aircraft carrier spares optimization project continued under Dragan's supervision with M.Eng. student Varun C. Ananda Rao. They are developing a software tool to optimally stock spare parts. A second project in interpreting and fusing data from disparate environmental conditions began during this time with Siddharth Patel, an M.Eng student. Both projects will be presented today.
Kinross	Zoha Sherkat-Masoumi completed the machine learning business case project in August. The project demonstrated ways of predicting

Member	Collaborations
	remaining useful life of haul trucks in time for maintenance to be scheduled, avoiding unexpected downtime. The results of this project will be presented by Vahid Najafi, today. We also began discussions on two potential projects. One is to develop a framework for selecting optimal components to be replaced together under a constrained budget at a company-wide level. The second is to develop a model that defines the optimal life of an asset, incorporating many different variables. These projects are planned to launch in early 2022.
Toronto Transit Commission	TTC set a distinct timeline of interest for the linetest and re-inspection projects for decision making. The process of cleaning and processing the data will be presented today..

Theoretical work

This section on theoretical work is oriented toward students' and postdoctoral fellows' research topics.

Name	Activity
Mohamed Abubakr, Ph.D. student	Mohamed is a first year PhD student. His current research interest is domain adaptation for continuously evolving environments with focus on failure detection under varying operating conditions.
Varun C Ananda Rao, M.Eng. student	Varun is a first year MEng student in the Mechanical and Industrial engineering program focused on supply chain management and optimisation under the ELITE emphasis. He is working on carrier optimisation project focusing on building a model that provides an optimised solution for maintaining inventory on the carrier. His work is focused on the implementation of the optimisation model and thereby creating an interface for users to easily obtain the solution.
Siddharth Behal, M.Eng. student	Siddharth Behal is currently working towards his MEng degree in Aerospace Engineering with dual emphasis in ELITE and Advanced Manufacturing. His background is in Space Engineering and Rocketry. Currently, he is working on a project dealing with RUL prediction for military vehicles and how their parameters could be optimized for increased reliability, maintainability and availability. Areas of Expertise and Interests: Engineering Analysis, Data Fusion, Product Development, Management Consulting, Business Process Management, Manufacturing Digitization
Kuilin Chen, Ph.D. candidate	Kuilin is a fourth-year Ph.D. student. His current research interest is few-shot learning. One paper on incremental few-shot learning is

Name	Activity
	published in ICLR 2021. He submitted a paper on few-shot learning with stochastic weight averaging to ICLR 2022.
Blair Cui, M.Eng. student	Blair is a first year M.Eng student in Mechanical and Industrial Engineering program with emphasis in data analytics and healthcare engineering. She is currently working on development of KPI (Key Performance Indicator) for mining sector project. Her work is focused on evaluating the factors affecting the maintenance effectiveness and verify the indicators' appropriateness for mining industries.
Michael Gimelfarb, Ph.D. candidate	Michael has continued his doctoral work on knowledge transfer in reinforcement learning and has been a postgraduate affiliate of the Vector Institute since April 2020. Currently, his research focuses on transferring skills robustly and safely in a risk-aware setting. His current work leverages robust and risk sensitive MDPs, representation learning and planning under uncertainty.
Scott Koshman, Ph.D. student	Scott continues his research on equipment health monitoring (EHM) for Halifax Class Frigates, under the supervision of Professor Fae Azhari. His recent focus has been data conditioning, the optimum application of parallel assets to the analysis of largish data sets (40+ billion transactions), and the fusion of data across databases. He works with data from diverse sources including EHM systems, an ERP, internal reporting, and external public environmental data. This research will inform approaches for the development of maintenance optimization models given certain types of imperfect data inputs. Scott is a senior member of American Society of Quality and recently recertified both his Certified Reliability Engineer and Certified Manager of Quality / Organizational Excellence credentials. As a member of the International Council of Systems Engineering (INCOSE), he recently participated in the 1st Virtual INCOSE International Workshop (IW2021). In his workplace, he has also been the lead for the development of graduate courses and the transition to a virtual delivery model in the subject area of global strategic security studies.
Danial Khorasanian, postdoctoral fellow	Danial Khorasanian has started his postdoc in MIE department of UofT since Sep 2021. His main research is about dynamic routing in hazardous materials transportation. He is also co-leading two groups of M.Eng students in multiple projects in cooperation with Nestle and Unilever companies. These projects are in the areas of forecasting, supply chain management, and image processing.
Yunhan Kim, visiting Ph.D. student	Yunhan Kim is currently working toward the Ph.D. degree in the Department of Mechanical Engineering, Seoul National University, South Korea. He is a visiting scholar at the Department of Mechanical and Industrial Engineering from University of Toronto, Canada, in 2021. His research topics are prognostics and health

Name	Activity
	management for mechanical and manufacturing systems using vibration signal analysis and deep learning.
Yang Li, visiting Ph. D. student	Yang Li is currently working toward the Ph.D. degree with the school of Mechanical Engineering, Southeast University, Nanjing, China. And he is also as a visiting PhD. Student in the Department of Engineering Mechanical & Industrial Engineering, University of Toronto, St. George, Canada. His main research interests include acoustic emission signal processing, electromechanical equipment damage detection and identification.
Ziyi Lyu, M.Eng. student	Ziyi is a second year M.Eng. student in the Mechanical and Industrial engineering and works on time series forecasting of the oil rig maintenance worker's daily workloads. She is focusing on developing ARIMA-GARCH, Prophet and regression models to improve the forecasting accuracy.
Eileen Mendoza, M.Eng. student	Eileen is a part-time M.Eng student in Mechanical and Industrial Engineering with Emphasis in Sustainable Energy and ELITE. She is currently working with a team of M.Eng students to develop a technique for predictive maintenance of wind turbine bearings using vibration analysis and machine learning. Her work is focused on optimizing signal processing techniques to aid in premature fault detection of bearings.
Gabriel Merisanu, M.Eng student	Gabriel is a first year M.Eng student in Mechanical and Industrial Engineering pursuing an emphasis in Data Science and ELITE. With 3 years of professional experience in consulting and project management, Gabriel is passionate about leveraging this past industry knowledge while exploring greater meaning within his career and the overall positive impact engineering can have on society at large. Gabriel's research is primarily focused on developing a new KPI framework and toolset used to bridge the communication gaps between maintenance teams and strategic leadership within the mining industry. Due to his experience in consulting and passion for management, Gabriel is focused on delivering end-to-end solutions that successfully integrate technology and human factors.
Seyedvahid Najafi, Ph.D. student	Vahid is a full-time Ph.D. student and works on the maintenance modeling and optimization of multi-unit series systems. He has developed condition-based maintenance strategies with general repair to minimize the operation and maintenance cost of multi-unit systems using deep reinforcement learning and dynamic programming. He has recently passed his second committee meeting in Nov 2021.
Dhavalkumar Patel, M.Eng. student	Dhaval started his M.Eng program in the Department of Mechanical and Industrial Engineering in September 2021. Dhaval's work at the centre is focused towards the exploration and application of Machine

Name	Activity
	Learning algorithms for predictive maintainence. Currently, he is working on the development of a GAN, Generative Adversarial Network, inspired fault classification model for Wind Turbine bearings based on the data provided by the industry partner.
Jeong Cheol Seok, M.Eng. graduate	Jeong Cheol Seok graduated in Summer 2021 as a MEng student in Mechanical and Industrial program and he is currently working as a machine learning engineer at one of the largest retail companies in Canada. During his time at C-MORE, he developed a MLP-LSTM-MLP architecture for prediction of engine failure in a mining site at Chile, achieving 87% recall and 95% precision. Additionally, he took on another project that utilizes machine learning algorithm to determine the actual wind speed and direction using the LiDAR measurement, floating on sea surface. The machine learning model result showed great improvement from the benchmarked analytical model.
Pooyan Sharifi, M.Eng. graduate	Pooyan completed his M.Eng program in Industrial Engineering with an emphasis in Data Science and Analytics in August 2021. During his time at the University of Toronto he completed his M.Eng project with CMORE. His project involved the asset management and criticality analysis of a medium sized power generation facility in the GTA. Pooyan also completed his undergraduate degree at the University of Waterloo in Chemical Engineering with a specialization in Process Modelling, Optimization and Control. He has over two years of experience working full-time in power generation as well as experience working in other industries including automotive manufacturing and the defence industry. The blend of these skills were useful for tackling his M. Eng project. Pooyan is currently looking for opportunities in Data Science and Analytics in a process/industrial setting which best combine his skills and experience.
Zoha Sherkat Masoumi M.Eng. graduate	Zoha completed her MEng student project on sensor-based maintenance for Kinross Gold Corporation. She is aiming to predict engine failures for trucks in real mine operating environment with the help of sensor readings. She is now a Ph.D. student in Mechanical and Industrial Engineering.
Hazel Shi, M.Eng. student	Hazel is a first year M.Eng student in Mechanical and Industrial Engineering with emphasis in data analytics. She is currently working on the toy signal generation for the machine learning model to detect the faulty state of wind turbine bearings.
Jahyun Shin, M.Eng. graduate	Lucrece completed her deep learning research project that automatically detects and classifies restricted items for an airport's X-ray baggage scanner. She has completed her M.Eng program and has since graduated.

Name	Activity
Avi Sokol, Ph.D. student	As a flex-time Ph.D. student and a full-time employee, Avi continues to research integration of Reinforcement Learning and Inventory Control to reduce waste in supply chains and the benefits of Reward Decomposition. Avi applied Reward Decomposition in Q-learning and SARSA models to solve Inventory Control problems. Both models produced the expected results and converged to the near-optimal solution. Currently Avi explores the application of Reward Decomposition in Deep Q-Learning models. In the past 6 months Avi passed his qualifying exam and is expected to have first annual meeting in the beginning of 2021.
Sophie Tian, M.ASc. student	Sophie is a second year M.ASc. student working on the combustion signature analysis project in collaboration with the NRC. She has been focused on proposing a time series classification model to identify fire hazard characteristics.
Katie Xu, M.ASc. student	Katie is a second year M.ASc. student working on the use of machine learning for process monitoring and control in 3D printing. In recent months, she has been focused on creating a dataset for this purpose.
Zihan Zhang, M.ASc. graduate	During the past 6 months, Zihan has finished her MASc thesis titled *Pattern-driven stochastic process for degradation forecasting with applications to rechargeable batteries*. She attended MIMAR conference and gave an oral presentation about her thesis work. Besides, the extended paper is under the second review of *IEEE Transaction on Industrial Informatics*. She is now pursuing her PhD degree in Industrial Engineering at Georgia Tech.

Visits and interactions with consortium members and others

June 2021 – December 2021

Biweekly throughout **National Research Council of Canada**
Sophie Tian, Prof. Chi-Guhn Lee, Nour Elsagan, Yoon Ko, Dexen Xi, Hamed Maaref, Danial Khorasanian held research progress update meetings between the fire safety team at NRC and the team at the University of Toronto on combustion signature analysis project and safe routing project. A progress update will be presented today.

Weekly from June to August **Capital Power**
Janet and Pooyan met with Bill Mercer, Rob Mozzoni, Adrian Martinez and Iain Ogilvie of Capital Power to report progress on the criticality project. We completed analysis on component cooling water pumps, boiler feed pumps and groundwater pumps. The project transferred to Daisy when Pooyan graduated, and the results on the controlvalves will be presented today.

June 2021
Ali conducted a 5-day CMRP preparation training program online to a group of asset managers from different industries

June 16, 2021 **Kinross**
Vahid, Zoha and Janet met with Theresa Taylor from Kinross to discuss next steps of the machine learning case studies project

June 21, 2021 **Titan**
Chi-Guhn and Janet met with Joe Xu, Yinxing Ma and Qiang Zhao of Titan to discuss future collaborations with prospective MASc and PhD students.

June 21-25, 2021
Chi-Guhn Janet, Ali and Jim Reyes-Picknell delivered a 5-day course on Emerging Technologies in Physical Asset Management to Canadian audiences through the School of Continuing Studies.

June 24, 2021 **Kinross**
Vahid, Zoha and Janet met with Emilio Sarno, Theresa Taylor, Mitchell Code and Adrian Eden to share results on the machine learning case studies project and to discuss tasks for the end of Zoha's project.

June 29, 2021 **MIMAR**

Vahid presented a paper titled “A condition-based maintenance policy for a two-unit system subject to dependent soft and hard failures: A reinforcement learning approach” at The 11th International Conference on Modelling in Industrial Maintenance and Reliability.

Biweekly from July-Oct 2021 **Danish Hydrocarbon R&T Center**

Ziyi Lyu, Janet Lam, Waqas Khalid held regular meetings to discuss feasibility of the developed forecasting models (ARIMA-GARCH, regression) for the oil rig’s mechanical emergency maintenance daily workloads. The results will be presented today.

July 2021 **MITACS**

Andrew served as a reviewer of Mitacs Accelerate research proposal titled: “Applied intelligence for scheduling and dispatching operations in industrial applications”.

July 8, 2021 **UTERC**

Chi-Guhn gave a keynote presentation titled “Markov Decision Process to Reinforcement Learning: My Personal Journey” at the U of T Engineering Research Conference.

July 12, 2021 **TTC**

Janet met with Jennifer Lu from TTC to discuss next steps of the linetest and reinspection projects.

July 16, 2021 **Titan**

Chi-Guhn, Janet and Zihan met with Joe Xu, Yinxing Ma, Qiang Zhao, Luffy Zhou, of Titan to discuss potential projects for collaboration.

July 23, Aug 6, Sep 2, 2021 **PEMAC**

Andrew attended meetings as a member of the PEMAC Awards committee.

July 27, 2021 **Danish Hydrocarbon R&T Center**

Janet and Ziyi met with Waqas Khalid from DTU to launch the workload forecasting project.

August 13-15, 2021 **SDPC**

Sharareh gave an invited talk titled “Predictive maintenance in the Industry 4.0 context” at the IEEE International Conference on Sensing, Diagnostics, Prognostics, and Control, hosted online by Harbin Institute of Technology in Weihai China.

August 18, 2021

Janet met with Ernesto Primera from Chevron to discuss ways to working together, increasing our footprint in Latin America.

Weekly from September to December **Nestle**

Danial Khorasanian, Aloagbaye Momodu, Professor Chi-Guhn Lee, Mike Ma, Celine Sim, Negar Taherian, Dhillon Tajinder met to discuss research projects defined in the areas of advertisement analysis, forecasting, and image processing.

Weekly from September to December **Unilever**

Danial Khorasanian, Aloagbaye Momodu, Nisarg Patel, Abhi Gandhi, Jing Guo met to discuss the forecasting project and the data provided by Unilever.

August 23-27, 2021

Chi-Guhn, Janet, Ali and Jim Reyes-Picknell delivered a 5-day course on Emerging Technologies in Physical Asset Management to International audiences through the PAMCO.

September 2021
Ali delivered a 5-day training course on CAMA designation preparation.

Ali delivered a 2-day Life Cycle Costing program, and two 3-day Spare Parts Management programs to engineers from Arcelormital Dofasco, and Maple Leaf Canada.

September 1, 2021 **CEA**
Janet met with Dan Gent from Canadian Electricity Association to discuss C-MORE's role as reviewer in their new Reliability Award.

September 1-4, 2021 **Canada-Korea Conference**
Chi-Guhn attended the Canada-Korea Conference in Halifax, NS. There, he delivered a technical talk and attended two round-table sessions.

September 7, 2021 **International Business Conferences**
Andrew delivered a webinar titled "Optimizing preventive maintenance, inspection and predictive maintenance decisions" at the International Business Conferences, India.

September 15-December 1, 2021
Andrew taught the graduate course class MIE 1723 Engineering Asset Management, at the University of Toronto.

September 16, 2021 **Kinross**
Vahid met with Emilio and Vatsal from Kinross to discuss the results of a business case study project carried out by Zoha, and talked about the need for continuing similar projects.

September 22, 2021 **Telus**
Chi-Guhn and Janet met with Chris Ospina and Kenneth Liang of Telus to discuss ways of working together.

September 22, 2021 **University of the West Indies**
Andrew served as an External Examiner for PhD thesis at University of the West Indies titled: Multi-Criteria Maintenance Optimization for Improving the Performance of Offshore Production Systems.

September 24, 2021 **Kinross**
Chi-Guhn and Janet met with Emilio and Vatsal of Kinross to discuss the potential of longer-term projects with students who are assigned for 1 or more years to the project.

September 29, 2021 **PEMAC**
Janet delivered a presentation titled "Minimizing risk of failure under constrained resources: A case study with a municipal transit company" at MainTrain 2021 virtual conference.

September 30, 2021 **KORES**
Chi-Guhn delivered a lecture on machine learning technologies for physical asset management at Korean Resources Corporation.

September 30, 2021 **Titan Technologies**
Chi-Guhn, Janet, Mohamed met with Qiang Zhao and Joe Xu from Titan to discuss insights about the data they have already provided. We gave recommendations for the data format and the features they will provide in the future. Finally, we briefly discussed future directions regarding signal processing and algorithms for the bearing fault detection problem. Updates on this project will be presented today.

October 2021
The article "Optimal replacement, retrofit, and management of a fleet of assets under regulations of an emissions trading system" by Rajabian, Ghaleb, and Taghipour. was featured in the October issue of Industrial and Systems Engineering Magazine.

Ali conducted a 5-day CMRP preparation training program online to a group of asset managers from different industries

Ali delivered a 5-day training course on CAMA designation preparation.

Andrew served as a reviewer for "Optimization of Maintenance Operations at a Local Oil Company in Trinidad", West Indian Journal of Engineering.

The 3rd Edition of Maintenance, Replacement and Reliability: Theory and Applications, A K S Jardine and Albert H C Tsang, CRC Press, was published.

October 6, 2021 **UofT**
Chi-Guhn delivered a seminar on reinforcement learning for the machine intelligence option of the Engineering Science Division.

October 12, 2021 **AOMS**
Chi-Guhn and Janet met with Doug Alguire and Kevin Joseph of AOMS Technologies to discuss ways or working together. AOMS focusses on fibre optic sensing applications. A potential space of collaboration is investigating thermal and mechanical stresses within electrical transformers and transmission lines.

October 14, 2021 **UKMOD**
Janet and Siddharth Behal met with Tim Jefferies from UKMOD to discuss project progress and key findings to better understand the client requirements. Also, established future milestones in terms of relating the output back to input.

October 28, 2021 **Metrolinx**
Janet met with Xavier Hall of Metrolinx to share updates from each organization and to discuss ways of working together.

October 29, 2021 **DTU**
Janet met with Jingrui Ge of DTU to discuss future collaboration with the Danish Hydrocarbon R&T Center.

November 2021
Ali conducted a 5-day CMRP preparation training program online to a group of asset managers from different industries

Ali delivered a 5-day training course on CAMA designation preparation.

Ali delivered a 5-day Reliability Management course to Socar Chemical in Azerbaijan

November 15, 2021 **WSP**
Chi-Guhn and Janet met with Anna Robak and Will Nash of WSP to discuss a joint proposal through the NSERC Mission Alliance program. A proposal was submitted on Nov 30 as a result of this discussion.

November 15-19, 2021 **UofT**

Andrew, Sharareh and Don Barry delivered a 5-day course on University of Toronto's School of Continuing Studies Program: Physical Asset Management.

November 10, 2021 Canadian & Cold Regions Rail Research Conference
Sophie Tian presented preliminary results on using machine learning methods to detect fire hazard characteristics in freight transportation fires.

November 17, 2021 Kinross
Janet and Vahid met with Emilio and Vatsal to discuss more detail on a couple of projects identified for the new year. Two of the three proposed projects will be pursued to some degree.

November 22, 2021 DTU
Several members of the C-MORE team met with the Danish Hydrocarbon R&T Center to share our research programs, and identify potential collaborative projects. Julie Agergaard from DTU will be presenting her research today as a product of these conversations.

C-MORE publications and presentations in 2021

Journal and conference papers published or accepted

[1] Aboussalah, A., Xu, Z. and Lee, C.-G., "What is the Value of the Cross-Sectional Approach to Deep Reinforcement Learning?" Quantitative Finance, forthcoming, 2021

[2] Arasteh, M., Alizadeh, S., and Lee, C.-G. "Gravity Algorithm for the Community Detection of Large-scale Network," Journal of Ambient Intelligence and Humanized Computing, (DOI 10.1007/ s12652-021-03374-8), 2021

[3] Asadayoobi N., Jaber M., and Taghipour S. "A new learning curve with fatigue-dependent learning rate." Applied Mathematical Modeling. V. 93: 644-656, 2021.

[4] Azimpoor S. and Taghipour S. "Joint Inspection and Product Quality Optimization for a System with Delayed Failure." Reliability Engineering & System Safety. V. 215, 107793, 2021.

[5] Azimpoor S., Taghipour S., Farmanesh B., and Sharifi M. "Joint Planning of Production and Inspection of Parallel Machines with Two-phases of Failure." Reliability Engineering & System Safety. V. 217, 108097, 2022.

[6] Babatunde H.G., and Lee, C.-G., "A Marginal Log-Likelihood Approach for the Estimation of Discount Factors of Multiple Experts in Inverse Reinforcement Learning," The 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2021), September 27 - October 1, 2021

[7] Babatunde H.G., and Lee, C.-G., "Discount Factor Estimation in a Model-Based Inverse Reinforcement Learning Framework," the 2nd Workshop on Bridging the Gap between AI Planning and Reinforcement Learning (PPL) during ICAPS (The International Conference on Automated Planning and Scheduling) 2021, August 2021.

[8] Barazgan-Lari M. and Taghipour S., "A Data Mining Approach for Forecasting Machine Related Disruptions." Proceedings of the Annual Reliability and Maintainability Symposium, 2021.

[9] Barazgan-Lari M., Taghipour S., and Habibi, M., “Real-Time Contamination Zoning in Water Disrtibution Networks for Contamination Emergencies: A Case Study.” Environmental Monitoring and Assessment. V. 193 (6): 1-22, 2021.

[10] Barazgan-Lari M., Taghipour S., Zaretalab A., Sharifi M. “Production scheduling optimization for a parallel machines subject to physical distancing due to COVID-19 pandemic.” Operations Management Research. Accepted on November 3, 2021

[11] Eslami Baladeh A. and Taghipour S., “Dynamic k-out-of-n System reliability under uncertain conditions: A stochastic programming approach.” Proceedings of the Annual Reliability and Maintainability Symposium, 2022.

[12] Ghaleb M., Taghipour S., “Real-time production scheduling with random machine breakdowns using deep reinforcement learning.” Proceedings of the Annual Reliability and Maintainability Symposium, 2021.

[13] Ghaleb M., Taghipour S., Zolfagharinia H. “Real-time integrated production-scheduling and maintenance-planning in a flexible job shop with machine deterioration and condition-based maintenance.” Journal of Manufacturing Systems. V. 61, 423-449, 2021

[14] Gimelfarb, M., Barreto, A., Sanner, S. and Lee, C.-G., “Risk-Aware Transfer in Reinforcement Learning using Successor Features,” Advances in Neural Information Processing Systems, Dec 6-14, 2021.

[15] Gimelfarb, M. Sanner, S. and Lee, C.-G., “Bayesian Experience Reuse for Learning from Multiple Demonstrators,” International Joint Conference on Artificial Intelligence (IJCAI) 2021, Montreal, August 21-26, 2021.

[16] Gimelfarb, M. Sanner, S. and Lee, C.-G., “Contextual Policy Transfer in Reinforcement Learning Domains via Deep Mixtures-of-Experts,” Conference on Uncertainty in Artificial Intelligence, Online, July 27-30, 2021.

[17] Huchuk, B.; Sanner, S.; and O'Brien, W. “Development and Evaluation of Data-driven Controls for Residential Smart Thermostats.” Energy and Buildings,111201. 2021. In press.

[18] Huchuk, B.; Sanner, S.; and O'Brien, W., “Evaluation of Data-driven Thermal Models for Multi-hour Predictions using Residential Smart Thermostat Data.” Journal of Building Performance Simulation. 2021. In press.

[19] Jeong, J.; Jaggi, P.; and Sanner, S., “Symbolic Dynamic Programming for Continuous State MDPs with Linear Program Transitions.” In Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI-21), Online, 2021. To appear.

[20]Khaw, Y. M.; Jahromi, A. A.; Arani, M. F. M.; Sanner, S.; Kundur, D.; and Kassouf, M. “A Deep Learning-Based Cyberattack Detection System for Transmission Protective Relays.” IEEE Transactions on Smart Grid. 2021. In press.

[21] Lam, J, “Minimizing risk of failure under constrained resources: A case study with a municipal transit Company.” MainTrain 2021 Conference, virtual, Sep 28-29, 2021.

[22]Lyu, S.; Rana, A.; Sanner, S.; and Bouadjenek, M. R., “A Workflow Analysis of Context-driven Conversational Recommendation.” In Proceedings of the 30th International Conference on the World Wide Web (WWW-21), Ljubljana, Slovenia, 2021. To appear.

[23] Mai, Z.; Li, R.; Jeong, J.; Quispe, D.; Kim, H.; and Sanner, S., "Online Continual Learning in Image Classification: An Empirical Survey." Neurocomputing. 2021. Accepted.

[24] Mizan T. and Taghipour S. "A Causal Model for Short-Term Time Series Analysis to Predict Workload." Journal of Forecasting. Vol. 40 (2): 228-242, 2021.

[25] Najafi, S, Zheng, R. and Lee, C.-G., "An Optimal Opportunistic Maintenance Policy for Two-unit Series System with General Repair using Proportional Hazard Models," Reliability Engineering and System Safety, Vol. 215, November, 2021.

[26] Olson, A. W.; Calderon-Figueroa, F.; Bidian, O.; Silver, D.; and Sanner, S., "Reading the City through its Neighbourhoods: Deep text embeddings of Yelp reviews as a basis for determining similarity and change." Cities, 110: 103045. March 2021. In press.

[27] Patton, N., Jeong, J., Gimelfarb, M. Sanner, S., "End-to-End Risk-Aware Planning by Gradient Descent," PRL Workshop – Bridging the Gap Between AI Planning and Reinforcement Learning, Aug 5-6, 2021.

[28] Rajabian A., Ghaleb M., and Taghipour S. "Optimal replacement, retrofit, and management of a fleet of assets under regulations of an emissions trading system." The Engineering Economist. V. 66 (3), 225-244, 2021.

[29] Safaei F., Ahmadi J., and Taghipour S. "A maintenance policy for a k-out-of-n system under enhancing the system's operating time and safety constraints, and selling the second-hand components." Reliability Engineering & System Safety. V. 218, Part A, 108103.

[30] Sharifi M., Taghipour S., "Joint Optimization of the Production Scheduling, Maintenance, and Inventory." Proceedings of the Annual Reliability and Maintainability Symposium, 2021.

[31] Sharifi M. and Taghipour S. "Optimal Production and Maintenance Scheduling for a Degrading Multi-Failure Modes Single-Machine Production Environment." Applied Soft Computing. V. 106: 107312, 2021.

[32] Sharifi M. and Taghipour S. "Optimizing a Redundancy Allocation Problem with Open-Circuit and Short-Circuit Failure Modes at the Components and Subsystems Levels." Engineering Optimization. V. 53 (6), 1064-1080, 2021.

[33] Sharifi M., Taghipour S., and Abhari A. "General Transitions Probabilities and Maintenance Inspection Interval Optimization of a weighted k-out-of-n System." Proceedings of the Annual Reliability and Maintainability Symposium, 2022.

[34] Sharifi M., Taghipour S., Abhari A. "Inspection Interval Optimization for a k-out-of-n Load Sharing System under a Hybrid Mixed Redundancy Strategy." Reliability Engineering & System Safety. Vol. 213: 107681, 2021

[35] Sharifi M., Abhari A., and Taghipour S., "Modeling Real-Time Application Processor Scheduling for Fog Computing." Proceedings of Annual Modeling and Simulation Conference, 2021.

[36] Shim, D.; Mai, Z.; Jeong, J.; Sanner, S.; Kim, H.; and Jang, J., "Online Class-Incremental Continual Learning with Adversarial Shapley Value." In Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI-21), Online, 2021.

[37] Sui, Y.; Wu, G.; and Sanner, S., “Representer Point Selection via Local Jacobian Expansion for Post-hoc Classifier Explanation of Deep Neural Networks and Ensemble Models.” In *Proceedings of the 35th Annual Conference on Advances in Neural Information Processing Systems (NeurIPS-21)*, Online, 2021. To appear.

[38] Xi, D., Ko, Y., Elsagan, N. Tian, S., and Lee, C.-G., “Prediction of Emergency Response Strategies Based on Combustion Signatures from FTIR Spectroscopy Using Machine Learning Techniques,” Canadian Cold Regions Rail Research Conference (CCRC) 2021, November 9-10, 2021

[39] Xing, W. Lyu, T. Chu, X, Rong, Y., Lee, C.-G., Sun, Q. and Zou, Y. “Recognition and Classification of Single Melt Tracks using Deep Neural Network: A Fast and Effective Method to Determine Process Windows in Selective Laser Melting,” Journal of Manufacturing Processes, (https://doi.org/10.1016/j.jmapro.2021.06.076) Vol. 68, Part A, pp.1746-1757, August, 2021

[40] Yang, B., Lee, C.-G., Lei, Y. Li, N. and Na, L., “Deep partial transfer learning network: A method to selectively transfer diagnostic knowledge across related machines,” Mechanical Systems and Signal Processing, (DOI 10.1016/j.ymssp.2021.107618) Vol. 156, July 2021

[41] Yang, B., Lei, Y. Xu, S. and Lee, C.-G., “An Optimal Transport-embedded Similarity Measure for Diagnostic Knowledge Transferability Analytics across Machines,” IEEE Transactions on Industrial Electronics, July, 2021.

[42] Yang, B., Lei, Y. Xu, S. and Lee, C.-G., “Multi-source Transfer Learning Network to Complement Knowledge for Intelligent Diagnosis of Machines with Unseen Faults,” Mechanical Systems and Signal Processing, (DOI 10.1016/j.ymssp.2021.108095), Vol. 162, January 2022

[43] Yang, H.; Sanner, S.; Wu, G.; and Zhou, J. P., “Bayesian Preference Elicitation with Keyphase-Item Coembeddings for Interactive Recommendation.” In Proceedings of the the 29th Conference on User Modeling, Adaptation and Personalization (UMAP-21), Online, 2021. To appear.

[44] Yang, H.; Shen, T.; and Sanner, S., “Bayesian Critiquing with Keyphrase Activation Vectors for VAE-based Recommender Systems.” In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-21), Online, 2021. To appear.

[45] Zhang, Z., Lee, C.-G. A novel GRU-driven degradation process for end-of-life forecasting. The 11th IMA International Conference on Modelling in Industrial Maintenance and Reliability. June 2021

Journal and conference papers submitted or under review

[1] Asadayoobi N., Taghipour S., and Jaber M., “Predicting human reliability based on probabilistic mission completion time using Bayesian network.” Reliability Engineering & System Safety. Under second round of review.

[2] Barazgan-Lari M. and Taghipour S. “A Comprehensive Comparison of Machine Learning-Based Approaches for Near-Shore Wave Height Prediction Using the Data Recorded in Neighboring Off-Shore Wave-Recording Station.” Ocean Engineering. Under review.

[3] Barazgan-Lari M. and Taghipour S. “A Hybrid Data-Driven Approach for Forecasting Charactristics of Production Disruptions and Interruptions.” International Journal of Information Technology & Decision Making. Under review.

[4] Barazgan-Lari M. and Taghipour S. “Forecasting the production downtimes' characteristics in the plastic injection industry: A predictive data mining-based approach.” Journal of Forecasting. Under review.

[5] Barazgan-Lari M., Taghipour S., Zaretalab A., Sharifi M. “A Parallel-Machine Production Planning-Scheduling Model Subject to Production Disruptions with the Consideration of Physical Distancing.” IMA Journal of Management Mathematics. Under review.

[6] Chen, K. and Lee, C.-G., “Risk-Aware Transfer in Reinforcement Learning using Successor Features,” NeurIPS 2021, under review, 2021.

[7] Ghaleb M. and Taghipour S. “Evidence-based study of the impact of maintenance practices on asset sustainability.” International Journal of Production Research. Under review.

[8] Ghaleb M. and Taghipour S. “Sustainability indicator evaluation system for monitoring and assessing the impacts of maintenance practices on the overall sustainability score of an asset.” Computers & Industrial Engineering. Submitted.

[9] M.I.M. Wahab, Lee, C.-G., and Sarkar, P. “A Real Options Approach to Value Manufacturing Flexibilities with Regime-switching Product Demand with Regime Switching,” Flexible Services and Manufacturing Journal, under review, 2021.

[10] Mizan T. and Taghipour S. “An Ensemble Model to Minimize Fluctuation Influences on Short-Term Medical Workload Prediction.” Scientia Iranica. Under second round of review.

[11] Momodu, A. and Lee, C.-G., “Trinomial Tree Methods for Pricing Double-Obstacle Options with Uncertain Volatility,” Financial Markets and Portfolio Management, under review, 2021.

[12] Patton, N., Jeong, J., Gimelfarb, M. Sanner, S., “RAPTOR: End-to-end Risk-Aware MDP Planning and Policy Learning by Backpropagation,” AAAI Conference on Artificial Intelligence 2022.

[13] Russel A., Taghipour S., and Sharifi M. “Mathematical Programming and Metaheuristics for Solving Continuous-Time Scheduling Optimization Problems in Low-Volume Low-Variety Production Systems.” Engineering Optimization. Under review.

[14] Safaei F. and Taghipour S. “Optimal preventive maintenance for repairable products with three types of failures sold under a renewable hybrid FRW/PRW policy.” Reliability Engineering & System Safety. Under revision.

[15] Sharifi M., Sayyad A., and Taghipour S. “Solving A New joint Reliability-Redundancy Allocation Problem with Common Cause multi-state Failures Using Immune Algorithm.” Part O, Journal of Risk and Reliability. Under second round of review.

[16] Sharifi M. and Taghipour S. “Redundancy Allocation Problem of a Multi-State System and Components with Time-Dependent Performance Rate.” Expert Systems with Application. Under second round of review.

[17] Sharifi M., Taghipour S., and Abhari A. “Condition-Based Non-Identical Inspection Intervals Optimizing of a k-out-of-n Load Sharing System with a Hybrid Mixed Redundancy Strategy:

A Heuristic and Meta-Heuristic Approach. Knowledge-Based Systems." Under third round of review.

[18] Sharifi M., Zaretalab A., Akhavan Niaki S.T., Taghipour S., and Pourkarim Guilani P. "A Multi-Objective Model for Optimizing the Redundancy Allocation, Component Supplier Selection, and Reliable Activities for Multi-State Systems." Reliability Engineering & System Safety. Under review.

[19] Sharifi M., Zaretalab A., Shahriari M., and Taghipour S. "Calculating Transition Probabilities for Different Systems Subject to Periodic Inspection." Quality Technology & Quantitative Management. Under review.

[20]Xing, W. Lyu, T. Chu, X, Rong, Y., Lee, C.-G., Sun, Q. and Zou, Y. "Recognition and Classification of Single Melt Tracks using Deep Neural Network: A Fast and Effective Method to Determine Process Windows in Selective Laser Melting," Journal of Manufacturing Processes, revision under review, 2021.

[21] Yousefi, R. and Wang, J., Joint monitoring and learning in change detection, Submitted to Management Science (Oct, 2021)

[22]Zhang, Z., Jung, Y., Jang, J. and Lee, C.-G., "A Novel Pattern-driven Degradation Process Considering Long-term Dependencies," IEEE Transactions on Industrial Electronics, under review, 2021.

Conference presentations

[1] Asadayoobi N., Taghipour S., and Jaber M.Y. "A Probabilistic Model for Predicting Task Accomplishment Time based on Bayesian Network." Canadian Operations Research Society (CORS) Annual Conference. Virtual, June 7-10, 2021.

[2] Chen, K. and Lee, C.-G., "Incremental few-shot learning via vector quantization in deep embedded space," International Conference on Learning Representation (ICLR) 2021, May, 2021

[3] Najafi, S. and Lee, C.-G., "A Condition-based Maintenance for a Two-unit System subject to Dependent Soft and Hard Failures: A Reinforcement Learning Approach," IMA International Conference on Modelling in Industrial Maintenance and Reliability (MIMAR) 2021, June 29-July 1, 2021

[4] Taghipour S. "Role of emerging technologies in physical asset management. 15th International Physical Asset Management Conference." Virtual, March 9 and 10, 2021, Tehran, Iran.

[5] Tian S, Lee CG, Xi D, Yoon K, Elsagan N, "Prediction of Emergency Response Strategies Based on Combustion Signatures from FTIR Spectroscopy Using Machine Learning Techniques", Canadian & Cold Regions Rail Research Conference 2021.

[6] Zhang, Z.and Lee, C.-G., "A Novel Pattern-driven Stochastic Degradation Process for Battery Forecasting," IMA International Conference on Modelling in Industrial Maintenance and Reliability (MIMAR) 2021, June 29-July 1, 2021.

Predictions of emergency mechanical workloads on oil rigs in the North Sea based on ARIMA-GARCH and Facebook Prophet models

Ziyi Lyu, Janet Lam, Waqas Khalid

Introduction

Nowadays, optimizing limited resources has become a rising concern under the fast development of the global economy. The Danish Hydrocarbon Research and Technology Center of the Technical University of Denmark has launched a research project regarding forecasting the mechanical maintenance workers' workloads regarding the emergency work orders of the Maersk oil company. Predicting the future workloads and required capacity (the number of workers needed) plays a significant role in resources (human, machine) management, scheduling and maintenance. This paper will discuss several potential forecasting models developed in R and Python environments regarding the time series provided by the company and future improvement of the models.

Model Development

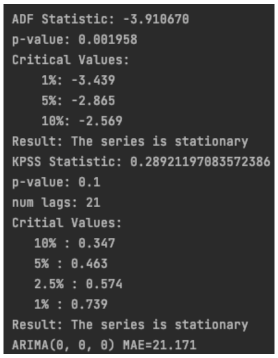
1. Data Preparation

The historical dataset for the CM work centre (generates emergency work orders) is selected and distributed from 2018.01.01 to 2020.02.15. The historical records before or beyond this specified time might not contribute to the development of the forecasting model because the company experienced an immense change in its SAP system. Furthermore, the work orders with more than 400 work hours are removed from the original dataset (before distribution) because the un-expecting long work duration might be due to human input errors and historical scheduling problems.

2. Data Analysis

2.1. Stationality

It is necessary to perform stationality checking tests of the time series before developing the models. The Augmented Dickey-Fuller (ADF) and Kwiatkowski-Philips-Schmidt-Shin (KPSS) tests are conducted to determine the stationality of the time series. The historical records are stationary and do not require any transformations. Results of the tests can be seen in Table 1.



```
ADF Statistic: -3.910670
p-value: 0.001958
Critical Values:
    1%: -3.439
    5%: -2.865
    10%: -2.569
Result: The series is stationary
KPSS Statistic: 0.28921197083572386
p-value: 0.1
num lags: 21
Critial Values:
   10% : 0.347
   5% : 0.463
   2.5% : 0.574
   1% : 0.739
Result: The series is stationary
ARIMA(0, 0, 0) MAE=21.171
```

Table 1: Results from ADF and KPSS tests

3. ARIMA Model

The ARIMA forecasting model is a common forecasting technique for time series. The model requires three parameters: p, d, q. Parameter p stands for the number of lagged observations (autoregressive coefficient), q is the number of lagged forecasted errors (moving average coefficient)[1][2]. Parameter d stands for the number of differencing orders needed to transform time series from non-stationary to stationary. From stationality testing results, the time series is stationary. Thus, d should be set to 0 in the model. The other parameters are determined through correlations graphs (auto-correlation (ACF) and particle auto-correlation graphs (PACF)) and grid search methods.

3.1 ACF, PACF and Parameter Grid Search

From Table 2, the time series does not have a clear trend and seasonality.

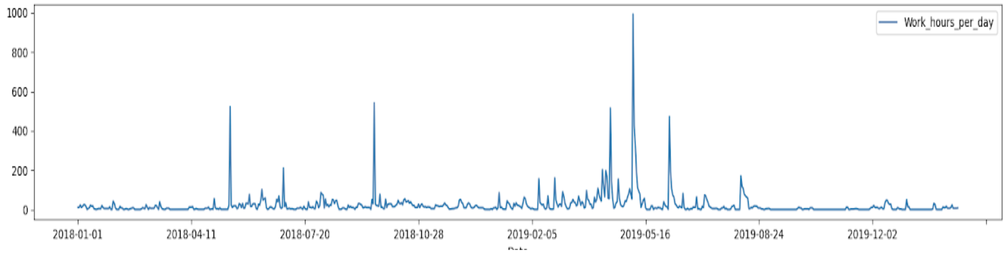


Table 2: Time series plot of the mechanical maintenance emergency work orders hours

The correlation graphs can be used to determine the range of the model parameters. The potential candidates of p are set to [0,1,2] from the ACF graph with the considerations of program runtime and efficiency. Similarly, as parameter p, the range of q is set to [0,1,2] for further investigation.

The time series is split into training and testing datasets to check the validity of each ARIMA model. The training dataset includes the first 80% (621 days) of the historical records, while the testing dataset includes the rest 20% of the data (155 days).

The optimal model is selected based on two evaluation metrics: mean absolute error (MAE) and root mean squared error (RMSE). The metrics are calculated based on in-sample forecasts produced by each model against the testing dataset. The metrics performance results are listed in Table 5. From the testing results, the ARIMA(2,0,2) fits the time series best among all the ARIMA models. The in-sample forecasting results comparison of the optimal ARIMA model against the testing dataset is shown in Table 6.

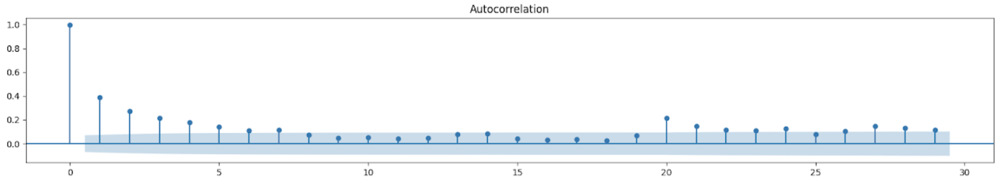


Table 3: ACF of the time series

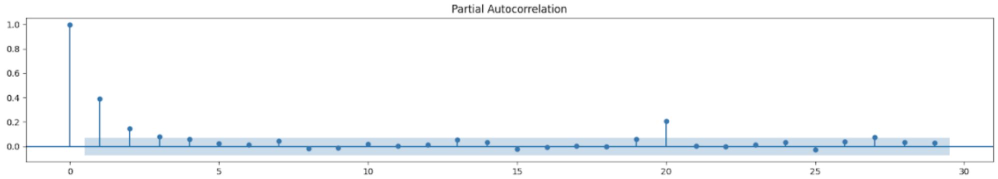
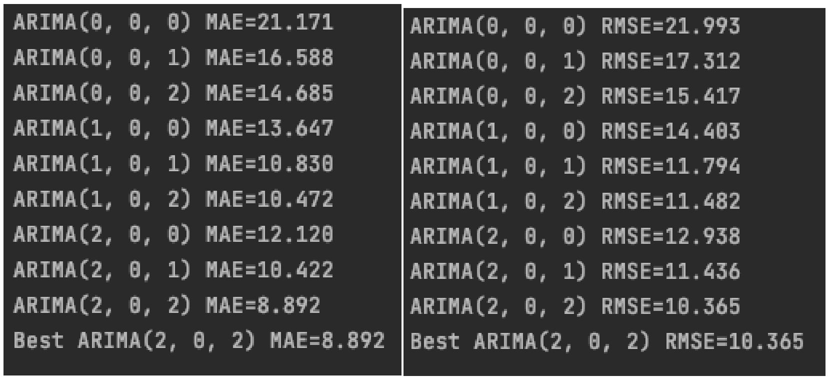


Table 4: PACF of the time series



```
ARIMA(0, 0, 0) MAE=21.171
ARIMA(0, 0, 1) MAE=16.588
ARIMA(0, 0, 2) MAE=14.685
ARIMA(1, 0, 0) MAE=13.647
ARIMA(1, 0, 1) MAE=10.830
ARIMA(1, 0, 2) MAE=10.472
ARIMA(2, 0, 0) MAE=12.120
ARIMA(2, 0, 1) MAE=10.422
ARIMA(2, 0, 2) MAE=8.892
Best ARIMA(2, 0, 2) MAE=8.892
```

```
ARIMA(0, 0, 0) RMSE=21.993
ARIMA(0, 0, 1) RMSE=17.312
ARIMA(0, 0, 2) RMSE=15.417
ARIMA(1, 0, 0) RMSE=14.403
ARIMA(1, 0, 1) RMSE=11.794
ARIMA(1, 0, 2) RMSE=11.482
ARIMA(2, 0, 0) RMSE=12.938
ARIMA(2, 0, 1) RMSE=11.436
ARIMA(2, 0, 2) RMSE=10.365
Best ARIMA(2, 0, 2) RMSE=10.365
```

Table 5: Evaluation metrics results of the time series

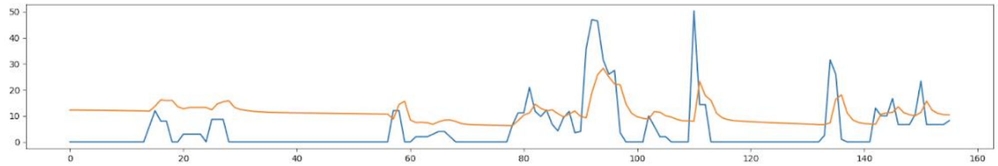


Table 6: In sample forecast of ARIMA(2,0,2) against testing dataset

3.2 Out of Sample Forecast

The ARIMA(2,0,2) failed to mimic the volatility of the historical data when making predictions of the future two years (730 days). Thus, a superior forecasting model is needed for forecasting the behaviour of the time series.

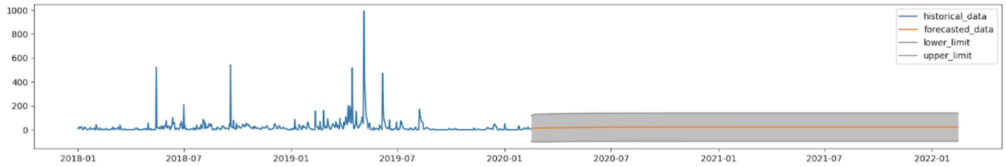


Table 7: out of sample forecast of ARIMA(2,0,2) for the next 730 days of the workloads

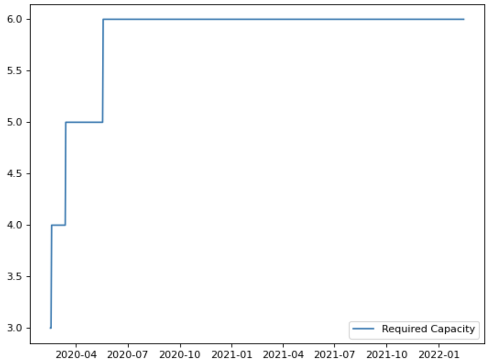


Table 8: out-of- sample forecast of ARIMA(2,0,2) for the next 730 days of the required capacity (number of mechanical maintenance workers needed)

4. ARIMA-GARCH Model

4.1 Parameter Determination

The generalized AutoRegressive Conditional Heteroscedasticity (GARCH) model is widely used for predicting the volatility of the time series. It takes the ARIMA model residuals to simulate the time series variance. From Table 9, the residuals of the ARIMA(2,0,2) model is volatile, which can be modelled to increase the forecasting accuracy or the of the time series.

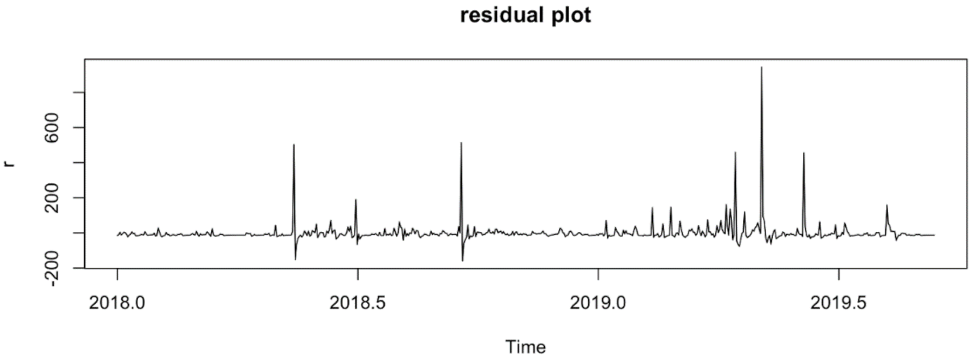


Table 9: Residual plot of the ARIMA(2,0,2) model

The combination of the two models can be applied to the time series simultaneously to the time series in the R's rugarch package. There are two parameters for the GARCH model: p and q. Parameter p is an autoregressive component that considers the effect of residuals from previous timestamps [3]. Parameter q is a moving average coefficient, considering the impact of prior variance [3]. The ACF and PACF graphs of squared residuals in Table 10 helps to determine the p and q of the GARCH model. Considering the computation time of the GARCH grid search developed in R, the ranges of p and q are both set from 0 to 9. From Table 10, when lags are equal to 20, the spikes are out of the band for both PACF and ACF plots. Thus, the combination of p and q equal to 20 are parameters candidates.

Akaike Information Criterion (AIC) is the evaluation metric for GARCH model selection. The model with the lowest AIC score will fit the time series superior to others [4]. From the output of the methods, GARCH(1,2) model has the lowest AIC score (8.920981) with the assumption of the standardized residual follows a normal distribution [5].

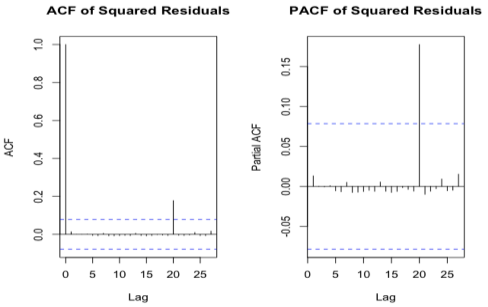
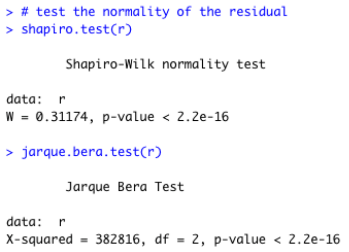


Table 10: ACF and PACF plots of the squared residuals produced by the ARIMA(2,0,2) model

However, from the Shapiro and Jarque-bera tests results shown in Table 11, the null hypothesis of the normal distribution is rejected. Moreover, the histogram of the residuals indicates that the residual is right-skewed and has a fatter tail. Thus, t or skewed student t distributions (sstd) fits the shape of the residuals better. Among all the models, GARCH(1,4) with sstd distribution has the lowest AIC score (8.141846).



```
> # test the normality of the residual
> shapiro.test(r)

        Shapiro-Wilk normality test

data:  r
W = 0.31174, p-value < 2.2e-16

> jarque.bera.test(r)

        Jarque Bera Test

data:  r
X-squared = 382816, df = 2, p-value < 2.2e-16
```

Table 11: Normality test results

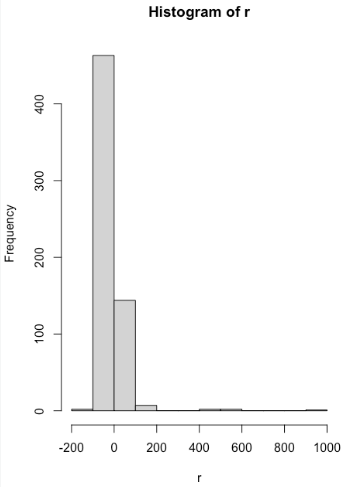


Table 12: Residual Histogram

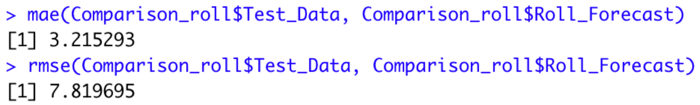
4.2 Model Evaluation

The GARCH roll method is used to backtest the specified ARIMA(2,0,2)-GARCH(1,4) with the sstd distribution model against the historical testing dataset as specified in the previous section. The in-sample forecasting horizon is 155 days. The moving window for the rolling estimation is 180 days. From the evaluation results of MAE and RMSE in Table 12 and the in-sample forecast plot in Table 13, the developed ARIMA(2,0,2)-GARCH(1,4) model fits the time-series better than the ARIMA(2,0,2) model.

4.3 Out of Sample Prediction

However, the GARCH roll forecasting method is only applicable for forecasting one step ahead of the time series. The GARCH bootstrap method can make out of sample forecasting with the considerations of the performance of the time series and its volatility. The GARCH boot uses the training dataset as inputs to learn the behaviour of the historical data and the hybrid solver to avoid the case when the program reports failing to coverage warnings.

There are two primary sources of uncertainty about n.ahead (forecast horizon) forecasting from GARCH models: one is from estimating the future density, and the other is due to parameter uncertainty [6]. The bootstrap method is based on resampling standardized residuals from the empirical distribution of the fitted model to generate future realizations of the series and sigma [6]. The partial method is chosen to consider the distribution uncertainty to avoid the expensive and lengthy parameter distribution estimation. The n.bootpred indicates the number of samples used to make every out of sample forecast. In this paper, the n.bootpred is set to 180 days, and n.head is 730 days (two years). The forecasting results (left of the horizontal bar) of the GARCH boot method can be found in Table 14. The correspondent required capacity is shown in Table 15.



```
> mae(Comparison_roll$Test_Data, Comparison_roll$Roll_Forecast)
[1] 3.215293
> rmse(Comparison_roll$Test_Data, Comparison_roll$Roll_Forecast)
[1] 7.819695
```

Table 13: MAE and RMSE results of ARIMA(2,0,2)-GARCH(1,4) with sstd

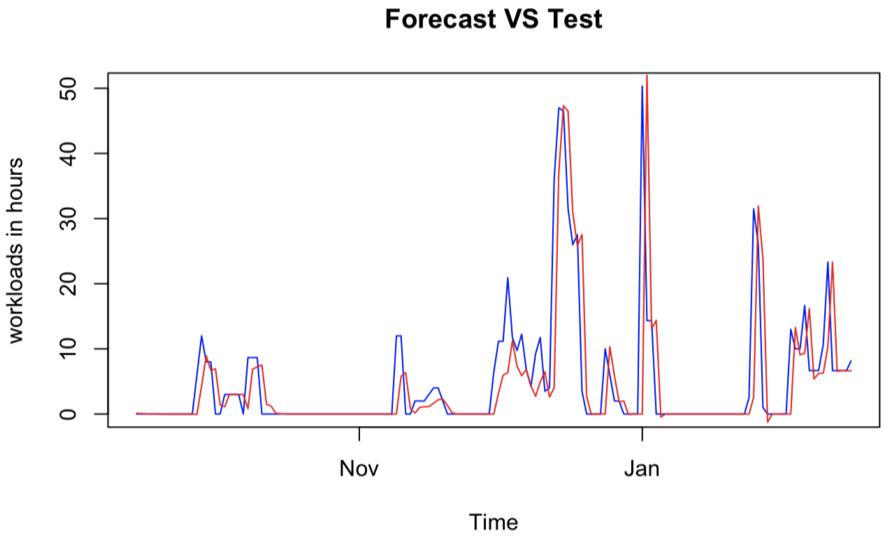


Table 14: GARCH roll in sample forecast result (red) against testing dataset (blue)

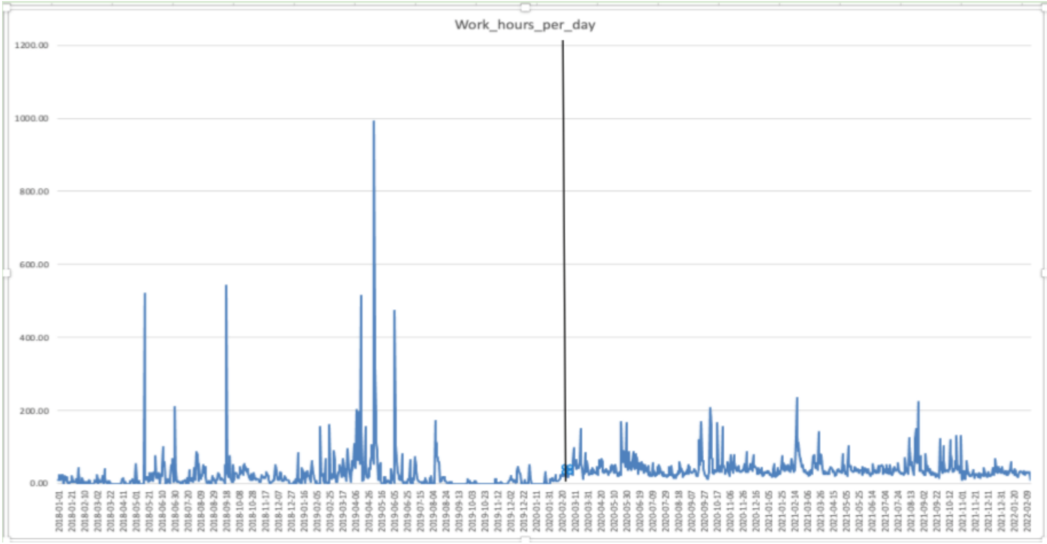


Table 15: Next two years forecast results of the GARCH boot with historical data

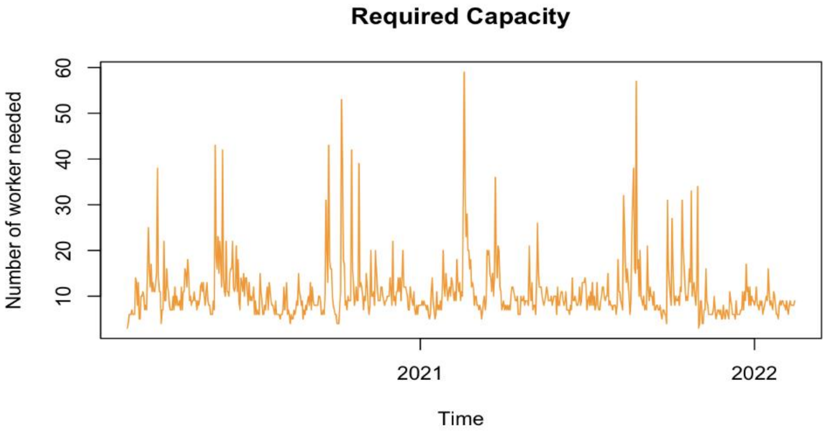


Table 16: Required capacity for next two years according to the forecasted workloads

5. Future Steps

Besides the presented models, there are certainly superior models that are more practical to use under actual circumstances. For example, the Facebook Prophet model allows the user to add extra regressors such as weather conditions, holiday effects to the model. Moreover, the combinations of ARIMA and machine learning methods can take care of both the linear and non-linear pattern in the historical data.

References

[1] "Introduction to ARIMA models", *People.duke.edu*, 2021. [Online]. Available: https://people.duke.edu/~rnau/411arim.htm. [Accessed: 05- Oct- 2021].

[2] "What Is ARIMA Modeling?", *Master's in Data Science*, 2021. [Online]. Available: https://www.mastersindatascience.org/learning/what-is-arima-modeling/. [Accessed: 05- Oct- 2021].

[3] R. Kurmar K, "Time Series Model(s)—ARCH and GARCH", *Medium*, 2020. [Online]. Available: https://medium.com/@ranjithkumar.rocking/time-series-model-s-arch-and-garch-2781a982b448. [Accessed: 05- Oct- 2021].

[4] Zach, "What is Considered a Good AIC Value? - Statology", *Statology*, 2021. [Online]. Available: https://www.statology.org/what-is-a-good-aic-value/. [Accessed: 05- Oct- 2021].

[5] "Fit a GARCH with skewed t-distribution | Python", *Campus.datacamp.com*, 2021. [Online]. Available: https://campus.datacamp.com/courses/garch-models-in-python/garch-model-configuration?ex=4https://campus.datacamp.com/courses/garch-models-in-python/garch-model-configuration?ex=4. [Accessed: 05- Oct- 2021].

[6] "ugarchboot-methods function - RDocumentation", *Rdocumentation.org*, 2021. [Online]. Available: https://www.rdocumentation.org/packages/rugarch/versions/1.4-4/topics/ugarchboot-methods. [Accessed: 05- Oct- 2021].

Key Performance Indicators within the Mining Industry' Progress Report

Gabriel Marisanu, Blair Cui

Initial problem statement

For the mining industry, the library of KPIs has not kept up with data that has become increasingly available through digitization, and there is currently a communication disconnection between maintenance teams and strategic leadership. As a consequence, maintenance leaders have a difficult time demonstrating the added value that maintenance activities create with an organization. Therefore, leading to challenges in securing the resources required for things such as continuous improvement projects. Through developing a new KPI framework and toolset to evaluate performance, our team's objective is to directly link the maintenance activities of an organization to a strategic goal that is being targeted by senior leadership and shareholders. Hence, demonstrating the strong link that exists between maintenance performance and the overall organization's performance.

Research completed to date

We have researched and analyzed journal articles and academic papers, to better understand the current state of the mining industry. Based on this initial analysis, we were able to draw a couple of conclusions as follows.

I) Maintenance is an essential part in the mining industry, which accounts for 20-50 % of the production costs within the organizations. Reducing maintenance expenditure will contribute more profits than increasing sales.

II) Maintenance teams within the mining industry are relying a lot on lagging KPIs to evaluate performance. Although there is a lot of data being generated by the equipment within this industry, this information is not being leveraged efficiently. Also, for the special characteristics of the mining industry, the existing KPIs are often better suited to manufacturing contexts.

III) The communication gaps between strategic management and maintenance teams within mining organizations, does not entirely stem from a lack of updated performance evaluation tools. Organizations must develop a framework, where high level strategic goals are broken down by functional groups in order for the various key players within the organization to agree on the permanence metrics that must be pushed to achieve the strategic goals.

IV) Additionally, we further investigated the OEE metric that is widely used in the industry, listed pros and cons associated with this. So that we can learn from the previous experience, try to avoid the limitations, and keep the merits in our future designing process.

Research goals for the upcoming presentation

For the next phase, and the upcoming presentation, we will search for the annual reports for each organization that are a part of the C-MORE consortium, and by reviewing the annual reports, identify the high level strategic goal for these organizations Kinross, Lundin Mining, Suncor, Cameco, Exxon mobil:

I) We will then transform this strategic goal into a quantifiable framework of KPIs that can be implemented at each of the hierarchical levels within the organization.

II) Look up the related papers and datasets about the KPIs of each organization currently using, evaluate them within each level of the organization. We will conduct a comparison through exploratory analysis, between the KPIs that are currently used in industry and new alternatives that may be available. An emphasis will be placed on leveraging machine learning or data science methodologies to implement more predictive KPIs, as opposed to the lagging indicators currently used.

III) During Gabriel’s time as an Energy and Sustainability associate, he had the opportunity of working with the Engineering Operations team at Cadillac Fairview to develop a KPI framework used to achieve the organization’s strategic goal of being an industry leader in energy management and tenant comfort. We will use the data from this project as a case study to provide the members of the consortium with an overview of the direction that we would like to take this research project in, additionally this will provide a great opportunity to generate a discussion with the members of the consortium at the end of the presentation.

Vehicle Performance Analysis & Prediction using Machine Learning

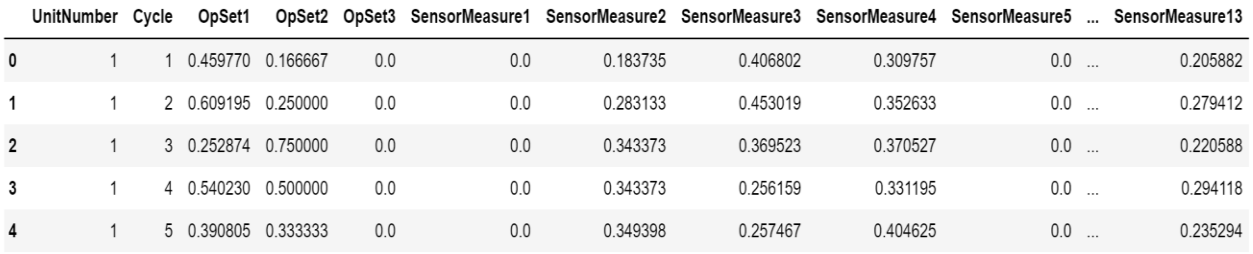
Siddharth Behal

Introduction

With the advent of advanced computational machines and reliance on higher quality standards in manufacturing, capability to optimize the performance of products has already become an important aspect in the current scenario. Additionally, when we consider the critical nature of work expected out of military vehicles on which soldier's lives depend, it becomes even more important to carefully monitor their performance and recommend changes before a critical failure could be encountered. This project aims at modelling the performance of a given vehicle using its historical data, current operational parameters, and environmental conditions for remaining useful life prediction and linking it back to the controllable user settings to extract the maximum life out of the vehicles. Here, remaining useful life indicates the expected lifetime where the probability of failure is quite low.

Dataset

The dataset utilized for this study was provided by CMAPSS (Commercial Modular Aerospace Propulsion System Solution) developed by NASA to generate data for propulsion systems. It was a multi-variate time series data. This simulated data gives a good representation of our problem where twenty-one sensor measurements are included for more than 100 turbofan engines running for n- number of cycles until a failure is detected. The failure mode for our dataset was due to a singular cause that is HPC(High-Pressure Compressor Stage) degradation. There were three operational settings which provides close resemblance to the controllable parameters for a given vehicle and which could be later utilized for linking the RUL. A sample shot of the data is presented below after sorting it through and converting into a csv file which could be used for training the machine learning algorithms.



	UnitNumber	Cycle	OpSet1	OpSet2	OpSet3	SensorMeasure1	SensorMeasure2	SensorMeasure3	SensorMeasure4	SensorMeasure5	...	SensorMeasure13
0	1	1	0.459770	0.166667	0.0	0.0	0.183735	0.406802	0.309757	0.0	...	0.205882
1	1	2	0.609195	0.250000	0.0	0.0	0.283133	0.453019	0.352633	0.0	...	0.279412
2	1	3	0.252874	0.750000	0.0	0.0	0.343373	0.369523	0.370527	0.0	...	0.220588
3	1	4	0.540230	0.500000	0.0	0.0	0.343373	0.256159	0.331195	0.0	...	0.294118
4	1	5	0.390805	0.333333	0.0	0.0	0.349398	0.257467	0.404625	0.0	...	0.235294

Fig.1 Sample Data

Here the unit number represents the engine number while the cycles represent the time parameter which portrays the age/life of the engine.

Methodology

The methodology employed for performing this study was as follows:

1. **Data Preparation –** In order to utilize data for the research purposes, we needed to convert it into labelled data as there were no labels in the raw data provided by CMAPSS which would have required unsupervised learning algorithms. However, in order to avoid unnecessary complexity, data was labelled using appropriate headers based on the text file attached with the data files. The next part of data preparation was to split the data into train and test data to be utilized for training and validating different machine learning algorithms.
2. **Model Selection –** Once the data was manipulated, multiple algorithms were tried and tested to predict the remaining-useful life of the engines. The models were trained and validated against test data. Also, they were compared using the two main metrics of performance, i.e., mean absolute error and mean-squared error for regression analysis.
3. **Tuning the Hyperparameters –** On selecting a suitable model, the next step was to tune the hyperparameters of the model to extract the best results from the model without over-fitting or under-fitting. This required careful selection of rolling window size in the case of similarity-based models and number of estimators, max depth and others in the case of random forest.
4. **Linking the Input parameters –** Since our end goal was to link the remaining useful life back to the input control variables, the approach chosen to complete this step included developing a separate relationship between the output variable (Health Indicator) and the input variables (Operational Settings). This relationship was attained using a random forest model which was again tuned using the same approach mentioned above.

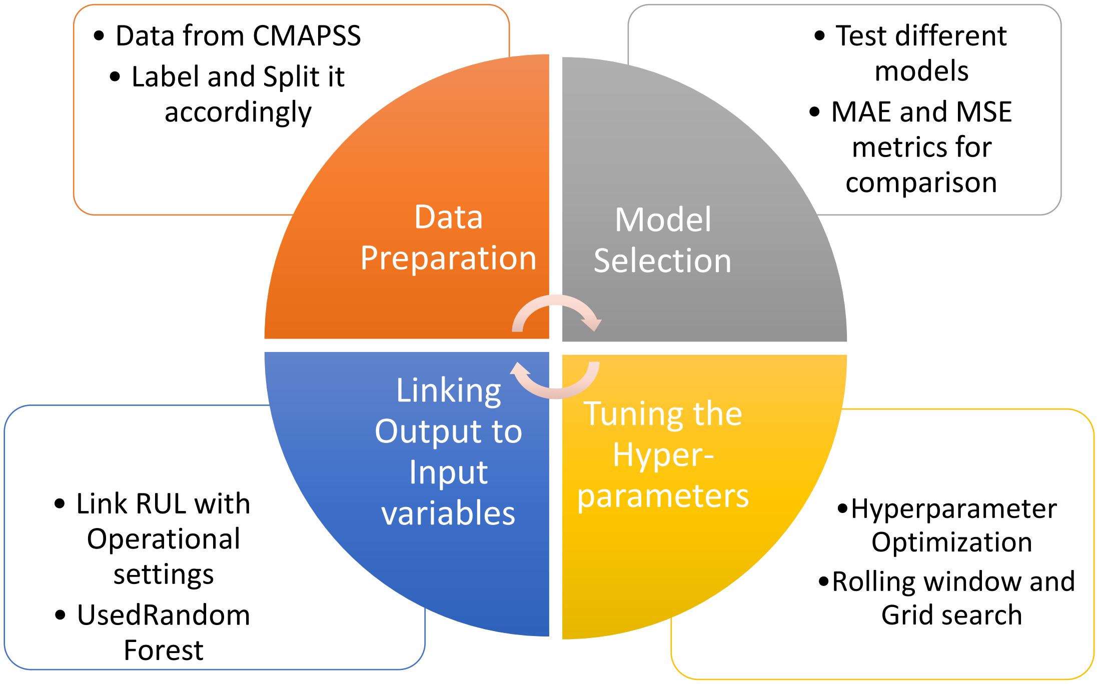

• Data from CMAPSS
• Label and Split it accordingly
Data Preparation
Model Selection
• Test different models
• MAE and MSE metrics for comparison
Linking Output to Input variables
Tuning the Hyper-parameters
• Link RUL with Operational settings
• UsedRandom Forest
•Hyperparameter Optimization
•Rolling window and Grid search

Fig.2 Study Methodology

Similarity Based RUL Model [Selected Model Description]

Similarity based RUL model was developed using an additional parameter derived from all the important features of the model. Feature extraction was done to minimize the number of variables on which this parameter would depend. The selection procedure of these parameters involved finding the Trendability score of each sensor with the increasing number of cycles. Parameters with values lower than 0.2 were ignored owing to the fact that their contribution to the deteriorating health of the engine would have been insignificant.

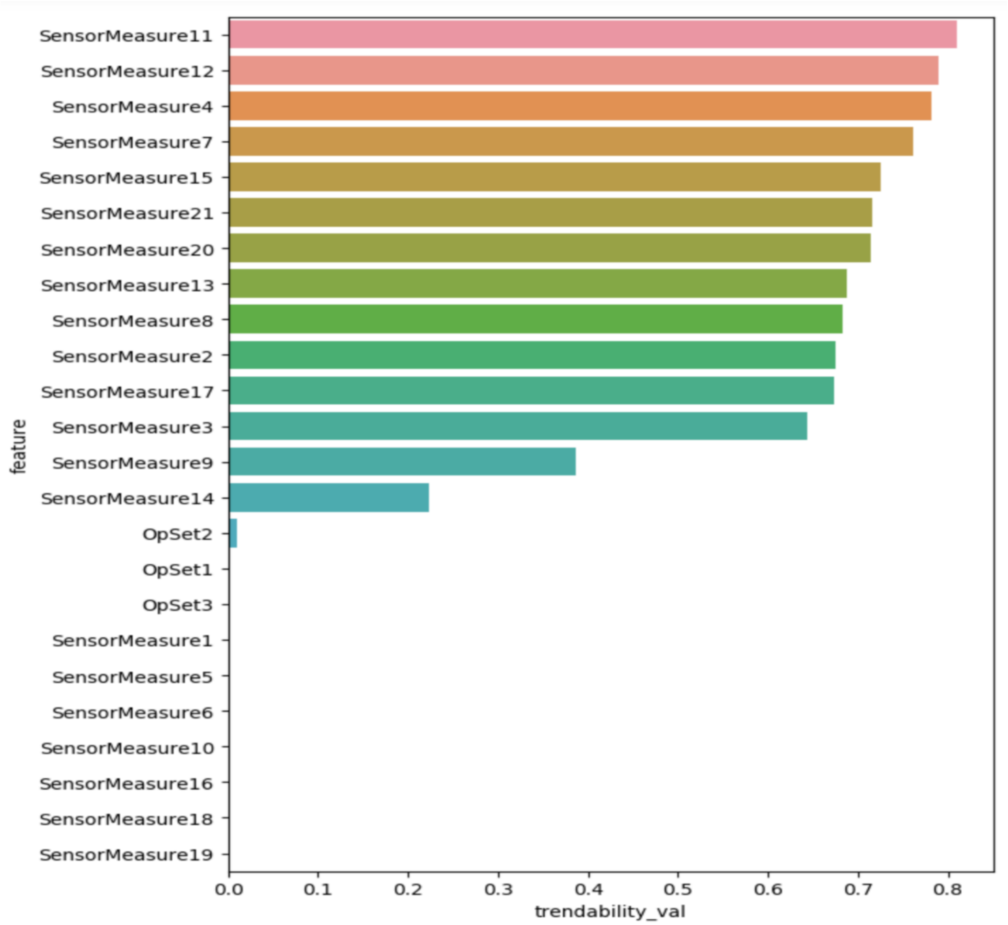


Fig.3 Feature Selection using Trendability scores

This additional parameter is named as the 'health indicator'(HI) which accounts for the variation in all selected sensor measurements. To find this parameter, first a linear model was fitted to the deteriorating total life of the engine in the test data and all selected features which were having Trendability score above the mentioned threshold value. Once the model coefficients were obtained, they were multiplied by each corresponding parameter value to generate a more sophisticated health indicator encompassing the effect of all parameters as shown in the figure below.

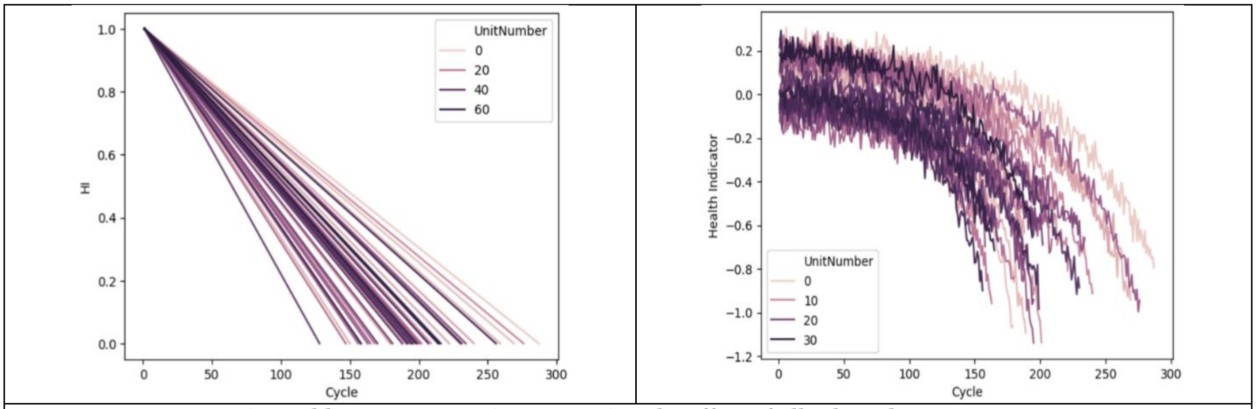


Fig.4 Old HI vs New HI incorporating the effect of all selected parameters

However, since the data contained a lot of variations, the health indicator required noise cleaning with the help of rolling window technique where the data from previous specified cycles were used to minimize the overall variations. Larger variations were causing the model to fit at much higher-order polynomials and then also it would overfit. Therefore, this technique helped overcome all such issues. The selected window size was 9 for this study.

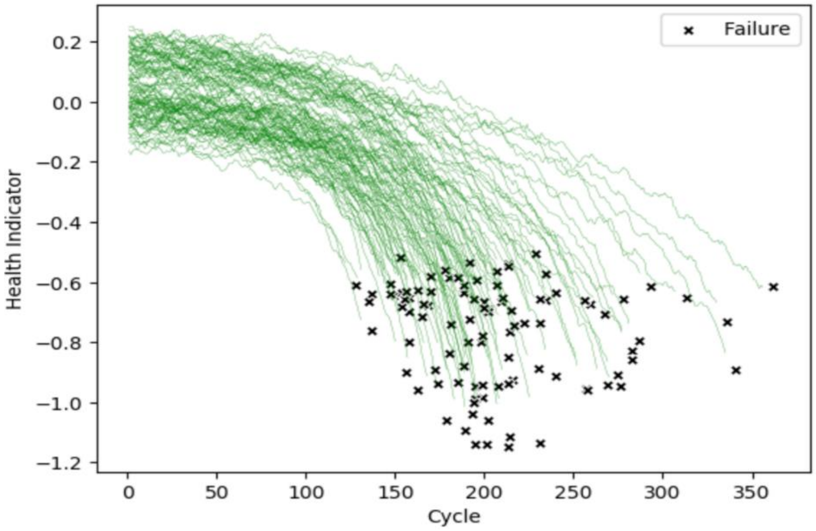
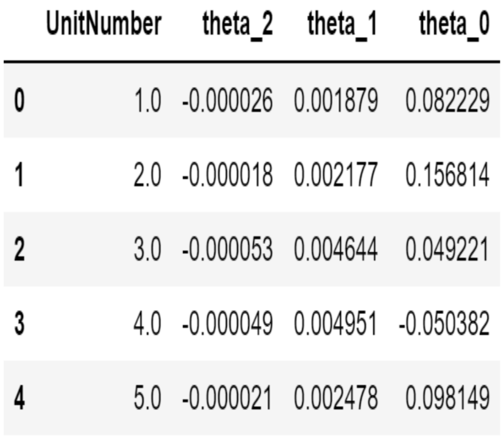


Fig.5 Smoothened HI variations w.r.t. number of cycles

On obtaining the smoothened health indicator curves for all engines, we fitted a second order polynomial to predict health indicator as direct function of number of cycles. This accounted for the effect of age on health indicator.

$$\boxed{y = \theta_0 + \theta_1 x + \theta_2 x^2}$$

Where, y is health indicator and x are number of cycles



	UnitNumber	theta_2	theta_1	theta_0
0	1.0	-0.000026	0.001879	0.082229
1	2.0	-0.000018	0.002177	0.156814
2	3.0	-0.000053	0.004644	0.049221
3	4.0	-0.000049	0.004951	-0.050382
4	5.0	-0.000021	0.002478	0.098149

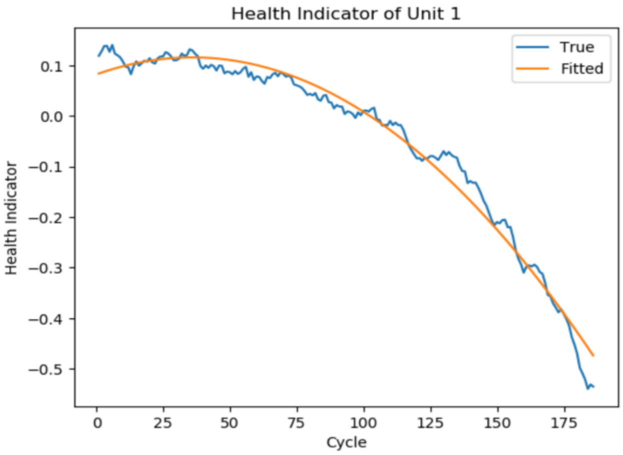
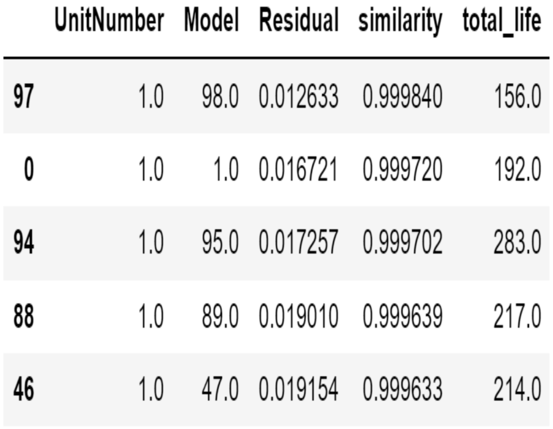


Fig.6 Sample Coefficients for the polynomial and Fitted curve

After generating these models, the next step was to calculate the remaining useful life on the test data. The technique utilized here was similarly implemented on the test data to obtain relationship between the health indicator for the test engines and the number of cycles they have been used. Since the test data had engines at some intermediate stage where they were operational, the health indicator value represented the current state of the engine. The final step was to compare those models with the train data models and find the 10 most similar engine models in the train dataset. Later their averaged total life was used to calculate the remaining useful life of the engine by subtracting the current life of the engine from the averaged value.



	UnitNumber	Model	Residual	similarity	total_life
97	1.0	98.0	0.012633	0.999840	156.0
0	1.0	1.0	0.016721	0.999720	192.0
94	1.0	95.0	0.017257	0.999702	283.0
88	1.0	89.0	0.019010	0.999639	217.0
46	1.0	47.0	0.019154	0.999633	214.0

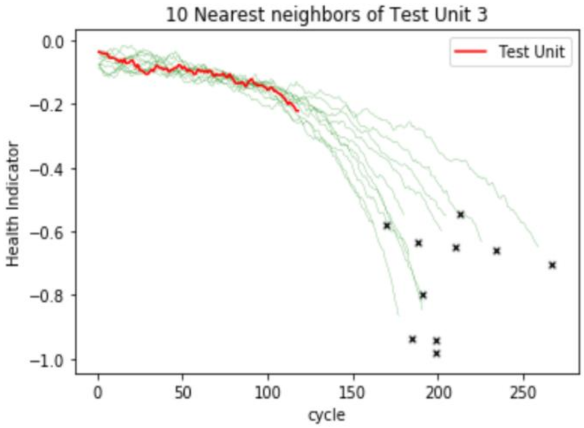


Fig.7 Similarity index evaluation for different models and visualizing 10 most similar to Engine 3

The outcome of the RUL for all engines in the test data was compared to the original RUL and depicted in the figure below.

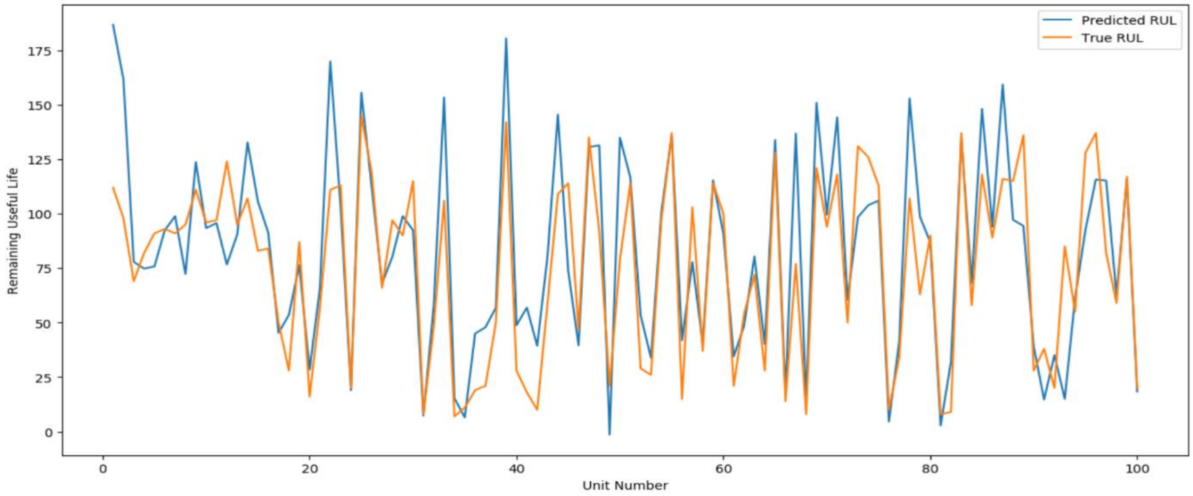


Fig.8 True and Predicted RUL vs Engine Number - From Similarity Based Model

To complete the last objective of the methodology, we utilized random forest algorithm to link the evaluated HI with the operational settings to generate recommendations for the user. These recommendations could include the range within which the health indicator could improve for the given engine and essentially its RUL as well. The random forest model resulted in good agreement with operational setting of the test data for a given health indicator value. This step was completed by tuning the hyperparameters with the help of grid search function and resulted in a MAE of 0.041 and MSE of 0.002.

Results

The reason for selecting this model in comparison to other is shown below in the form of a table where the performance metrics MAE and MSE are compared for all the tested models. Since this model was giving superior results, it was finally selected to proceed further with the objective of linking HI and input user-controlled inputs which were the two operational settings in our case.

Table 1 Comparison of different Models for RUL prediction

Model	Mean Squared Error (MSE)	Mean Absolute Error (MAE)
Random Forest	3179.90	49.34
Linear Regression	2747.63	44.35
Logical Regression	2454.39	40.50
Similarity Based Model	**638.94**	**18.62**

The results from the scatterplot of random forest model implemented after obtaining HI from similarity-based RUL model is presented below which is a clear indication of the successful modelling.

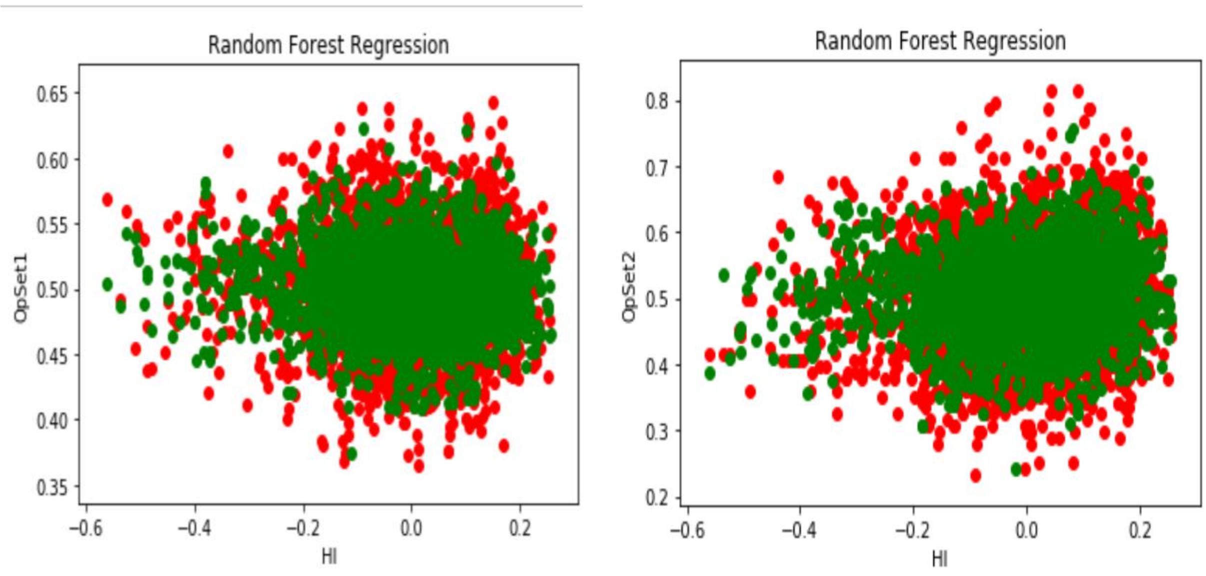


Fig.9 Results from Random Forest Regression on Operational setting 1 and 2 w.r.t. HI

Hence, the final solution to the above problem could be stated in the form of the following steps:

1. Train the model from the historical available data from various sensor measurements accounting for engine health and environmental conditions along with the user operational settings for controllable parameters.
2. Use this model as a frame of reference for a particular fleet of similar vehicles to predict their RUL by just evaluating its current health indicator value and its trajectory recorded at multiple maintenance junctions along its lifecycle.
3. On obtaining the RUL and current HI, we can recommend the changes in operational settings to get a desired RUL from the vehicle such as if required to complete a certain mission with fixed deliverables.
4. Additionally, these same recommendations could also lead to evaluating the time periods where we certainly require maintenance to improve the health indicator and effectively its RUL.

Pilot implementation of the carrier aircrafts spares optimization problem

Varun Chikkakuragodu Ananda Rao, C-MORE MEng student
Dragan Banjevic, C-MORE consultant

Background

The problem of the spares optimization for the carrier aircrafts was initiated by Tim Jefferis in June of 2020 in a discussion with Janet Lam. Dragan Banjevic investigated possible approaches to the problem and produced a report that describes practical requirements and constraints, as well as an approach to solution. Details can be found in the C-MORE December 2020 Meeting Report. In spring of 2021 it was decided to continue with the project by creation of a pilot implementation. C-MORE MEng student has been assigned to work on the implementation, with Dragan's support.

The problem can de described, in brief, as follows.

1. We have to decide what spares to take onto the Carrier to support the embarked aircraft for the mission time in terms of flight hours, availability, reliability, etc.
2. Components fail at random so the required numbers cannot be predicted with certainty. The components cannot be repaired during mission time so they should be stored in some quantities. Consumable components that are replaced on regular basis are separately considered and are not included in the problem.
3. There is a possibility for replenishment of some components during mission time using fly ins, depending on the components' sizes. Bulky components cannot be replenished.
4. Every component takes some storage space. The total storage space is limited. Some other constraints can be considered, such as cost. In the current work we consider the storage limitation as the only constrain.
5. Some components can be cannibalized.

Mathematical model (following December 2020 report)

1. The carrier embarks n aircrafts that will go on flying missions on time to time. Every flying mission (sortie) requires k operational aircrafts, $k \leq n$, at any time during total mission time $t = T$ (total time for all missions).

2. Every aircraft requires M replaceable nonrepairable components (component classes) to be able to fly. The components fail at random, with constant failure rates (per hour) $\lambda_1, \lambda_2, \ldots, \lambda_M$. We assume that all components are different and that we need one of each. If more than one component of the same class is needed (e.g., on the left and right wing), we can consider it as a single component with doubled failure rate. In that sense, we consider an aircraft as a series system with M different components. We also assume that failures of different components are independent.
3. We first classify components by replenishment levels, that is, by time intervals in which they cannot be replenished. E.g., bulky components cannot be replenished through the total mission time T, except at the beginning of the mission. If, for example, the part of the stock can be replenished two times during the entire mission, making each replenishment interval of length $T/3$ (we assume that replenishment takes place at the beginning of an interval). The intervals need not be of the same length. Thus we would have two replenishment levels, one with the single interval, and the other with three replenishment intervals $[0, t_1), [t_1, t_2), [t_2, T)$. All components that have the same replenishment strategy belong to that replenishment level. Let for a component in level i, replenishment interval lengths are $t_{i1}, t_{i2}, \ldots$, with $\sum_j t_{ij} = T$.
4. Assume we have L replenishment strategies/levels $A_1, A_2, \ldots, A_L$ and that every component belongs to one of them. Let $t_{i1}, t_{i2}, \ldots$ are replenishment intervals lengths for level i. Denote by a_{ij} the component j belonging to level i, or $a_{ij} \in A_i$, and by λ_{ij} its appropriate failure rate. Let $m_{ijl}, m_{ijl} \geq 0$ is the number of stocked units of the component a_{ij} in an interval of length t_{il}.
5. Let the total storage space is C. Assign fixed storage space $C_i, \sum_i C_i = C$ to level i. If one unit of the component a_{ij} requires space c_{ij} for storage, than at the replenishment point of the interval t_{il} the occupied space is $\sum_j c_{ij} m_{ijl}$, with constrain $\sum_j c_{ij} m_{ijl} \leq C_i$ for every l. It means that at every replenishment point we have up to C_i space available. The storage space constraints can be combined with constraints on the stock cost and handling, penalty for missing stock, etc., but we will not discuss it here.
6. Our objective is to find stock $\{m_{ijl}\}$ and stock boundaries C_i that maximize the overall reliability $R = \prod_i R_i$ of the stock under the constrains $\sum_j c_{ij} m_{ijl} \leq C_i, i, l = 1, 2, \ldots$.

Optimization algorithm

1. In our approach to optimization in 6. above, we can proceed in two steps. First, we find the optimal stock $\{m_{ijl}\}$ that depend on fixed storage boundaries $\{C_i\}$, and then optimize the storage boundaries $\{C_i\}$ under the condition $\sum_i C_i = C$. For the first step, the optimization is reduced to separate optimizations for every time interval t_{il}, with just **one condition**: for given (i, l) find $\{m_{ijl}\}$ that maximizes $R_{il} = \prod_j R_{ijl}(m_{ijl})$ under the constrain $\sum_j c_{ij} m_{ijl} \leq C_i$. In reality, we would not expect more than two or three boundaries C_i, so the second step would not be a big problem.
2. **Method 1**. The dimensionality of the problem (the number of components) can be large (as is it in this real study) and can affect the calculation. It can be reduced significantly by grouping of components that take similar space inside single level. Even, as a starting point of the calculation, we can assume that the stock sizes for a component is proportional to its failure rate – larger the failure rate, larger the stock. If $m_{il} = m_i$ is the total stock size for interval l in level i we can take $m_{ijl} \approx \frac{\lambda_{ij}}{\sum_j \lambda_{ij}} m_i$, and then calculate m_i from the condition $\sum_j c_{ij} m_{ijl} \approx C_i$, or $m_i \approx \frac{\sum_j \lambda_{ij}}{\sum_j c_{ij} \lambda_{ij}} C_i$, thus using actual storage sizes c_{ij}. Then, $m_{ijl} \approx \frac{\lambda_{ij}}{\sum_j c_{ij} \lambda_{ij}} C_i$. In this approach, stock sizes do not depend on l (time intervals t_{il}), but on available space C_i.

After using those initial values m_{ijl}^0, we can find initial values C_i^0 by minimizing in C_i. This method ("Method 1") uses predetermined (user given) storage spaces for each level.

3. **Method 2**. In this method we use a formula to calculate storage spaces for each level, given total storage space C. Then we calculate stock sizes for components using formulas from Method 1. Single component j from level i would need storage space per time unit $c_{ij}\lambda_{ij}$ (to fulfill demand), so the total space demand at level i is $\sum_j c_{ij}\lambda_{ij}$ (sum over all components in level i), and for all levels is $\sum_{i,j} c_{ij}\lambda_{ij}$. We can calculate the initial level storage C_i as proportional to its demand out of the total, or $C_i \approx \frac{\sum_j c_{ij}\lambda_{ij}}{\sum_{i,j} c_{ij}\lambda_{ij}} C$, and then we can calculate m_i as before, $m_i \approx \frac{\sum_j \lambda_{ij}}{\sum_j c_{ij}\lambda_{ij}} C_i = \frac{\sum_j \lambda_{ij}}{\sum_{i,j} c_{ij}\lambda_{ij}} C$.
4. So, we have two types of stock sizes for levels. Method 1: Given (predetermined) stock sizes, and Method 2: Calculated stock sizes using the formula from 3. The final method ("Method 3 –optimal") will be the optimal one, when we optimize the reliability of the stock, given the total stock size C. This method is currently under development. We can do optimization for components in every level for given stock sizes for levels, if they are fixed, and the overall optimization of levels stock sizes and components inside, if the level stock sizes are not fixed.

Pilot implementation

For pilot implementation we use Excel, to demonstrate ideas presented above. Real implementation would require more advanced tools, including a user-friendly interface and ability to handle a large amount of calculation/optimization. The Excel implementation use six pages, first three providing input data, and last three, outputs for different methods of calculation.

Cover page

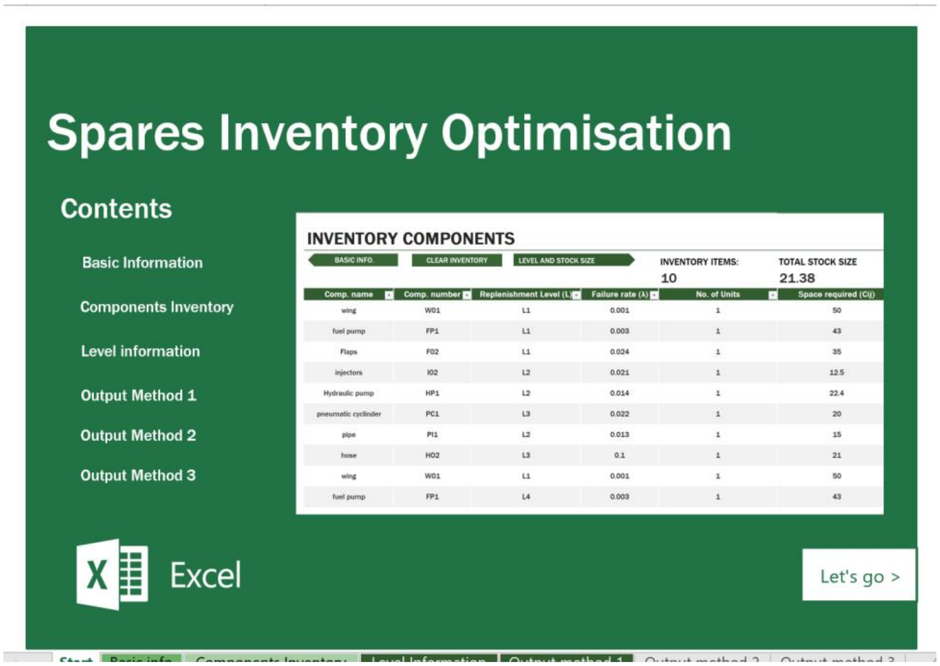
This is the first page that the user sees. This page includes the cover page which includes quick links to input data, components inventory, level information and output sheets. This sheet also provides an overview of the components inventory section.

Basic information

Screenshot shows an example of basic information sheet. This sheet includes primary input details that is required for the optimization such as the number of aircrafts embarked for the mission, mission time, total number of components per aircraft, total number of replenishment levels and total storage space available on the carrier. This page again includes quick links to other pages.

Components inventory

This screenshot is the components inventory page. This table includes the component name, component number, replenishment level, failure rate, number of components, storage space of each component, storage space for each level and cannibalization. The user can input the data in this table and once done, the output is calculated. It also shows the number of inventory items that the user has entered and the total stock size of the components on the top right of the page.



Spares Inventory Optimisation

Contents

Basic Information

Components Inventory

Level information

Output Method 1

Output Method 2

Output Method 3

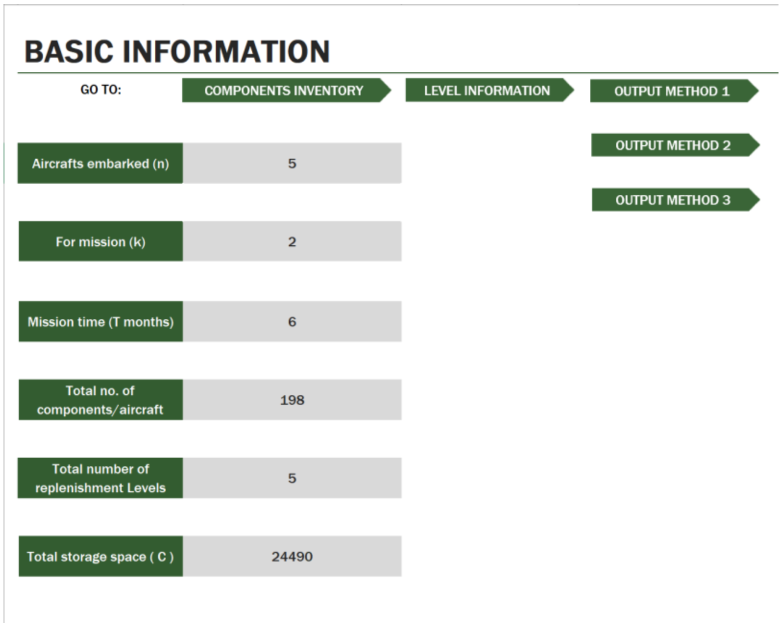
INVENTORY COMPONENTS

INVENTORY ITEMS: 10

TOTAL STOCK SIZE 21.38

Comp. name	Comp. number	Replenishment Level (L)	Failure rate (λ)	No. of Units	Space required (Cij)
wing	W01	L1	0.001	1	50
fuel pump	FP1	L1	0.003	1	43
Flaps	F02	L1	0.024	1	35
injectors	I02	L2	0.021	1	12.5
Hydraulic pump	HP1	L2	0.014	1	22.4
pneumatic cyclinder	PC1	L3	0.022	1	20
pipe	PI1	L2	0.013	1	15
hose	H02	L3	0.1	1	21
wing	W01	L1	0.001	1	50
fuel pump	FP1	L4	0.003	1	43

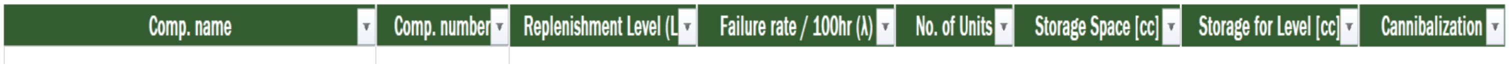
Figure 1 Cover page



BASIC INFORMATION

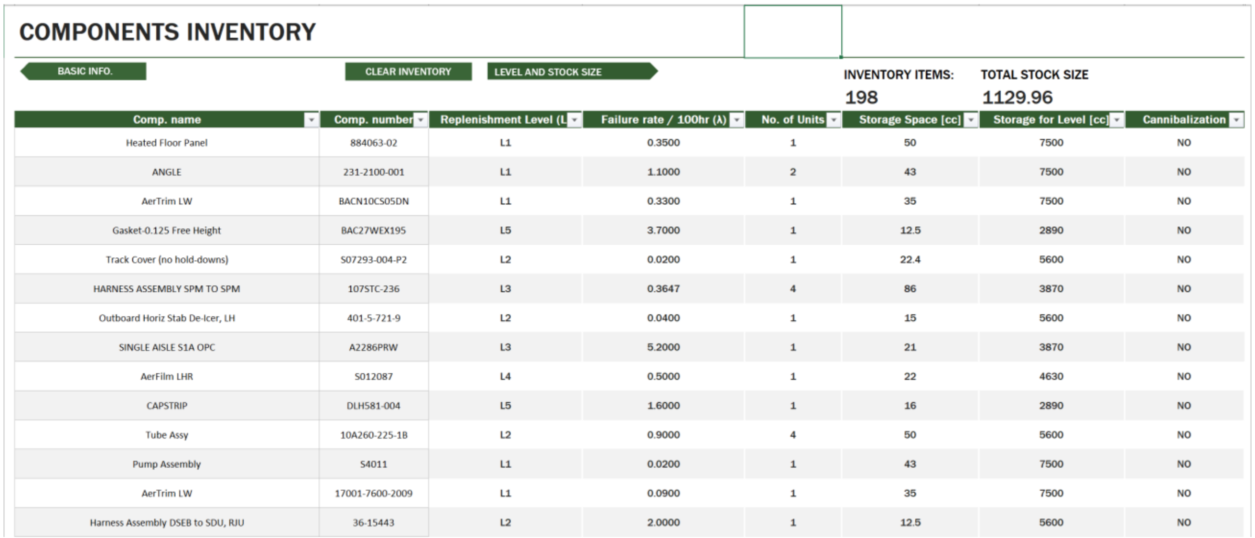
Aircrafts embarked (n)	5
For mission (k)	2
Mission time (T months)	6
Total no. of components/aircraft	198
Total number of replenishment Levels	5
Total storage space (C)	24490

Figure 2 Basic information



Comp. name	Comp. number	Replenishment Level (L	Failure rate / 100hr (λ)	No. of Units	Storage Space [cc]	Storage for Level [cc]	Cannibalization

Figure 3 Input data


COMPONENTS INVENTORY

INVENTORY ITEMS: 198

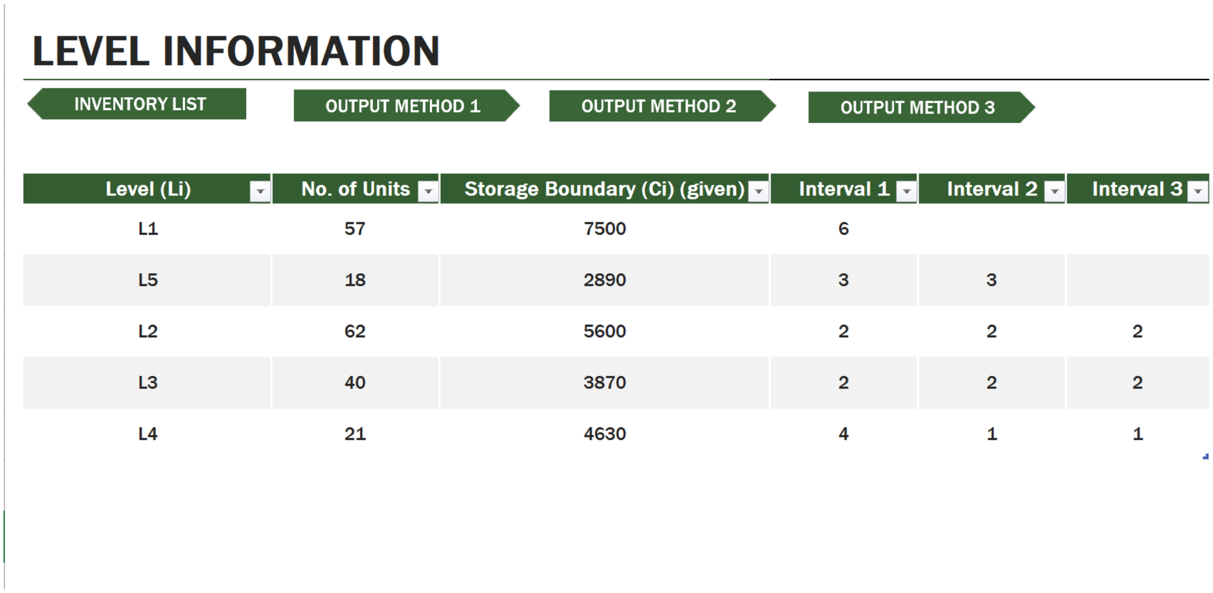
TOTAL STOCK SIZE 1129.96

Comp. name	Comp. number	Replenishment Level (L	Failure rate / 100hr (λ)	No. of Units	Storage Space [cc]	Storage for Level [cc]	Cannibalization
Heated Floor Panel	884063-02	L1	0.3500	1	50	7500	NO
ANGLE	231-2100-001	L1	1.1000	2	43	7500	NO
AerTrim LW	BACN10CS05DN	L1	0.3300	1	35	7500	NO
Gasket-0.125 Free Height	BAC27WEX195	L5	3.7000	1	12.5	2890	NO
Track Cover (no hold-downs)	S07293-004-P2	L2	0.0200	1	22.4	5600	NO
HARNESS ASSEMBLY SPM TO SPM	107STC-236	L3	0.3647	4	86	3870	NO
Outboard Horiz Stab De-Icer, LH	401-5-721-9	L2	0.0400	1	15	5600	NO
SINGLE AISLE S1A OPC	A2286PRW	L3	5.2000	1	21	3870	NO
AerFilm LHR	S012087	L4	0.5000	1	22	4630	NO
CAPSTRIP	DLH581-004	L5	1.6000	1	16	2890	NO
Tube Assy	10A260-225-1B	L2	0.9000	4	50	5600	NO
Pump Assembly	S4011	L1	0.0200	1	43	7500	NO
AerTrim LW	17001-7600-2009	L1	0.0900	1	35	7500	NO
Harness Assembly DSEB to SDU, RJU	36-15443	L2	2.0000	1	12.5	5600	NO

Figure 4 Components inventory

Level information

This section belongs to the replenishment level information page. In this page the user inputs the different interval information. The interval information is nothing but the time at which components of each level is to be replenished. This information is crucial for the calculation of stock sizes and subsequent optimisation of the storage.

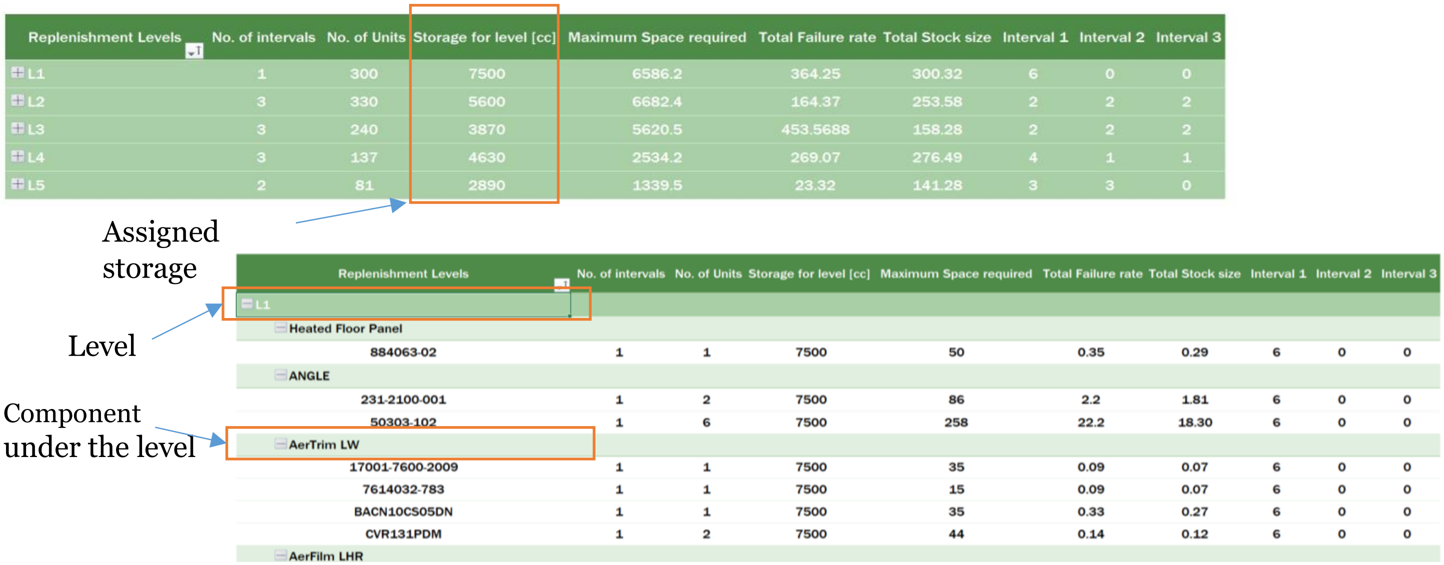

LEVEL INFORMATION

Level (Li)	No. of Units	Storage Boundary (Ci) (given)	Interval 1	Interval 2	Interval 3
L1	57	7500	6		
L5	18	2890	3	3	
L2	62	5600	2	2	2
L3	40	3870	2	2	2
L4	21	4630	4	1	1

Figure 5 Level information

Stock size output - predetermined storage

This is the first output page. This sheet summarises all the input data with its corresponding output information. Most important information that is depicted is the total stock size. Total stock size is the sum of stock sizes of each component in each replenishment level. Initially, the table is presented in a consolidated view. If the user wishes to see output of each component, dropdown option in the replenishment column can be used.



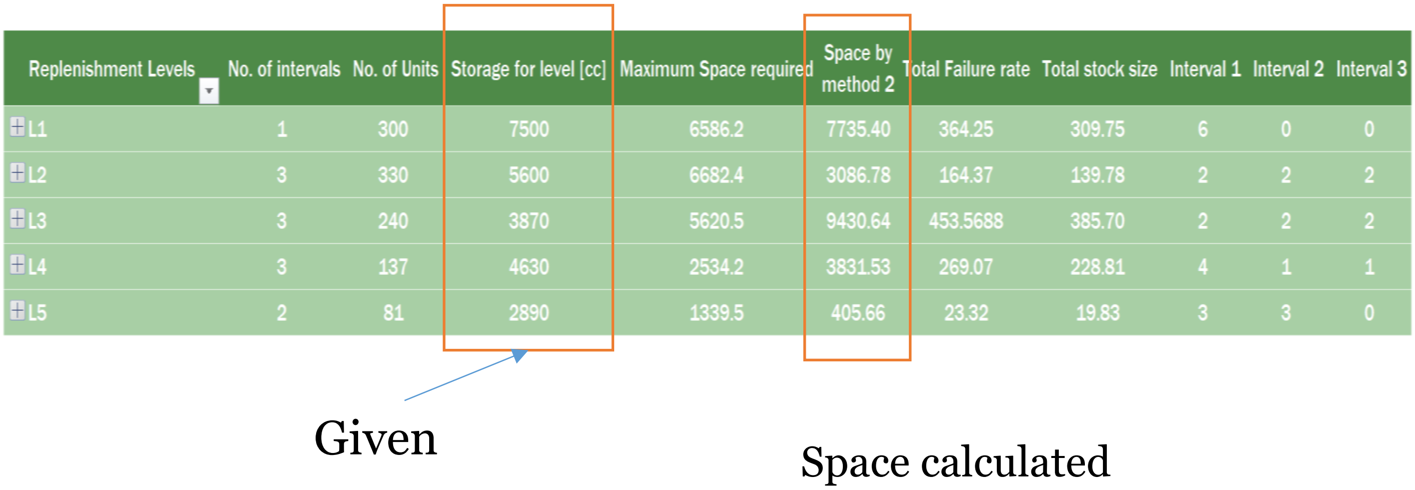
Replenishment Levels	No. of intervals	No. of Units	Storage for level [cc]	Maximum Space required	Total Failure rate	Total Stock size	Interval 1	Interval 2	Interval 3
⊞L1	1	300	7500	6586.2	364.25	300.32	6	0	0
⊞L2	3	330	5600	6682.4	164.37	253.58	2	2	2
⊞L3	3	240	3870	5620.5	453.5688	158.28	2	2	2
⊞L4	3	137	4630	2534.2	269.07	276.49	4	1	1
⊞L5	2	81	2890	1339.5	23.32	141.28	3	3	0

Replenishment Levels	No. of intervals	No. of Units	Storage for level [cc]	Maximum Space required	Total Failure rate	Total Stock size	Interval 1	Interval 2	Interval 3
⊟L1									
⊟Heated Floor Panel									
884063-02	1	1	7500	50	0.35	0.29	6	0	0
⊟ANGLE									
231-2100-001	1	2	7500	86	2.2	1.81	6	0	0
50303-102	1	6	7500	258	22.2	18.30	6	0	0
⊟AerTrim LW									
17001-7600-2009	1	1	7500	35	0.09	0.07	6	0	0
7614032-783	1	1	7500	15	0.09	0.07	6	0	0
BACN10CS05DN	1	1	7500	35	0.33	0.27	6	0	0
CVR131PDM	1	2	7500	44	0.14	0.12	6	0	0
⊟AerFilm LHR									

Figure 6 stock size - predetermined storage

Stock size output – Calculated storage boundary

This is the Second output page includes calculated storage boundary in comparison to given storage boundary. Based on the calculated storage boundary, new total stock size for each component and each replenishment level is computed. This total stock size can be compared to the stock size calculated using given data.

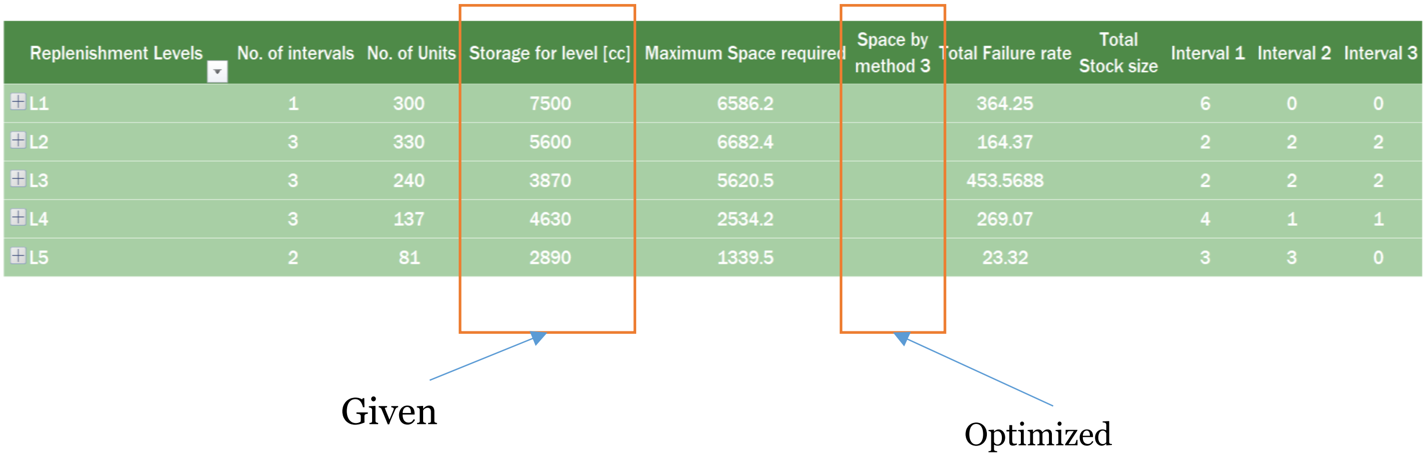


Replenishment Levels	No. of intervals	No. of Units	Storage for level [cc]	Maximum Space required	Space by method 2	Total Failure rate	Total stock size	Interval 1	Interval 2	Interval 3
⊞L1	1	300	7500	6586.2	7735.40	364.25	309.75	6	0	0
⊞L2	3	330	5600	6682.4	3086.78	164.37	139.78	2	2	2
⊞L3	3	240	3870	5620.5	9430.64	453.5688	385.70	2	2	2
⊞L4	3	137	4630	2534.2	3831.53	269.07	228.81	4	1	1
⊞L5	2	81	2890	1339.5	405.66	23.32	19.83	3	3	0

Figure 7 Calculated storage boundary

Stock size output – optimised

In the third output page, the table includes optimized space calculated using the reliability formula. This optimized space can further be used to calculate optimized stock size for each component and replenishment level. However, this step of the output section has to be implemented. Below screenshot of the output table shows the expected output table design after the calculations.



Replenishment Levels	No. of intervals	No. of Units	Storage for level [cc]	Maximum Space required	Space by method 3	Total Failure rate	Total Stock size	Interval 1	Interval 2	Interval 3
⊞L1	1	300	7500	6586.2		364.25		6	0	0
⊞L2	3	330	5600	6682.4		164.37		2	2	2
⊞L3	3	240	3870	5620.5		453.5688		2	2	2
⊞L4	3	137	4630	2534.2		269.07		4	1	1
⊞L5	2	81	2890	1339.5		23.32		3	3	0

Figure 8 Optimized stock size

Optimisation Algorithm

We consider gradient decent algorithm for the purpose of optimising the space allotment for the components.

Consider single time interval of length t on a single level, with a constraint on storage space C_{up}. Consider N components with failure rates $\lambda_1, \lambda_2, ..., \lambda_N$. If there are several equal components, n_i, failure rate is used for the group as $\lambda_i' = n_i\lambda_i$. Total reliability for all components is $\mathrm{R} = R(\boldsymbol{x}) = \prod R_i(x_i)$, and total storage space occupied is $C = C(\boldsymbol{x}) = \sum c_i x_i$, $\boldsymbol{x} = (x_1, x_2, ..., x_N)$.

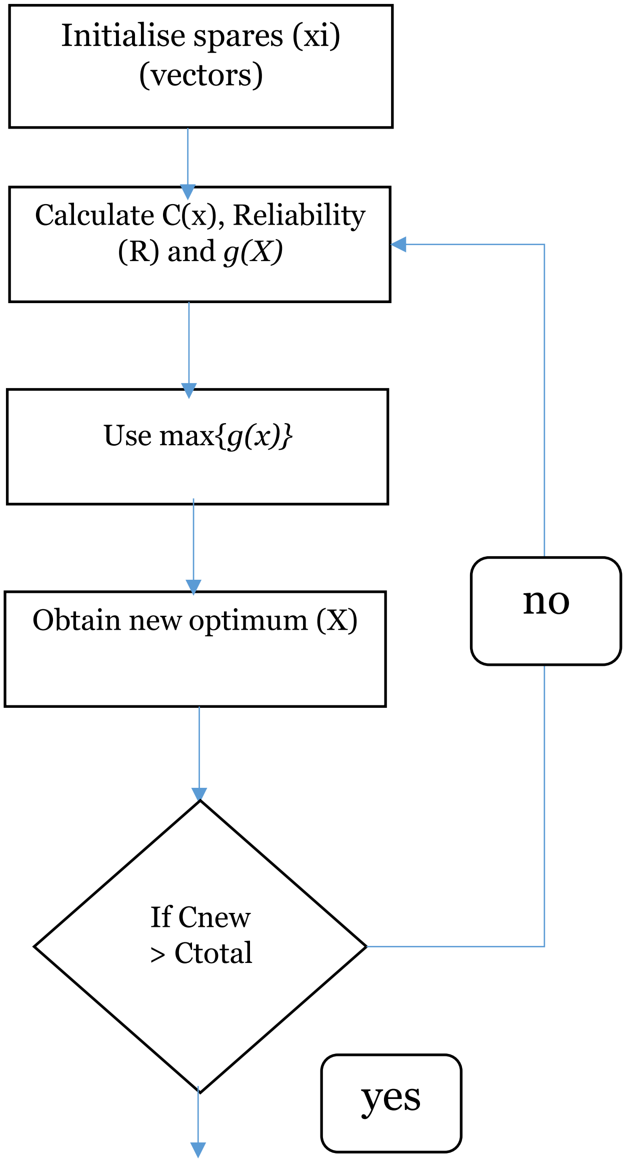
We use max of *g(x)* – gradient function to determine the optimal case.

$$R_i(x_i) = \sum_{l=0}^{x_i} \frac{(k\lambda_i t)^l}{l!} e^{-k\lambda_i t} \qquad g_i = g_i(x_i^0) = \frac{R_i(x_i^0 + 1) - R_i(x_i^0)}{c_i R_i(x_i^0)} :$$

g(x) is dependent on the reliability function (R), where:

$$R_i(x_i + 1) - R_i(x_i)$$

To consolidate al the optimisation steps into a simple flow chart, consider this:


Initialise spares (xi) (vectors)
Calculate C(x), Reliability (R) and *g(X)*
Use max{*g(x)*}
Obtain new optimum (X)
If Cnew > Ctotal
no
yes

Sensitivity analysis

Mission time interval
Reliability and space used are proportional to each other for given time interval. For given space, reliability and time interval are inversely proportional. For given reliability, as time interval increases, the number of spares to be loaded increases which results in higher space requirement.

Failure rate of Components
Failure rate is inversely proportional to reliability. Failure rate for nominal case is represented in the orange curve. If the failure rate of components increases by 1.5 times, then reliability or the probability of running out of spare parts increases. In other words, more components need to be stocked as the failure rate of those components is high.

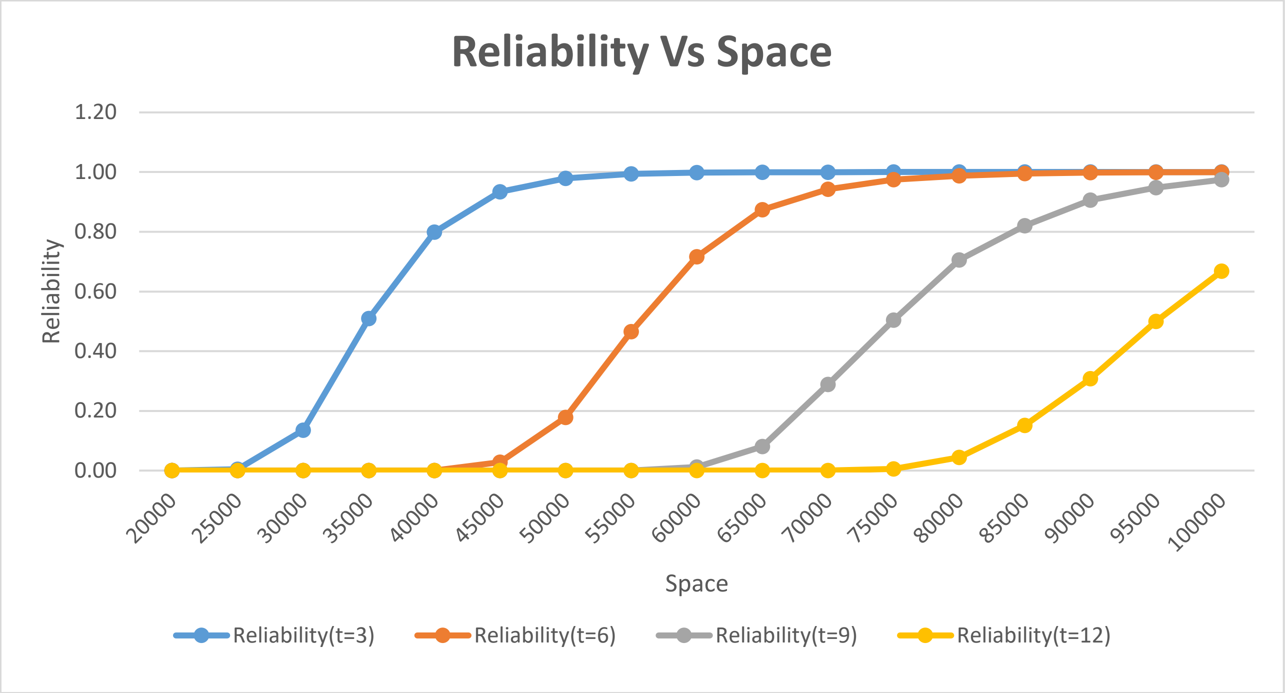


Figure 9 Mission time interval

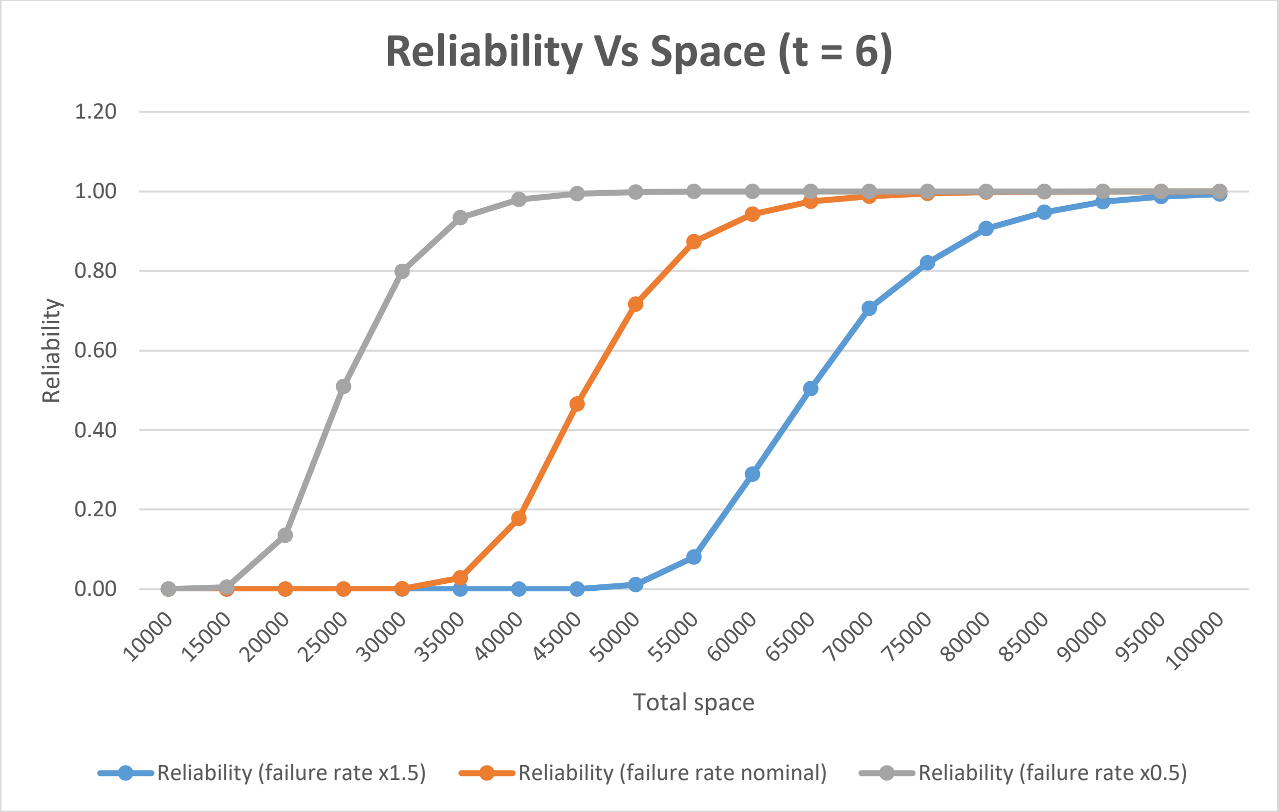


Figure 10 Failure rate of components (a)

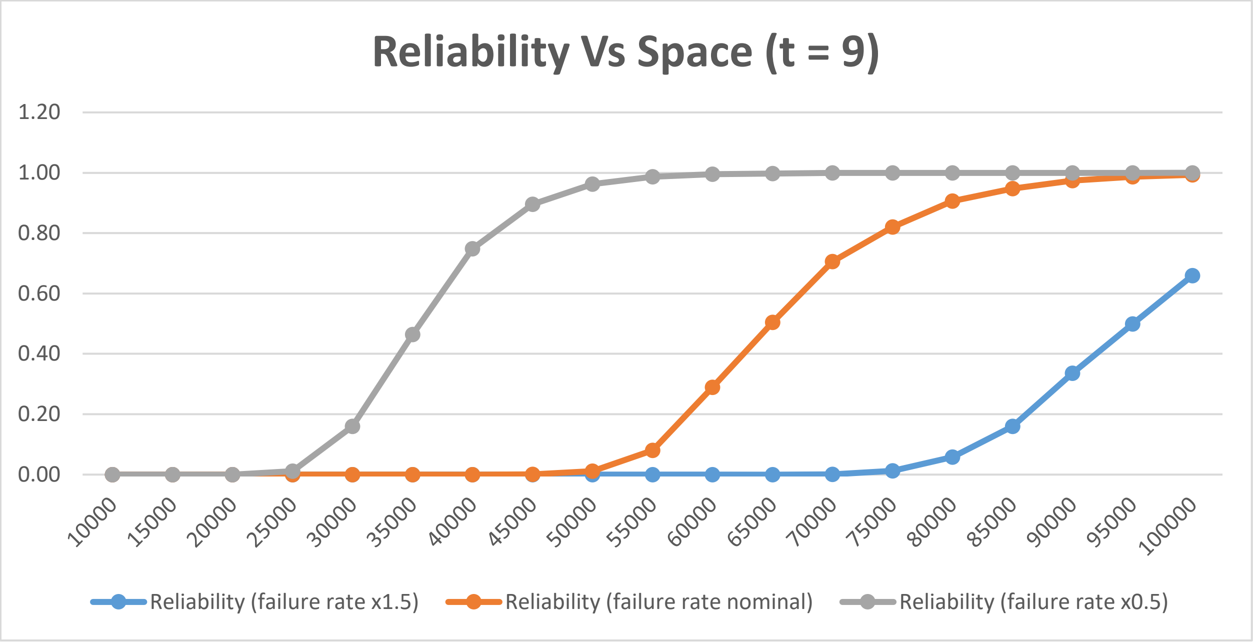


Figure 11 Failure rate of components (b)

Number of Components

In figure 12, the number of components is inversely proportional to the reliability. As the number of components to be stored increases, the reliability goes down. Consider 35000 cube feet of space, when there are 150 components, reliability is close to 1. However, as the number of components increase, say 175 components, reliability drops to 0.8.

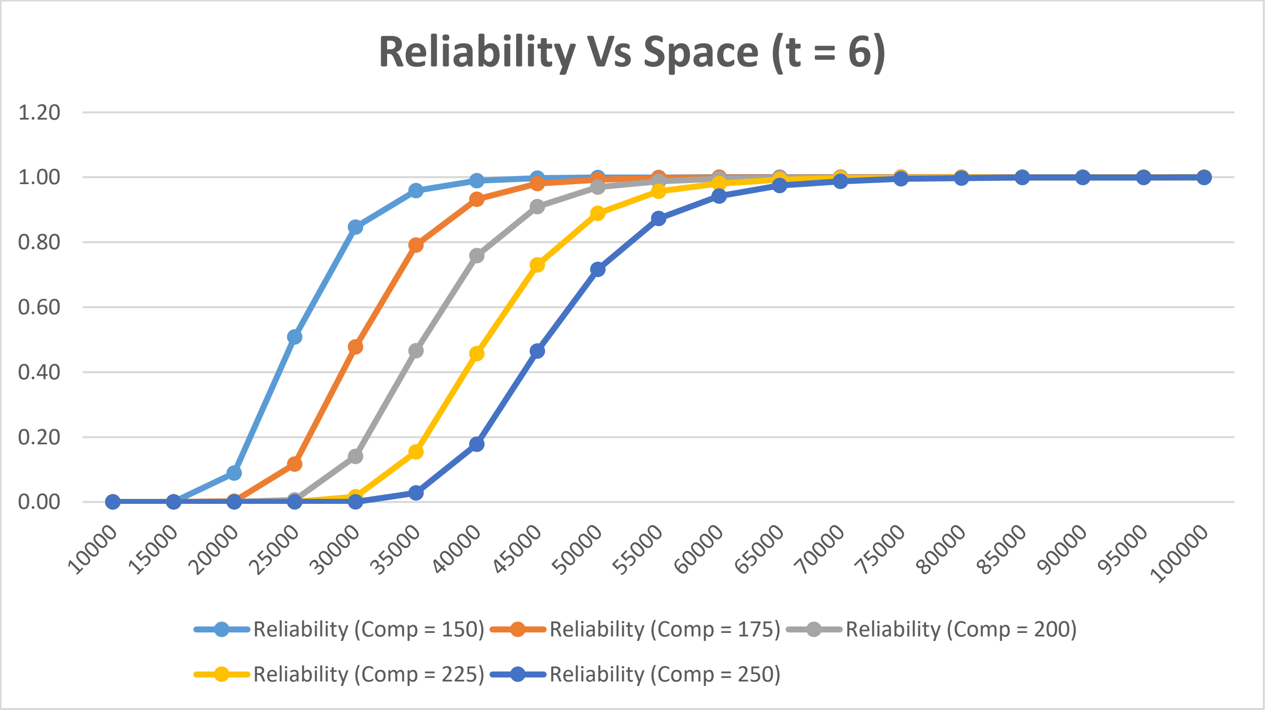


Figure 12 Number of components (a)

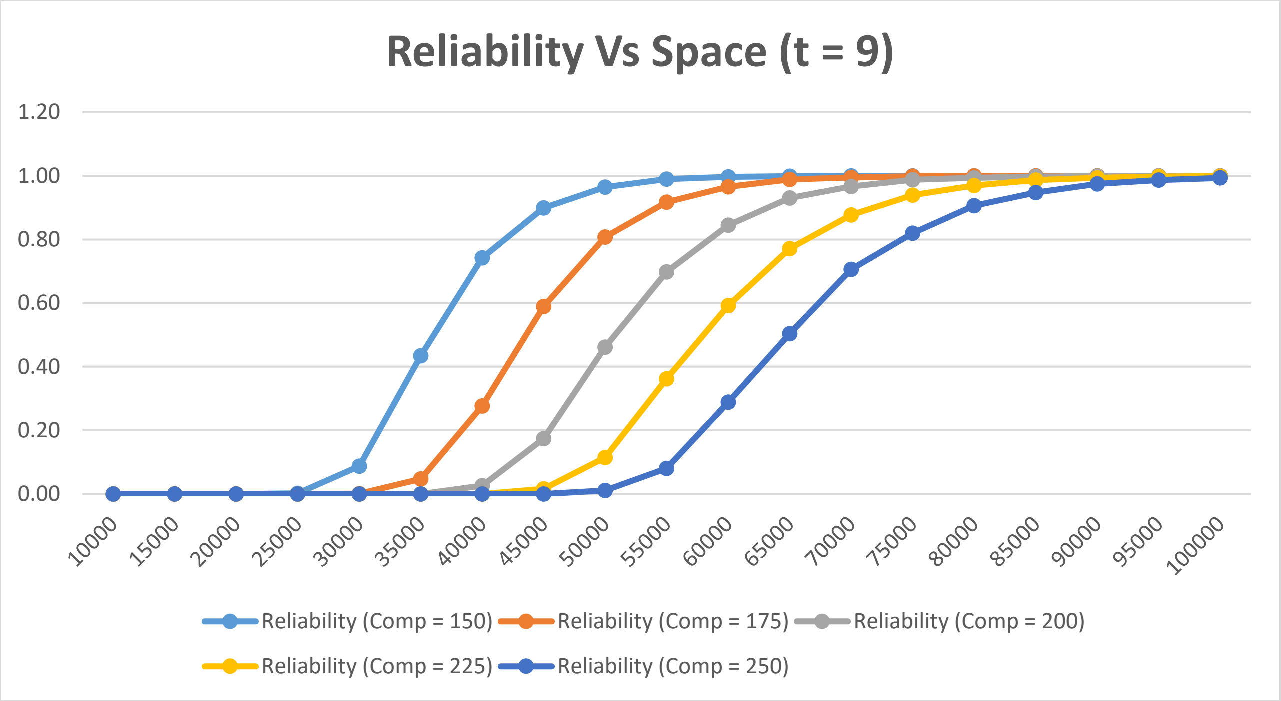


Figure 13 Number of components (b)

Percentage of components that can be cannibalised

Consider three cases of cannibalisation, nominal case where 50 % of components can be "re-used" or "shared' among the aircrafts. Secondly, where all components, 100%, can be cannibalised. Finally, 0% of components or no components can be cannibalised. Hence, as the percentage of components that can be cannibalised increased, reliability increases. However, when non of the components can be cannibalised, a greater number of spare components must be stocked on-board and thus, space required increases.

Number of mission aircrafts

Consider the total number of available aircrafts to be 10 (n = 10) and number of functional aircrafts is varied from 4 to 10. Firstly, when there are 4 out of 10 aircrafts functional, the space required to store spare components is low and in turn high reliability is achieved even with low space available. However, when all the available aircrafts are employed, a greater number of spare components are required to be stocked which in turn consumes more space. Hence, reliability is inversely proportional to the number of aircrafts used for a given space.

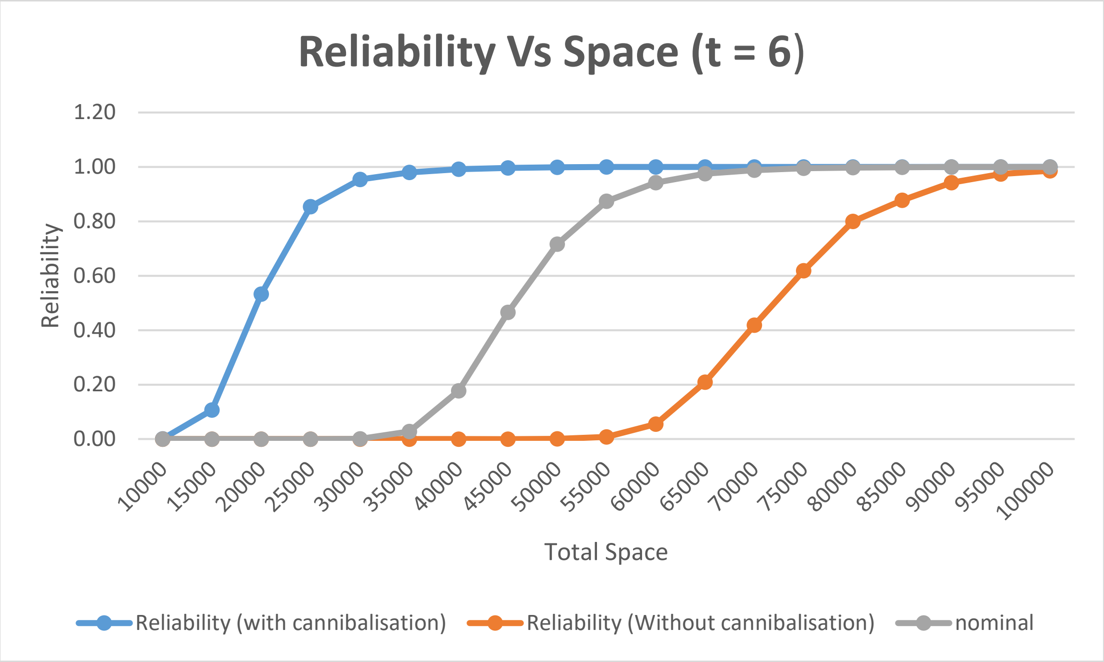


Figure 14 Percentage of components that can be cannibalised (a)

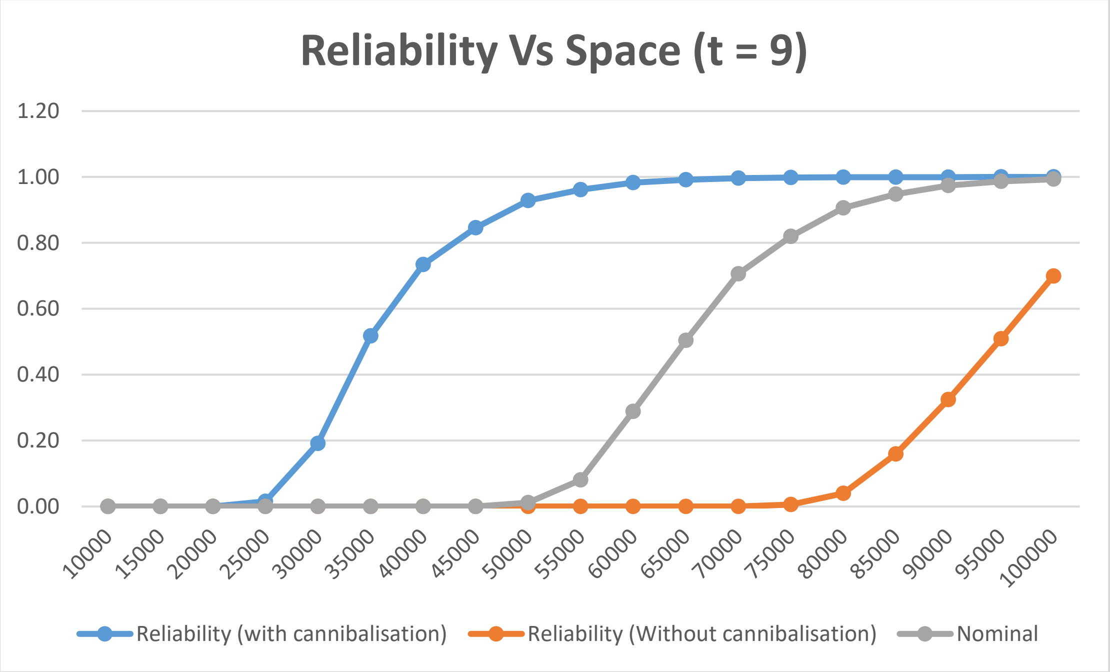


Figure 15 percentage of components that can be cannibalised (b)

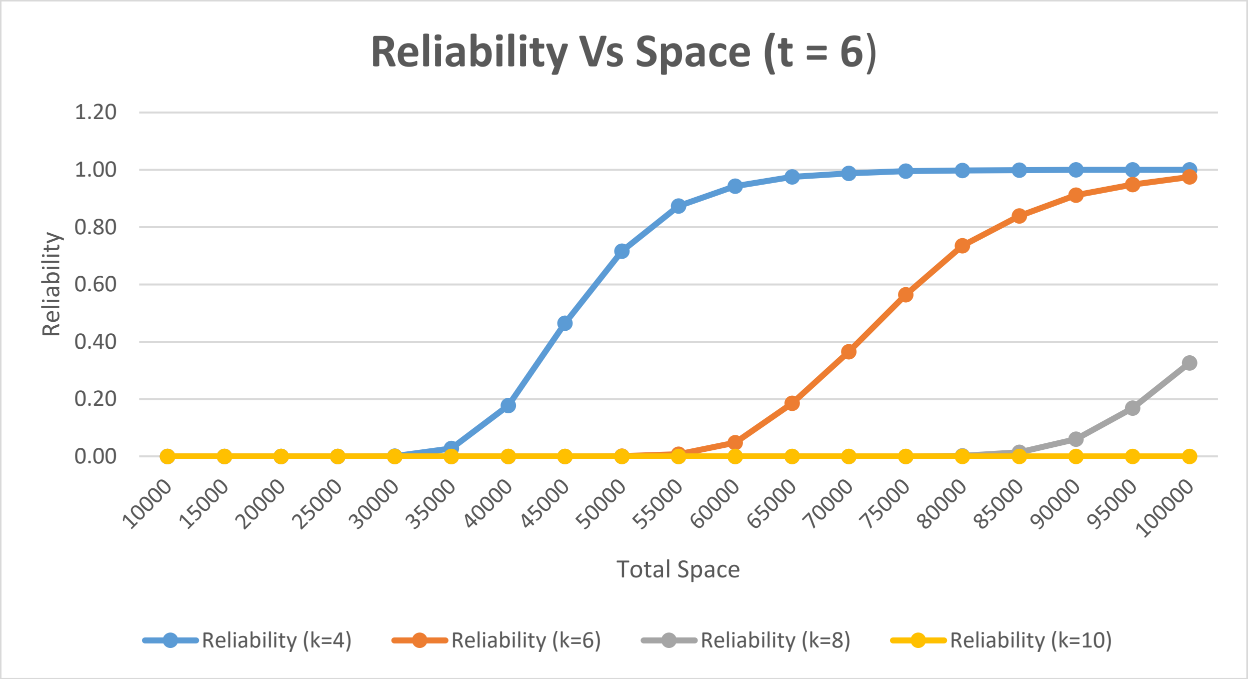


Figure 16 Number of mission aircrafts (a)

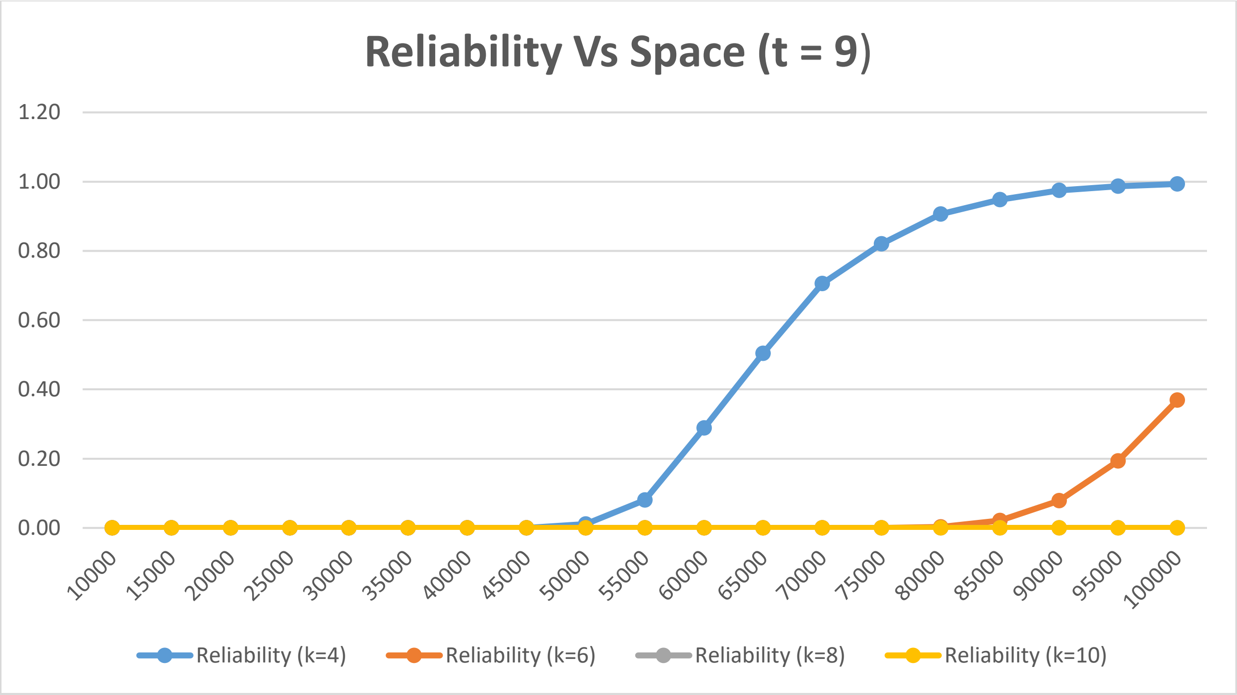


Figure 17 Number of mission aircrafts (b)

Machine learning algorithm for sensor-based maintenance

Zoha Sherkat Masoumi

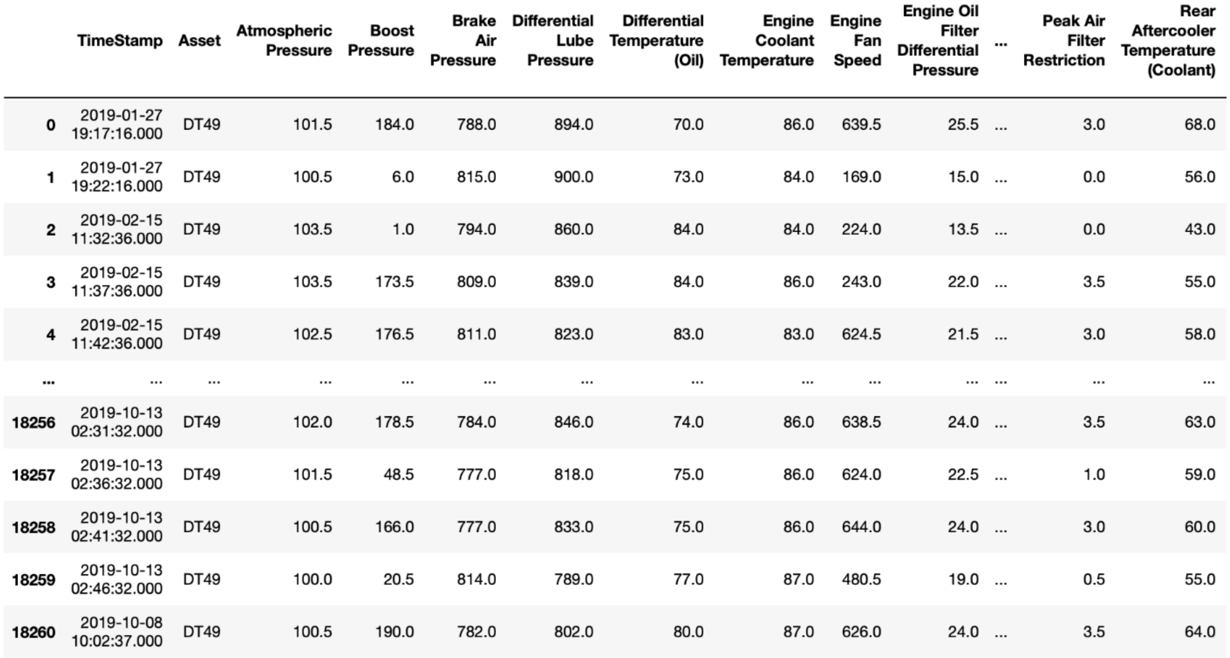
1 Introduction

Kinross Gold corporation is a Canadian gold mining company operating in Americas, Russia and West Africa. Trucks are used to carry gold and silver from mines to facilities and distribution centres. Trucks unscheduled failures interrupt mining plans which result in excessive downtime and cost. In addition, trucks are expensive assets, and unscheduled failures will reduce their life expectancy leading to deterioration. On average, there were about 262 unscheduled failures per truck in 2019 and eight main failure modes (Dispatch hardware, Electrical, Engine, Hydraulics, Lubes/Oil, Operator damage, Structure, and Tire repair). As a result, the company aims to detect failures before they occur with the help of sensors. In this project, sensor recordings from 2019 are utilized to train machine learning algorithm to predict engine failures that are expected to take place in 3 hours.

2 Data

This case study is based on the data for 13 mining trucks for the year of 2019. For each truck, the unscheduled failure information and sensor recordings are available in two different tables. Sensor recording table contains determined time (Time Stamps) during which the sensors have recorded data. There are total of 33 sensors, with 8 sensors having missing values while 25 having sufficient data which will be considered for future analysis. For every truck, there were between 18,000 to 38,000 sensor recordings for 2019. Table 1 displays sensor recording table for truck number 49 ,DT49, after formatting. Unscheduled failure table includes name of the asset, comment on its failure mode, failure start time and its duration.

Table 1: Sensor recordings table for DT49



	TimeStamp	Asset	Atmospheric Pressure	Boost Pressure	Brake Air Pressure	Differential Lube Pressure	Differential Temperature (Oil)	Engine Coolant Temperature	Engine Fan Speed	Engine Oil Filter Differential Pressure	...	Peak Air Filter Restriction	Rear Aftercooler Temperature (Coolant)
0	2019-01-27 19:17:16.000	DT49	101.5	184.0	788.0	894.0	70.0	86.0	639.5	25.5	...	3.0	68.0
1	2019-01-27 19:22:16.000	DT49	100.5	6.0	815.0	900.0	73.0	84.0	169.0	15.0	...	0.0	56.0
2	2019-02-15 11:32:36.000	DT49	103.5	1.0	794.0	860.0	84.0	84.0	224.0	13.5	...	0.0	43.0
3	2019-02-15 11:37:36.000	DT49	103.5	173.5	809.0	839.0	84.0	86.0	243.0	22.0	...	3.5	55.0
4	2019-02-15 11:42:36.000	DT49	102.5	176.5	811.0	823.0	83.0	83.0	624.5	21.5	...	3.0	58.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
18256	2019-10-13 02:31:32.000	DT49	102.0	178.5	784.0	846.0	74.0	86.0	638.5	24.0	...	3.5	63.0
18257	2019-10-13 02:36:32.000	DT49	101.5	48.5	777.0	818.0	75.0	86.0	624.0	22.5	...	1.0	59.0
18258	2019-10-13 02:41:32.000	DT49	100.5	166.0	777.0	833.0	75.0	86.0	644.0	24.0	...	3.0	60.0
18259	2019-10-13 02:46:32.000	DT49	100.0	20.5	814.0	789.0	77.0	87.0	480.5	19.0	...	0.5	55.0
18260	2019-10-08 10:02:37.000	DT49	100.5	190.0	782.0	802.0	80.0	87.0	626.0	24.0	...	3.5	64.0

3 Model Output: Remaining Useful Life (RUL)

Remaining Useful Life (RUL) is the time (hours) remaining to the next failure. For each sensor recording, RUL has been calculated from unscheduled failure table. Figure 1 displays the RUL vs. sensor recording for truck 68 and 69 as well as their failure cycles. Some cycles are longer; however, there are many short cycles indicating that the machine failed shortly after being put back into operation. Truck 79, DT79, has the longest cycle without failure (867.43 hours). Based on the discussion with Kinross Gold Corporation representative, the company is interested in Engine failure mode;thus, in this case study only the engine failures are investigated. Figure 2 displays a sample RUL cycle for DT69; as shown, RUL starts with larger value, it decreases over time, and at failure it reaches zero. In this project, instead of predicting the time remaining to the next failure, it is aimed to detect failure within a prespecified threshold (3 hours) for following three reasons:

- Company's main interest is to detect failures within 3 hours in order to have enough time available for replacement of the truck and informing maintenance person
- Sensors are designed to detect failures and are expected to display normal recordings when RUL is more than 3 hours. Thus, sensors can not predict the exact time remaining to the failure with high precision, but they can detect anomalies when approaching failure.
- Binary classifiers are easier and faster to train; their results outperform the prediction models for this case study

As a result, all the output values are labeled as 0 when there are less than 3 hours remaining to the next engine failure and 1 otherwise. Threshold and labels are displayed on figure 2. For the failure to be predicted correctly, at least two of the labels corresponding to that failure must be accurately identified.

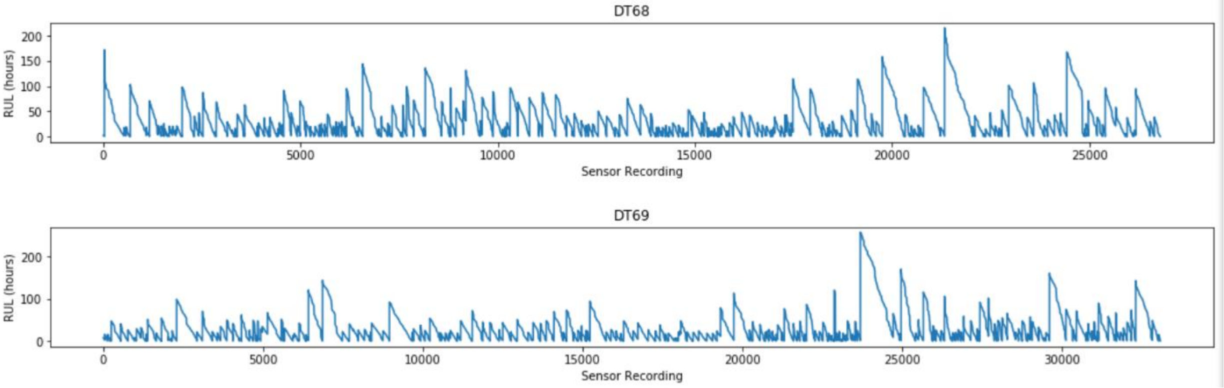


Figure 18 RUL vs. sensor recording for 'DT68' and 'DT69'

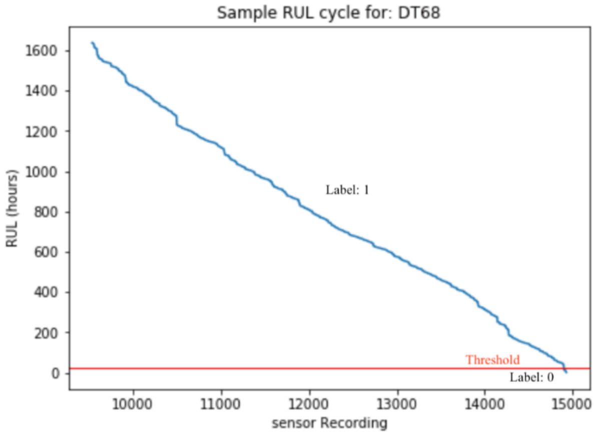


Figure 19: Sample RUL cycle for 'DT69'

4 Challenges

The following two are the main challenges of this case study. To start with, the data set is highly imbalanced. Figure 3 displays the number of unscheduled engine failures for the year of 2019: except truck 66 ,DT66, which has 46 engine failures, the remaining trucks have less than 23. In addition, on average, only 0.4 % of the recordings are failure samples (the average ratio of failure to non failure sample is 4:1000) and for some trucks this number reduces to 0.06 %. This will result in misclassifying the minority class; therefore, the output of the algorithms would be biased toward majority class. In many algorithms, minority class is identified as noise, and models are not trained on them. However, in many real world applications such as failure detection, the main objective is to correctly identify the minority class (failures) to avoid higher cost causing by this misclassification [2]. Furthermore, failure pattern is different for each truck. Some trucks have a long failure cycle and few short ones while others contain short and medium cycles only. Additionally, for many trucks, failures are not evenly distributed throughout the year. For example, figure 4 displays the RUL vs. time for four trucks. As shown, truck 70 and 73 have no failures during the first months and majority of their failures occur toward the end of the year. However, failures for truck 49 and 66 are approximately evenly distributed over 2019. Unfortunately, many machine learning algorithms, especially the time series, are sensitive to pattern in the data and perform poorly when the failure pattern on the training set and the test set differ.

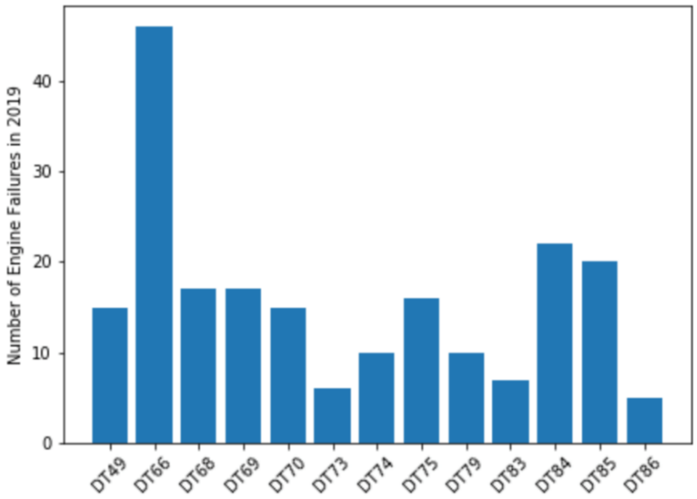


Figure 20: Number of engine failures for each truck in 2019

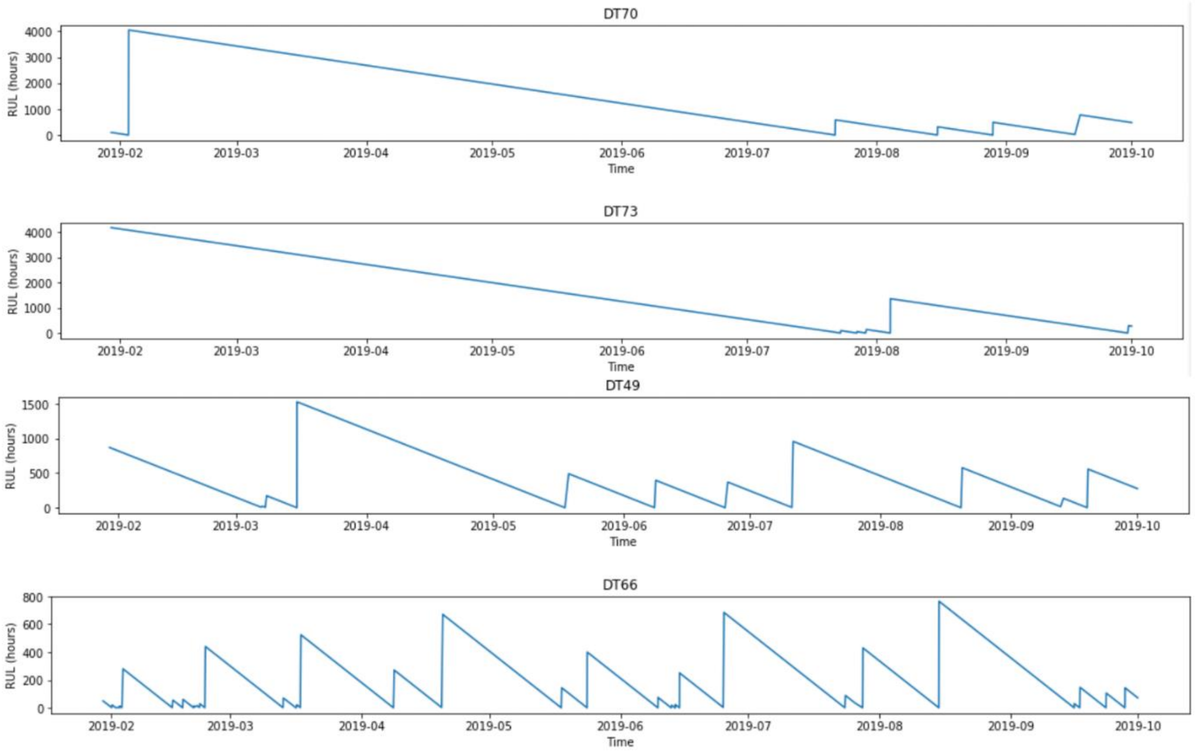


Figure 21: Sample RUL for Engine Failure

5 Approach

For this case study, I followed a three- stage approach. First stage incorporates input data transformation, in the second stage, input data is re-sampled and finally, machine learning model is applied to predict failures.

5.1 Data transformation

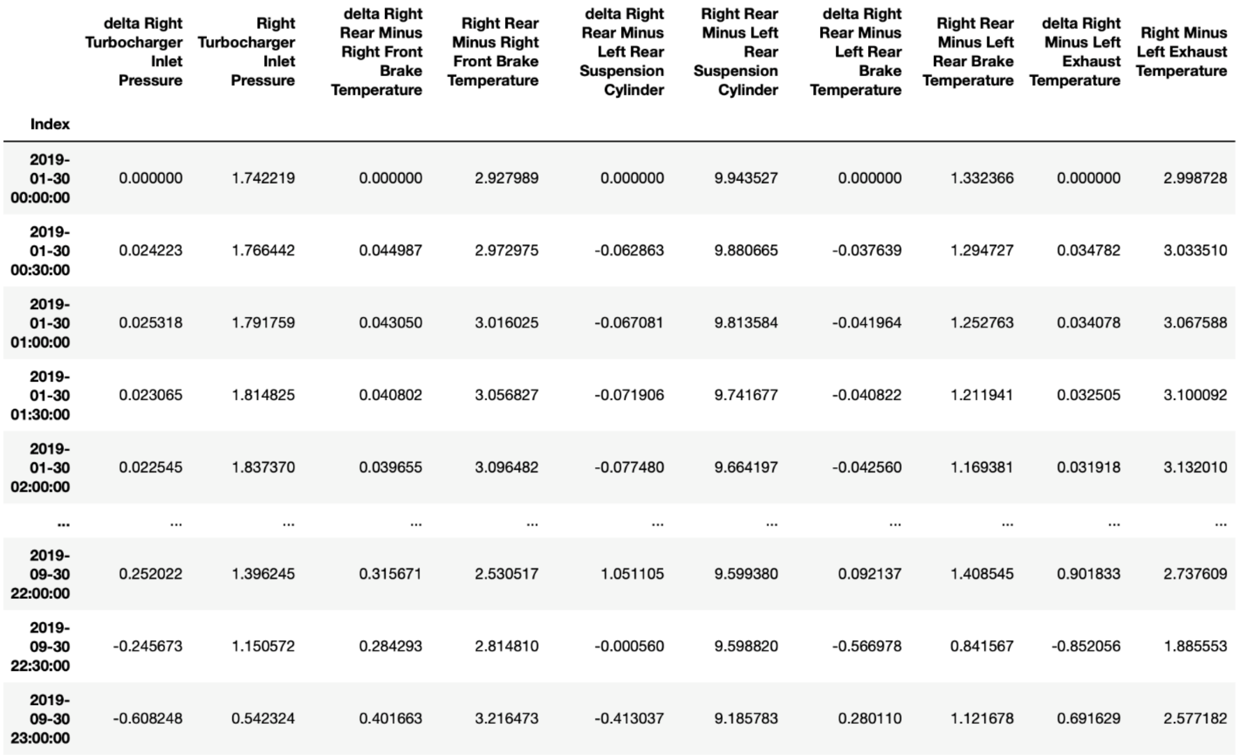
1. Outlier Detection: many machine learning models are sensitive to the range of input data and existence of outliers which leads to biased parameter estimation and cause poor results [3]. Thus, in step number 1, I removed data points that were below 0.5 percentile and above 99.5 percentile for each sensor column in the table. Overall, I dropped only one percent of the data points for each sensor because outliers were mainly on the extremes of the recordings.
2. Define Time Interval: The recordings are not available for the entire 2019 and for each truck recordings started and ended in different days or even months. Thus, to be consistent, I decided on the time intervals where data is available for all the trucks. For this study, I only considered data points between February and November of 2019.
3. Fix Time Interval: In the original table, for each truck, the time between two consecutive recordings is not equal, and this interval is varied from 5 minutes to few hours. This is misleading for machine learning algorithms since sensor recordings that are very close to each other will document similar results with low variance whereas the sensor recordings that are few hours away record very different values, affecting the training process. As a result, by interpolation, the time intervals between failures are fixed to 30 minutes.
4. Log transformation: It is easier to train machine learning algorithms on a near normal distribution than skewed data set. In the input table, sensor columns are skewed and distorted from normal distribution. The best method to reduce skewness in data set is log transformation. However, original log transformation can be applied to positive numbers only. To log transform recordings that contain negative values, a constant is added to each input data [5]. The constant number is minimum of the recording on the that column plus a small value such as 1. The log transformation formula is as follows:

   $$r_{\text{desired}} = \log(r + 1 - \min(r))$$

   where r is the sensor recording before transformation and r_{desired} is the recording after log transformation.
5. Adding Delta Columns: Differencing eliminates seasonality, and trends that is caused by changing levels in data [6]. As a result, the difference between consecutive recordings for each sensor is added as a separate feature. These new features are named "delta" plus sensor name displayed on table 2.

The first three steps in input data transformation are demonstrated in the following examples below: The first example is for "Rear After-cooler Temperature" sensor, the top graph on figure 5 displays the original recordings. Please note that each colour corresponds to one truck. As shown, majority of the data points are below 100, however, there are some very large recordings. Overall, its range is between 5 and 65533. After dropping the outliers (figure 5 middle graph), range of recordings changed from 65,528 to 60. However, it is possible to observe that for most trucks recordings are not available for the entire year and some others like 'DT70' (the truck displayed purple colour) recordings are available by the end of December. After the second (considering data points between February and November) and third step (fixing the time between recordings to 30 minutes) in data transformation, figure 5 at the bottom displays the final results. For better illustration, figure 6 displays "Peak Air Filter Restriction" sensor which follows similar procedure. In addition, step 4 in data transformation is displayed for "Brake Air Pressure" sensor. Before transformation, the range of the data is between 502 to 886 (figure 7 on the top) and after log transformation recordings are between 0 and 5.7 meanwhile, recordings for each truck is easier to distinguish (figure 7 at the bottom).

Table 2: New table with delta columns added



Index	delta Right Turbocharger Inlet Pressure	Right Turbocharger Inlet Pressure	delta Right Rear Minus Right Front Brake Temperature	Right Rear Minus Right Front Brake Temperature	delta Right Rear Minus Left Rear Suspension Cylinder	Right Rear Minus Left Rear Suspension Cylinder	delta Right Rear Minus Left Rear Brake Temperature	Right Rear Minus Left Rear Brake Temperature	delta Right Minus Left Exhaust Temperature	Right Minus Left Exhaust Temperature
2019-01-30 00:00:00	0.000000	1.742219	0.000000	2.927989	0.000000	9.943527	0.000000	1.332366	0.000000	2.998728
2019-01-30 00:30:00	0.024223	1.766442	0.044987	2.972975	-0.062863	9.880665	-0.037639	1.294727	0.034782	3.033510
2019-01-30 01:00:00	0.025318	1.791759	0.043050	3.016025	-0.067081	9.813584	-0.041964	1.252763	0.034078	3.067588
2019-01-30 01:30:00	0.023065	1.814825	0.040802	3.056827	-0.071906	9.741677	-0.040822	1.211941	0.032505	3.100092
2019-01-30 02:00:00	0.022545	1.837370	0.039655	3.096482	-0.077480	9.664197	-0.042560	1.169381	0.031918	3.132010
...	...	...	...	...	...	...	...	...	...	...
2019-09-30 22:00:00	0.252022	1.396245	0.315671	2.530517	1.051105	9.599380	0.092137	1.408545	0.901833	2.737609
2019-09-30 22:30:00	-0.245673	1.150572	0.284293	2.814810	-0.000560	9.598820	-0.566978	0.841567	-0.852056	1.885553
2019-09-30 23:00:00	-0.608248	0.542324	0.401663	3.216473	-0.413037	9.185783	0.280110	1.121678	0.691629	2.577182

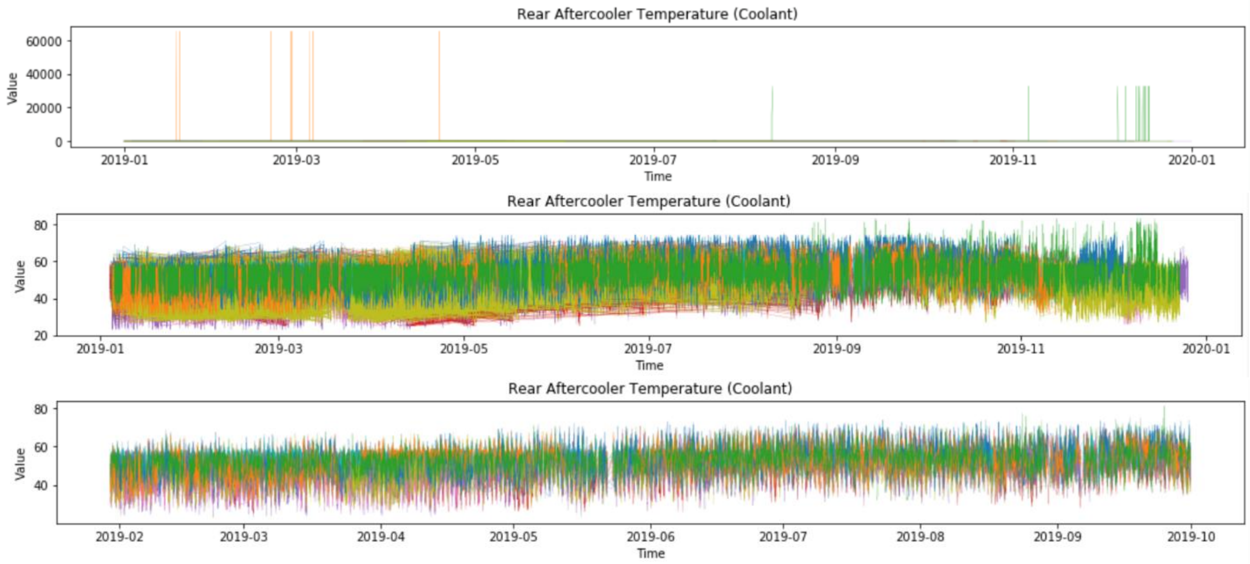


Figure 22 Rear Aftercooler Temperature Sensor: (1) Original Recordings (2) After step 1 (3) After step 2 and 3

5.2 Sampling Method

Data is highly imbalanced as noted above. It creates bias and results in majority prediction for machine learning algorithms. To avoid such issues, a combination of under-sampling and over-sampling is utilized. In random under-sampling, random samples from the majority class are removed with specified sampling strategy. For this case study, sampling strategy of 0.5 % is used which excludes sample from minority class until 0.5 % of the samples correspond to minority class. In addition, from the available over-sampling techniques, Boarder Line SMOTE outperformed others. Boarder Line SMOTE is a variation of SMOTE (Synthetic Minority Oversampling Technique). It oversamples the minority class by following below steps [7]

Step 1: a point in the minority class boarder is selected like the one labeled on figure 8
Step 2: near neighbour of the selected point another point from the same class is selected
Step 3: A line connects two points
Step 4: A new point is generated randomly and placed on the line

Above steps repeat until the ratio of samples in training set is as specified. For this case study, the ratio is 0.9 % (for each 100 majority class 90 points in minority class is oversampled). This method only over-samples the boarder points because these points could be misclassified with higher probability. At the same time in this method, noise points (noise point is labeled on figure 8 are not oversampled as by oversampling them, a bridge is created between classes and results in misclassification.

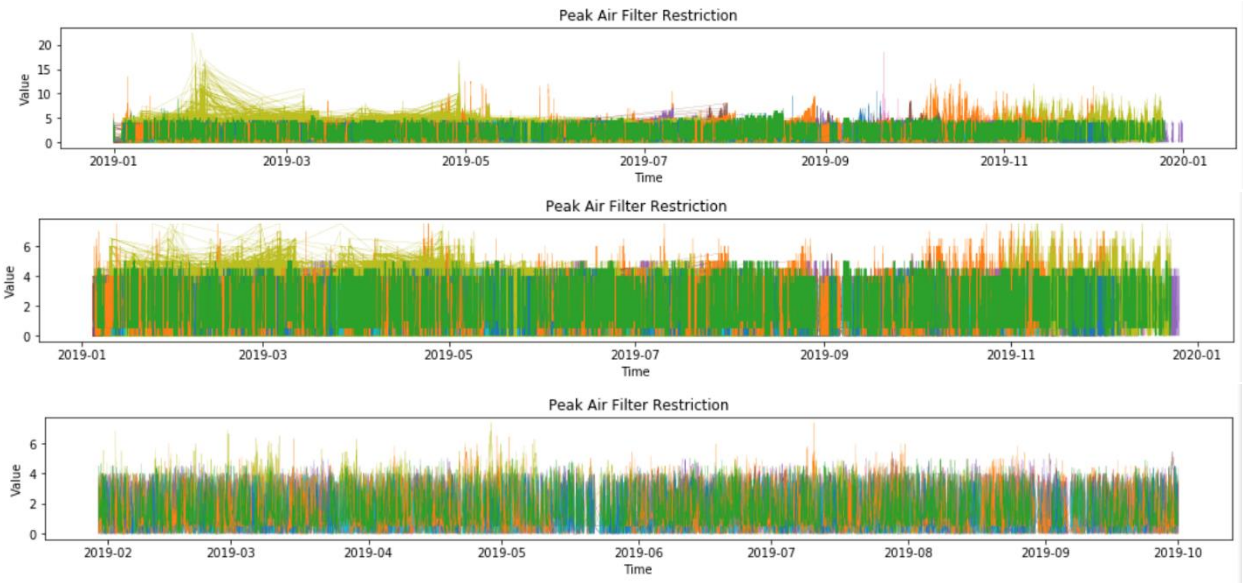


Figure 23: Peak Air Filter Restriction Sensor: (1) Original Recordings (2) After step 1 (3) After step 2 and 3

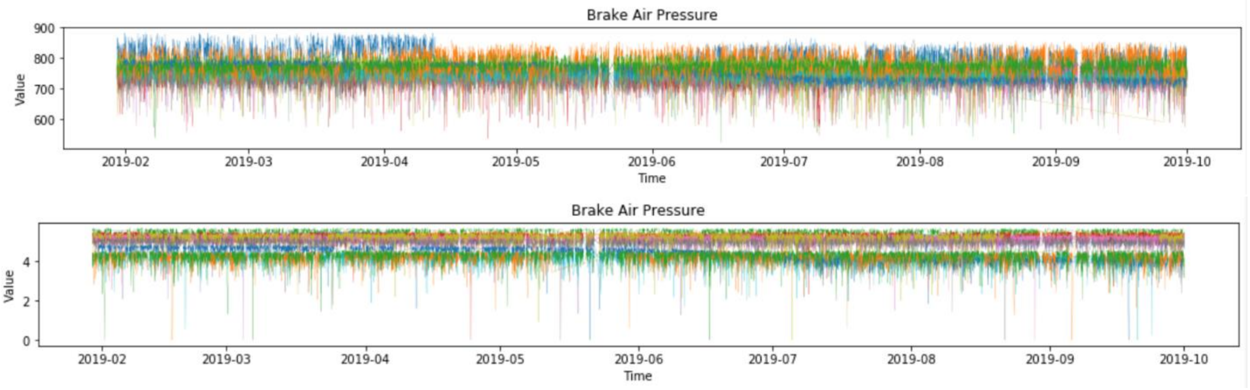


Figure 24: Brake Air Pressure Sensor: (1) Before log transformations (2) After step 1 (3) After log transformations

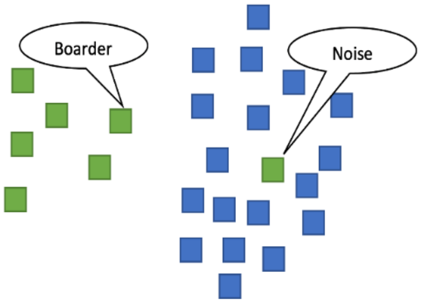


Figure 25: labeled boarder and noise points

5.3 Naïve Bayes Classifier

The highest performing for the current dataset is Naive Bayes Classifier. Naive Bayes Classifier is based on the Bayes theorem with the naive assumption that features are independent. Given a set of features x probability that belongs to class c ($P(c|x)$) equals to the provided equation:

$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)}$$

Where $P(x \mid c)$ is the probability of feature x given class c, $P(c)$ and $P(x)$ are the probability of class c and feature x in the dataset, respectively. For each class, the given probability is calculated and the class with the highest probability is assigned to the set of features. It is important to emphasize that the probabilities can be computed in parallel, which results in fast training [8].

6 Performance Metrics

Accuracy is the proportion of correctly predicted recordings to the total number of prediction and its equation is as follows [9]:

$$\text{Accuracy} = \frac{T_P + T_N}{T_P + F_P + T_N + F_N}$$

Accuracy is the most common matrix used to evaluate the performance of the model. However, it is misleading when it comes to imbalanced data. The main reason is that, in case of imbalanced data, many machine learning algorithms predict the majority class, and this will result in good accuracy matrix even though non of the minority class was predicted correctly. Thus, we will focus on precision and recall for our model. Precision indicates the proportion of failure/non-failure recordings over the failure/non-failure recordings classified by algorithm and its equation is as follows [9]:

$$\text{Precision} = \frac{T_P}{T_P + F_P}$$

the proportion of failure/non-failure recordings classified by algorithm over actual failure/non-failure recordings and its equation is as follows

$$\text{Recall} = \frac{T_P}{T_P + F_N}$$

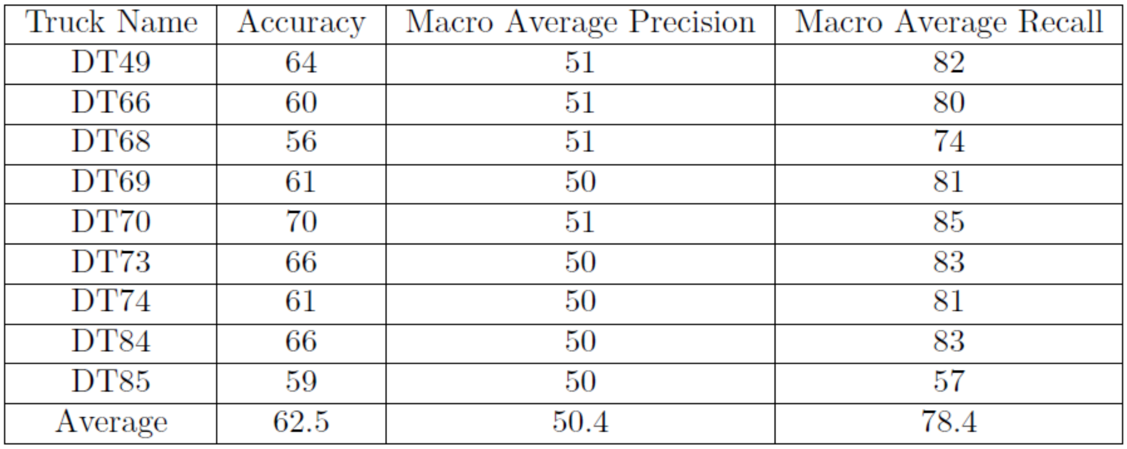
7 Results

As indicated earlier, the aim of our model is to utilize sensor recordings to predict failures that are expected to take place in 3 hours. It is important to note the following about model preparation:

Firstly, for each truck, the first 80 % of the sensor recordings are utilized for train and the remaining 20 % are used for test. In addition, each truck is also trained on the recordings for other trucks. Secondly, since the objective is to correctly identify failures, the model is evaluated only on the trucks that have failures on their test set and four trucks that do not satisfy this condition have been excluded from analysis. Lastly, sampling method and Naive Bays algorithm are applied automatically through pipeline, which results in higher performance.

After all, table 3 displays the algorithm results for each truck. The overall accuracy is 62.5 % and the macro average recall is 78.4%. As shown, the recall value for all the trucks except two of them is above 80% this indicates that the algorithm can detect majority of failures. However, the macro average precision is 50.4%. This results in false alarms and requires further investigation.

Table 3: Performance results of the machine learning model for different trucks



Truck Name	Accuracy	Macro Average Precision	Macro Average Recall
DT49	64	51	82
DT66	60	51	80
DT68	56	51	74
DT69	61	50	81
DT70	70	51	85
DT73	66	50	83
DT74	61	50	81
DT84	66	50	83
DT85	59	50	57
Average	62.5	50.4	78.4

8 Business analysis

In this section an overview of the cost and benefits of the proposed algorithm will be provided. Based on the information received, correctly detecting failures will cost 1 unit. In addition, the cost of false alarm is 0.05-0.2 unit while unscheduled failure will result in 1.3-1.5 cost unit. The algorithm would result in higher economic value when the cost of false alarm is lower, and the cost of unscheduled failure is higher. Therefore, let's assume that the cost of false alarm is 0.05 cost unit (its minimum value) and the cost of unscheduled failure is 1.5 cost unit (its maximum value). As noted earlier, the performance of the model is evaluated on the trucks that failed during test time (about 49 days). Overall, there are about 18 failures on 9 trucks; thus, average number of unscheduled engine failures per truck is 2. As a result, on average, there are four possible cases:

- Algorithm correctly detects both failures resulting in 2 unit cost + 0.05 unit cost for each false alarm
- Algorithm only detects one of the failures resulting in 2.5 unit cost (1 unit cost for detected failure and 1.5 unit cost for unscheduled failure) + 0.05 unit cost for each false alarm

- Algorithm is unable to detect any failures resulting in 3 unit cost (1.5 unit cost for each unscheduled failure) + 0.05 unit cost for each false alarm
- When no machine learning model is applied, on average for each truck, 2 unscheduled failures occur and leads to 3 unit costs and no false alarm cost.

Figure 9 displays cost vs. number of false alarms and each of the above cases are shown with a different colour. According to figure 9 the algorithm that can detect both failures and has no false alarm costs 1 unit less (3-2) and for each false alarm 0.05 unit cost is deducted from cost saved. When there are 20 false alarms cost of the algorithm is 3 unit and equals to no machine learning case, which is called the equilibrium point. Thus, any algorithm that can detect both failures but has more than 20 false alarms costs more than when no machine learning is used. In addition, if the algorithm is able to detect half of the failures and has no false alarms costs 0.5 unit less than no machine learning case (3-2.5). When there are more than 10 false alarms, its cost exceeds when no machine learning is applied. It is because the cost of 10 false alarm equals the savings from one failure prediction. For deeper understanding, these 20 false alarms account for about 70% macro average precision. Thus, any machine learning model requires the macro average precision of more than 70% to be economically worthwhile in the best case.

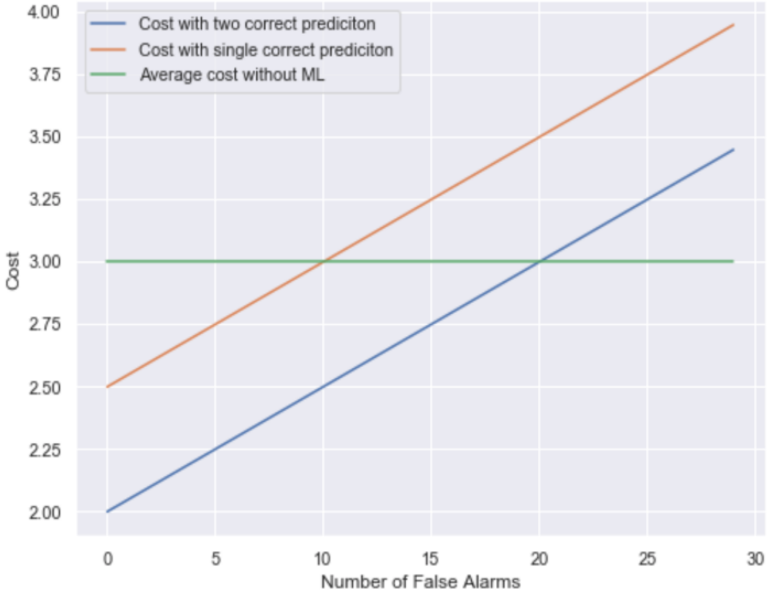


Figure 26: Cost vs. Number of False Alarms

9 Conclusions

In conclusion, the aim of this project was to detect engine failures within a prespecified threshold of three hours. This report presented a three-stage approach for a sensor based maintenance. In the first stage, different steps have been followed to transform the raw data. In the second stage, a combination of undersampling and oversampling has been utilized to balance dataset and in the last stage, Naive Bays algorithm has been applied to detect failure recordings. The macro average recall was 78.4% which indicates that the algorithm was able to detect majority of failures. However, the macro average precision is 50.4% which shows that the model has false alarms. These false alarms mainly resulted from the sampling methods used. However, sampling methods are necessary because they reduce bias of the model toward majority class. To avoid this, in the future, data from more trucks or few years data of the same truck could be used to have more failure samples and reduce the dependency of the model to resampling methods. Furthermore, another suggestion is to work toward reducing the cost of false alarms. Currently, the savings from each failure detection equals to the cost of 10 false alarms. Thus, for a machine learning model to be economically worthwhile, it requires to have a very high precision and recall which might not always be easily achievable. Moreover, one good practice about sensor-based maintenance is to fix the time intervals in which sensors record information. In the original data, the time between sensor recordings varied from 5 minutes to few hours, which was affecting the machine learning model. Therefore, to be consistent, I used interpolation to fix the time between sensor recordings, however, decline in the quality of input data was inevitable. In addition, an expert knowledge could be very helpful to find the sensors that are proven to have impact on engine failures and only include those that are related to engine failure. For example, one of the sensors that could help the machine learning algorithm is the age of the engine. That is mainly because it is proven that many engine failures are related to engine deterioration and age.

References

[1] Kinross Gold. About us. https://www.kinross.com/about/default.aspx.

[2] Shaza M. Abd Elrahman and Ajith Abraham. A review of class imbalance problem. *Journal of Network and Innovative Computing*, 1:332-340, 2013.

[3] Irad Ben-Gal. *Outlier Detection*, pages 131-146. Springer US, Boston, MA, 2005.

[4] Mathieu Lepot, Jean-Baptiste Aubin, and Francois H.L.R. Clemens. Interpolation in time series: An introductive overview of existing methods, their performance criteria and uncertainty assessment. *Water*, 9(10), 2017.

[5] Jason Osborne. Notes on the use of data transformations. *Practical Assessment, Research, and Evaluation*, 8, 2002.

[6] Ameeth Kanawaday and Aditya Sane. Machine learning for predictive maintenance of industrial machines using iot sensor data. In *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pages 87-90, 2017.

[7] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: A new over-sampling method in imbalanced data sets learning. In De-Shuang Huang, Xiao-Ping Zhang, and Guang-Bin Huang, editors, *Advances in Intelligent Computing*, pages 878-887, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[8] Daniel Berrar. Bayes' theorem and naive bayes classifier. In *Encyclopedia of Bioinformatics and Computational Biology*, volume 1, pages 403-412, 2018.

[9] Jonas Paul Winkler, Jannis Gronberg, and Andreas Vogelsang. Optimizing for recall in automatic requirements classification: An empirical study. *In 2019 IEEE 27th International Requirements Engineering Conference (RE)*, pages 40-50, 2019.

Data preparation motivation and process for analytics

Janet Lam

Introduction

For the past several progress meetings, we've been presenting updated results for TTC's track maintenance division. As the models have already been developed, at each point, the major task is to bring the data up to the model's requirements. This presented an opportunity for us to discuss with our members and guests what happens behind the scenes when we start a data-related project. The objective of this report is to communicate the importance of the relationship and collaboration between C-MORE and the industry partner in getting the project underway.

In this report, we'll discuss how we get the data from its original form, to analysis-readiness. It will feature snapshots of the data from our TTC linetest and reinspection projects, along with the rational for data requirements. TTC's data is comparatively a high-quality dataset. The columns are mostly complete, and there are clear definitions for most features. Most importantly, TTC provided a primary point person who was available to answer all of the questions we had about the dataset, allowing us to make definitive decisions about the data.

Equipment histories and probability distributions

Regardless of whether we are taking a traditional or machine learning approach, equipment histories make up the centre of our information. We need to determine the nature of when our equipment fails in order to make predictions about it. Thus, defining an equipment history is the first step.

The beginning and the end

In order to clearly define a history, we need to determine the beginning and the end of each life. A simple way to define these points is when a component is replaced; it serves as the end of the previous life and the beginning of the next. We rely on collecting at least a few data points of the "same type" of histories in order to make statistical inferences. Thus, we make some simplifying assumptions that components from different (but similar) equipment, and sequential components on a single equipment behave the same.

The nature of the replacement also matters; we may have a replacement due to failure, or due to non-failure. As we all know, most of our replacements occur at a fixed point in time, or some other reason that isn't a failure. It's important to note these differences, because our overall objective is to determine the life distribution of our component of interest. If a component is removed before it fails, we know that the life of that component is at least as long as its current age, but could have lasted longer.

Probability distribution

Ultimately, we take a series of realized lifetimes, and determine the distribution of them. A very simple interpretation is to see how often a certain lifetime occurs, and use that to answer "what lifetime is likely to occur?", or "how likely is a component of this age to fail?". Figure 27 illustrates how a collection of lifetimes can be transformed into a probability distribution.

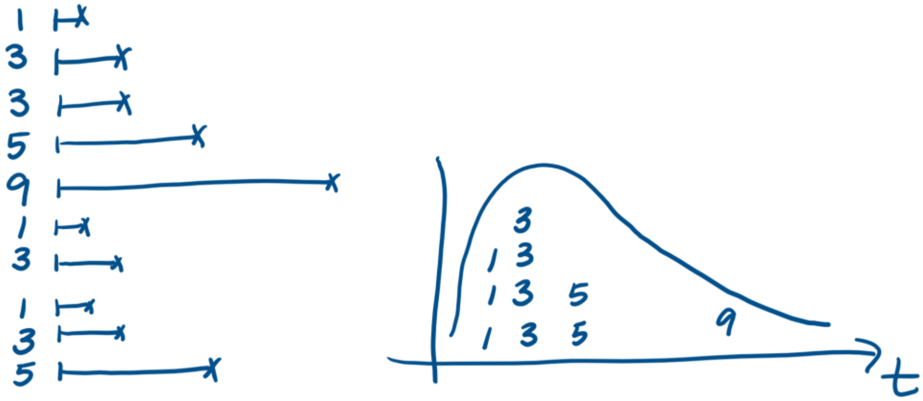


Figure 27 Lifetimes to distribution

Potential challenges

As we know from practice, real data is never neat and tidy like the illustration and examples we see in presentations. Some of the issues we've come across in the past is that the line between failure and non-failure are not as clear as we'd like it to be. Not only that, the idea of replacing a component can be fuzzy too.

Most systems are quite complex, and even components are made up of many sub-components. How much needs to be replaced in order to definitively say that the component has been replaced? When a maintenance technician does routine minor maintenance orders a proactive replacement because a component is wearing, is it a failure or not?

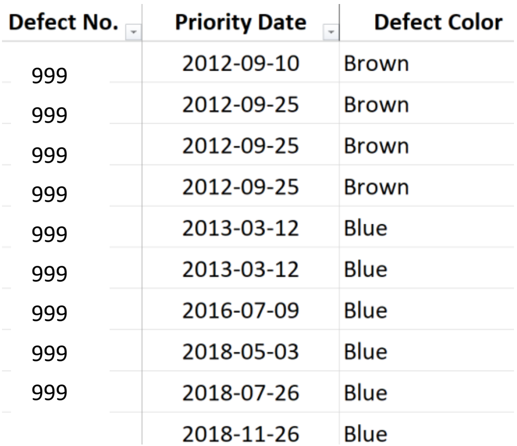
This is where the collaboration between C-MORE and the industry partner becomes critical. You are the experts in what happens on-site, and we rely on you to help us define what is a beginning, an end, a failure, or not.

Beginning and ends in TTC's reinspection project

In TTC's re-inspection project, the objective was to determine the probability that a defect would transition into a higher-priority state between inspections. This resulted in definitions that are somewhat unusual, but still applicable. First, a component in question was no longer a physical asset, but the concept of a defect. We wanted to generate a probability distribution for defects, and the end of a defect's life was defined as when it changed priorities. Figure 28 illustrates a defect that changes priority from "Brown" to "Blue" in March of 2013. In this case, we would determine the length of time the defect was in the Brown state, and consider that a history.

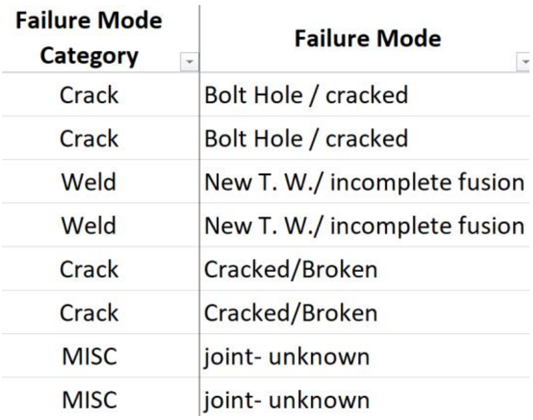
The nature of a failure

As mentioned in the previous section, determining whether a failure occurred or not is not as clear as one would think. In some cases, the end of a history can be categorized into different failure modes. In particular, if there is reason to suspect that certain types of failure behave very different (statistically) than others, there is value of distinguishing these. Another reason for looking at different failure modes is because the industry partner simply has an interest in analysing the different modes. This is another place in which partner collaboration is critical. Most CMM systems track very detailed failure modes, and it's necessary to group them into fewer large categories. As the expert in the physical asset itself, C-MORE relies on the partner to categorize the modes into larger classes. Figure 29 demonstrates the work that TTC did to classify specific failure modes into three broad categories of interest.



Defect No.	Priority Date	Defect Color
999	2012-09-10	Brown
999	2012-09-25	Brown
999	2012-09-25	Brown
999	2012-09-25	Brown
999	2013-03-12	Blue
999	2013-03-12	Blue
999	2016-07-09	Blue
999	2018-05-03	Blue
999	2018-07-26	Blue
	2018-11-26	Blue

Figure 28 Defect changing priority



Failure Mode Category	Failure Mode
Crack	Bolt Hole / cracked
Crack	Bolt Hole / cracked
Weld	New T. W./ incomplete fusion
Weld	New T. W./ incomplete fusion
Crack	Cracked/Broken
Crack	Cracked/Broken
MISC	joint- unknown
MISC	joint- unknown

Figure 29 Failure mode classification

Data anomalies and errors

Every dataset is prone to anomalies and errors. Cleaning up these errors can be quite time consuming; some of it can be automated, but it always required close supervision and interaction from a real human being. Here, we discuss some of the common issues that we've come across.

Dates

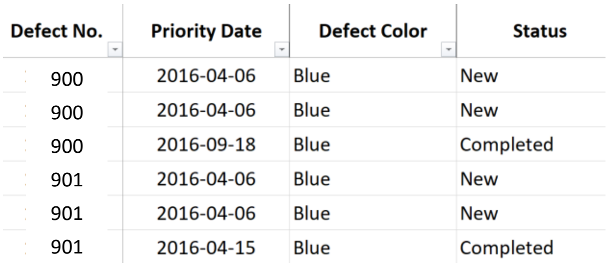
Every dataset we receive needs to be verified for date formats. There's often a mismatch between the partner system and our systems. It's a routine correction that needs to be made, most times.

Duplicate entries

As we peruse the dataset, we often come across duplicate entries. Usually, we are cleared to remove these, but we need to confirm with the partner before we remove them, as well as how to identify a duplicate entry versus an actual repeated event. Figure 30 demonstrates two defects that have duplicate entries in the dataset. These duplicates were eventually removed, but in some cases, it's not as clear that the entries are duplicates. For example, two entries on the same date may have different statuses. Then we'd have to make a decision on which of the two entries to keep, or to keep both.

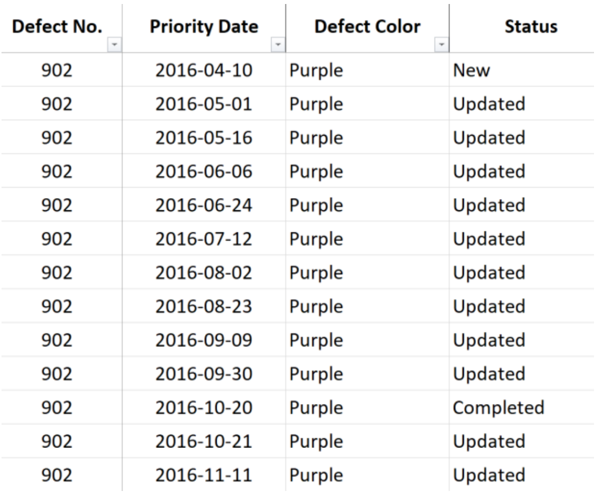
Entries that continue after ending

Sometimes, simply due to the computer system, we it's possible the entries come out of order. Then, we may have scenarios where a history is determined to have ended, but then we get another entry that indicates a continuation. This is another scenario in which expert knowledge from the partner is critical. Figure 31 is an example of a defect that seems to have been resolved in October of 2016, but then continues to persist into November.



Defect No.	Priority Date	Defect Color	Status
900	2016-04-06	Blue	New
900	2016-04-06	Blue	New
900	2016-09-18	Blue	Completed
901	2016-04-06	Blue	New
901	2016-04-06	Blue	New
901	2016-04-15	Blue	Completed

Figure 30 Duplicate entries



Defect No.	Priority Date	Defect Color	Status
902	2016-04-10	Purple	New
902	2016-05-01	Purple	Updated
902	2016-05-16	Purple	Updated
902	2016-06-06	Purple	Updated
902	2016-06-24	Purple	Updated
902	2016-07-12	Purple	Updated
902	2016-08-02	Purple	Updated
902	2016-08-23	Purple	Updated
902	2016-09-09	Purple	Updated
902	2016-09-30	Purple	Updated
902	2016-10-20	Purple	Completed
902	2016-10-21	Purple	Updated
902	2016-11-11	Purple	Updated

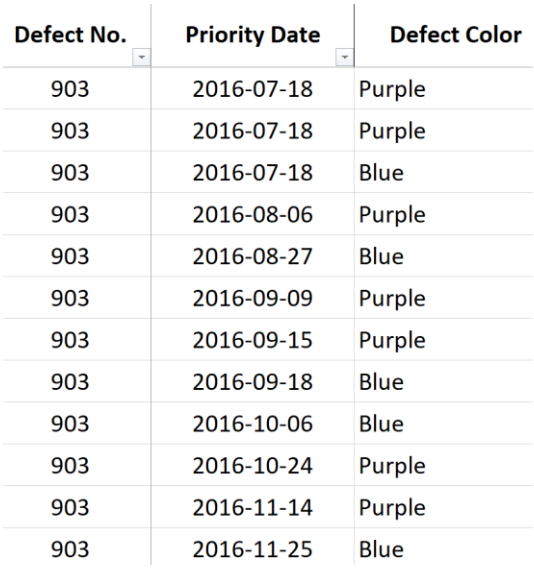
Figure 31 A completed defect continues

Inscrutable data

Sometimes, we have data that we simply cannot interpret. In these cases, we need to work with the partner to decide our way forward. Figure 32 shows a defect that changes back and for multiple times. Leaving the data as it is would result in a large bias to the left of the true lengths of defects. However, it's not clear whether the whole history should be purple, or blue, or experience a switch somewhere in the middle. In other similar scenarios, we address the simpler data concerns discussed above, and when all that remains are issues that require more significant support from the partner, we work on these together.

Conclusions

This report discussed the rationale and objectives for data in preparation for analysis. By discussing a case-study with the TTC, we saw how even high-quality data requires significant pre-processing, and how C-MORE must work closely with the industry partner in order to get our projects underway.



Defect No.	Priority Date	Defect Color
903	2016-07-18	Purple
903	2016-07-18	Purple
903	2016-07-18	Blue
903	2016-08-06	Purple
903	2016-08-27	Blue
903	2016-09-09	Purple
903	2016-09-15	Purple
903	2016-09-18	Blue
903	2016-10-06	Blue
903	2016-10-24	Purple
903	2016-11-14	Purple
903	2016-11-25	Blue

Figure 32 Flip-flopping information

Fault Detection for Wind Turbine Bearings Under Varying Rotational Speed - Progress Report

Mohamed Hassan, Hazel Shi, Eileen Mendoza, Dhavalkumar Patel

1.0 Project Overview

Bearings are one of the essential components in wind turbines. Accordingly, keeping them in good condition as long as possible and replacing them as early as possible at failure can cut down massive operation and maintenance costs. This is mainly due to reducing downtime and unplanned maintenance cost. Additionally, delayed replacement of defective bearing can lead to severe implications on a more expensive component such as the gearbox and shaft. Besides, it can significantly decrease wind turbine production.

At wind turbine, bearings are under continuous varying load, both in terms of direction and magnitude. This not only accelerates their wear out, complicates their root cause of failure and failure modes but also can lead to complicating fault detection and prognostics.

The aim of this project, it to develop a data-driven algorithm that can handle such tasks in an automated manner while fusing both state-of-the-art deep learning, signal processing, and domain of expertise.

2.0 Methodology

Condition monitoring of bearings can be achieved by utilizing vibration signals collected from the gearbox of a wind turbine. The signals are then processed to extract features from both the time and frequency domain, which can then be used in a machine learning (ML) algorithm to identify the health state of the bearings. Using large datasets to train the ML model, it is possible to develop an algorithm that will allow for premature fault detection under varying operation conditions, practice preventative maintenance, and reduce downtime for the wind turbine.

2.1 Toy Data Generation

Due to labeled data shortage and imbalance, it is necessary to generate toy data for the machine learning model. Such toy data can be extremely useful when combined with transfer learning alforthiums. The toy data generation could be separated as two stages, the signal generation and data storage.

The probability of the healthy state will be set and by sampling from the Bernoulli distribution, healthy samples labelled 0 and faulty samples labelled 1 will be generated with the corresponding probabilities. An if statement will be applied that for healthy samples, the signal will be generated with a formula added by some random noises. While for the faulty signal generation will be more complex. Some amount of the intervals within the total time will be created, and a gradually decaying fault amplitude will cumulate for each interval. Then this factor will be added to the normal formula with some noises.

To ensure the signals generated clear and structured, the data will be saved into the sqlite database. Two tables will be created within the database. The first table will be the raw data table which saves the unique ID and the corresponding signals. The second table which is a summary table that saves all the parameters used for generating the signal. The two tables share a same index so it will be easy to query for the related signal and its parameters.

2.2 Signal Processing

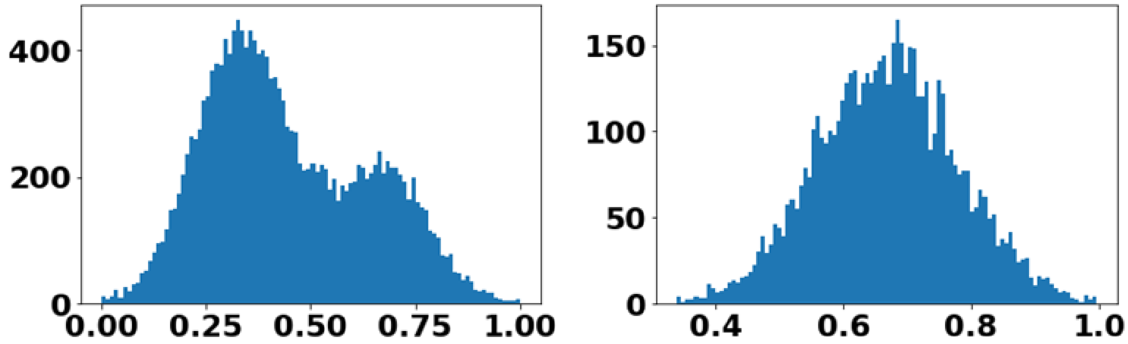
A common technique used in signal processing is the Fourier Transform, or FT. FT decomposes a signal into its component frequencies by testing for the presence of each frequency. This is done by multiplying the raw waveform by sine waves of discrete frequencies to determine if they match. A fast Fourier transform, or FFT uses a more efficient algorithm that takes advantage of the symmetry in sine waves. In python, the numpy library already contains modules to perform Fourier Analysis using FFT. Other statistical features from the signal can be extracted and used as labels for the health state. Matlab's signal processing toolbox provides functions that can be used to analyze and extract features from signals. To take advantage of this, the Matlab Engine API for python is used to perform other signal processing like producing a kurtogram to characterize the "peakedness" of a signal and using a bandpass filter to filter signal components within a range of frequencies, preventing unwanted frequencies. The filtered signal can then be used in envelope analysis or amplitude demodulation, which is another widely used technique in characterizing bearing faults.

Currently, the signal processing module is slow and inefficient and one focus for this project is to optimize this module of the code. To do this, bottlenecks in the code need to be verified using existing python profilers like the cProfile module. This will help identify which lines of the code takes the longest and needs to be optimized. Future work could also include finding alternatives for MATLAB functions called within Python to reduce run times further. This can be done by exploring available methods within Python's SciPy package.

2.3 Machine Learning

The goal of the project is to perform fault classification based on the signals gathered from the wind turbine. The operating parameters vary depending on the environmental conditions – the most important being the wind speed as it alters the operating RPM. Let's assume that the RPM is constant. In this case, the complexity of the fault classification problem reduces as clusters can easily be identified depending on other parameters. But as this is not the case for this problem statement, a model architecture is developed that focuses on eliminating the dependence of the classification problem on the RPM range.

For the latest implementation, a toy dataset was considered which could closely emulate one set of the original datasets expected from the industry partner. Two separate domains were considered for the distribution of the RPM with a considerable overlap. The ideal case would be to have a uniform distribution of the RPM for a provided range. However, those values would lead to poor performance of the model in terms of accuracy as well as computation time. Thus, a dataset was considered which is neither too simple nor too complicated for this initial experiments. The below plot demonstrates the normalized distribution where the values for domain 1 range from 300 to 1500 RPM with an increment of 200 whereas the values for domain 2 vary from 900 to 2100 RPM with an increment of 200 units.



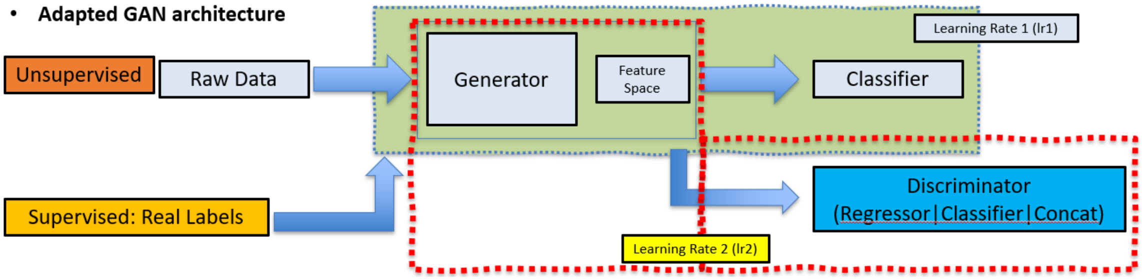
The x-axis in the above images represents the normalized RPM values and the y-axis represents the count of each value in the dataset. An overlap can be observed between these two domains as mentioned above. The left image is the distribution in the training dataset and the right image is that of the testing dataset.

Three different model architectures were considered for the classification problem. The first one corresponds to the conventional classification approach which comprises of 1D convolutional layers. However, based on the general understanding, this model will develop unique feature space for each RPM value and this might lead to higher variance. In order to eliminate this, another architecture was explored which is inspired from the Generative Adverserial Networks. The sole idea is to generate a feature space during the training phase which eliminates the dependence of the classification problem on the RPMs.

The architecture comprises of two networks 'Generator + Classifier' and 'Generator + Discriminator' respectively; both these networks compete against each other. The optimizer reduces the error of the classification results from the classifier whereas the loss is amplified for the results from the Discriminator. The discriminator focuses on identifying the domain that each example belongs to. Two variations can be developed from this network by varying the type of the discriminator. The classifier type adheres to the idea of identifying the domain label for the RPM in each example. On the other hand, the regressor type tries to predicts the actual RPM value. We are focused on a poor performance for this network; ie., either the loss increases or it remains constant which means that the network will try to predict a value as far as possible from the original value. This will lead to the generation of a feature space where the existing knowledge of the RPM has been lost because of the adversarial training but still the classifier performs well as the optimizer is focused on reducing the loss.

In addition to these three architectures, two learning methods were employed: Supervised Learning and Unsupervised Learning. Most of the data that is expected from the industry partner might not have labels. Hence, the performance of the model for the unsupervised case is also critical. The below image demonstrates the complete architecture.

- **Adapted GAN architecture**



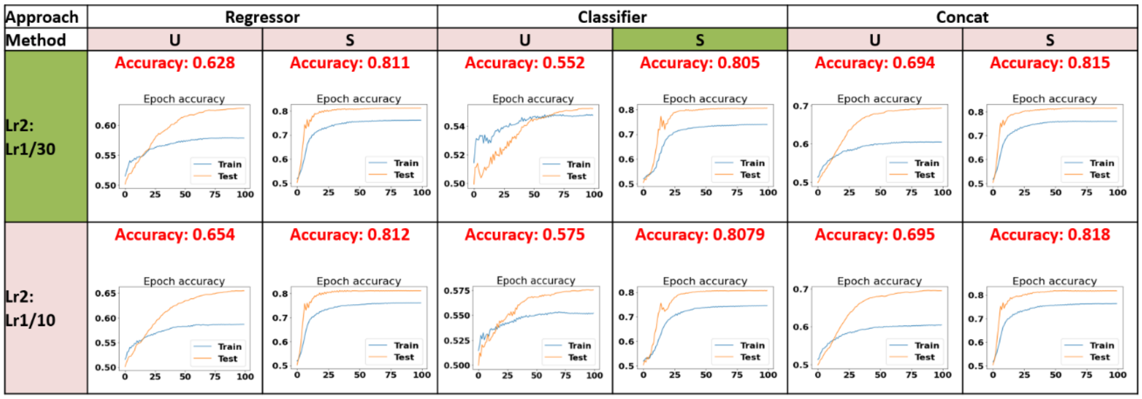
From the above image it can be seen that two different types of learning approaches were considered. For the unsupervised case the preprocessed data was fed as an input to the model, but real labels were also added for the supervised case. The three types of discriminators refer to the three different architectures: 'Regressor', 'Classifier' and 'Concat' . The Concat model represents the conventional classification model where the discriminator network is frozen and hence, the model comprises of just the first network.

For this model, a huge list of hyperparameters can be identified. However, for this study, only two parameters were considered – the two learning rates of the respective networks. The explored feature space is as follows:

1. Learning Rate 1 (Lr1): [0.0005, 0.001, 0.005, 0.01, 0.03]
2. Learning Rate 2 (Lr2): [Lr1/30, Lr1/10]

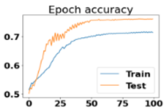
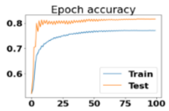
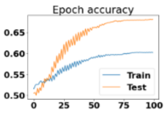
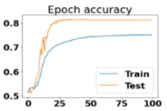
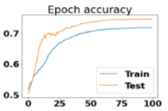
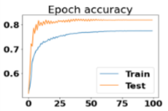
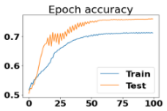
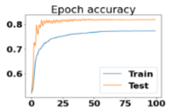
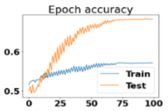
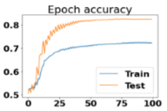
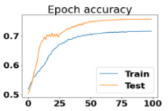
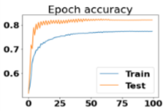
For this selected space there are total 10 iteration sets (Lr1, Lr2). The results are compressed in the below images as form of a table. There are three major columns one for each model architectures. In addition to that, the model results are separated into two columns based on the learning method. The results obtained for each set is juxtaposed in the below image for comparison.

A. Lr1: 0.0005

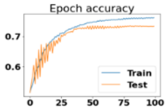
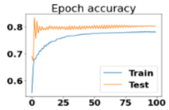
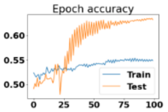
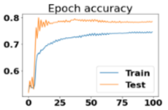
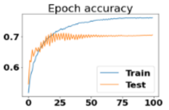
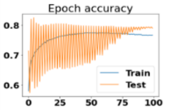
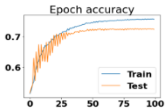
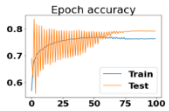
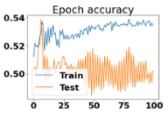
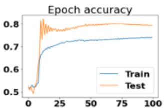
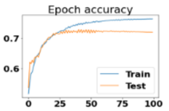
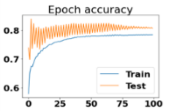


Approach	Regressor		Classifier		Concat	
Method	U	S	U	S	U	S
Lr2: Lr1/30	Accuracy: 0.628	Accuracy: 0.811	Accuracy: 0.552	Accuracy: 0.805	Accuracy: 0.694	Accuracy: 0.815
Lr2: Lr1/10	Accuracy: 0.654	Accuracy: 0.812	Accuracy: 0.575	Accuracy: 0.8079	Accuracy: 0.695	Accuracy: 0.818

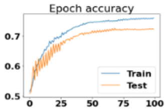
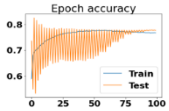
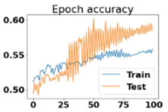
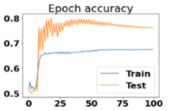
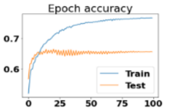
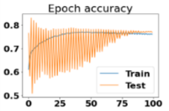
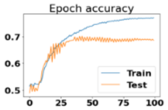
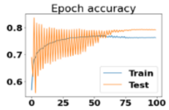
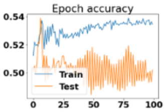
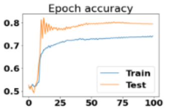
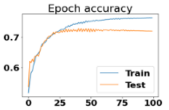
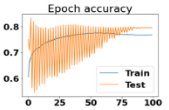
B. Lr1: 0.001

Approach	Regressor		Classifier		Concat	
Method	U	S	U	S	U	S
Lr2: Lr1/30	Accuracy: 0.760 	Accuracy: 0.817 	Accuracy: 0.681 	Accuracy: 0.814 	Accuracy: 0.747 	Accuracy: 0.818 
Lr2: Lr1/10	Accuracy: 0.760 	Accuracy: 0.819 	Accuracy: 0.683 	Accuracy: 0.824 	Accuracy: 0.756 	Accuracy: 0.820 

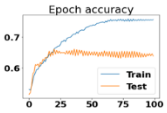
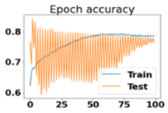
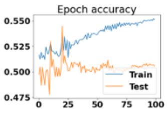
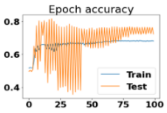
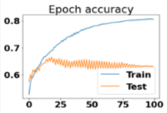
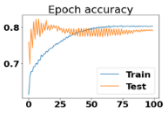
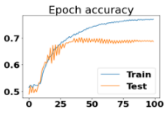
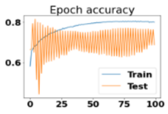
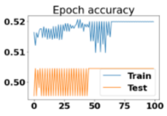
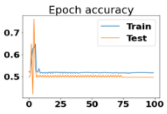
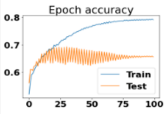
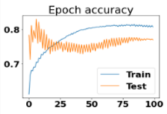
C. Lr1: 0.005

Approach	Regressor		Classifier		Concat	
Method	U	S	U	S	U	S
Lr2: Lr1/30	Accuracy: 0.733 	Accuracy: 0.802 	Accuracy: 0.632 	Accuracy: 0.786 	Accuracy: 0.706 	Accuracy: 0.790 
Lr2: Lr1/10	Accuracy: 0.724 	Accuracy: 0.791 	Accuracy: 0.50 (!) 	Accuracy: 0.794 	Accuracy: 0.719 	Accuracy: 0.807 

D. Lr1: 0.01

Approach	Regressor		Classifier		Concat	
Method	U	S	U	S	U	S
Lr2: Lr1/30	Accuracy: 0.723 	Accuracy: 0.776 	Accuracy: 0.548 	Accuracy: 0.762 	Accuracy: 0.656 	Accuracy: 0.770 
Lr2: Lr1/10	Accuracy: 0.710 	Accuracy: 0.793 	Accuracy: 0.50 (!) 	Accuracy: 0.783 	Accuracy: 0.683 	Accuracy: 0.793 

E. Lr1: 0.03

Approach	Regressor		Classifier		Concat	
Method	U	S	U	S	U	S
Lr2: Lr1/30	Accuracy: 0.638 	Accuracy: 0.762 	Accuracy: 0.505 	Accuracy: 0.764 	Accuracy: 0.629 	Accuracy: 0.790 
Lr2: Lr1/10	Accuracy: 0.686 	Accuracy: 0.687 	Accuracy: 0.50 (!) 	Accuracy: 0.495 	Accuracy: 0.656 	Accuracy: 0.769 

The inferences that can be laid out from the above results are:

1. As we observe results for each column for the increasing learning rates, it can be concluded that the classification accuracy increases as the learning rate is increased from 0.0005 to 0.001. But, as the learning rate 1 is increased further the results deteriorate.
2. For the supervised case, the conventional model performs best.
3. For the unsupervised case, as expected, the GAN inspired network where regressor was considered as the discriminator returned best results.
4. The classifier model didn't perform well. However, if the discriminator loss is analyzed, it can be seen that the loss increases in this model whereas for the regressor model, the discriminator loss remains mostly constant. This means that the classifier model imposed harsher adversarial training.
5. These observations can act as a reference point for further tuning and development of the architecture.

3.0 Future Work

- Test the current developed model on the real data to be provided by the company Titan.
- Fuse simulated data with real data to alleviate the challenges associated with labeled data shortage.
- Keep optimizing our signal processing code to be handle larger data set.
- Fuse the processed signal with the raw vibration signal, and use this fusion as input to our machine learning model.
- Continue exploring the existing new domain adaptation methods in the literature to deal with the varying operating RPM

Criticality Analysis and Asset Management of a Power Generation Facility

Pooyan Sharifi

Introduction

Capital Power is an independent power generation company who own and operate over 5100MW of power generation across 25 facilities in North America. Amongst these facilities there is an 875MW natural gas fueled power plant located in Brampton, Ontario. The Capital Power generation facility located in Brampton does not operate on a 24/7 basis but rather generates power during intermediate and peak demand periods.

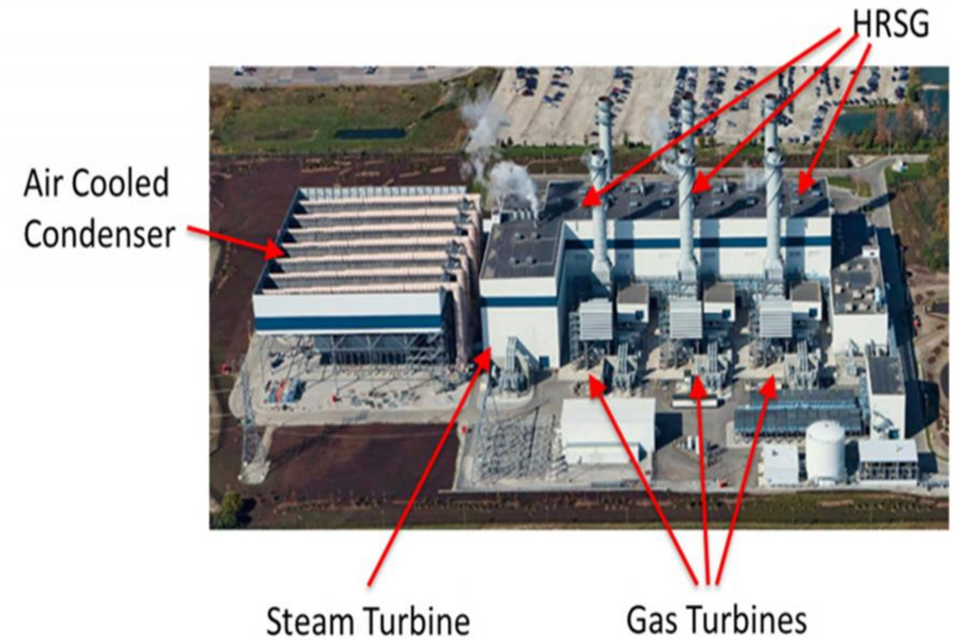


Figure 33: Aerial view of Capital Power Goreway Station

Within such a power generation facility there is an abundance of large-scale systems and sub-systems consisting of various equipment and parts. These equipment and parts can be critical to the operation of the plant. The facility consists of three gas turbines, one steam turbine and three Heat Recovery Steam Generators (HRSG 11, 12 and 13) with duct burners. These are the major power generating equipment in the facility. There are various other process equipment as part of the power generation facility such as boiler feed water pumps, condensate pumps, hydraulic lift oil pumps, lube oil pumps and automated control valves amongst many others.

The focus of this project is on the criticality of parts and equipment which together allow for operation of the power generation process. A critically analysis will be performed to identify these critical spare parts. Furthermore, we will utilize maintenance records and data provided by Capital Power to derive insights from this analysis such as identifying the number of spare parts to store in inventory. Ultimately a framework will be generated to aid in the decision-making process for Capital Power's future process requirements.

1.1 Identify Major Systems, Equipment, Parts

Given the large scale of a power generation process and the abundance of various equipment involved in the process it is essential to identify and prioritize equipment of interest. Through collaboration with Capital Power the Maintenance and Engineering specialists helped identify groups of major equipment of assets. These are suggested to be priority based on their importance in the functioning of the power generation process as well as intuitively based on a known history of maintenance work orders and costs. These major groups of assets will be identified and briefly described below. It is important to note that the goal of criticality will be achieved statistically though the use of data in the form of maintenance records on said equipment rather than from a process engineering perspective.

1.1.1 Boiler Feedwater Pumps & Motors

The Boiler Feedwater Pumps and Motors are part of the feedwater system to the Heat Recovery Steam Generator (HRSG) Intermediate Pressure (IP) and High Pressure (HP) steam drums for the production of steam.

The Feedwater Pumps provide feedwater at sufficient pressures and flow rates to maintain normal water level in the HP and IP drums from start-up to full load operation. Each HRSG (3 in total) is equipped with two 100% capacity Feedwater pumps. The Feedwater Pump is a barrel type, eight-stage centrifugal pump manufactured by Sulzer Pumps Inc each driven by a 3400-horsepower motor. In total there are six Feedwater Pumps, and their tag names are 11A, 11B, 12A, 12B, 13A, 13B. The number is referring to the associated HRSG which it provides feedwater to. Each HRSG is setup for a lead/lag pump setup where one pump is sufficient for full operation and the other is used as a backup. These pumps alternate lead/lag roles to more evenly distribute running hours and associated wear and tear between them.

Firstly, we examined operational data exploration of the equipment as provided by Capital Power. The operating run hours of each pump were determined via a status control bit which is saved on Capital Power's PI Historian SCADA system. This data was extracted on an hourly basis which allowed us to determine the operational run hours of the equipment. This control bit (1 when pump is running, 0 when off) was extracted hourly from 2009-2020. Additionally, the record of all maintenance work (planned and unplanned) conducted on the equipment was saved in Capital Power's Mainsaver database and extracted for our uses. The results are illustrated the figure below.

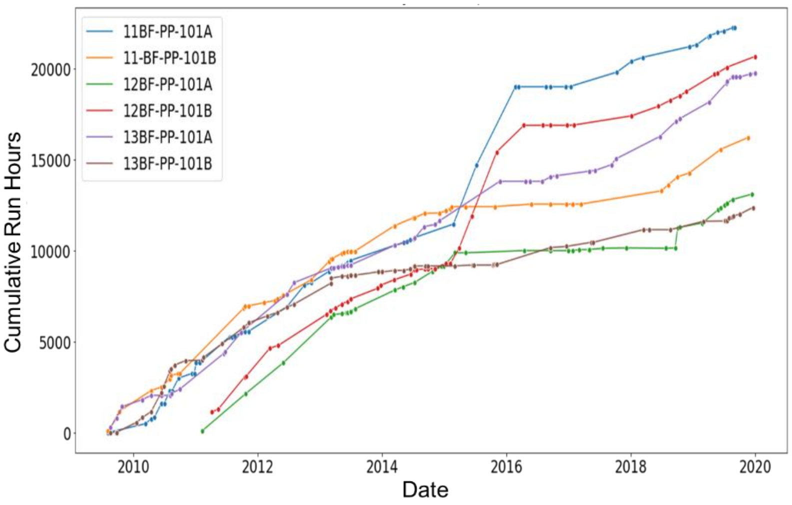


Figure 34: Cumulative Run Hours of Feedwater Pumps

Each line in the Figure above corresponds to a specific boiler feedwater pump. The line for each pump illustrates the run hours of the pump from 2009-2020 and each dot on the line corresponds to a maintenance event (i.e., preventative maintenance, corrective maintenance action). We observed the order of most run hours to be 11A, 12B, 13A, 11B, 12A, 13B. Additionally a long period of minimal run time for pump 11B between 2015-2017 and 13B between 2013-2016 was observed. This was corroborated with extensive corrective maintenance actions in the maintenance records between these dates.

Next, we will examine the number of work orders associated with each pump. Naturally we would assume those pumps with most run hours will have the most work orders due to increased wear and tear from extensive usage.

In general, it was observed that pumps that have greater run hours have more work orders associated with them. For example, pumps 11A and 12B exhibit this pattern while having the longest run hours of all pumps. However, for HRSG13, pump 13B has many work orders despite having the lowest run hours. This was likely due to various mechanical issues with the pump which required several work orders on the pump while it was out of service. Lastly, we will examine the nature of the work orders for each of the pumps.

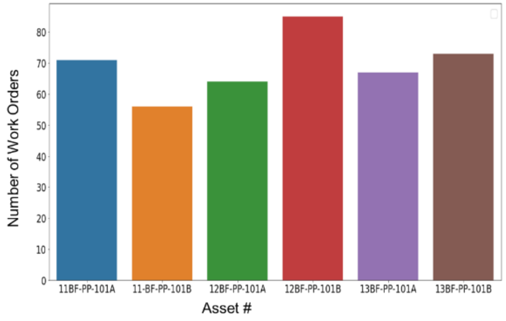
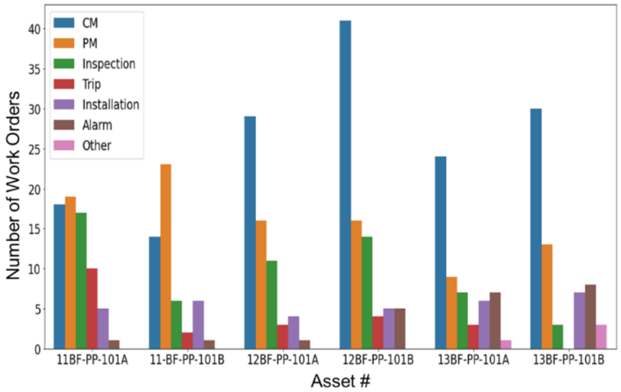
	

Figure 35: Number of Work Orders for Feedwater Pumps	*Figure 36: Work Order Classification for Feedwater Pumps*

We observed that most work orders are preventative maintenance (PM) and corrective maintenance (CM) actions. The difference being that PMs are planned and part of the PM schedule as designed by the Capital Power Maintenance group. On the other hand, a CM is a maintenance action triggered by an operations related issue such as a clogged filter or alarm from the distributed control system (DCS). The key being that the CM occurs when the equipment fails whereas a PM does not. This was a key distinction used for later analysis.

We also noted that there is a significant number of inspections done which include visual inspections or various types of diagnostic testing such as oil sample analysis and vibration measurements.

1.1.2 Condensate Pumps & Motors

The major purpose of the condensate pumps and motors are to deliver condensate collected in collections tanks (from condensed steam) to the LP steam drum as well as to supply cooling to makeup water and various other loads. These pumps and their motors are a major part of the overall condensate system which is essential for operation and power generation. There are a total of four 33.3% capacity condensate pumps named 15A, 15B, 15C and 15D. They are 8 stage vertical pumps each driven by a 500-horsepower motor. Three pumps are required for operation at full capacity (all turbines running) but less can be utilized when the plant is running at a partial load. A fourth pump is used as backup at full capacity acting as a fail-safe.

Firstly, we examined operational data exploration of the equipment as provided by Capital Power. The operating run hours of each pump were determined via a status control bit similar to what was conducted for the feedwater pumps. Hourly data was extracted from the PI SCADA system which allowed us to determine the operational run hours of the equipment. This control bit (1 when pump is running, 0 when off) was extracted hourly from 2010-2020. Additionally, the record of all maintenance work (planned and unplanned) conducted on the equipment was saved in Capital Power's Mainsaver database and extracted for our uses. The results are illustrated the figure below.

The order of most run hours for the pumps is 15A, 15B, 15C and 15D. It is observed that pumps 15C and 15D have relatively low run hours in comparison to 15A and 15B and there are quite significant difference in run hours between 15A and 15B. The higher run hours of pump 15A and 15B suggest they are the primary pumps ran during partial load while 15C and 15D are usually only turned on for full load operation. Next, we will illustrate the number of work orders for each pump.

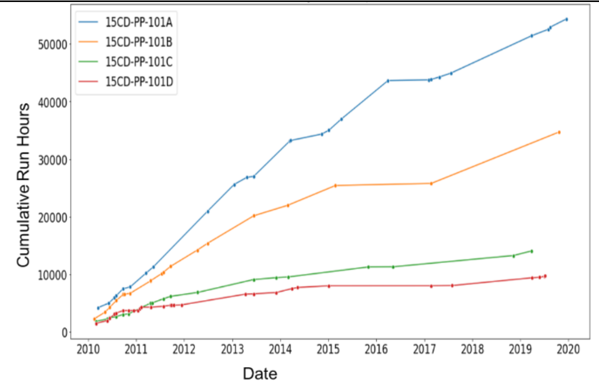


Figure 37: Cumulative Run Hours of Condensate Pumps

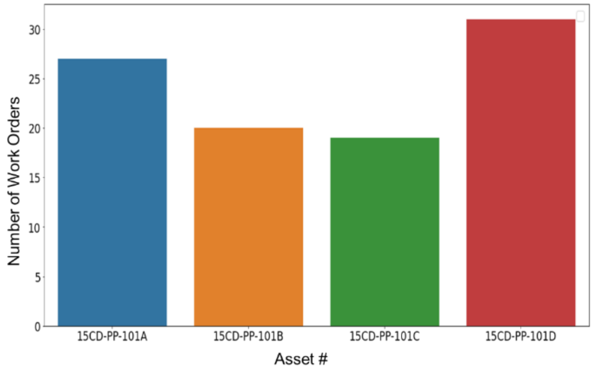


Figure 38: Total Work Orders of each Condensate Pump

It is observed that the number of work orders for each pump increases with the number of run hours on the pump. This is the trend observed for pumps 15A, 15B and 15C however pump 15D appears to be an exception to this trend. Pump 15D is observed to have the lowest number of run hours but the most work orders. Lastly the type of work order for each pump will be explored.

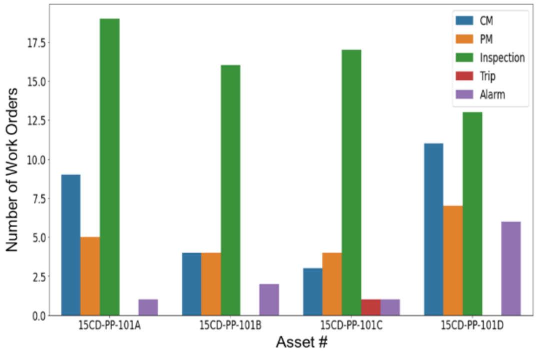


Figure 39: Work Order Classification for Condensate Pumps

The work orders for each pump are classified based on the type of work that is completed. As evident in the Figure above, there are more CMs than PMs for pumps 15A and 15D. This is likely explained by the fact that pump 15A has the highest run hours while 15D has various mechanical issues that requires more frequent repairs. We also observed that the condensate pumps all have a significantly higher number of inspections than the feedwater pumps.

1.1.3 CCW & GCW Pumps & Motors

The closed-cycle cooling water (CCW) and generator cooling water (GCW) pumps & motors are major parts of the cooling systems. Various equipment have cooling requirements for operation and these are provided from these pumps. There are two CCW pumps referred to as 19CCA and 19CCB. There are two GCW pumps referred to as 19GCA and 19GCB. Firstly, we will explore the run hours of each set of pumps from 2009-2020.

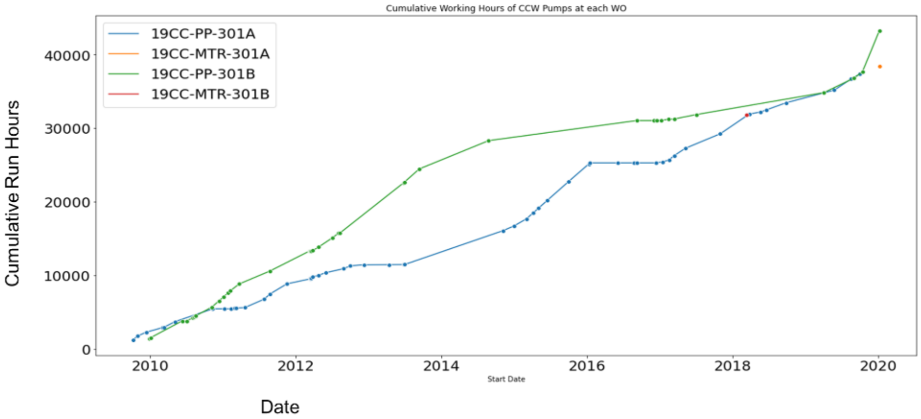


Figure 40: Cumulative Run Hours of CCW Pumps

It is observed that both pumps have a similar number of cumulative run hours. Each motor is tracked separately but only contained a single data point hence no line is apparent.

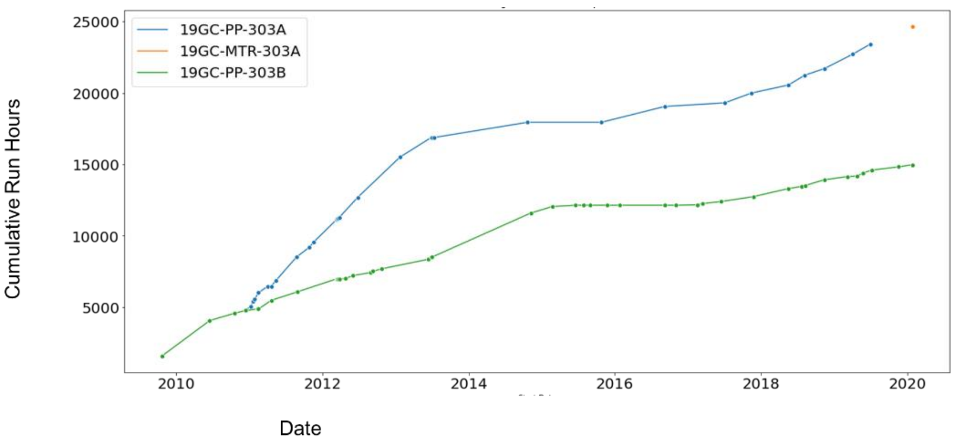


Figure 41: Cumulative Hours of GCW Pumps

For the GCW pumps we observed that pump 19GCA has more cumulative run hours than 19GCB. The motor for pump 19GCA has a single data point hence no line is apparent. No work orders were available for the motor on pump 19GCB. Next we will explore the number of work orders for each sets of pumps.

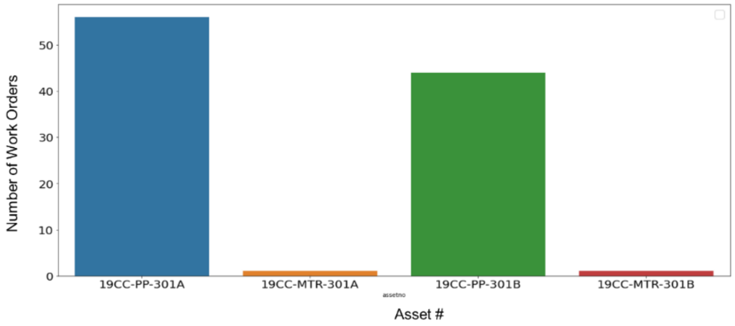


Figure 42: Work Orders for Each CCW Pump

Of the CCW pumps we observe that pump A has a larger number of work orders associated with it despite both pumps having a similar number of work orders. However, this difference isn't very large. Further breakdown as to the nature of the work orders could provide more insight.

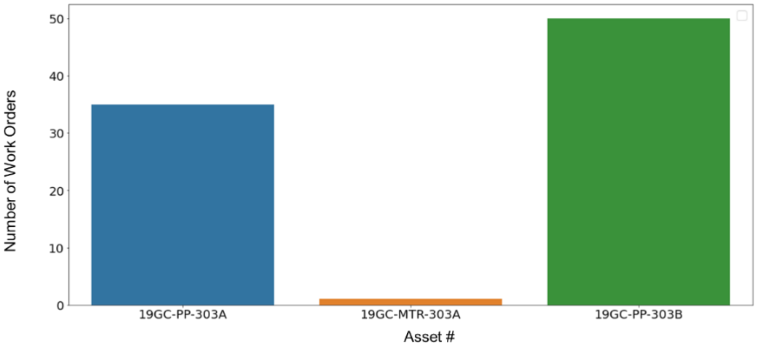


Figure 43: Work Orders for each GCW Pump

For the GCW pumps it is evident that pump 19GCB had more work orders than pump 19GCA. This does not appear to coincide with the fact that pump 19GCA has more run hours than the other pump. However, it appears that pump 19GCB has significant time period of no run hours and many work orders which suggest it might be a pump with various mechanical issues that is worked on while offline. This could explain the larger number of work orders with lesser number of run hours. Lastly, we will explore the type of work orders for each set of pumps.

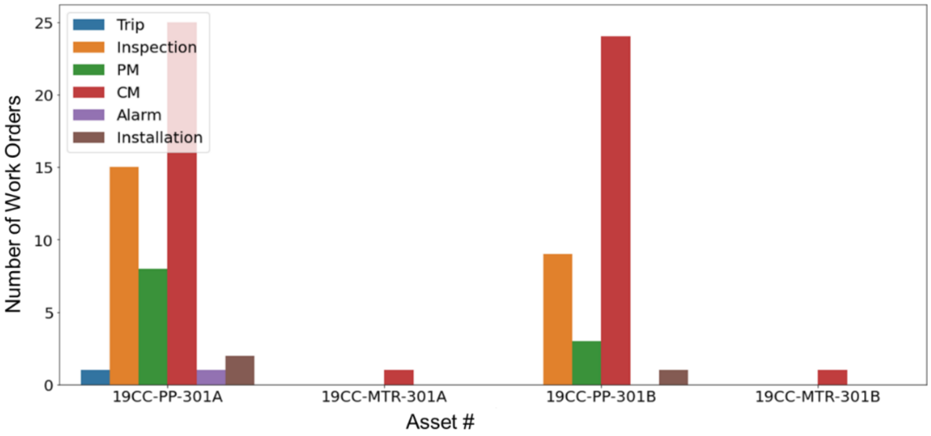


Figure 44: Work Order Classification of CCW Pumps

There appear to be significantly more CMs than other types of work orders for both the CCW pumps. The next most frequent work order type is inspection followed by PMs. As mentioned earlier the motors for these pumps each have a single data point available and they are CMs.

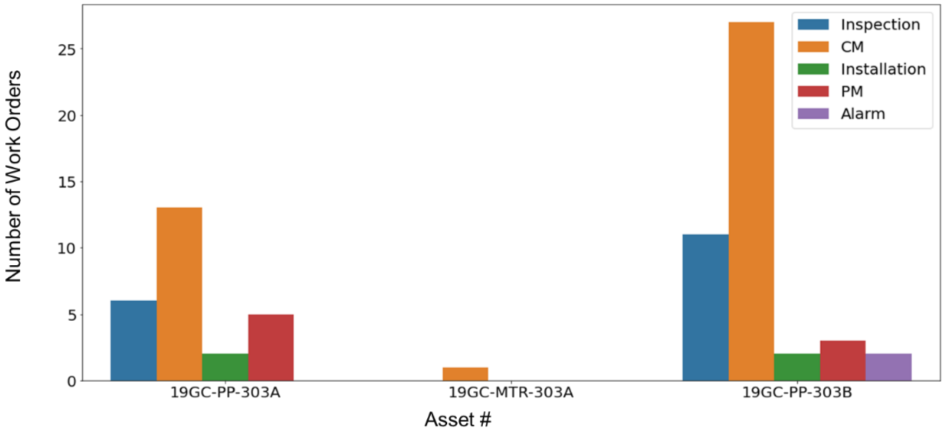


Figure 45: Work Order Classification of GCW Pumps

For the GCW pumps we observe a similar trend where most work orders are CMs. The next most frequent work order type is inspections followed by PMs.

1.1.4 Process Valves

As expected in a power generation plant there are a plethora of valves of many different sizes and types for different applications. The primary mediums in this plant include cooling water, condensate or hot water, steam and natural gas. These valves may operate differently such as manual valves, solenoid valves and modulating valves. Many of these valves are essential for proper plant operation and so they were identified by Capital Power as another asset group of interest. The majority of larger automated valves have all records of maintenance work stored on Capital Power's Contro Valve Portal. These records which contain information regarding the work orders and their components have been extracted into an excel format for further analysis in later sections.

Unlike other equipment such as pumps, the operation of valves is dependent on the run status of the system or generators. Hence the valve age can be estimated using the run hours of the generators. This can be deduced from the number of starts of the generator for solenoid valves (open/close) or the number of run hours for modulating valves.

2. Estimation of Failure Distributions for Major Systems

The raw data for work orders was filtered to keep only relevant information including asset number, work description and corrective action description. Next the work orders were classified based on the work description which is a text field entered by the Maintenance team at Capital Power. Work orders are classified into various categories such as PM, CM, Inspection, Trip, Alarm, etc. Based off the work description a new field was created which identifies the component of the system which the work order is for such as lube oil system, bearings, mechanical seals, etc. These are components which comprise the overall system or equipment. Finally with this data the cumulative run hours are determined at the date of each work order in order to generate a time series of life cycles for each component for the asset.

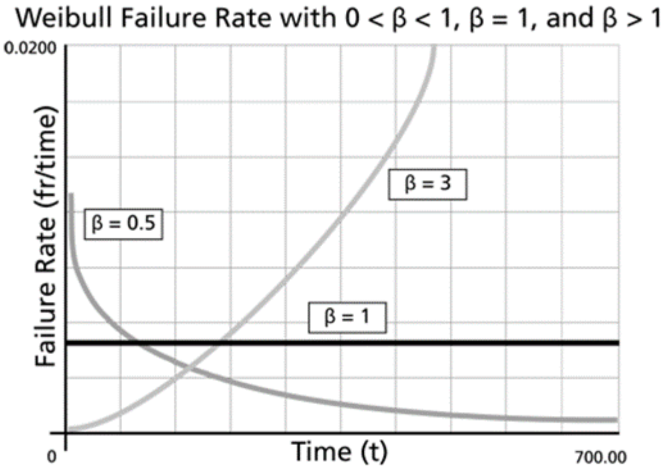
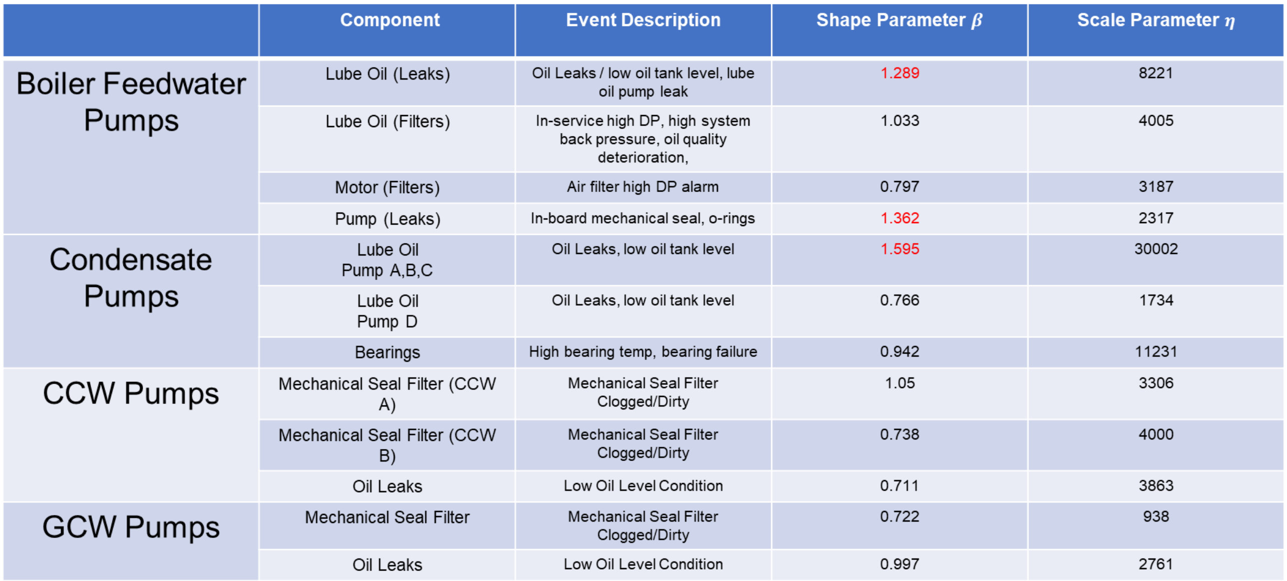


Figure 46: Weibull Failure Rate with Different Shape Parameter Values

A 2-parameter right censored Weibull distribution (right censored are the PMs) will be applied to the data in order to estimate important life characteristics for each of the components. The results of this analysis are summarized in the table below.

Table 4: 2 Parameter Right Censored Weibull Estimated Parameters



	Component	Event Description	Shape Parameter β	Scale Parameter η
Boiler Feedwater Pumps	Lube Oil (Leaks)	Oil Leaks / low oil tank level, lube oil pump leak	1.289	8221
	Lube Oil (Filters)	In-service high DP, high system back pressure, oil quality deterioration,	1.033	4005
	Motor (Filters)	Air filter high DP alarm	0.797	3187
	Pump (Leaks)	In-board mechanical seal, o-rings	1.362	2317
Condensate Pumps	Lube Oil Pump A,B,C	Oil Leaks, low oil tank level	1.595	30002
	Lube Oil Pump D	Oil Leaks, low oil tank level	0.766	1734
	Bearings	High bearing temp, bearing failure	0.942	11231
CCW Pumps	Mechanical Seal Filter (CCW A)	Mechanical Seal Filter Clogged/Dirty	1.05	3306
	Mechanical Seal Filter (CCW B)	Mechanical Seal Filter Clogged/Dirty	0.738	4000
	Oil Leaks	Low Oil Level Condition	0.711	3863
GCW Pumps	Mechanical Seal Filter	Mechanical Seal Filter Clogged/Dirty	0.722	938
	Oil Leaks	Low Oil Level Condition	0.997	2761

The first column describes the major group of equipment which was analyzed and identified in the previous section. For each of these equipment there were various components which had enough work order data in order for an analysis to be conducted. The event description for each component describes the various descriptions of work orders for these components or in some cases the condition which would trigger a CM. The final two column are the estimated Weibull parameters which describe the distribution.

For the boiler feedwater pumps the lube oil system has two components or failure modes. The first being leaks which often occur as the gasket, seal or pump being inspected by an operator and noticing a leak. Additionally, a low oil tank level will also be classified as a failure due to oil leak. The corrective action undertaken is to repair this related equipment. The shape parameter for this failure mode is 1.289 which suggests the failure rate increases over time. This could be due to gradual wear of related parts such as the seal or gasket. In addition to oil leaks another major component or failure mode is of the lube oil filters. These filters eventually build up debris over time which increase the differential pressure across the filter and reduce the flow rate of the lube oil. This reduction in flow could result in an increase in lube oil temperature which could result in faster lube oil degradation or tripping of equipment due to high lube oil temperature. The shape parameter was approximately equal to one which suggests a constant failure rate over time. This would coincide with the idea that filters typically get clogged in roughly the same number of operating hours over time. The next major component of the boiler feedwater pumps is the motor which had two distinguishable failure modes. The first being the air filters of the motor which typically failed and required replacement when clogged resulting in a high differential pressure alarm. This component had a shape parameter of 0.797 which suggests it has early-life failures. The next component is of pump leaks which involve failure of the mechanical seal or O-rings on the pump. These had a shape parameter of 1.362 which suggest a failure rate that increases over time. This could be due to gradual wear of the seal and O-rings over time.

Moving onto the condensate pumps we observed a failure mode related to the lube oil system. This failure mode corresponded to conditions described as oil leaks and low oil tank level by Maintenance operators. The corrective action was to refill oil into the system. For pumps A, B, C the shape parameter was 1.595 which indicates a failure rate that increases with time possibly due to aging of the equipment. For pump D the shape parameter was 0.766 which suggests early life failures which corresponds to the observation made from the time series data of many frequent failures in <500 run hours around the year 2019. The analysis for pump D was done separately since it behaved significantly differently from the other pumps and there was sufficient data for individual analysis. Lastly the bearings for all pumps had a shape parameter of nearly 1 which indicates a near constant failure rate. The bearing failure mode corresponded to high bearing temperature or outright bearing failure. The corrective action was almost always outright replacement of the bearings which is a significant maintenance cost.

The CCW pump A had a failure mode involving the mechanical seal filter which occurred when it got clogged or was identified by Maintenance personnel as dirty. The shape parameter was 1.05 which is nearly 1 which indicates a nearly constant failure rate. For CCW pump B the shape parameter was 0.738 which is less than 1 and suggest early life failures. Lastly the shape parameter of oil leaks for the CCW pumps is 0.711 which is less than 1 and suggests early life failures.

For the GCW pumps there are a total of two failure modes identified from the data. The first being the mechanical seal filter with a shape parameter of 0.722 which is less than 1 and suggests early life failures. Lastly the oil leaks failure mode had a shape parameter of 0.997 which is approximately 1 which suggests a constant failure rate.

3 Spares Recommendations

Utilizing the results of the failure distribution we can now aim to provide recommendations as per the number of spares to keep in inventory. Spares Management software offered by the University of Toronto can be used for this purpose. However additional decision criteria need to be decided upon or assumed to generate results. Additionally, further system information needs to be discovered about each component.

3.1 Optimization Criteria

First, we must determine what is the most appropriate optimization criteria to use which coincides with ultimate goals of Capital Power. The first option is Interval Reliability which optimizes such that it minimizes the likelihood of there being a no stock-out period over a specific period of time. The next option is to optimize for Instant reliability whose objective is to have no stock-out when stock is required. Another option is Availability which optimizes with the objective of maximizing system uptime. Lastly Cost Optimization is the final decision criteria whose objective is to minimize all costs including holding costs, downtime costs, emergency replacement costs, etc.). It was assumed that the ultimate objective of Capital Power is to minimize maintenance costs and downtime resulting in larger net profits. This naturally led to the decision to use Cost Optimization as our optimization criteria moving forward. It is important to note though that the use of other optimization criteria does not drastically change the results of the proceeding sections.

3.2 System Information

Further information regarding the system is required for the Spares Management software to be utilized. The following system information is required:

- Spare Type (Repairable spare, non-Repairable spare)
- Parts in Use

- Mean Time Between Replacement (from Weibull parameters)
- Planning Horizon (we decide or can use lead time of the part)
- Replacement Options
- Downtime vs. No Down Time
- Regular Cost of Spare Part
- Emergency Cost of Spare Part

3.3 Spares Results

The results from Spares Management software are summarized in table 2.

It was assumed that the number of spares in stock at the start of the planning horizon is set to 0. The holding cost of having spares in inventory was set to $0. All spares are the non-repairable type (replaced at failure). The emergency cost of spare was assumed to be twice of the regular cost a spare.

4 Recommendations

The next group of equipment which will undergo further analysis are the large, automated valves throughout the plant. These valves have had all their relevant data extracted from the Capital Power Contro valve portal. The remaining task is to identify valves of interest and extract operational run hours for each valve. Depending on the nature of the valve (modulating or solenoid) the age will be derived either from the operational run hours of the turbines or the number of starts.

Table 5: Spares Management Recommendations for Various Components

	Component	Parts in Use	Planning Horizon	MTBR	Regular Cost of Spare	Emergency Cost of Spare	Downtime if no spares available?	Recommended # of Spares	Total Cost
Boiler Feed Water Pumps	Lube Oil (Filters)	1	10,000 hours	4000 hours	$250	$500	N	7	$637
Condensate Pumps	Bearings	1	25,000 hours	11250 hours	$15,000	$30,000	N	2	$49829
CCW Pumps	Mechanical Seal Filter	1	10,000 hours	3310 hours	$300	$600	N	3	$1237
GCW Pumps	Oil Leak	1	10,000 hours	2760 hours	$350	$700	N	4	$1684
Boiler Feed Water Pumps	Leaks (mechanical seal)	1	10,000 hours	2120 hours	$10,000	$11,000	N	2	$50,319
Condensate Pumps	Lube Oil Pump D (Leaks)	1	10,000 hours	1734 hours	$350	$700	N	6	$2549
CCW Pumps	Oil Leak	1	10,000 hours	3863 hours	$600	$1200	N	2	$2320
GCW Pumps	Mechanical Seal Filter	1	2,500 hours	938 hours	$300	$600	N	3	$1176
Boiler Feed Water Pumps	Lube Oil (Leaks)	1	10,000 hours	8221 hours	$350	$700	N	1	$708
Condensate Pumps	Lube Oil Pump A,B,C (Leaks)	1	50,000 hours	33783 hours	$350	$700	N	1	$845

Prediction of Emergency Response Strategies Based on Combustion Signatures from FTIR Spectroscopy Using Machine Learning Techniques

Sophie Tian, Chi-Guhn Lee, Dexen Xi, Yoon Ko, Nour Elsagan

1.0 Introduction

In rail transport, safety is of the utmost importance. If freight catches on fire, it is crucial to identify the type of the fire and scope of the associated hazards in an accurate and timely manner to protect people and the environment from the fire. In the emergency response to freight transportation fire incidents, the first responders are trained to leverage the Emergency Response Guidebook (ERG), which allows them to identify response strategies based on the hazardous materials involved in the transportation incident, and to follow the appropriate responses in the guidebook to mitigate the freight fire incident. However, first responders routinely confront unknown hazards from freight fire incidents such as flammable, explosive, toxic, and corrosive materials. Such cases occur when the goods transported are unlabeled or the materials burning are unknown, and based on the ERG, the first responders are instructed to use the blanket guide: “Guide 111 - Mixed Load / Unidentified Cargo” which may be too general. As a result, there is a need to aid the first responders in the emergency response to such goods being transported.

The Fourier Transform Infrared Spectroscopy (FTIR) analyzer is a device capable of detecting an array of chemical components from the effluents of the burning material over time. The spectral data obtained from an FTIR produces unique fingerprints of the sample materials (Titus et al., 2019), making the FTIR well-suited to collect data for machine learning (ML) models where patterns and structures are expected to exist within the data. As an early work, Chen et al. (2000) used data from an FTIR analyzer capable of detecting the concentrations of 18 gas species to train an artificial neural network (ANN) for classifying the corresponding fires as either flaming, smoldering, or nuisance. The training data were composed of repeated tests on eight different materials or chemical compounds, and the validation data came from fire and non-fire cases of materials not observed in the training set. The trained three-layer ANN correctly predicted 96% of the 248 test cases, demonstrating the feasibility of using FTIR data to detect fires. To the best of our knowledge, there has been few other studies utilizing machine learning algorithms to make classification decisions with the FTIR on combustion processes.

Although ML has yet to be adopted as a candidate method in the analyses of combustion processes, it has been gradually introduced into and increasingly used in the field of fire sciences, especially in fire detection. In a comprehensive review of chemical-based indoor fire detection systems and associated algorithms, Fonollosa et al. (2018) identified that chemical gas sensors can improve fire sensitivity and early detection, but they suffer from a high rate of false alarms. The study suggested pattern recognition algorithms as the only path to improve false alarm immunity and cited numerous works that utilized ML techniques such as ANNs, K-nearest-neighbors (KNN) and decision trees to decrease the chances of false alarms. However, in their discussion of gas sensors for combustion products, the authors only discussed the use of chemical sensor components, namely electrochemical cells, metal oxide sensors (MOX), and non-dispersive infrared cells (NDIR) for fire detection, and explicitly excluded the FTIR analyzer from this discussion without providing rationale for this choice. This suggests a knowledge gap in the understanding of how FTIR analyzers compare to chemical sensor components in fire detection. These chemical gas sensors are similar to FTIR analyzers since they are both used to detect and to quantify target gases, and both take measurements over time. However, the FTIR analyzer has the advantage of being able to detect and collect the signatures of a variety of gases simultaneously, while typically an array of chemical sensor components needs to be manually selected and assembled to measure the target gases. Nevertheless, both tools produce multivariate, time series data. As a result, machine learning models that have been demonstrated to work well for gas sensor data should also work well for FTIR data. In the field of fire detection using images, Park et al. (2019) proposed a fire detection system incorporating a deep neural network for time series sensor data analysis as well as a convolutional neural network (CNN), which is a type of deep neural network used to efficiently analyze temporal or visual data, for fire detection in images among other multi-functional components. Overall, machine learning techniques have become increasingly recognized in the area of fire detection, using either time series data or image data.

In a broader context, ML techniques have been widely adopted to the classification of gases using data collected from gas sensors such as the electronic nose (E-nose). In a review of smart gas sensing technologies, Feng et al. (2019) outlined and compared various smart gas sensor arrays, signal processing methods and gas pattern recognition algorithms including support vector machines (SVM), KNN and ANNs. They highlighted that machine learning now plays a key role not only in making gas classification decisions but also in its ability to adapt for sensor drifts, which alleviates the need to manually manipulate the signals captured by the gas sensors. In recent years, as neural networks proved to be universal function approximators, gas classification also began to adopt neural networks to replace traditional approaches. For gas classification that uses time series as input data, an intuitive model to use is a CNN, which employs convolutional kernels to reduce the number of trainable parameters in the model and also to detect the same feature at different time points within a time series. Peng et al. (2018) proposed a deep-CNN model, GasNet, with up to 38 layers to classify four types of gases using data from eight MOS sensors, and this model significantly outperformed a trained SVM and ANN. Zhao et al. (2019) proposed a one-dimensional deep CNN, 1D-DCNN, to classify three pure gases and two binary mixture gases. The 1D-DCNN model was able to automatically extract important features from the data through convolutional kernels, and it significantly outperformed a SVM, a KNN, an ANN and a random forest as benchmarks. Another approach is to apply existing deep CNN architectures such as VGG (a deep CNN architecture named after its creator, the Visual Geometry Group) and Residual Network (ResNet) to gas classification as demonstrated in Han et al. (2019), however such deep architectures would require a large amount of training data, which is often difficult to obtain in gas classification. Since the field of gas classification gradually favored CNN architectures to capture the temporal information effectively, it is natural to approach the combustion signature classification task using combustion data measured by an FTIR via CNNs as well.

Another research area for consideration is the research on the time series classification task itself, which focuses on developing algorithms leveraging the temporal information, and much work has gone into algorithmic development and comparison. A recent paper by Ruiz et al. (2021) compared state-of-the-art multivariate time series classification (MTSC) methods on 26 equal-length time series datasets and highlighted four models: Canonical Interval Forests (CIF), the Hierarchical Vote Collective of Transformation-based Ensembles (HIVE-COTE), the Random Convolutional Kernel Transform (ROCKET), and InceptionTime as significantly outperforming the previous time series classification benchmark named dynamic time warping (DTW). In particular, ROCKET, developed by Dempster et al. (2020), was referred as the state-of-the-art method and the recommended starting point for benchmarking in future research on time series classification tasks.

In this work, our objective is to obtain the hazard characteristics of fires using the effluents collected with a FTIR analyzer, which we call combustion signatures, from fires conducted in a bench-scale test set-up, cone calorimeter, by training ML models with the combustion signature data. The combustion signatures used in this study are collected in a laboratory environment; however, our end goal is to augment the dataset, refine the trained model, and eventually apply the model to real freight fire scenarios. For the machine learning models, we use CIF and ROCKET as suggested by Ruiz et al. (2021), and we will evaluate the classification performance and the time required to train each model. This work a first step taken to understand the fire hazards, which will allow us to later link them to the emergency response strategies in the ERG where possible and provide more specific instructions to the first responders when the burning materials are unknown.

Our main contributions are summarized below:

- We review the applications of artificial intelligence and machine learning in the field of fire sciences and more broadly, classification of gases, to demonstrate that machine learning has been gradually adopted in these fields.
- We apply the classification of six hazard categories on the combustion analysis dataset using existing top-performing multivariate time series classification methods.
- We demonstrate empirically the potential of using machine learning in fire sciences and provide our results as a first step towards allowing first responders to identify fire hazards in unlabeled, burning freights

2.0 Materials and Methods

In this section, we will describe the experiment setup used to collect the combustion signature data, the classification method used, as well as setup for training the machine learning models.

2.1 Experiment Setup

Experiments were conducted using a Cone Calorimeter coupled with a FTIR gas analyser to analyze the emissions. The cone calorimeter is used to detect the fire behaviour of a fuel sample under ambient conditions, where a material of specific size is subjected to a pre-specified heat flux, and heat release and mass loss rates are measured. The heat flux was set to 50 kW/m2 in all tests used for this work.

MKS Multigas 2030 continuous FTIR gas analyzer was connected to the exhaust of the cone calorimeter to analyze the gaseous effluents. The gas analyzer was equipped with a 200 mL gas cell heated to 191°C and 5.110 m optical path with a liquid nitrogen cooled detector.

Before sampling, the gas analyzer was flushed with pure dry nitrogen gas (grade 5.0) at a rate of 3L/min for at least 1 hour before a background spectrum of nitrogen was recorded. Sampling was performed by switching the sampling port to a 5' heated sampling line (191°C). The heated sampling line was connected to a port sampling from the ventilation system. The gas ran through a stainless steel filter body, equipped with a bonded microfiber filter element .The sample gas was pulled into the FTIR at 3L/min via a MKS 2380 heated pump (191°C) at atmospheric pressure. The heated pump was fitted with two particulate stainless steel body filters with bonded microfiber/PTFE filter elements to prevent particulate matter from reaching the gas cell. A 5' heated line (191°C) was also used to connect the pump to the gas analyzer. The gas velocity was calculated to be 20.7 ft/sec. The expected lag between sampling and detection was therefore around 1 second.

Determination of the chemical concentrations present in the effluents was performed using the included MKS software (MG2000, Analysis Validation Utility and Gas Search Utility). Quantitative calibrations were performed by MKS and installed within the MG2000 software.

To confirm the validity of the analysis performed by MG2000, the analysis validation utility (AVU) was used on specific spectra with elevated concentration of targeted compounds. This was to determine an estimate on the detection limits, confidence limits, maximum bias and values specified in ASTM D6348 and EPA 320. The AVU was used strictly to confirm the analysis and check spectral residuals remaining. The values recorded by the AVU were not reported and used as a check. The Gas Search Utility was also used to help predict gases that may have been present and were subsequently added to the recipe.

As seen in Table 1, most materials were only tested once, four materials were tested twice, and three materials were tested three times. Overall, we obtained 30 samples from 20 distinct materials.

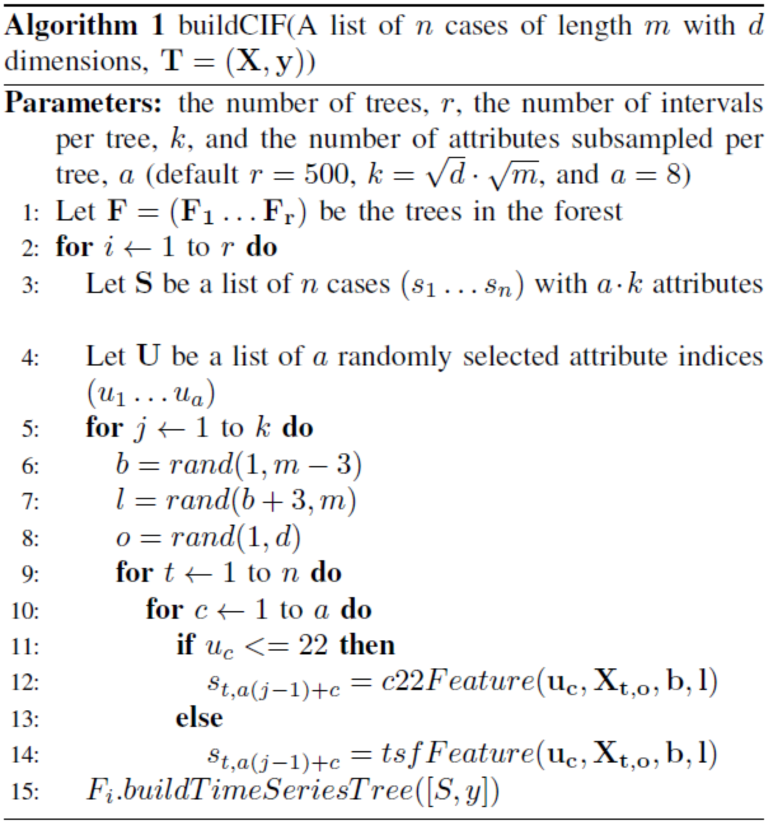
Table 1. List of materials tested, and the number of experiments conducted on each material.

	Materials with one experiment	Materials with two experiments	Materials with three experiments
Materials Tested	Blue CAT-6 wiring, carpet flooring, consumer electronics shell, electrical wiring, epoxy, heptane, intumescent caulking, melamine board, PMMA, polyethylene, polyurethanes foam, tar shingle, wall.	Crude oil, diesel, phenolic panel, polystyrene.	ABS, polyisocyanurate, PVC.
Number of materials	13	4	3

2.2 Classification Methods

2.2.1 Canonical Interval Forest

The Canonical Interval Forest is an interval-based forest of trees ensemble method proposed by Middlehurst et al. (2020). For each tree, the model randomly samples an interval from the time series data, computes a set of summary statistics using the selected interval, and concatenates the summary statistics into one vector. A decision tree, which makes predictions by recursively splitting the data on different attributes using a tree structure, is then built using the concatenated feature vector. This process is repeated to obtain r decision trees, resulting in a forest of trees. A classification decision will be made based on the majority vote of the forest of trees. The detailed process to build the CIF model is shown in Algorithm 1.



Algorithm 1 buildCIF(A list of n cases of length m with d dimensions, $\mathbf{T} = (\mathbf{X}, \mathbf{y})$)

Parameters: the number of trees, r, the number of intervals per tree, k, and the number of attributes subsampled per tree, a (default $r = 500$, $k = \sqrt{d} \cdot \sqrt{m}$, and $a = 8$)

```
Let $\mathbf{F} = (\mathbf{F_1} \ldots \mathbf{F_r})$ be the trees in the forest
for $i \leftarrow 1$ to $r$ do
    Let $\mathbf{S}$ be a list of $n$ cases $(s_1 \ldots s_n)$ with $a \cdot k$ attributes

    Let $\mathbf{U}$ be a list of $a$ randomly selected attribute indices $(u_1 \ldots u_a)$
    for $j \leftarrow 1$ to $k$ do
        $b = rand(1, m-3)$
        $l = rand(b+3, m)$
        $o = rand(1, d)$
        for $t \leftarrow 1$ to $n$ do
            for $c \leftarrow 1$ to $a$ do
                if $u_c <= 22$ then
                    $s_{t,a(j-1)+c} = c22Feature(\mathbf{u_c}, \mathbf{X_{t,o}}, \mathbf{b}, \mathbf{l})$
                else
                    $s_{t,a(j-1)+c} = tsfFeature(\mathbf{u_c}, \mathbf{X_{t,o}}, \mathbf{b}, \mathbf{l})$
    $F_i.buildTimeSeriesTree([S, y])$
```

Algorithm 1. The model build procedure for CIF (Middlehurst et al., 2020).

2.2.2 The Random Convolutional Kernel Transform (ROCKET)

ROCKET, developed by Dempster et al. (2020), leverages a large number of random convolution kernels to learn feature vectors and uses a linear classifier to make the classification decisions. Unlike CNN models, ROCKET is not a deep learning model and does not learn the weights of the convolutional kernels. Instead, it randomly initializes 10,000 convolutional kernels of various parameters such as length and weight of the kernel and apply every kernel to each instance of the data. This random initialization step makes the model extremely fast to run because no training stage is required to identify important patterns from the time series data.

After passing the inputs through the random convolutional kernels, the results, which we call the feature maps, will be used to compute two summary statistics. One statistic is the maximum value and the other is the proportion of positive values within each feature map. As a result, from the 10,000 kernels, we obtain 10,000 feature maps and subsequently 20,000 summary statistics. These summary statistics are then concatenated into a vector of length 20,000 and used to train a linear classifier such as a ridge regression or a logistic regression classifier. In Ruiz et al. (2020), the authors found that ROCKET achieved state-of-the-art accuracy even with small datasets, whether in the number of training examples or the length of the time series, in addition, this model was the fastest classifier among the classifiers tested. As a result, this model is a very good candidate for our small combustion analysis dataset.

2.2.3 Training Setup

The classification goal in this study is to answer six binary questions for each sample of burning material, namely: is this sample flammable; toxic; explosive; water extinguishable; corrosive and oxidizing. Each hazard category is independent of each other and as a result, we train one machine learning model per category and present their results separately. Since each material in the combustion signature dataset may take different lengths of time to complete burning, we perform zero-padding on the dataset which means that all experiments have been aligned to take the length of the longest experiment by adding zeros to the end of shorter experiments. Using the combustion signature dataset with 30 data samples, we perform a 70-30 train-test split, meaning 70% of the data (or 21 data samples) will be used for training the models, and 30% of the data (or 9 data samples) will be used to test the generalization properties of the trained models.

For the machine learning models tested, which are CIF and ROCKET, we use the implementations from the Sktime library (Markus et al. 2021), which is a Python library for time series analysis. For each model, we use the default hyperparameter values used in the original papers, for example, we use 500 trees for CIF, and 10,000 random convolutional kernels for ROCKET.

For each experiment, we record the training accuracy, test accuracy and more importantly, the time taken to train and test the model. As concluded by Ruiz et al. (2021), we expect the classification accuracies of the three models to be similar while taking drastically different times to train.

3.0 Results and Discussion

Table 2 shows the classification accuracies of ROCKET and CIF, and the time requirements (in seconds) to train each model. The rows show results from each hazard category as well as the average result across all hazard categories. For each row, the superior test accuracy and training time are bolded.

Overall, we observe that all models trained achieved training accuracies of 100% and test accuracies that are less optimal. This indicates overfitting, which means that all models have memorized the training dataset too well which prevented it from generalizing to unseen data points. There exist measures to alleviate the problem of overfitting, including increasing the size of the overall dataset through either data collection or through data augmentation techniques that generate synthetic data through manipulating the existing dataset, or regularizing the model to reduce the complexity of the model. Since the combustion analysis dataset is very small, the ideal solution to alleviate overfitting would be to increase the dataset through both additional data collection and synthetic data generation. We leave the exploration of such tasks as the next step in our future work.

From the test accuracies in individual hazard categories, we observe that ROCKET and CIF tied in three categories: explosivity, water reactiveness, and oxidizing; ROCKET outperformed CIF in one category:

flammability; and CIF outperformed ROCKET in the remaining two categories: toxicity and corrosiveness. As a result, CIF achieved a slightly higher average test accuracy across all hazard categories. However, looking at the training time requirements, we observe that ROCKET is roughly 1,000 times faster than CIF for all hazard categories, taking 3-4 seconds to complete training, while CIF takes about one hour to train.

Overall, we believe that ROCKET is the clear winner, since its test accuracies are comparable to those of CIF, and it is also 1,000 times faster to run. Considering that the current results are obtained from a preliminary 30-sample dataset, we expect the performance of both models to be improved when a larger dataset is available. In addition, CIF would naturally take longer to train with a larger dataset, making ROCKET even more favorable. These results are analogous to those presented by Ruiz et al. (2021), and we once again demonstrate the superiority and speed of ROCKET due to its random convolutional kernels.

Table 2. Results on the combustion analysis dataset from a 70-30 data split.

	ROCKET (SOTA)			Canonical Interval Forest		
Hazard Category	Train Accuracy (%)	Test Accuracy (%)	Train Accuracy (%)	Test Accuracy (%)	Train Accuracy (%)	Test Accuracy (%)
flammability	100	67	100	67	100	67
toxicity	100	56	100	56	100	56
explosivity	100	89	100	89	100	89
water reactiveness	100	67	100	67	100	67
Corrosiveness	100	78	100	78	100	78
Oxidizing	100	100	100	100	100	100
Average Performance						

4.0 Conclusions

In this work, we presented a review of applications of machine learning in fire sciences and gas classification. We presented the combustion signature dataset collected using the FTIR and cone calorimeter and used this dataset to demonstrate that ML can be applied to the characterization of freight fire hazards using two existing implementations of multivariate time series classification methods. However, in this study we have been limited by the small scale of the dataset, with only 30 data samples available. All data were obtained in the laboratory, indicating a need to test the models developed using real-world data before adoption. We have also been limited by a lack of repetition in experiments used. Previously, published classification tasks in related fields such as fire detection or gas classification typically only collected data on a few distinct materials and would run tests on these limited materials for a large number of times, for example, 1000 times per material. However, in our case, we only tested 20 distinct materials and collected a total of 30 data samples, meaning that most materials were only burned once, and no repetitions of tests are available. Since this is a first attempt at using machine learning models on fire hazard detection, using such a complicated and small dataset limited the performance of the machine learning models tested. As a result, the most imminent work is to collect more data in the laboratory and also explore data augmentation methods. As a future step, we will also develop our own multivariate time series classification method for the combustion analysis dataset and to contribute to the research community. These tasks will aid us in eventually applying our designed ML model trained on an augmented, large-scale dataset, to real freight fire scenarios and provide decision support to first responders when they encounter unlabeled, burning goods.

5.0 Acknowledgement

This project was supported by collaborative research funding from the National Research Council of Canada's Artificial Intelligence for Logistics Program. This project was supported by the Vector Scholarship in Artificial Intelligence, provided through the Vector Institute.

References

Chen, Y., Serio, M. & Sathyamoorthy, S. (2000), 'Development of a fire detection system using ft-ir spectroscopy and artificial neural networks', Fire Safety Science 6, 791–802.

Dempster, A., Petitjean, F. & Webb, G. I. (2019), 'ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels', CoRR abs/1910.13051. URL: http://arxiv.org/abs/1910.13051.

Feng, S., Farha, F., Li, Q., Wan, Y., Xu, Y., Zhang, T. & Ning, H. (2019), 'Review on smart gas sensing technology', Sensors 19(17). URL: https://www.mdpi.com/1424-8220/19/17/3760.

Fonollosa, J., Sol´orzano, A. & Marco, S. (2018), 'Chemical sensor systems and associated algorithms for fire detection: A review', Sensors 18(2). URL: https://www.mdpi.com/1424-8220/18/2/553.

Han, L., Yu, C., Xiao, K. & Zhao, X. (2019), 'A new method of mixed gas identification based on a convolutional neural network for time series classification', Sensors 19(9). URL: https://www.mdpi.com/1424-8220/19/9/1960.

Markus Löning, Tony Bagnall, Matthew Middlehurst, Sajaysurya Ganesh, George Oastler, Jason Lines, ViktorKaz, Franz Király, Martin Walter, Patrick Rockenschaub, jesellier, Taiwo Owoseni, Lovkush, AidenRushbrooke, oleskiewicz, Yi-Xuan Xu, Patrick Schäfer, RNKuhns, Hongyi, … Ayushmaan Seth. (2021). alan-turing-institute/sktime: v0.7.0 (v0.7.0). Zenodo. https://doi.org/10.5281/zenodo.5093663.

Middlehurst M, Large J, Bagnall A (2020). The canonical interval forest (CIF) classifier for time series classification. In: Proceedings of the IEEE international conference on big data.

Park, J. H., Lee, S., Yun, S., Kim, H. & Kim, W.-T. (2019), 'Dependable fire detection system with multifunctional artificial intelligence framework', Sensors 19(9). URL: https://www.mdpi.com/1424-8220/19/9/2025.

Peng, P., Zhao, X., Pan, X. & Ye, W. (2018), 'Gas classification using deep convolutional neural networks', Sensors 18(1). URL: https://www.mdpi.com/1424-8220/18/1/157.

Ruiz, A. P., Flynn, M., Large, J., Middlehurst, M. & Bagnall, A. (2021), 'The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances', Data Mining and Knowledge Discovery 35, 401–449.

Titus, Deena, E. James Jebaseelan Samuel, Selvaraj Mohana Roopan (2019). Chapter 12 nanoparticle characterization techniques. Ashutosh Kumar Shukla, Siavash Iravani, eds., Green

Synthesis, Characterization and Applications of Nanoparticles. Micro and Nano Technologies, Elsevier, 303–319. doi:https://doi.org/10.1016/B978-0-08-102579-6.00012-5. URL https://www.sciencedirect.com/science/article/pii/B9780081025796000125.

Zhao, X., Wen, Z., Pan, X., Ye, W. & Bermak, A. (2019), ‘Mixture gases classification based on multi-label one-dimensional deep convolutional neural network’, IEEE Access 7, 12630–12637.

Dataset for 3D Printer Process Modelling

Katie Xu

Introduction

3D printing is a class of fabrication methods for making customized 3D objects. Generally, the target geometry is divided into a series of 2D profiles (layers) and objects are built up layer-by-layer. Fused deposition modeling (FDM) is a type of 3D printing where layers are formed by melting a thermoplastic material and depositing it in the desired locations where it solidifies and becomes part of the object. 3D printing has important advantages over traditional manufacturing processes such as the ability to fabricate complex geometries, reduced waste material, and lower tooling costs. However, the need for part-specific parameter tuning and the influence of external disturbances lead can lead to high rejection rates. This limits the usefulness of 3D printing in many practical applications. The goal of this project is to develop a closed-loop system to monitor and control the quality of parts made using a FDM 3D printer.

Over the past 6 months, efforts for this project have been focused on collecting data about physical system. This includes designing the dataset as well as setting up the hardware and software necessary for data collection. The resulting dataset will provide insight into the dynamics of the system, which is essential for achieving effective closed-loop control.

The project is in collaboration with Professor Zou and his students from the Department of Material Science and Engineering.

Dataset Design

The physical system, described in detail in the next section, consists of a FDM 3D printer with a camera situated directly above the print bed, pointing down. The system is capable of taking a top-down picture of the part after each layer has been printed. Using this system, we are able to monitor the progress of the print with a sampling frequency of once per layer.

Requirements

In order for the dataset to be useful in developing a closed-loop controller, it must contain information about the following two things:

1. Layer-wise transition dynamics. This is how the state of the system changes over time, as well as how state changes are influenced by different inputs. This requirement informs the selection of inputs for the dataset.

2. A reward process. This is for evaluating or predicting the quality of a print, and is necessary in order to make a controller aimed at maximizing print quality. This requirement informs the labelling system to be used for this dataset as well as the geometry of the samples to be printed.

Input selection

Input selection is of critical importance for creating a useful dataset because it is the only way we can influence the distribution of the data contents. For this dataset, the inputs are the process parameters. Specifically, print speed and flow compensation are changed while all others are kept constant. Print speed, in mm/s, is the speed of the print head during printing. Flow compensation, in %, changes the rate at which material is pushed through the nozzle.

It is important to select parameters which produce a wide variety of state transitions, as well as overall part qualities. In order to observe a variety of state transitions, 270 samples with randomized input trajectories will be produced. Specifically, the print speed and flow compensation will be chosen randomly for each layer. Speed is sampled uniformly from the set of speeds between 50 mm/s and 175 mm/s at 5 mm/s intervals. Flow compensation is sampled also uniformly, between 50% and 150% at 5% intervals. To observe a wide variety of part qualities, some samples will also be made with constant parameters throughout all layers. This is done three times for all combinations of speeds (mm/s) in {50, 75, 100, 125, 150, 175} and flow compensations (%) in {5, 75, 100, 125, 150} for a total of 90 samples. These samples ensure variety of overall quality because they include the extremities of the parameter space.

Finally, it is important to note that notwithstanding the above, parameters for the first layer are always kept constant at 20mm/s print speed and 100% flow compensation. This is to ensure the success of the first layer which serves as a foundation for the rest of the sample, and also helps control against external factors such as operator error, sensor drift, or calibration issues.

Labels

For this project, we are defining quality mainly in terms of the material properties of the final part – specifically, the ultimate tensile strength (UTS). Because we are interested in the material quality, all samples will be geometrically identical tensile specimens. In addition to UTS measurements, the quality of each image and part will be assessed qualitatively and assigned scores based on several criteria. The reason for this is twofold. First, UTS measurements are quite sparse – only one per sample – and it is desirable to have a denser reward signal if possible. Second, samples with very high UTS values often have print artifacts as well, which would render it unusable in practice. The specific grading scheme is still being finalized but some potential criteria include the presence of print artifacts, geometric distortion, the type of failure during the tensile test (brittle/ductile/failed before test even started), and the level of over- or under-extrusion.

Summary

To summarize, the dataset will consist of 360 samples with different print speeds and flow compensation percentages: 270 samples will be made with randomized input trajectories and 90 will be made with constant input trajectories. For each sample, the UTS will be measured and quality scores will be assigned. Furthermore, images of each layer will be captured and assigned quality scores.

System Overview

The data collection system consists of a Ender 3 v2 3D printer, a Raspberry Pi 3 Model B+), and a Raspberry Pi HQ Camera with a 6mm wide angle lens. The camera and the Raspberry Pi are mounted onto the frame of the printer as shown in Figure 1. The flow of information between these a parts is shown in Figure 2.

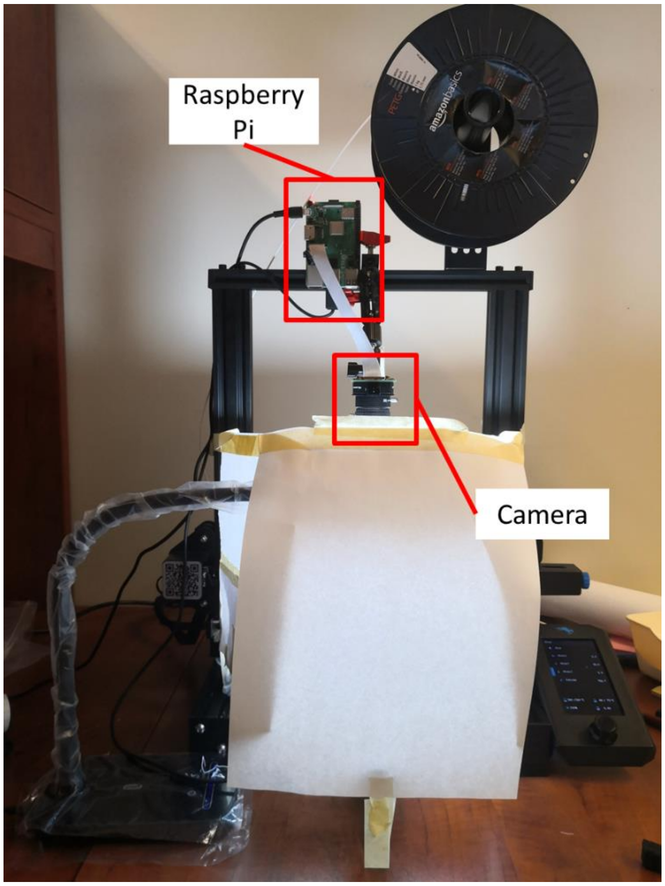
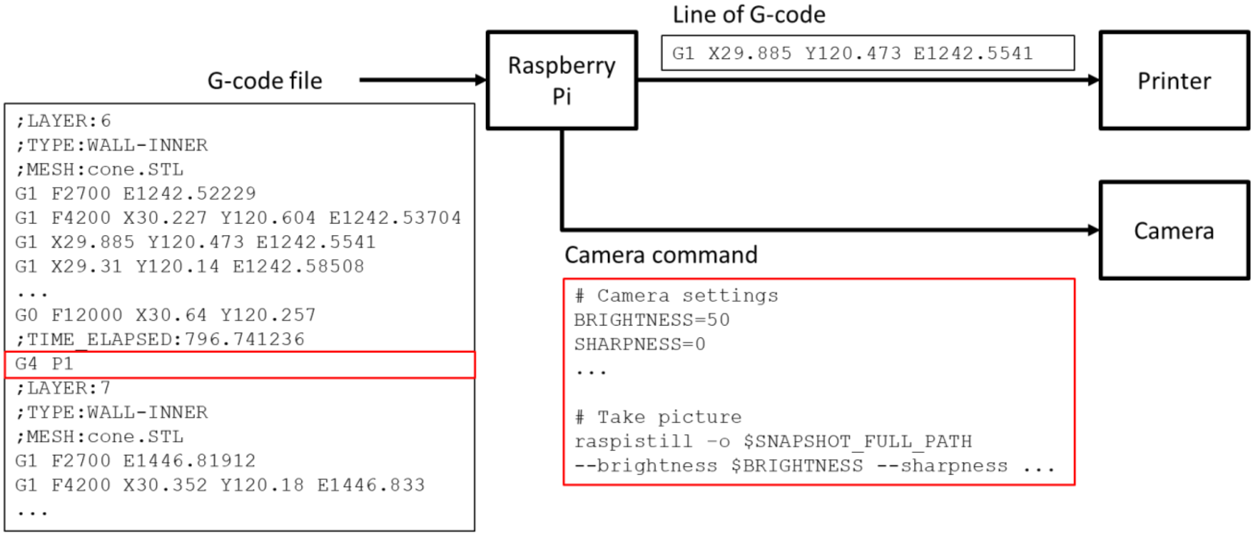


Figure 1: Physical system



```
G1 X29.885 Y120.473 E1242.5541
```

```
;LAYER:6
;TYPE:WALL-INNER
;MESH:cone.STL
G1 F2700 E1242.52229
G1 F4200 X30.227 Y120.604 E1242.53704
G1 X29.885 Y120.473 E1242.5541
G1 X29.31 Y120.14 E1242.58508
...
G0 F12000 X30.64 Y120.257
;TIME_ELAPSED:796.741236
G4 P1
;LAYER:7
;TYPE:WALL-INNER
;MESH:cone.STL
G1 F2700 E1446.81912
G1 F4200 X30.352 Y120.18 E1446.833
...
```

```
# Camera settings
BRIGHTNESS=50
SHARPNESS=0
...

# Take picture
raspistill -o $SNAPSHOT_FULL_PATH
--brightness $BRIGHTNESS --sharpness ...
```

Figure 2: Flow of information

G-code is a text file with a series of low-level commands for the printer, and it is generated offline for each sample. The G-code files are then uploaded to the Raspberry Pi, which reads it line by line and either forwards the command to the printer or triggers the camera sequence when it encounters a special G4 P1 command. Details of each part of this procedure are presented in the following sections.

Making the G-code

Cura is used to generate the G-code for parts with constant input trajectories. A different G-code file is for each parameter set.

For parts with randomized input trajectories, Cura is not capable of parameters separately for every layer. Instead, this is accomplished by splicing G-code from different parts together. For example, suppose that a given part requires the following parameters: layer 1 speed = 20mm/s, layer 1 flow = 100%, layer 2 speed = 100 mm/s, and layer 2 flow = 120%. Then the G-code for this part is constructed by combining the first layer from the G-code file with speed = 20 mm/s and flow = 100%, together with the second layer from a separate G-code file with speed = 100 mm/s and flow = 120%.

Printing the parts

The Raspberry Pi is responsible for reading the uploaded G-code file and forwarding commands to the printer. This is managed by a program called OctoPrint which handles communications between the Pi and the printer, as well as to a graphical user interface which can be accessed remotely.

The material used for printing is PETG (polyethylene terephthalate glycol).

Taking the pictures

In addition to forwarding G-code commands to the printer, the Raspberry Pi is also responsible for coordinating the printing with image capturing. This means pausing the print after each layer, moving the part into the frame of the camera, and triggering the camera before resuming the print. This occurs whenever the G4 P1 command is encountered, and is handled by a plugin for OctoPrint called Octolapse.

Preliminary Image Processing

The raw image captured using the system described in the previous section is shown in Figure 3.

From this image, a few steps are taken to extract the region of interest around the tensile specimen. Ideally, this could be accomplished by cropping at a fixed location for all images. However due to the location of the camera drifting slightly over time, this does not work well. Instead, the location of the tensile specimen and rotation of the image are determined automatically for each part using the first layer image. This is accomplished through the following steps:

- Preprocessing
- Edge detection
- Contour detection – to locate the specimen
- Line detection – to detect the angle of rotation

This procedure is implemented in Python, using OpenCV. Each step is described in the corresponding section below.

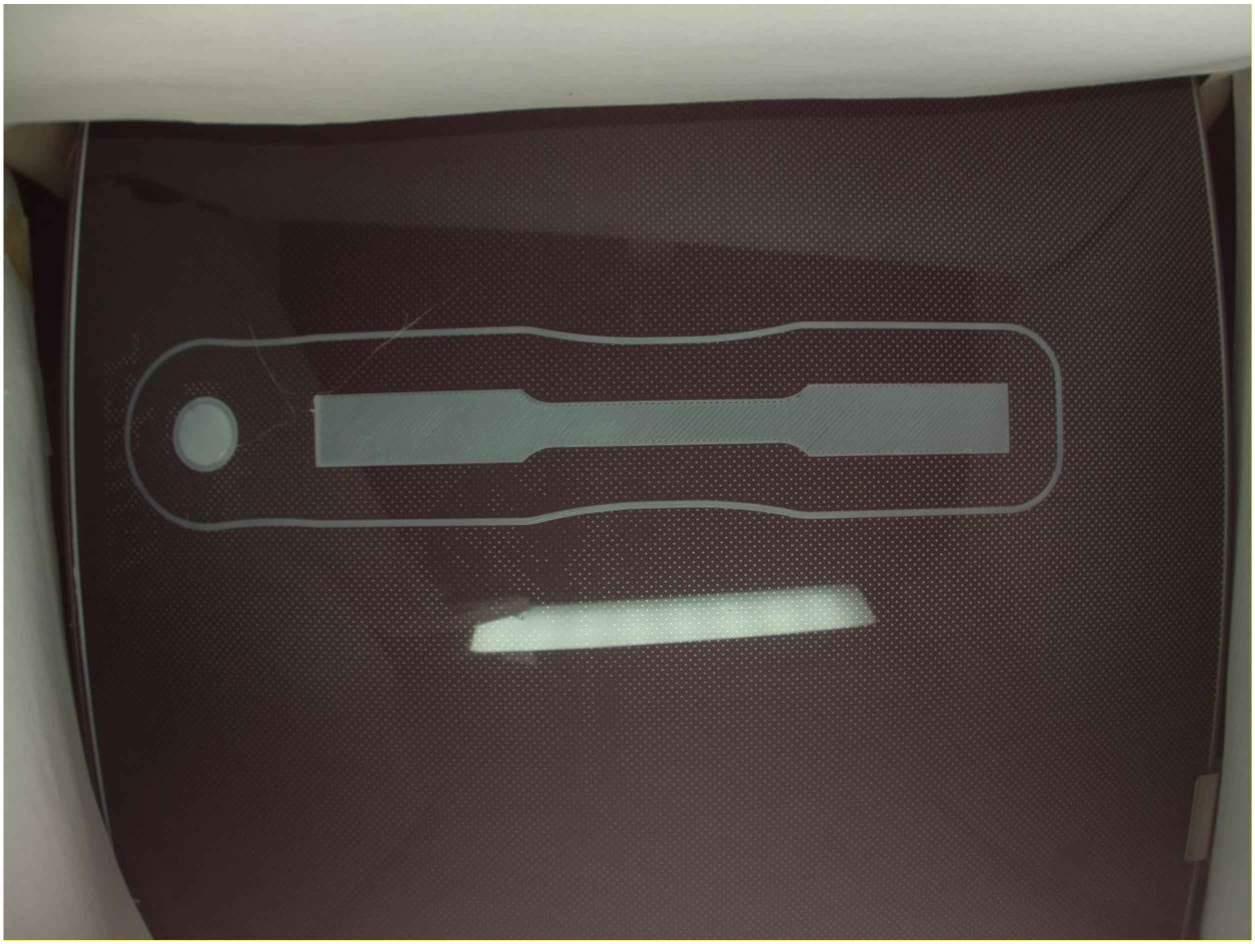


Figure 3: Raw image

Preprocessing

First, the image is roughly cropped, scaled down, converted to greyscale, blurred, and the grey levels in the image were normalized. The result of this step is shown in Figure 4. The reason for this preprocessing procedure is to ensure effective edge detection.

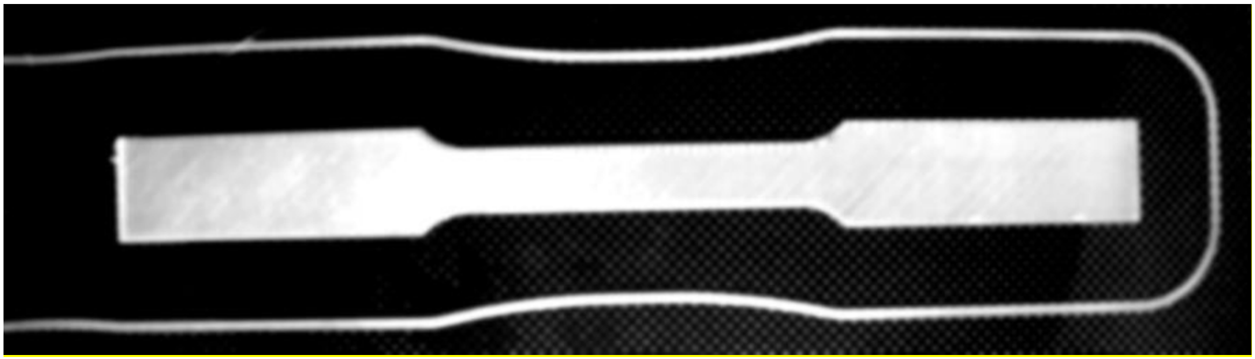


Figure 4: Preprocessing result

Edge detection

Canny edge detection is used to extract edges from the preprocessed image. This method involves applying Sobel filters to compute pixel gradients in both the X and Y directions followed by non-maxima suppression and hysteresis thresholding to retain only “strong” edges.

The result of this procedure is fairly good but the detected edges sometimes have small discontinuities which can create problems in the contour detection stage. To remedy this issue, morphological transformations are applied to close small gaps. The result of the edge detection and closing operation is shown in Figure 5.

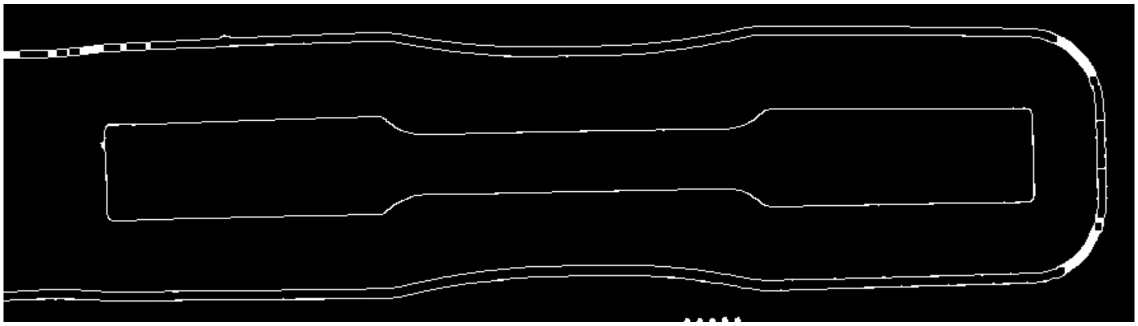
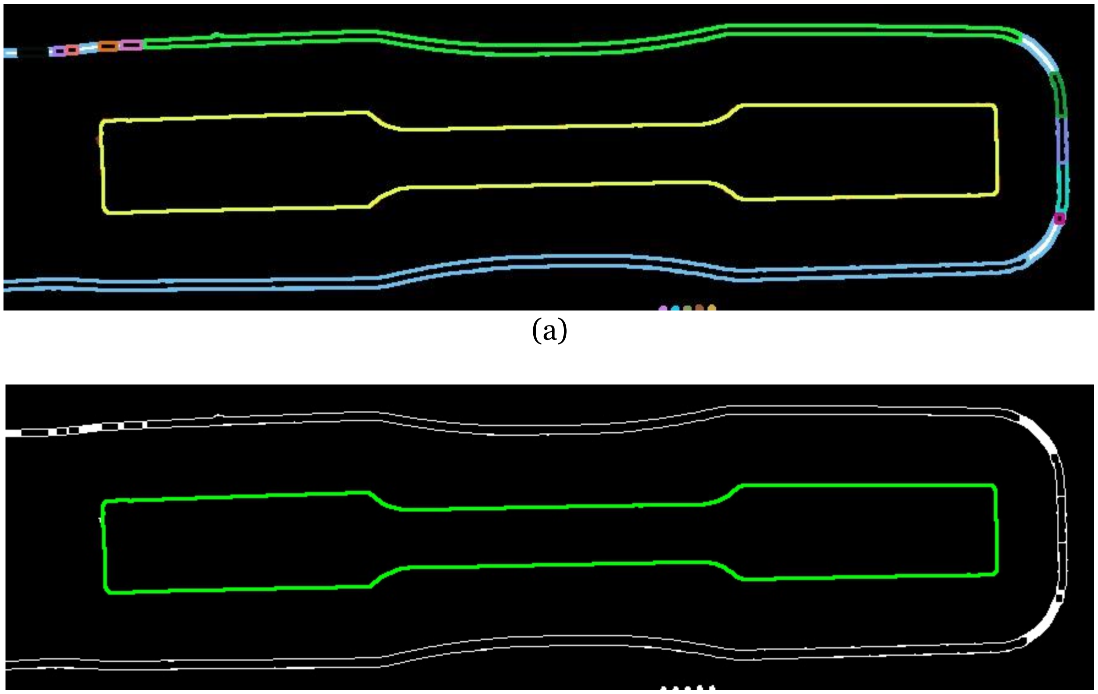


Figure 5: Closed edge image

It is essential that edge detection is performed well because the following two steps operate based on the resulting edge image.

Contour detection

Following edge detection, Suzuki and Abe's [1] border following algorithm for contour detection was used to extract closed contours from the edge image. This results in many contours being found, as shown in Figure 6a. To select the contour corresponding to the tensile specimen, the areas of all contours are computed and the one whose area is closest to the nominal area of the tensile specimen is selected (Figure 6b). The nominal area is determined by manually measuring one of the images. Finally, the centroid of the selected contour is found. This will be the center of the cropping box.



(a)

(b)

Figure 6: Contour detection result (a) all contours (b) selected contour

Line detection

Returning again to the output of edge detection, straight lines in the image are found using the Hough transform. Roughly speaking, this involves iterating through the space of all possible lines and selecting the ones which pass by a sufficient number of edge pixels. Following the Hough transform, lines which are approximately horizontal (±5°) are kept and all others are discarded. Figure 7 shows the detected lines for a sample image. The average angle of these lines is used to compute the rotation of the image.

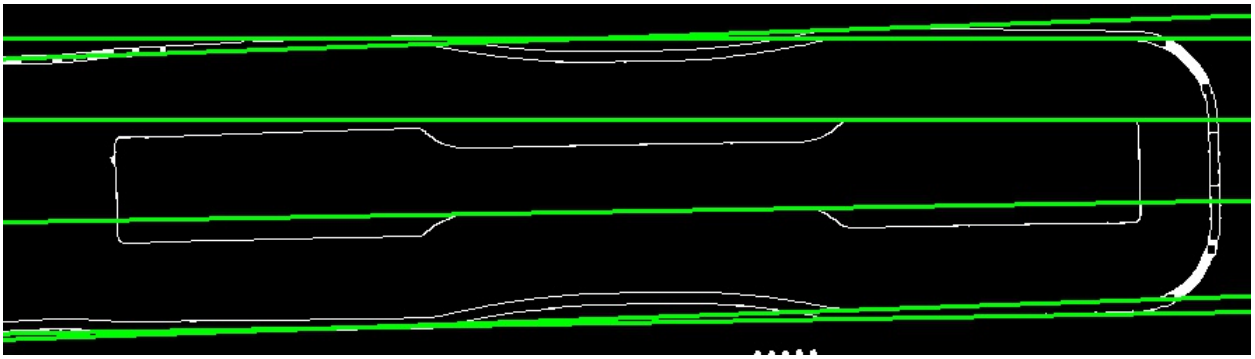


Figure 7: Hough lines

Cropping and rotation

Finally, the image is rotated through the angle found in the line detection step and cropped to a box of fixed size centered around the centroid found in the contour detection step. The result is shown in Figure 8.

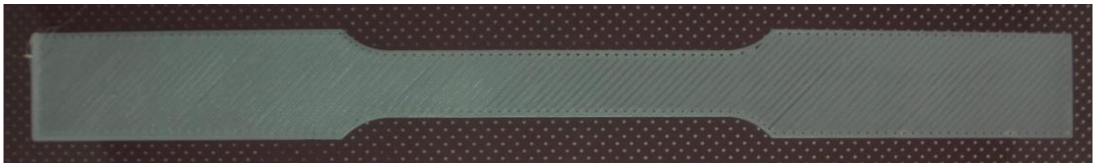


Figure 8: Cropped image

Limitations

For this image processing procedure to effectively locate the part, parameters must be tuned carefully for the dataset. This means that relatively small changes to the physical system may render the current procedure unusable. Moreover, parameter tuning is not very intuitive so significant trial and error could be involved.

For more robust specimen localization, alternate approaches such as object detection could be applied. However, this is not done for this project because these processing steps are strictly for data cleaning purposes, as an alternative to manual cropping. The resources necessary to build an object detector and to create the labels needed for training would exceed that which is needed for fully manual cropping. In practice, robust localization may indeed be required, at which point further work should be done to either improve or replace this procedure.

Conclusion

This report outlines the methodology for creating a dataset suitable for modelling the dynamics of a FDM 3D printing process. The dataset will consist of 360 tensile specimens. For each sample, the UTS will be measured and it will evaluated on several quality criterial. Throughout the printing process, top-down images of each layer will be captured and the quality of the layer will be scored based on the image. Finally, images will be cropped to the region of interest containing the tensile specimen. This is done by detecting the location and rotation of the tensile specimen in the image.

Of the 360 samples, 270 of them will be printed with randomized input trajectories, with the parameters for each layer being sampled uniformly from the parameter space. The remaining 90 samples will be printed with constant input trajectories with parameter values fixed at evenly spaced intervals in the parameter space.

References

[1] S. Suzuki and K. be, “Topological structural analysis of digitized binary images by border following,” *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 1, pp. 32–46, 1985. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0734189X85900167

Unsupervised few-shot learning

Kuilin Chen

Introduction

Few-shot learning (Fei-Fei et al., 2006) aims to learn a new classification or regression model on a novel task which is not seen during training, given only a few examples in the novel task. Meta-learning (Finn et al., 2017; Snell et al., 2017) is a popular approach for few-shot learning by training the model with episodes of few-shot samples to mimic the test setting. However, existing few-shot learning methods are not ”true” few-shot learning because they still require a lot of labeled data for pre-training or meta-training. Recently, several unsupervised meta-learning approaches have attempted addressed this problem, by constructing episodes via pseudo-labeling (Hsu et al., 2019) or augmenting images (Khodadadeh et al., 2019). However, the performance of unsupervised meta-learning approaches is still far from their supervised counterparts.

Recent works (Chen et al., 2019; Tian et al., 2020b) show that vanilla transfer learning outperforms meta-learning methods on a wide range of few-shot learning benchmarks. It demonstrates that it is possible to learn transferable representation from unlabeled meta-training data without using episodic meta-training. Therefore, we develop a self-supervised learning method that learns representation from unlabeled meta-training data to achieve comparable performance of SOTA supervised few-shot learning methods.

Methodology

Given unlabeled meta-training data, we treat each sample as a class. For example, if we have K training samples in the meta-training dataset, we train a K-way classification model on the meta-training data. The model is trained by minimizing the following loss function

$$\begin{aligned}\mathcal{L} &= -\frac{1}{n}\sum_{i=1}^{n}\log\frac{\exp(\mathrm{sim}\left(\mathbf{z}_i,\mathbf{p}_i\right)/\tau)}{\sum_{j=0,j\neq i}^{K}\exp(\mathrm{sim}\left(\mathbf{z}_i,\mathbf{p}_j\right)/\tau)}\\ &= \underbrace{-\frac{1}{n}\sum_{i=1}^{n}\frac{1}{\tau}\mathrm{sim}\left(\mathbf{z}_i,\mathbf{p}_i\right)}_{\text{compactness}}\underbrace{+\frac{1}{n}\sum_{i=1}^{n}\log\sum_{j=0,j\neq i}^{K}\exp(\mathrm{sim}\left(\mathbf{z}_i,\mathbf{p}_j\right)/\tau)}_{\text{uniformity}}\end{aligned}$$

where $\mathbf{z} = \boldsymbol{f}(\mathbf{x})$ is the encoded features of augmented input x, and $\mathbf{p}_{\boldsymbol{i}}$ is the class template for class i. We use Gaussian Blur, color jittering, random crop to augment the original images so that the model can learn invariant features of unlabeled data.

In addition, we also mimic the unseen meta-test data via mixing up the meta-training samples. Mixing up the image pixels does not result in realistic images. Inspired by generative models, we mix up intermediate features of an image from the encoder so that the mixed-up image looks more realistic.

Another challenge of few-shot learning is that the classifier trained from few-shot training samples could be very biased because the few-shot training data does not well represent the true training data distribution. Therefore, we also add noise to the few-shot training samples to make the few-shot training samples better represent the training data distribution. As a result, the final classifier is less biased and achieves strong performance on the test data.

Results

We conduct few-shot classification experiments on two widely used few-shot image recognition benchmarks: miniImageNet, and FC100. We do not use the labels in the meta-training dataset.

miniImageNet is a 100-class subset of the original ImageNet dataset (Deng et al., 2009) for few-shot learning (Vinyals et al., 2016). Each class contains 600 images in RGB format of the size 84 × 84. miniImageNet is split into 64 training classes, 16 validation classes and 20 testing classes, following the widely used data splitting protocol (Ravi and Larochelle, 2017).

FC100 is another derivative of CIFAR-100 with minimized overlapped information between train classes and test classes by grouping the 100 classes into 20 superclasses. They are further split into 60 training classes (12 superclasses), 20 validation classes (4 superclasses) and 20 test classes (4 superclasses).

Method	Backbone	miniImageNet 5-way	
		1-shot	5-shot
Supervised			
Matching Net (Vinyals et al., 2016)	ResNet-12	63.08 ± 0.80	75.99 ± 0.60
Proto Net (Snell et al., 2017)	ResNet-12	60.37 ± 0.83	78.02 ± 0.57
MAML (Finn et al., 2017)	ResNet-12	56.58 ± 1.84	70.85 ± 0.91
AdaResNet (Munkhdalai et al., 2018)	ResNet-12	56.88 ± 0.62	71.94 ± 0.57
TADAM (Oreshkin et al., 2018)	ResNet-12	58.50 ± 0.30	76.70 ± 0.30
Baseline++ (Chen et al., 2019)	ResNet-12	60.83 ± 0.81	77.81 ± 0.76
TapNet (Yoon et al., 2019)	ResNet-12	61.65 ± 0.15	76.36 ± 0.10
Variational FSL (Zhang et al., 2019)	ResNet-12	61.23 ± 0.23	77.69 ± 0.17
MetaOptNet (Lee et al., 2019)	ResNet-12	62.64 ± 0.61	78.63 ± 0.46
Ensemble (Dvornik et al., 2019)	ResNet-18	59.48 ± 0.65	75.62 ± 0.42
DSN (Simon et al., 2020)	ResNet-12	62.64 ± 0.66	78.83 ± 0.45
DKT (Patacchiola et al., 2020)	ResNet-12	61.29 ± 0.57	76.25 ± 0.51
Self-supervised			
SimCLR (Chen et al., 2020a)	ResNet-12	55.76 ± 0.88	75.59 ± 0.69
MoCo v2 (He et al., 2020)	ResNet-12	57.73 ± 0.84	77.51 ± 0.63
BYOL (Grill et al., 2020)	ResNet-12	56.17 ± 0.89	76.17 ± 0.66
Ours w.o. mixup	ResNet-12	59.21 ± 0.85	78.79 ± 0.57
Ours w.o. fake samples	ResNet-12	60.44 ± 0.85	80.34 ± 0.58
Ours	ResNet-12	$\mathbf{61.26 \pm 0.81}$	$\mathbf{80.89 \pm 0.59}$

Method	Backbone	FC100 5-way	
		1-shot	5-shot
Supervised			
Proto Net (Snell et al., 2017)	ResNet-12	37.5 ± 0.6	52.5 ± 0.6
TADAM (Oreshkin et al., 2018)	ResNet-12	40.1 ± 0.4	56.1 ± 0.4
Baseline++ (Chen et al., 2019)	ResNet-12	43.1 ± 0.7	55.7 ± 0.7
MetaOptNet (Lee et al., 2019)	ResNet-12	41.1 ± 0.6	55.5 ± 0.6
MTL (Sun et al., 2019)	ResNet-12	45.1 ± 1.8	57.6 ± 0.9
MABAS (Kim et al., 2020)	ResNet-12	42.3 ± 0.7	58.1 ± 0.7
Fine-tune (Dhillon et al., 2020)	WRN-28-10	38.2 ± 0.5	57.2 ± 0.6
Self-supervised			
SimCLR (Chen et al., 2020a)	ResNet-12	36.2 ± 0.7	49.9 ± 0.7
MoCo v2 (He et al., 2020)	ResNet-12	37.7 ± 0.7	53.2 ± 0.7
BYOL (Grill et al., 2020)	ResNet-12	37.2 ± 0.5	52.8 ± 0.6
Ours w.o. mix-up	ResNet-12	39.7 ± 0.7	57.5 ± 0.7
Ours w.o. fake samples	ResNet-12	42.6 ± 0.7	59.7 ± 0.7
Ours	ResNet-12	**43.3 ± 0.7**	**60.0 ± 0.7**

Our results indicate that our method can achieve comparable results of supervised few-shot learning methods, while our method does not use the label in the meta-training dataset. In addition, our method achieves stronger results than existing self-supervised learning methods, such as SimCLR, MoCo v2 and BYOL, because existing self-supervised learning methods are not designed for few-shot learning. Our method mimic the unseen meta-test data via mixing up meta-training samples. That's why our method outperforms existing self-supervised learning algorithms.

References

Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your maml. In International Conference on Learning Representations, 2018.

Ben Athiwaratkun, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. There are many consistent explanations of unlabeled data: Why you should average. In International Conference on Learning Representations, 2019.

Edmon Begoli, Tanmoy Bhattacharya, and Dimitri Kusnezov. The need for uncertainty quantification in machine-assisted medical decision making. Nature Machine Intelligence, 1(1):20{23, 2019.

Y Bengio, S Bengio, and J Cloutier. Learning a synaptic learning rule. In IJCNN-91-Seattle International Joint Conference on Neural Networks, volume 2, pages 969 vol. IEEE, 1991.

Luca Bertinetto, Joao F Henriques, Philip Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In International Conference on Learning Representations, 2019.

Glenn W Brier. Varification of forecasts expressed in terms of probability. Monthly weather review, 78(1):13, 1950.

Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In International Conference on Learning Representations, 2019.

Morris H DeGroot and Stephen E Fienberg. The comparison and evaluation of forecasters. Journal of the Royal Statistical Society: Series D (The Statistician), 32(1-2):1222, 1983.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A largescale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pages 248-255. Ieee, 2009.

Guneet Singh Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. In International Conference on Learning Representations, 2020.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In International Conference on Learning Representations, 2021.

Jan Drugowitsch. Variational bayesian inference for linear and logistic regression. stat, 1050:16, 2014.

Nikita Dvornik, Cordelia Schmid, and Julien Mairal. Diversity with cooperation: Ensemble methods for few-shot classification. In Proceedings of the IEEE International Conference on Computer Vision, pages 3723-3731, 2019.

Nanyi Fei, Zhiwu Lu, Tao Xiang, and Songfang Huang. fMELRg: Meta-learning via modeling episode-level relationships for few-shot learning. In International Conference on Learning Representations, 2021.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Proceedings of the 34th International Conference on Machine Learning-Volume 70, pages 1126{1135. JMLR. org, 2017.

Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, pages 9537-9548, 2018.

Sebastian Flennerhag, Andrei A Rusu, Razvan Pascanu, Francesco Visin, Hujun Yin, and Raia Hadsell. Meta-learning with warped gradient descent. In International Conference on Learning Representations, 2020.

Victor Garcia and Joan Bruna Estrach. Few-shot learning with graph neural networks. In 6th International Conference on Learning Representations, ICLR 2018, 2018.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry Vetrov, and Andrew Gordon Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. Advances in Neural Information Processing Systems, 2018:8789{8798, 2018.

Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4367-4375, 2018.

Micah Goldblum, Steven Reich, Liam Fowl, Renkun Ni, Valeriia Cherepanova, and Tom Goldstein. Unraveling meta-learning: Understanding feature representations for few-shot tasks. In Hal Daume III and Aarti Singh, editors, Proceedings of the 37th International Conference on Machine Learning, volume 119 of Proceedings of Machine Learning Research, pages 3607{3616, Virtual, 13-18 Jul 2020. PMLR.

Shaogang Gong, Stephen McKenna, and John J Collins. An investigation into face pose distributions. In Proceedings of the Second International Conference on Automatic Face and Gesture Recognition, pages 265{270. IEEE, 1996.

J Gordon, J Bronskill, M Bauer, S Nowozin, and RE Turner. Meta-learning probabilistic inference for prediction. In International Conference on Learning Representations (ICLR 2019). OpenReview. net, 2019.

Erin Grant, Chelsea Finn, Sergey Levine, Trevor Darrell, and Thomas Griths. Recasting gradient-based meta-learning as hierarchical bayes. In International Conference on Learning Representations, 2018.

UNIVERSITY OF TORONTO
CENTRE FOR MAINTENANCE OPTIMIZATION AND RELIABILITY ENGINEERING (C-MORE)

STNG Project: Engine Failure Prediction using Model Ensemble

Prepared by:
JC Seok

Supervisor:
Professor Chi-Guhn Lee
Dr. Janet Lam
Dr. Dragan Banjevic
Mohamed Abubakr Hassan

Jul 2021

1.0 Problem Statement

Data-driven techniques, especially artificially intelligence (AI) have been getting huge amount of attention in the manufacturing and production sector, due to their ability to model highly nonlinear, complex, and multi-dimensional systems. Tremendous research of deep learning techniques has been started to be applied in machine health monitoring and on the Remaining Useful Life (RUL) prediction. Precise RUL prediction can significantly improve the reliability and operational safety, avoid fatal breakdown, and reduce cost overall. Cummins and its client, STNG, proposed a project on investigating the remaining useful life and predict failures in the N – timeframe of the engines that are used in the mining site. The project explores data-driven techniques for developing a RUL prediction model and identifying decision boundaries for determining the critical state of the engines.

2.0 Related Work

There have been many state-of-the-art applying data driven approaches to the C-MAPSS Dataset for prediction of RUL [1].

Huang et al. [2] implemented an MLP approach for modeling the remaining useful life of the laboratory-tested bearings and received a superior performance compared to the reliability-based approaches. The ANN model developed took measurement values at present and previous as inputs and got the equipment life as percentage as output.

Recurrent neural networks (RNN) are commonly used for problems involving time series data, because of their ability to process and pass information over time. The authors of [3] used multi-layer LSTM followed by a feed forward neural network to map the features from LSTM to the predicted RUL.

Additionally, MLP-LSTM-MLP structure has been proposed from the authors of [4], where they use MLP layers initially to handle complex feature processing, use LSTM for capturing the temporal dependencies and finally MLP for prediction using temporally smoothed data.

Convolutional neural networks are often used for dealing time series data due to their ability to model correlations in a temporal window. The authors of [5] proposed a deep learning CNN architecture that contain multiple CNN layers followed by a fully connector layers to improve the performance.

To combine the advantages of LSTM and CNN, a hybrid architecture is proposed in [6], developing CNN layer and deep LSTM and a followed by a fully connected layer.

Classical deep learning algorithms tend to encounter the vanishing/exploding gradient problem found in ANN with gradient-based learning methods and backpropagation. To overcome the problem of vanishing gradient problem, a new residual CNN (ResCNN) model is proposed for RUL estimation in [7].

2.0 Data Description

The provided engine data from STNG contains four different categories which are as follows:

Table 1 - Provided Dataset Breakdowns

Description	Count
In-Operation Engines	105
Failed Engines	39
Passed Engines	9
Infant Mortality	7

The in-operation engines represent engines that have not yet failed and are in-operation. Hence, these engines cannot be used to predict the RUL as the target variables are currently unknown. The failed engines represent engines that have failed before the service life of 18000 hours, and passed engines represent engines that have passed or operated over the service life of 18000 hours. Lastly, the infant mortality represents engines that have failed before the operational hours of 5000. For model development, the only data that could be utilized are “Failed Engines” as the target variable of RUL is known for ones that have failed. The target variable of RUL is also known for engines classified as “Infant Mortality”, but are excluded in the analysis, as they represent outliers in the failed engine sets and the data does not exhibit relationships with failures, making the prediction poor.

Once the engines reach its service life of 18000 hours or failed, the engines are repaired and replaced. Afterwards, the repaired engines are put back to another vehicle for operation. This process repeats for minimum of 3 cycles until the engines are unsalvageable.

There are two primary datasets which are readings from sensor such as temperature and pressure of the engine, derived in real time, and the oil data that contains features such as masses of different elements in the engine oil, extracted periodically.

The sensor dataset has 221 readings / features, however, most of the sensors and features such as signals in bit, do not provide useful information for RUL prediction. The 24 features are selected amongst the 221 readings. The temporal variable for the dataset is the operational hours.

The oil dataset contains 31 features of different element masses that represent additive, coolant leak, wear, and contamination. The oil change occurs every 480 hours of operation in average and is performed throughout the engine life cycle. Instead of using the operational hours as the temporal variable, we use the oil change cycles to achieve the consistent sampling rate between data points.

Additionally, few general datasets were provided which are as follows:

1. Engines Hour – The history of all engines including its engine id, site, operational hour before failure, failure date, start date, and vehicle number
2. Engine Operational Hour –The actual operational hour of each engine per day
3. Failure Mode – Failure mode for engines that were provided

2.1 General EDA (Exploratory Data Analysis)

The Figure 1 shows the operational hours at failure of the engines the datasets were provided for. The graph shows bimodal distribution, which shows two peaks at 5000 – 6000 hours and another peak at 11000 – 13000 hours. The common practice is to split the distribution and train separately, but when we analyze historic operational hours before failures of all engines, the distribution is no longer bimodal, but slightly left skewed.

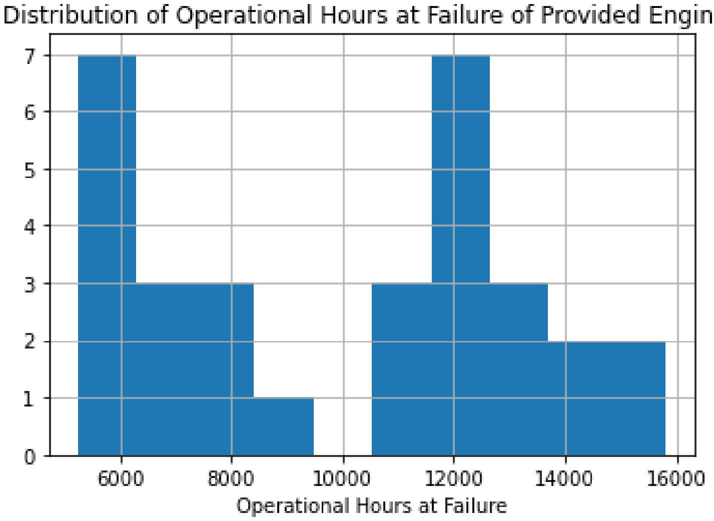


Figure 1 - Operational hours at failure for provided engines

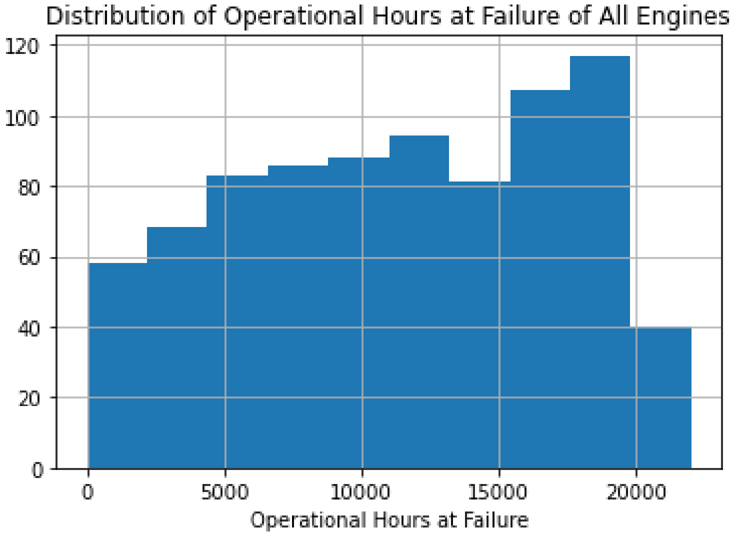


Figure 2 - Operational hours of all engines

As mentioned in the previous section, after engine failure or upon reaching its service life, these engines get repaired and get used on another vehicle. The figure below shows the average operational hours per cycle for engines with different accumulated hours (i.e., sum of operation hours of previous life cycle, zero if the engine is new and was never overhauled). As the historic hours of the engine increase, the average operational hours before failure significantly drop. Even after replacement of major parts, we expect the engines to have degradation, resulting in failures earlier than its previous cycle. In the model implementation, the engine accumulated hours are included as an input variable.

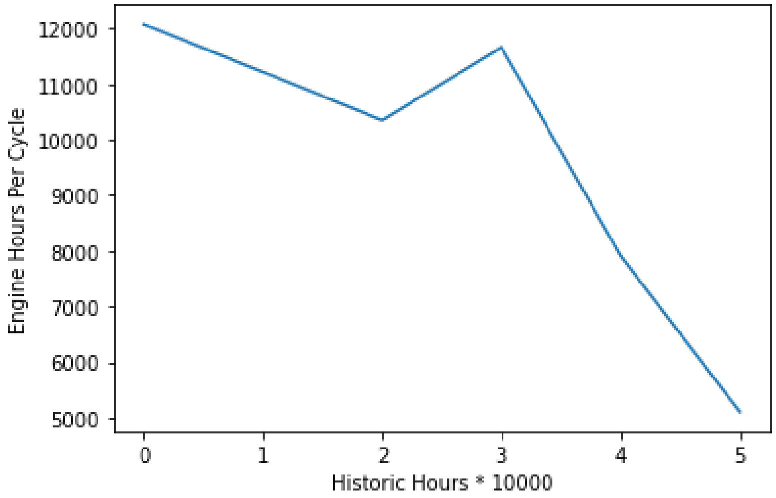


Figure 3 - Average operational hours before failure per its accumulated hours throughout its entire life

The engines get operated in 7 different sites. We can examine the distribution of operational hours are different between sites. However, site cannot be used as an input feature since sensor records and oil analysis were provided from one site.

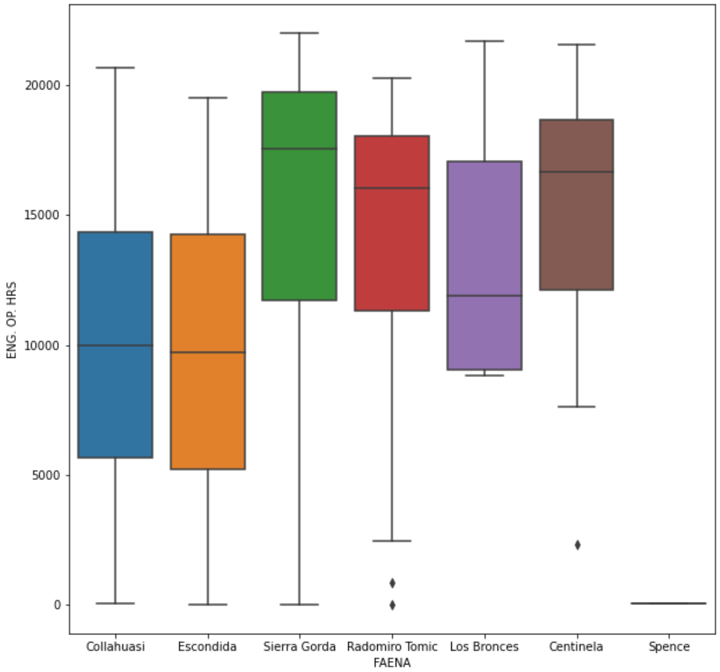


Figure 4 - Boxplot of operational hours before failure per site

When the engines fail, they could have different failure modes. For the dataset provided, there were only two failure modes, which are seized-piston and seized crankshaft. However, according to STNG, there could be 5 different failure modes, such as coolant leak.

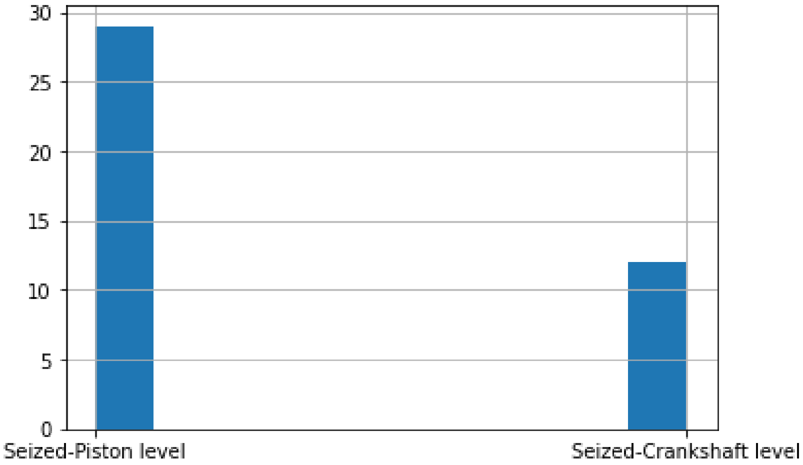


Figure 5 - Failure modes of the provided engines

3.0 Proposed Methodology

This project proposes a model ensemble approach to address the engines failures modeling by establishing a prediction framework that integrates both sensor and the engine data. The framework gathers the prediction from sensor and the oil data and uses the minimum as the final prediction.

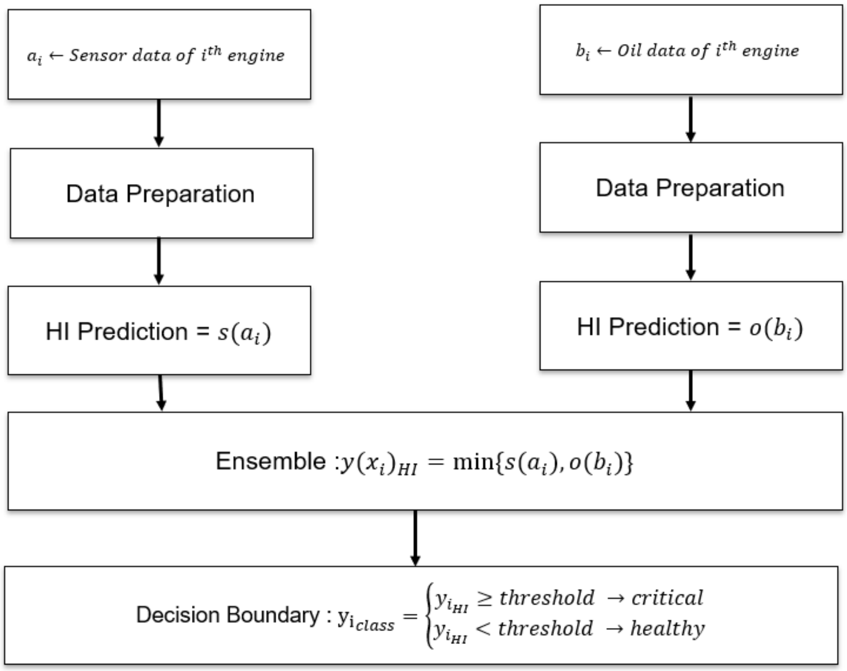

$a_i \leftarrow Sensor\ data\ of\ i^{th}\ engine$
$b_i \leftarrow Oil\ data\ of\ i^{th}\ engine$
Data Preparation
Data Preparation
HI Prediction = $s(a_i)$
HI Prediction = $o(b_i)$
Ensemble :$y(x_i)_{HI} = \min\{s(a_i), o(b_i)\}$
Decision Boundary : $y_{i_{class}} = \begin{cases} y_{i_{HI}} \geq threshold \rightarrow critical \\ y_{i_{HI}} < threshold \rightarrow healthy \end{cases}$

Figure 6 – Proposed model framework

The final prediction of health index (HI) is then converted as classes of healthy and critical using a decision boundary to invoke the operators to make corrective action or preventive measures on critical engines.

3.1 Target Variable

Instead of predicting for Remaining Useful Life (RUL) in hours directly, the model predicts for health index (HI) which is remaining useful life (RUL) in hours normalized into scale [0,1]. Since all engines show different levels of degradation even at same remaining useful life, the health index or percentage reflects the actual state of the engine more than remaining useful life in hours. This can be formulated as follows:

$$Health\ Index = 1 - \frac{current\ operational\ hours}{operational\ hours\ at\ failure} \quad (1)$$

3.2 Metrics

For evaluation of the proposed model, root mean squared error (RMSE) is used; however, the most important predictions are when the machines are at a critical state. Instead of evaluating the entire interval, the RMSE of prediction less than 0.25 and actual target variable less than 0.25 is used as evaluation. This evaluates the recall and precision for the machines at critical state of 25% health level

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2} \ for\ y_i < 0.25\ \ or\ \hat{y}_i < 0.25 \ \ (2)$$

3.3 Data Split Method

We realize that combining all engines and randomly splitting on the data instants level causes major data leakage. Thus, the dataset is split between train and test per engine. Afterwards, k-fold partition is applied on the train set and 5-fold cross validation is applied to get the train and validation set for training.

3.4 Model Prediction Interpretation

Upon prediction of the health index or remaining useful life, we propose to use the threshold value to classify between critical and healthy state of the engine. This allows the operators to make corrective plans and additional maintenance when the model starts predicting as critical state.

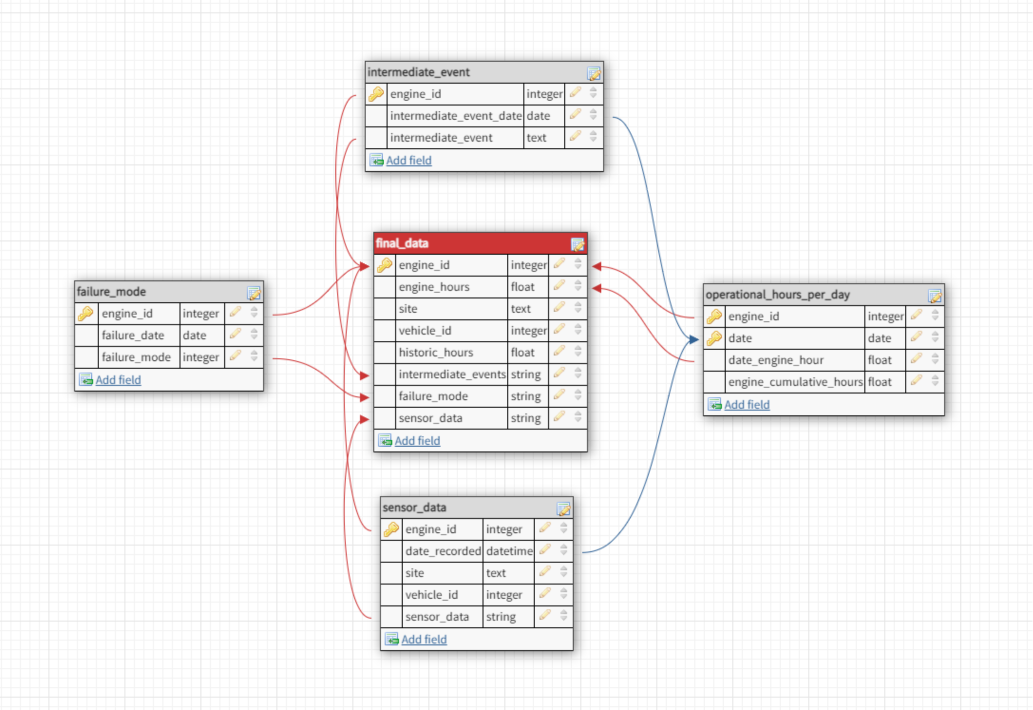
The threshold is defined through the procedures as the following:

1. Incrementally iterate through failure thresholds
2. Convert the numerical health index (HI) prediction to discrete classes using the threshold
   - If $\hat{y}_{HI} \geq HI_{threshhold} \rightarrow y_{class} = 0$
   - If $\hat{y}_{HI} < HI_{threshhold} \rightarrow y_{class} = 1$
3. Measure the average precision, recall, and F1 score for different critical thresholds

4.0 Sensor Model

4.1 Data Preparation

The initial step of the data preparation is cross-referencing across four different datasets. The figure below shows the data schema for constructing the final dataset.



intermediate_event	
engine_id	integer
intermediate_event_date	date
intermediate_event	text

failure_mode	
engine_id	integer
failure_date	date
failure_mode	integer

final_data	
engine_id	integer
engine_hours	float
site	text
vehicle_id	integer
historic_hours	float
intermediate_events	string
failure_mode	string
sensor_data	string

operational_hours_per_day	
engine_id	integer
date	date
date_engine_hour	float
engine_cumulative_hours	float

sensor_data	
engine_id	integer
date_recorded	datetime
site	text
vehicle_id	integer
sensor_data	string

Figure 7 - Table schema of the sensor model

The measurement data is complex and high dimensional. The initial dataset contains 221 features, but after removing the columns with greater than 30% null values, more than 150 features were eliminated. Additionally, there are many features that have low variance which mean that they are ineffective against the performance. After analyzing each variable, the total final selected features are 24 variables.

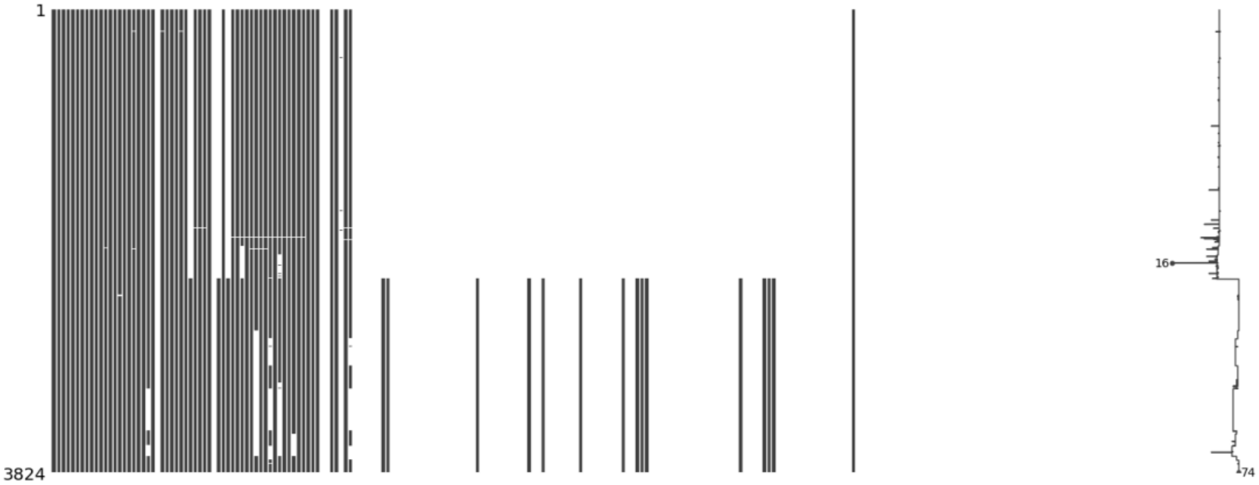


Figure 8 - Missing data representation

The sensor data does not have consistent sampling rate and the actual operational hours were from another dataset, which contains the operational hours per day. Thus, the data points were imputed to an average, every 15 operational hours. The 15 hours is chosen as it is the average hours the engines operate per day.

The measurement data exhibits great outliers such as spikes and noise. After analyzing the data, the spikes occurred randomly and does not have any relationship with the failure. Therefore, the filtering method using Z score threshold is used to eliminate the noises.

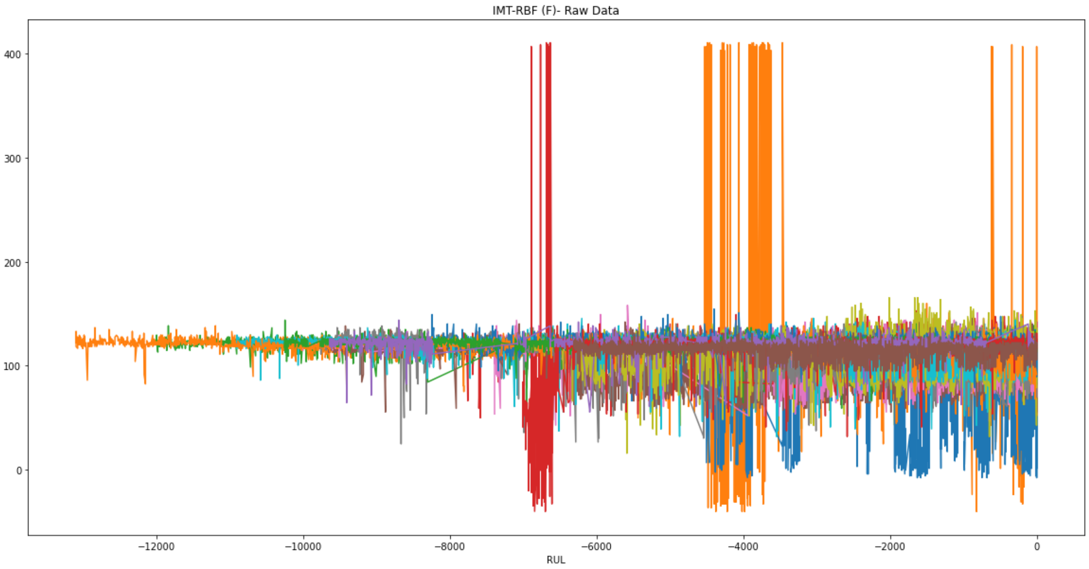


Figure 9 - Raw IMT-RBF (F) data

Lastly, the normalization is applied as it helps to accelerate the convergence rate. The min-max normalization is used to convert the sensor data scale within the range of [0,1].

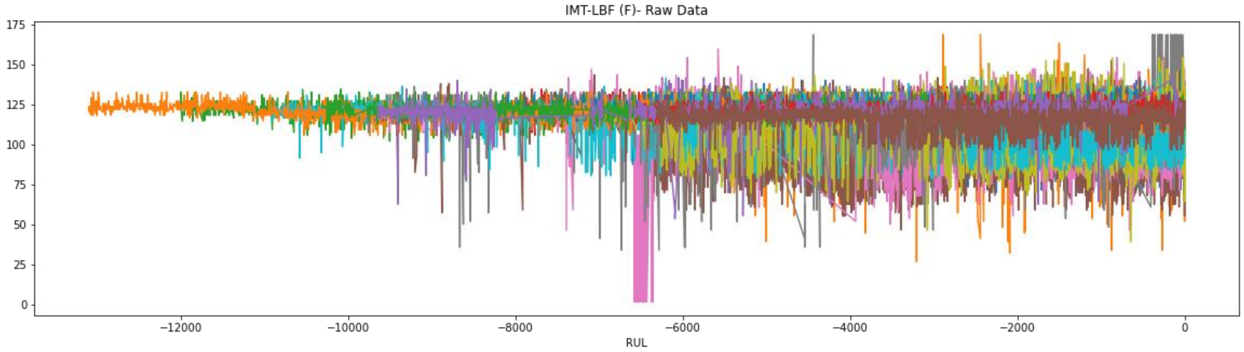


Figure 10 - Raw IMT-LBF (F) data

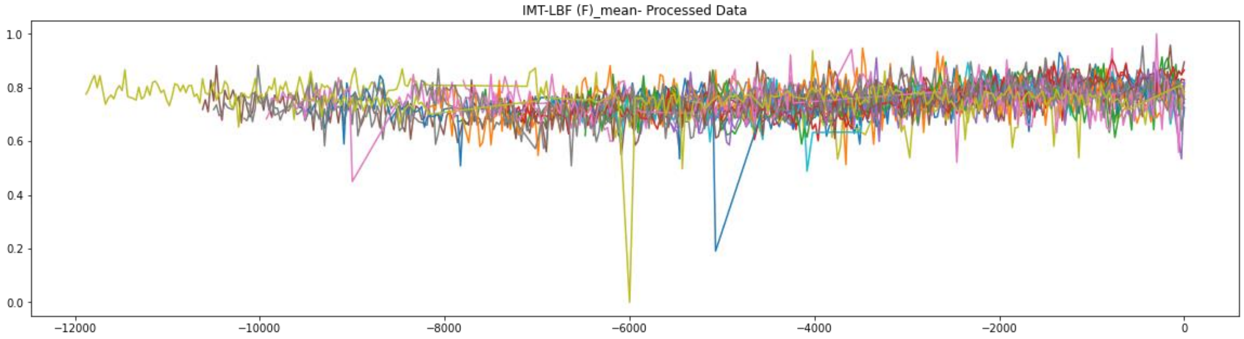


Figure 11 - Preprocessed IMT-LBF(F) time series data

4.2 EDA – Sensor Data

During the month of June and July, we can examine a minor drop in temperature compared to other months of the year. The model is initially developed with the months as a feature, but was removed in the model final as it did not have a significant predictive power,

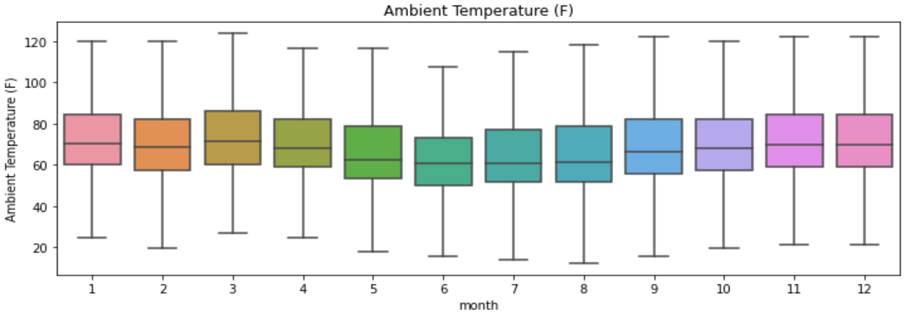


Figure 12 - Boxplot of ambident temperature (F) per month of the year

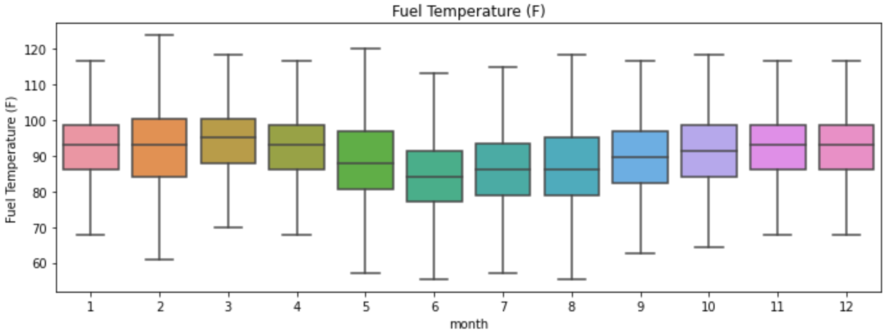


Figure 13 - Boxplot of fuel temperature (F) per month of the year

The graph below are few variables that show a slight increasing trend before failure. The x-axis is the negative remaining useful life. For example, in the figure, data points at -10,000 RUL represents readings 10,000 hours before failure, and data points at 0 RUL represents failure.

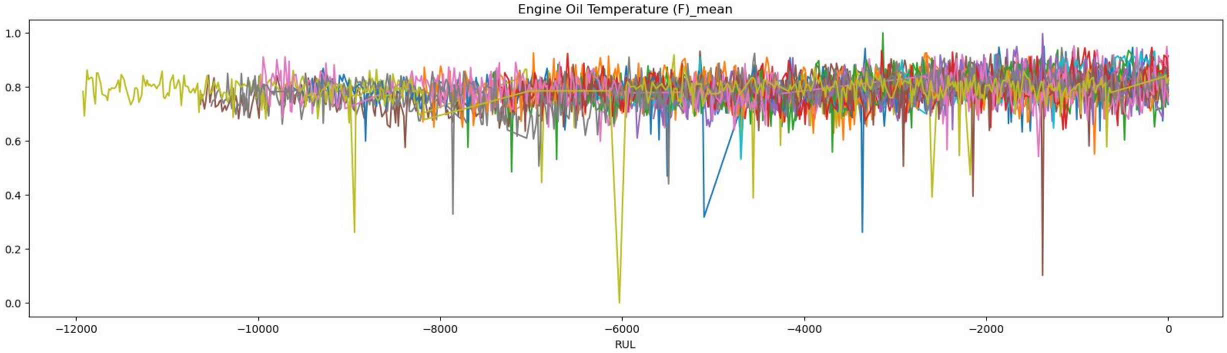


Figure 14 – Time series data of Engine oil temperature (F)

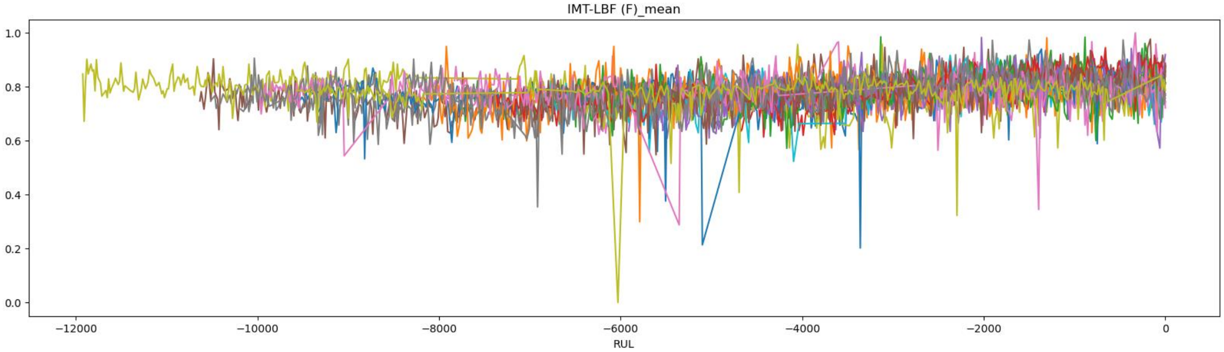


Figure 15 - Time series data of IMT-LBF (F)

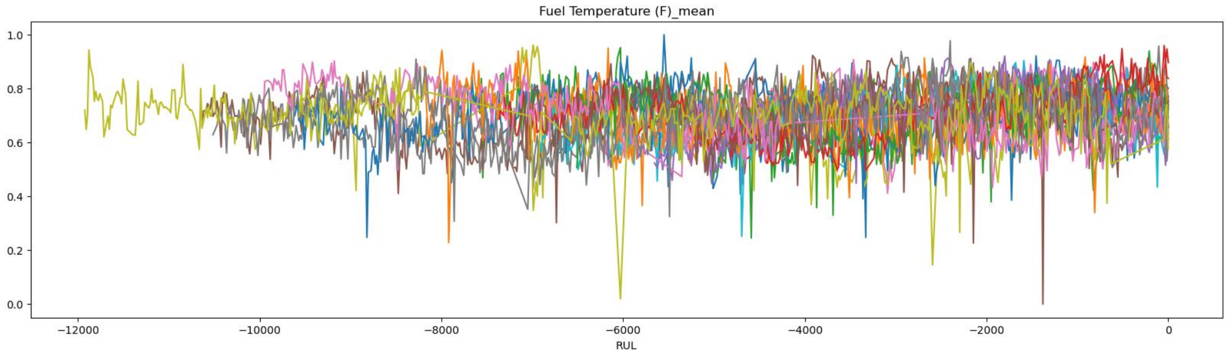


Figure 16 - Time series data of fuel temperature (F)

4.3 Failure Modes

In the provided engine datasets, the failure modes are seized piston and crankshaft. If the readings or responses between these two engines are different, then we would require building a separate model per failure modes.

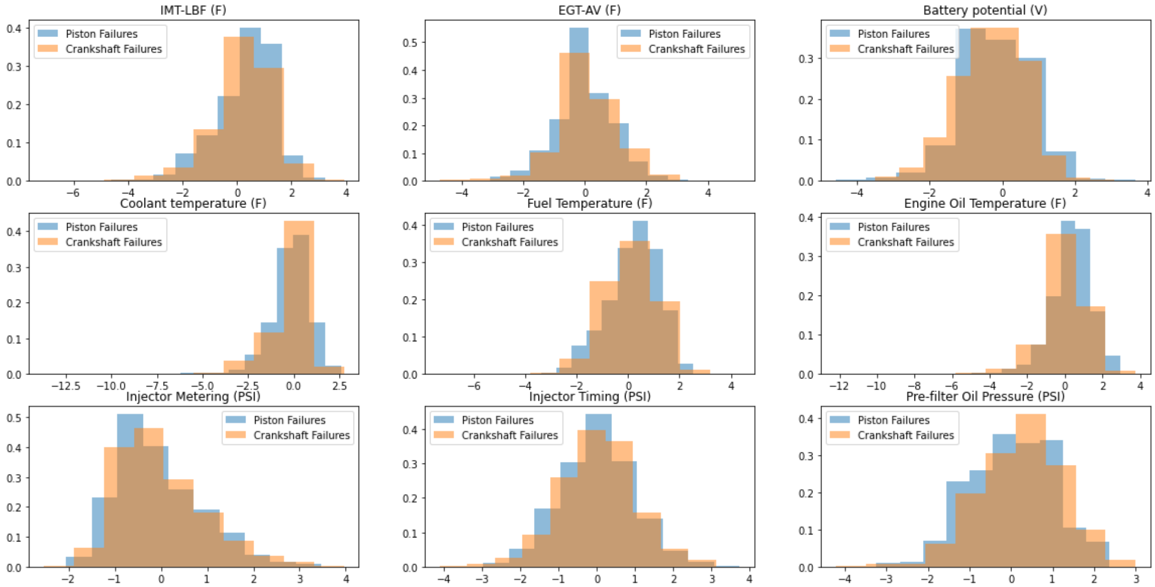


Figure 17 - Histogram of variables, comparing between two failure modes

The figure shows the histograms of piston and crankshaft failure responses at the last 2000 hours before failure for different variables. The responses are very identical between the two failures.

To confirm that the responses from the failures are identical, a Logistic Regression is implemented for the last 2000 hours before failure between the crankshaft and piston failures. The confusion matrix of the logistic regression is shown below.

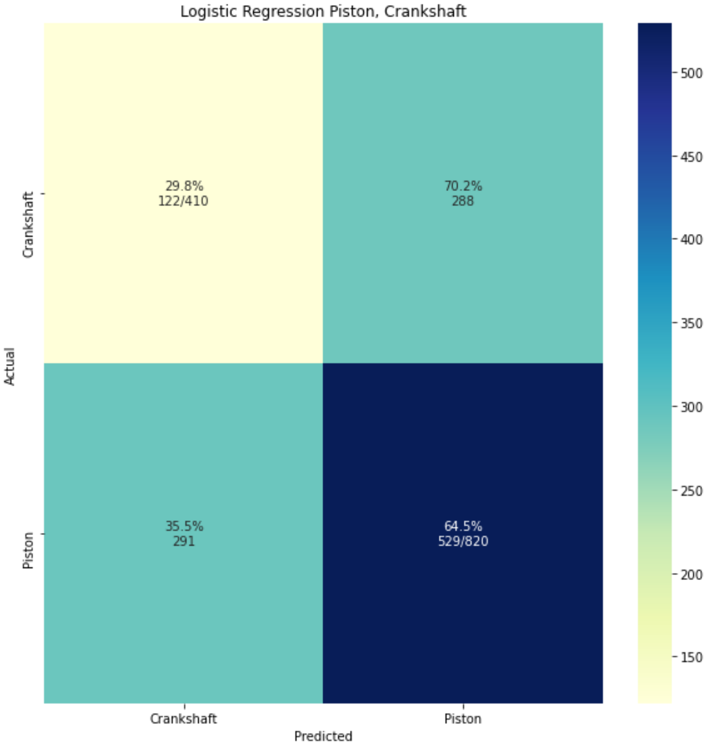


Figure 18 - Confusion matrix of predictions using logistic regression

The confusion matrix indicates that the model performs poorly and is not able to predict between the two failures. However, the downside of logistic regression is that we are currently classifying per data points, not the entire time series or the engine. We thus explored dynamic time warping for classification of the time series.

Dynamic time warping is an algorithm that measures the similarity between the two temporal sequence. It finds the optimal non-linear alignment between the two time series and try to minimize the distance between the two temporal sequence or time series. The way it calculates the similarity is as follows:

Let $X(x_1, x_2, \dots x_n)$ and $Y(y_1, y_2, \dots y_m)$ be the two time series with the length of n and m. The initial step is to construct a matrix of $n \times m$ whose i^{th} and j^{th} element is the Euclidean between x_i and y_i. The goal is to find a path through this matrix that minimizes the cumulative distance between the two time series. This path is represented as $W = w_1, w_2 \dots w_K$, where the elements w_k represents distance between two points i and j. The final output is the minimum Euclidean distance between the two time series, $DTW(X, Y) =$

$argmin_w(\sum_{k=1}^{K} w_k)$. The optimal path is found using dynamic programming following the algorithm:

$$DTW(X,Y) = w(x_i, y_j) + min \begin{cases} DTW(i-1,j) \\ DTW(i-1,j-1) \\ DTW(i,j-1) \end{cases} \quad (3)$$

For our application, running the dynamic time warping algorithm outputs a similarity or distance matrix between all engines. The distance matrix is then used on K-Nearest Neighbor for final classification.

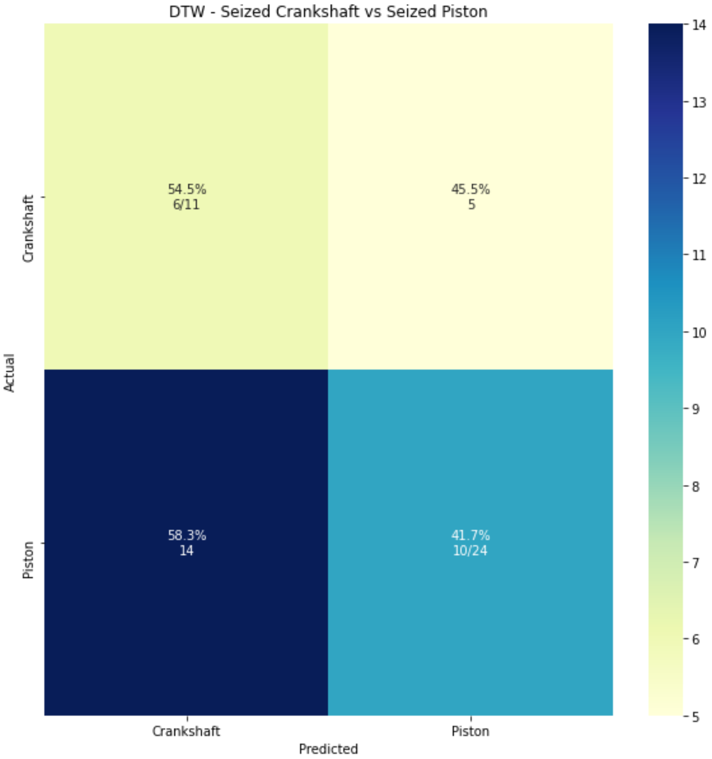


Figure 19 - Confusion matrix of classification using dynamic time warping (DTW)

The confusion matrix indicates that the differences in the responses between crankshaft and piston failures are similar. We thus can develop a single model for predicting the RUL between the two failures.

4.4 Model Implementation

The chosen architecture is MLP-LSTM-MLP due to its performance compared to other proposed architectures. Researchers from University of Loraine proposed the architecture for predicting RUL using C-MAPSS Turbo engine dataset and achieved high performance. Additionally, similar structure MLP-RNN-MLP was proposed as an autoencoder for effectively denoising the time series data for automatic speech recognition (ASR) [8].

The architecture presented in paper from University of Loraine is served as a baseline model and was tuned further. The major difference is that the proposed architecture uses fixed length of sequence whereas the architecture from Loraine uses the entire time series of the engine from start to failure for training. The reason for not using the entire time series for our application is that even though we use LSTM which effectively handles vanishing / exploding gradients, we can still face such problems at long sequences. We thus use fixed sequences instead of entire time series.

LSTM layers alone may not be sufficiently expressive to capture the complexities of noise environments and the features. Conversely, MLP are well fitted to perform such a task. The idea of this architecture is to feed the raw inputs to the MLP before the LSTM layers. Having the raw inputs processed by the MLP layer initially allows the model to learn a good representation from a noisy and complex data, while LSTM captures the dependencies in the time sequence. Afterwards, the final layer is added to predict the RUL from these processed representations.

With multiple hidden layers, we denote i^{th} hidden layer's activation as $h^{(i)}(x_t)$. For hidden layers, the activation is computed as:

$$h^{(1)}(x_t) = \sigma\left(W^{(1)}(x_t) + b^{(i)}\right)$$

$$h^{(i)}(x_t) = \sigma(W^{(i)}h^{(i-1)}(x_t) + b^{(i)})$$

Each hidden layer $h^{(i)}$ has a corresponding weight matrix $W^{(i)}$ and the bias vector $b^{(i)}$. In our application, the tanh() serves as the activation function, σ.

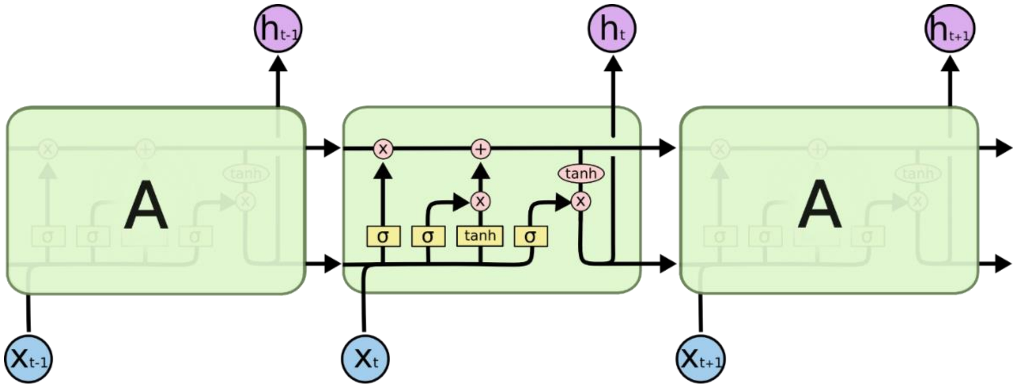


Figure 20 - Illustrated guide to LSTM [9]

For the LSTM model, the middle layer has a temporal connections with the following procedure:

1. The first step is to decide what information we are going to discard from the cell state through the "forget gate layer". It looks at the last LSTM unit (h_{t-1}) and the current input x_t and outputs a number in the range of (0,1) for each cell state C_{t-1}. The output of 1 result in completely keeping the value while a 0 represents completely removing it. The sigmoid function for the forget gate is:

$$f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f)$$

Herein, σ is the sigmoid function, and W_f and b_f are the weight matrices and bias, respectively of the forget gate.

2. The next step is deciding what new information we are going to store in the cell state. This consists of two parts. First, an input gate layer which is a sigmoid layer decides which values to update. Then, a tanh layer creates a vector of new vectors, $\tilde{C}_t$ that could be added to the state.

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i)$$
$$\tilde{C}_t = tanh(W_C * [h_{t-1}, x_t] + b_C)$$

3. In the next step, we update the old cell state C_{t-1} into the new cell state C_t by multiplying the old state by the forget gate vector and add the new information. This is be formulated as:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

4. The final step is to decide on the output. Initially, the sigmoid layer decides what parts of cell state we are going to output. Then, we feed the cell state through tanh (to output between -1 and 1) and multiply by the output of the sigmoid gate. This is formulated as:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$
$$h_t = o_t * \tanh(C_t)$$

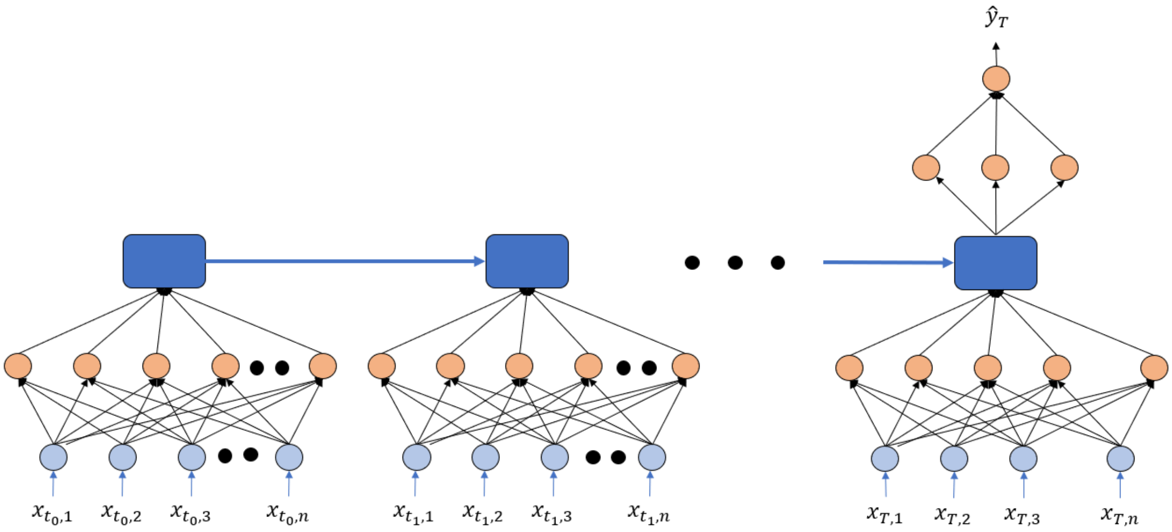


Figure 21 - Architecture of the MLP-LSTM-MLP with a sequence data

The Figure 21 illustrates the architecture of the proposed model with a single sequence data. Note that the MLP layers are shown as one hidden layer for simplicity. For our model, we use 3 hidden layers for first and last MLP.

The input vector x_t with n-dimensions (where n is the number of input variables) are initially processed by the first MLP layers, and the resulting sequence of feature vectors with each sequence of length T is then processed by a single LSTM layer. Finally, the output of the last LSTM cell of each sequence is passed to another MLP layers which outputs the scaler $\hat{y}_t$, the predicted RUL. Note that the weights of the MLPs are shared across all time steps.

4.4 Results

The performance is evaluated using K-Fold of 5 with mean and standard deviation of RMSE error per engines. After selecting the best architecture which is, MLP-LSTM-MLP, the hyperparameters were tuned using Bayesian optimization.

Table 2 - Performance of various architectures

Models	RMSE	Standard Deviation
DANN	0.1267	0.0030
LSTM	0.1232	0.0019
DCNN	0.1320	0.0020
CNN-LSTM	0.1244	0.0022
ResNet-3	0.1284	0.0014
MLP-LSTM-MLP	**0.1185**	**0.0022**

Table 3 – Hyperparameter of MLP-LSTM-MLP

Description	Value
Learning Rate	0.00036
Number of neurons in MLP layers	256/128/128
Number of LSTM cells	64
Number of neurons in MLP layers	128/64/64
Activation function for MLP layers	tanh ()
Batch Size	16
Sequence Length	14
MLP Dropout Rate	0.2

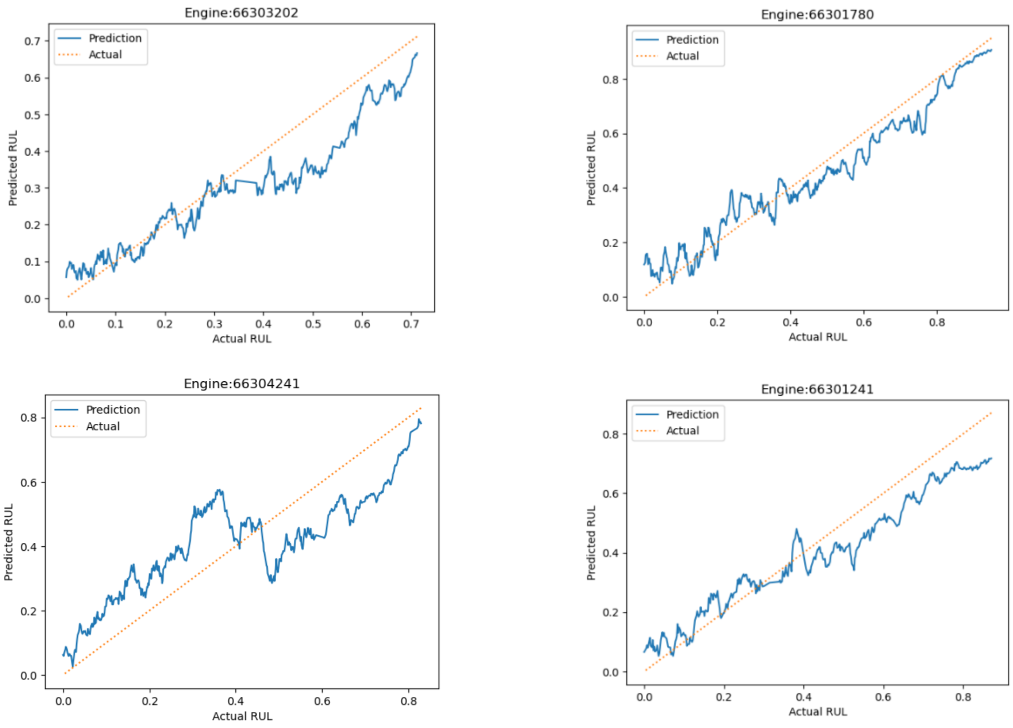


Figure 22 -RUL prediction for engines that failed after 8000 hours

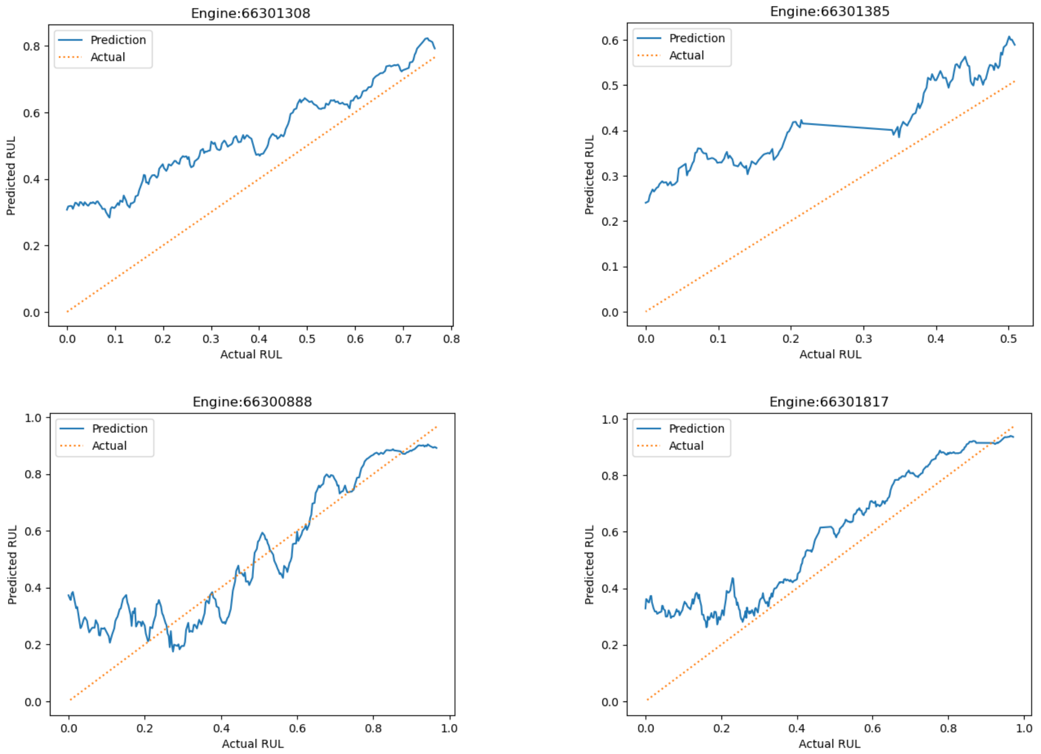


Figure 23 - RUL prediction for engines that failed after 8000 hours

The Figure 22 and Figure 23 shows engine prediction that failed before 8000 hours and after 8000 hours respectively. We observe that the model significantly underperforms on engines that failed before the operational hours of 8000. The RMSE of the engines that failed before and after 8000 hours are given in the following table:

Table 4 - Performance comparison between standard failures and premature failures

Engine Types	**RMSE**
Engine Failure Before 8000 Hours	0.211
Engine Failure After 8000 Hours	0.0846

The Figure 24 shows the precision, recall and F1 score at different threshold of all engines when used as a validation set. The threshold requires to be high to achieve an acceptable level of precision and recall. On the other hand, if we separate the engines that have failed before 8000 and after 8000, we can observe the performance at each threshold improves significantly for engines above 8000 hours of operation, but engines below 8000 hours of operation underperforms at an unacceptable level. The conclusion from the sensor model is that it does not have the capability to predict the engines that failed prematurely. We thus disregard the model prediction for engines before 8000 hours of operations and only focus on prediction once the engines operate above 8000 hours.

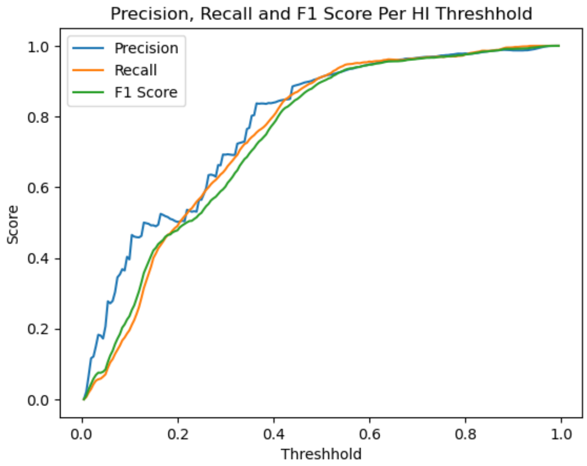


Figure 24 - Precision, Recall, and F1 Score of all Engines per Threshold

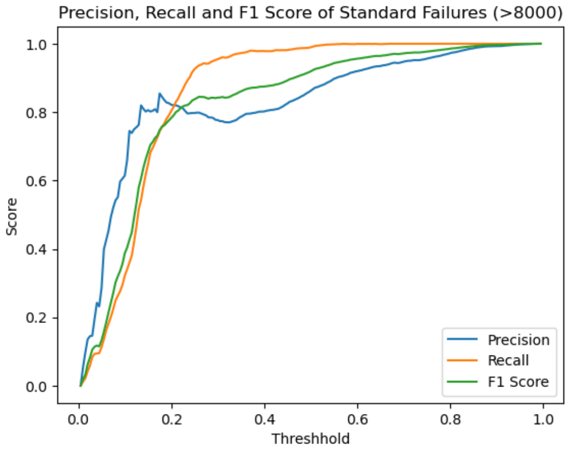


Figure 25- Precision, Recall, and F1 Score of all Engines Failed after 8000 hours of operation

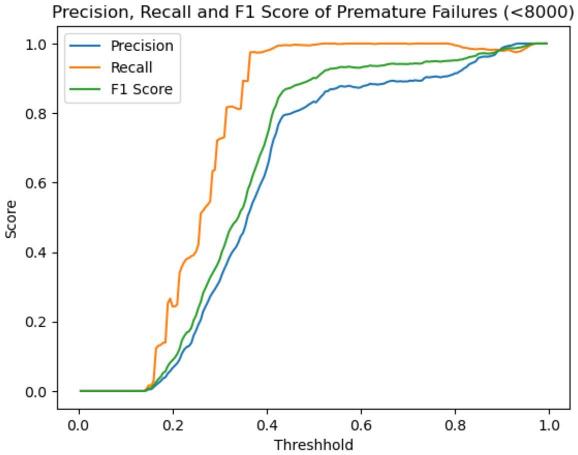


Figure 26 - Precision, Recall, and F1 Score of all Engines Failed before 8000 hours of operation

For the sensor model, the proposed threshold is 0.165. At this threshold, the model can achieve high precision and recall despite having low threshold. The performance metric at this threshold is given in the following table.

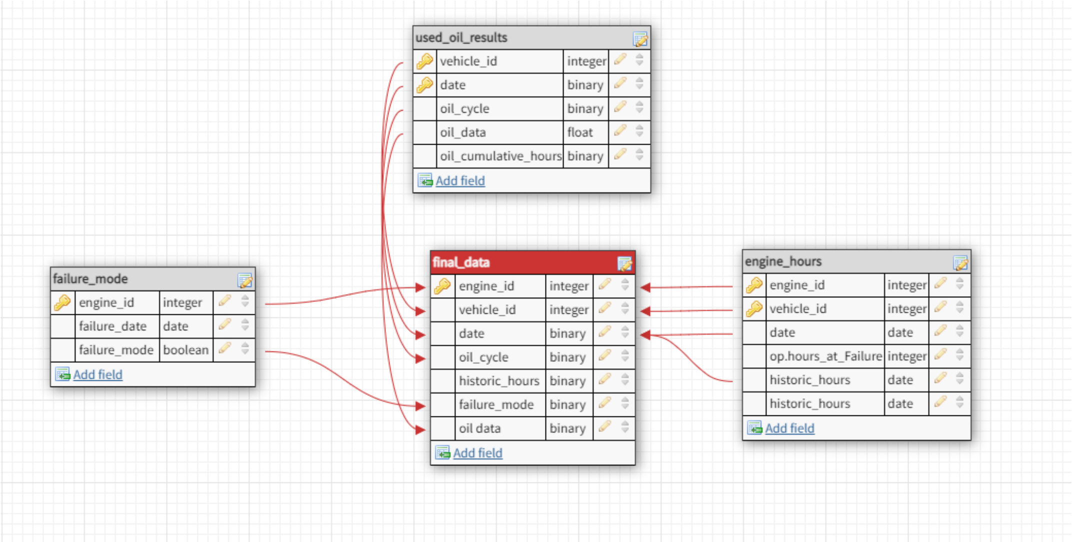
Table 5 - Performance at 0.165 threshold for engines that operated above 8000 hours

Performance Metric	**Result**
Precision	0.867
Recall	0.733
F1-Score	0.744

5.0 Oil Model

5.1 Data Preparation

The initial step of the data preparation is combining the three datasets into one. The figure below shows the data schema for constructing the final dataset.



used_oil_results	
vehicle_id	integer
date	binary
oil_cycle	binary
oil_data	float
oil_cumulative_hours	binary

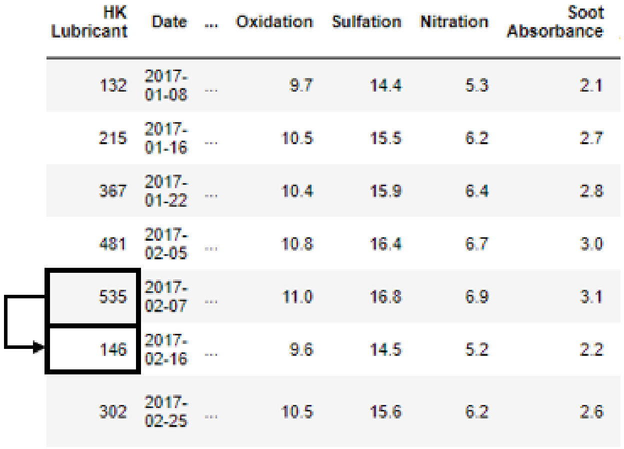
failure_mode	
engine_id	integer
failure_date	date
failure_mode	boolean

final_data	
engine_id	integer
vehicle_id	integer
date	binary
oil_cycle	binary
historic_hours	binary
failure_mode	binary
oil data	binary

engine_hours	
engine_id	integer
vehicle_id	integer
date	date
op.hours_at_Failure	integer
historic_hours	date
historic_hours	date

Figure 27 - Final dataset schema

Throughout the life cycle of the engine, the oil changes are performed on the engine, which reinstates the readings to its original state. In the figure below, a variable, HK Lubricant represents the accumulated hours on the engine oil. When the HK Lubricant drops, it indicates that the oil change has been done and the readings are in the new cycle of engine oil.



HK Lubricant	Date	...	Oxidation	Sulfation	Nitration	Soot Absorbance
132	2017-01-08	...	9.7	14.4	5.3	2.1
215	2017-01-16	...	10.5	15.5	6.2	2.7
367	2017-01-22	...	10.4	15.9	6.4	2.8
481	2017-02-05	...	10.8	16.4	6.7	3.0
535	2017-02-07	...	11.0	16.8	6.9	3.1
146	2017-02-16	...	9.6	14.5	5.2	2.2
302	2017-02-25	...	10.5	15.6	6.2	2.6

Figure 28 - Example of the dataset in tabular form

Due to constant oil change, the data shows high seasonality. To de-seasonalize the data, the average, and the max values at per engine oil cycle is used for the analysis. The mean and the max values were aggregated from the data as it possesses high variance.

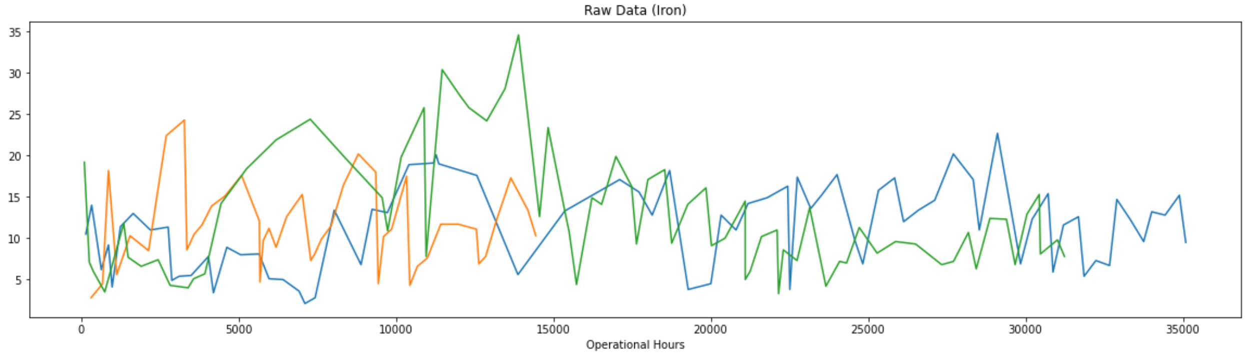


Figure 29 – raw time series data of iron element

The outliers in the data were filtered using the Z-score threshold of 2. The outliers did not exhibit any patterns nor indicated any signs of degradation as it occurred at random times.

Finally, the data was normalized using min-max normalization.

5.2 EDA

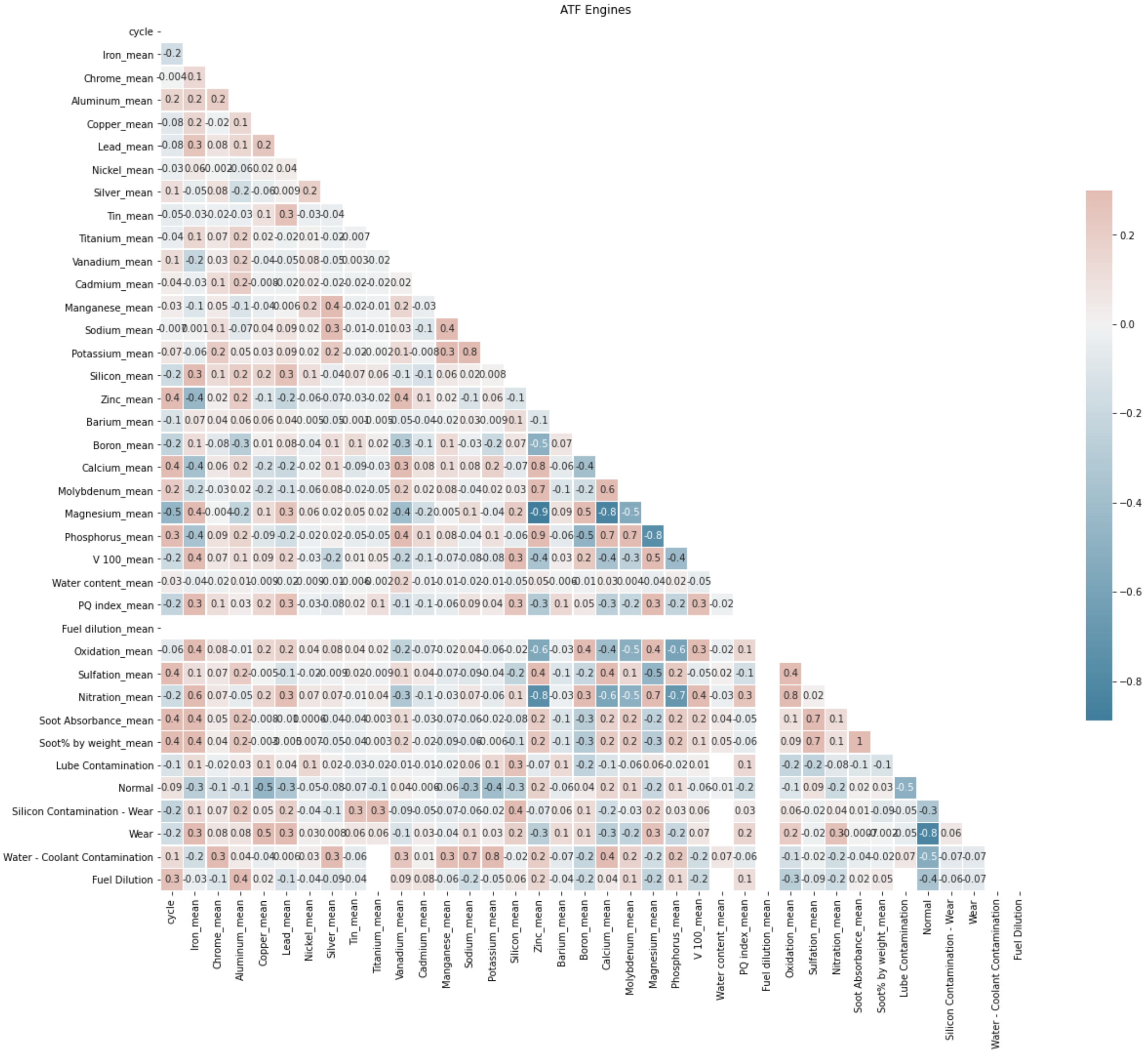


Figure 30 - Correlation matrix of the variables

The findings from the correlation matrix are as follows:

- As cycles increase, the soot absorbance and soot% by weight, increases with relatively high correlation
- Sulfation increases which represents degradation over time.
- Some additives like Magnesium and Boron decreases while calcium and molybdenum increases.

- Most of the wear elements have flat or zero correlation with cycle, but aluminum, vanadium and silver show a slight positive correlation.
- Iron, PQ index and viscosity decreases over time even though the failure is from wear.

The figure below shows some of the variables with high correlation that represent wear. The x-axis in the negative remaining useful life in cycles where 0 represents failure and -20 RUL represents 20 engine oil changes or cycles before failure.

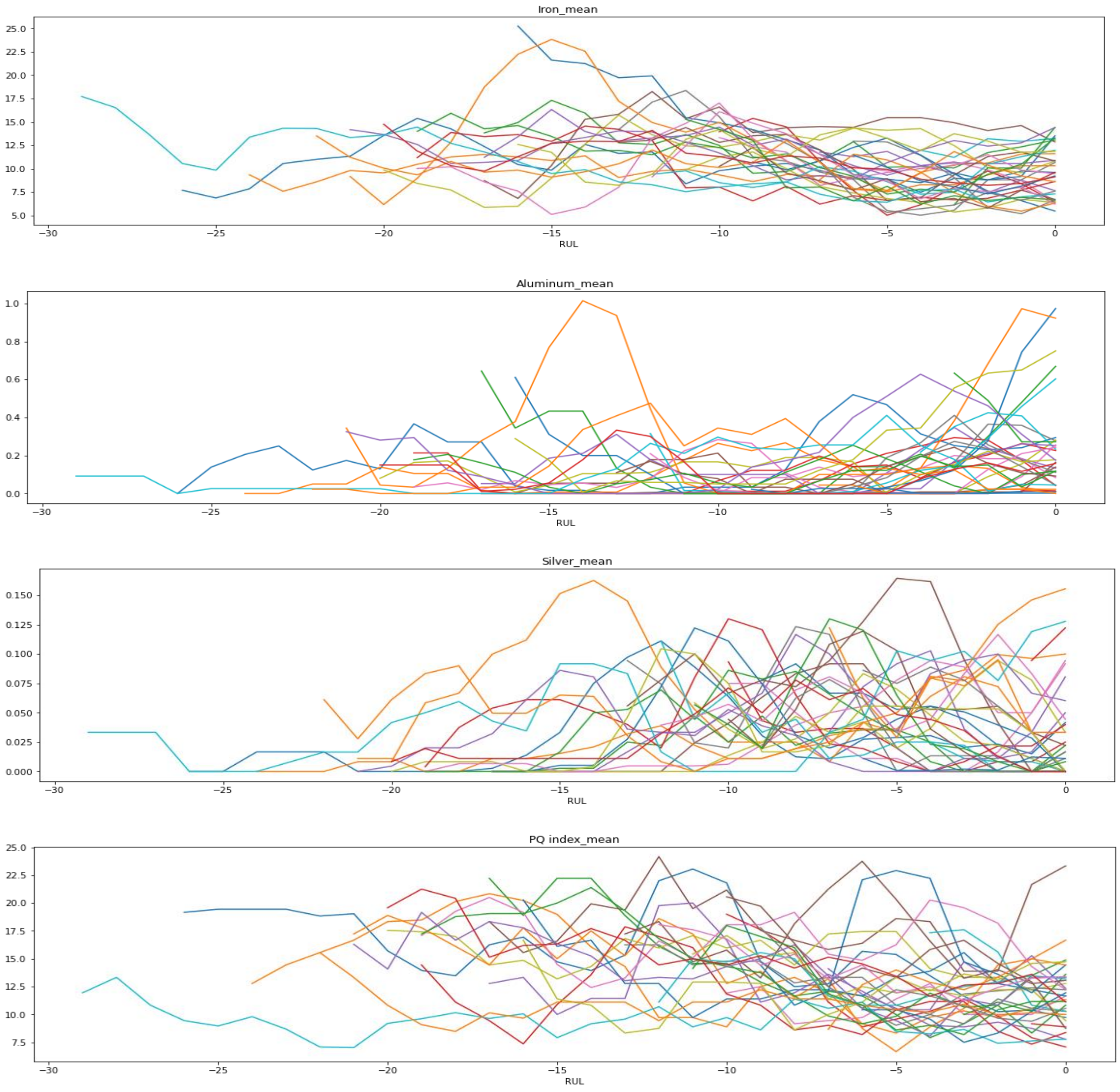


Figure 31 - Responses of input variables associated with wear

Aluminum and silver elements slightly increase over time, but PQ index and iron decreases over time which is abnormal. With failure modes from piston and crankshaft, we expect the mass to increase over time, but the data is showing the opposite.

The figure below shows variables that represent contamination and degradation.

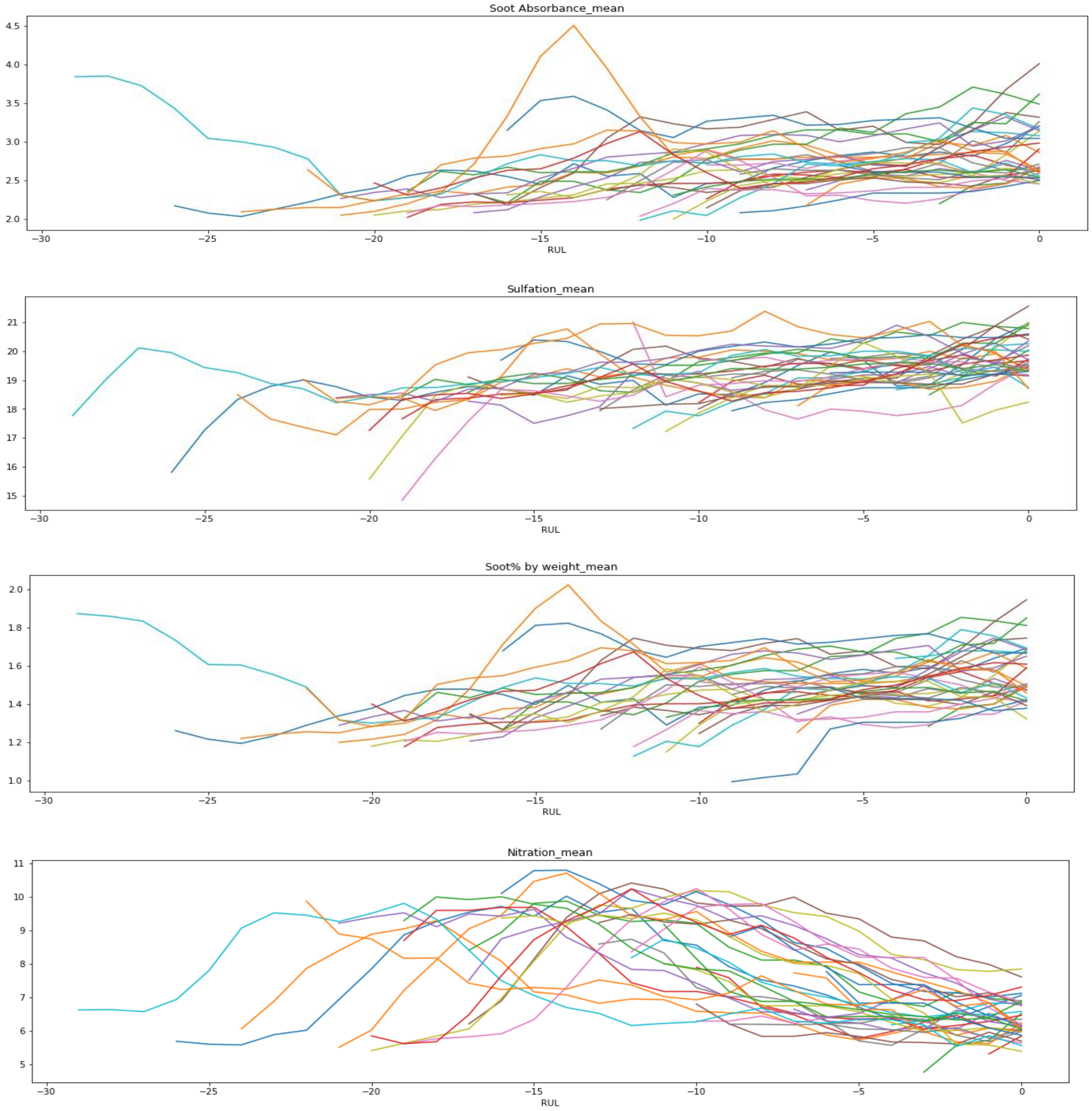


Figure 32 - Responses of input variables associated with contamination and degradation

We can examine the contamination, soot absorbance, soot% by weight, and sulfation increasing as it reaches close to the failure. Nitration on the other hand reduces over time which is unexpected.

5.3 Failure Modes

In the provided engine datasets, the failure modes are seized piston and crankshaft. If the readings or responses between these two engines are different, then we would require building a separate model per failure modes.

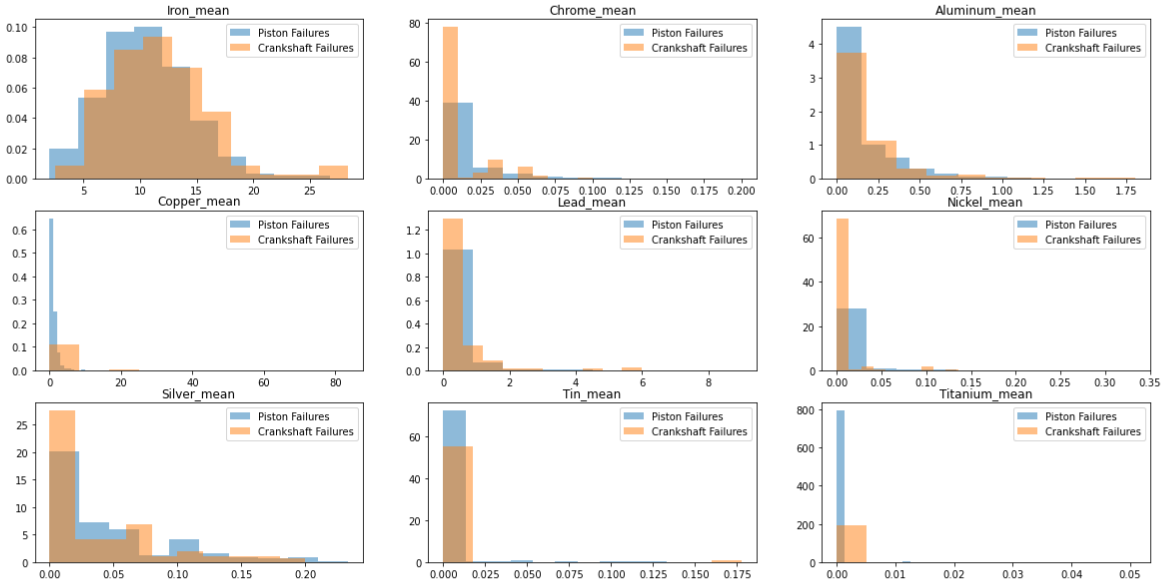


Figure 33 - Histogram of wear variables, comparing piston and crankshaft failures

The figure above shows the histograms of variables that are associated with wear directly. We can examine a very similar distribution between the two failures across all wear variables. The responses between these two failures are very identical.

To ensure that the responses from the failures are identical, a Logistic Regression is implemented for the last 2000 hours before failure between the crankshaft and piston failures. The confusion matrix of the logistic regression is shown below.

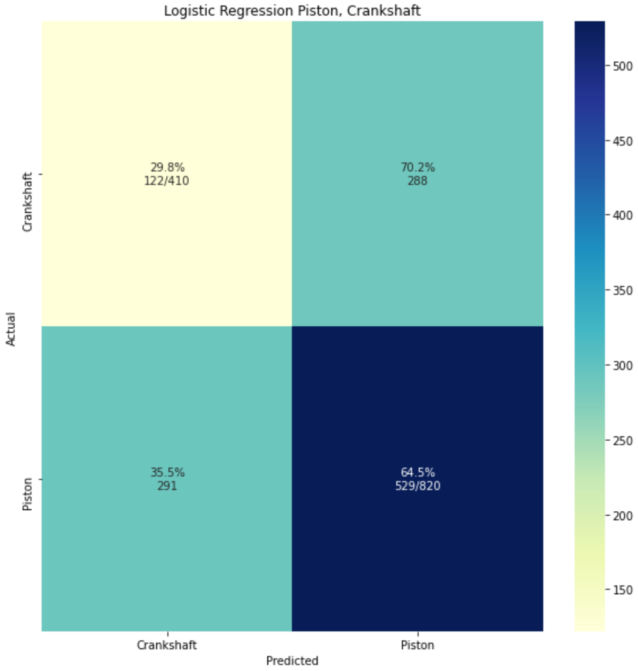


Figure 34 - Confusion matrix of predictions using logistic regression

The confusion matrix indicates that the model performs poorly and is not able to predict between the two failures. However, the downside of logistic regression is that we are currently classifying per data points, not the entire time series or the engine. We thus use dynamic time warping for classification of the entire time series.

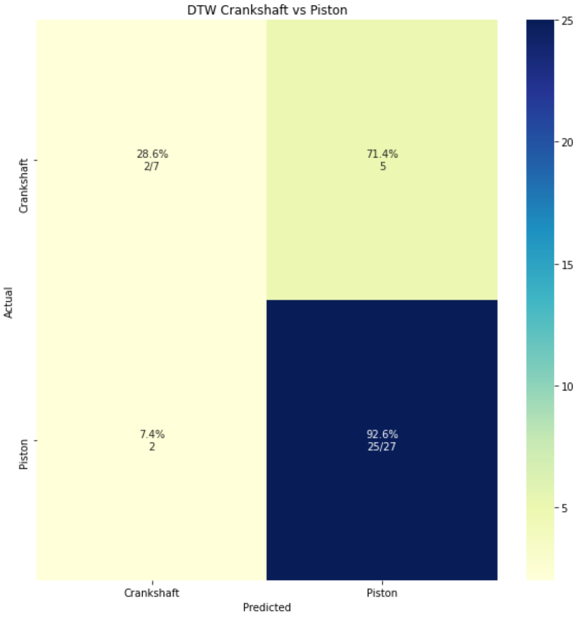


Figure 35 - Confusion matrix of the failure classification using DTW and KNN

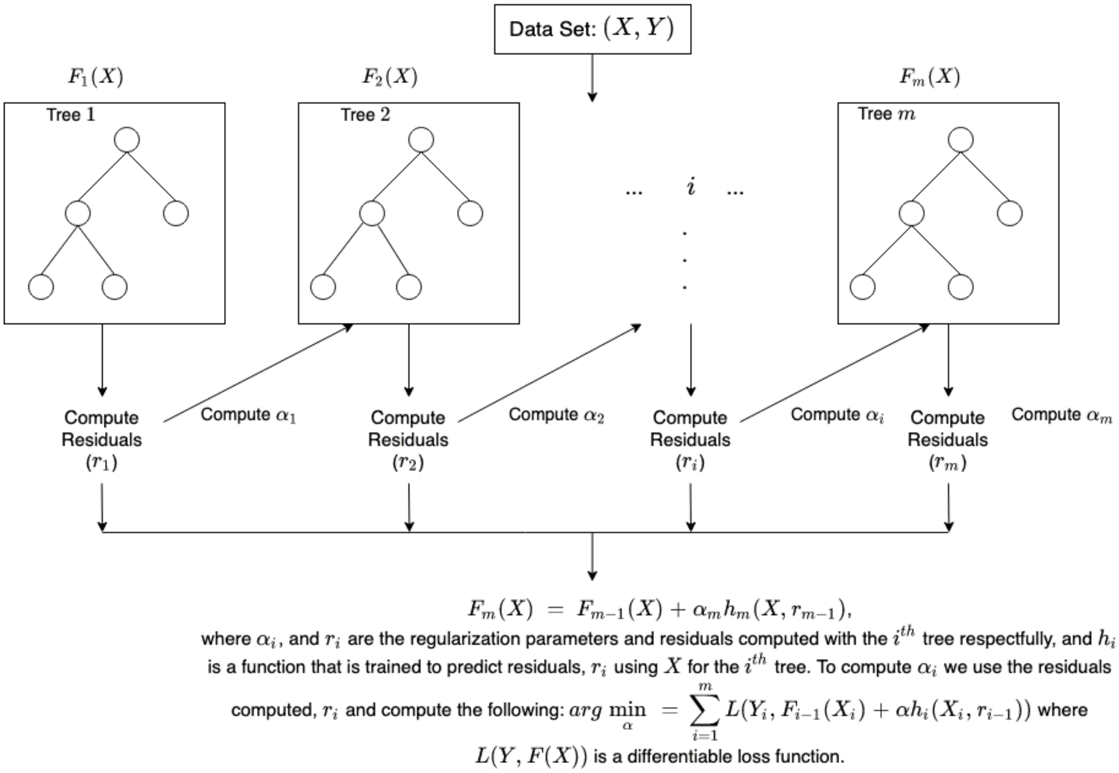
The Figure 35 shows the confusion matrix of classification model using KNN and dynamic time warping. The predictions between the two classes are poor which represent that these two failures are similar. However, for the oil data, the classes are imbalanced which could result the KNN model to overfit. When there are more data samples to train from, the classification task should be implemented again to validate this hypothesis.

5.4 Model Training

As opposed to the sensor data, the sampling rate of the oil data is low, resulting in small number of data points per engine. In average, there are 13 data points per engines. We thus use the machine learning algorithm, XGBoost is chosen for the prediction of RUL.

XGBoost is an algorithm that is further developed from gradient boosting. The gradient boosting uses the weak learners which are the regression tree that maps an input data point to its leaf that contain the continuous score. The XGBoost algorithm minimizes a regularized (L1 and L2) objective function that combines both loss function and penalty term for model complexity. The training proceeds iteratively, adding new trees that predict the residuals or errors of prior trees that are then combined with previous trees to make the final prediction. It's called gradient boosting because it uses a gradient descent algorithm to minimize the loss when adding new models.

Below is a brief illustration on how gradient tree boosting works.



$$F_m(X) = F_{m-1}(X) + \alpha_m h_m(X, r_{m-1}),$$

where α_i, and r_i are the regularization parameters and residuals computed with the i^{th} tree respectfully, and h_i is a function that is trained to predict residuals, r_i using X for the i^{th} tree. To compute α_i we use the residuals computed, r_i and compute the following: $arg \min_{\alpha} = \sum_{i=1}^{m} L(Y_i, F_{i-1}(X_i) + \alpha h_i(X_i, r_{i-1}))$ where $L(Y, F(X))$ is a differentiable loss function.

Figure 36 - Illustration on how XGBoost works

5.5 Oil Model Results & Threshold

The oil model is evaluated across 5 runs while maintaining the same parameters and the mean RMSE and average of standard deviations of RMSE error between engines in test sets are given in the following table. After selection of the best algorithm, XGBoost, the hyperparameters were optimized using Bayesian Optimization.

Table 6 - Performance of oil model for various machine learning algorithm

Models	**RMSE**	**Standard Deviation**
Lasso	0.2020	0.0030
ENet	0.2017	0.0064
Bayesian Ridge	0.1990	0.0074
Support Vector Regression (SVR)	0.2250	0.0111
GBoost	0.1720	0.0064
LightBoost	0.1740	0.0049
XGBoost	**0.1664**	**0.0054**

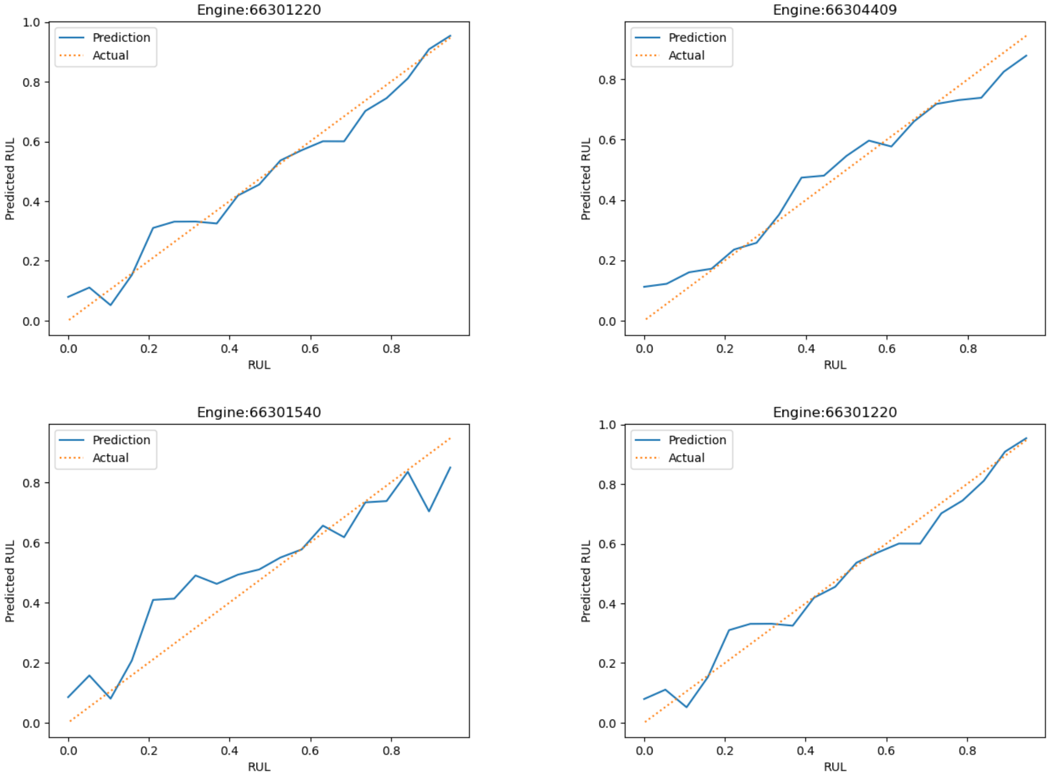


Figure 37 - RUL prediction of oil model for engines that failed after 8000 hours

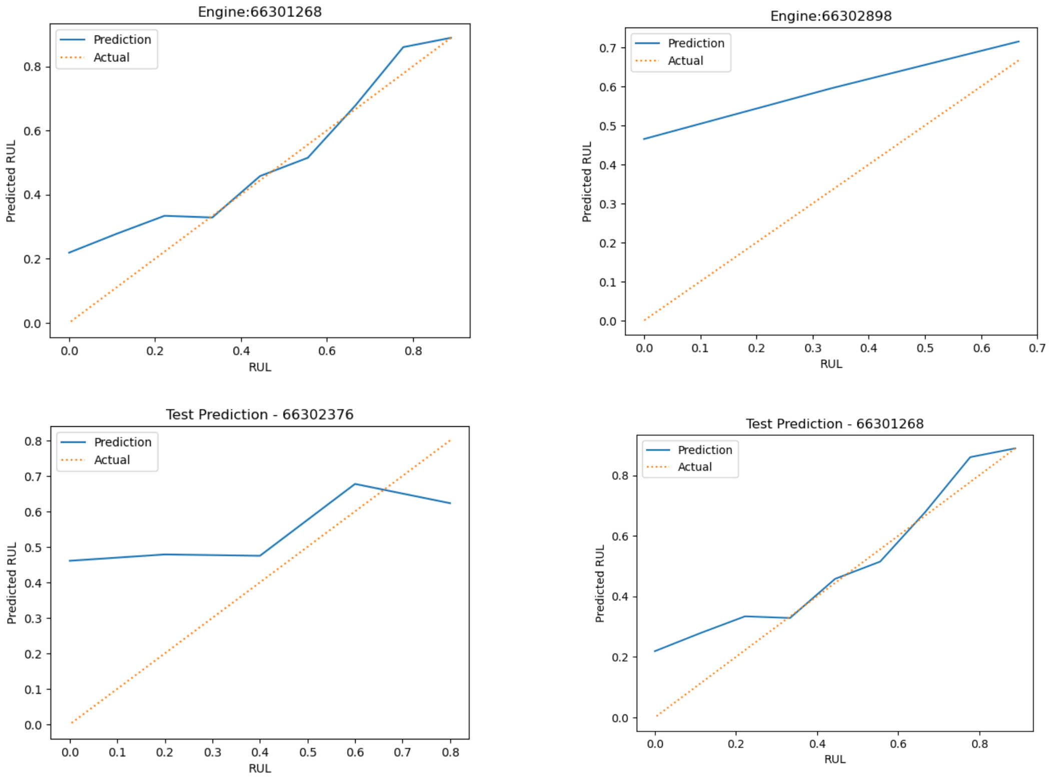


Figure 38 - RUL prediction of oil model for engines that failed before 8000 hours

The Figure 37 and Figure 38 shows that the oil model is also not able to predict well for the engines that prematurely failed; however, one key finding is that these engines tend to have high degradation rate as it reaches close to the critical state despite having low number of operated cycles as shown in Figure 39. In other words, if the prediction is converted to cycles remaining from health percentage, we can observe the pre-maturely failed engines having low cycles remaining even with short operational time. The conversion of health index to cycles remaining is formulated as follows:

$$RUL_{cycle} = \frac{x_{cycles}}{1 - RUL_{HI}} - x_{cycles} \ (4)$$

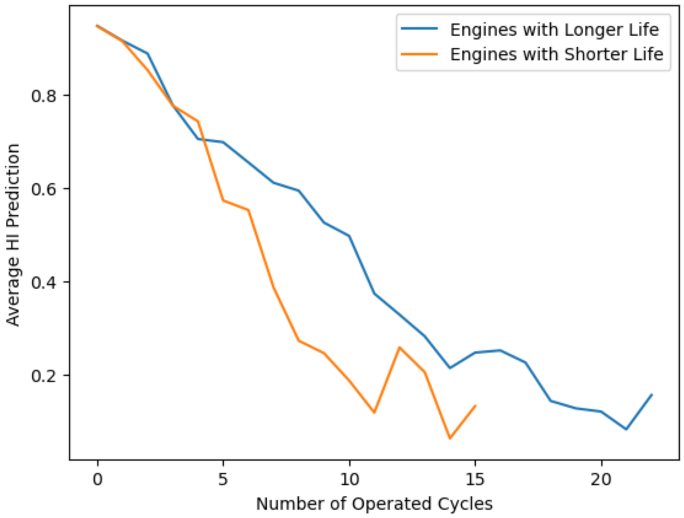


Figure 39 - Average HI prediction per number of operated cycles

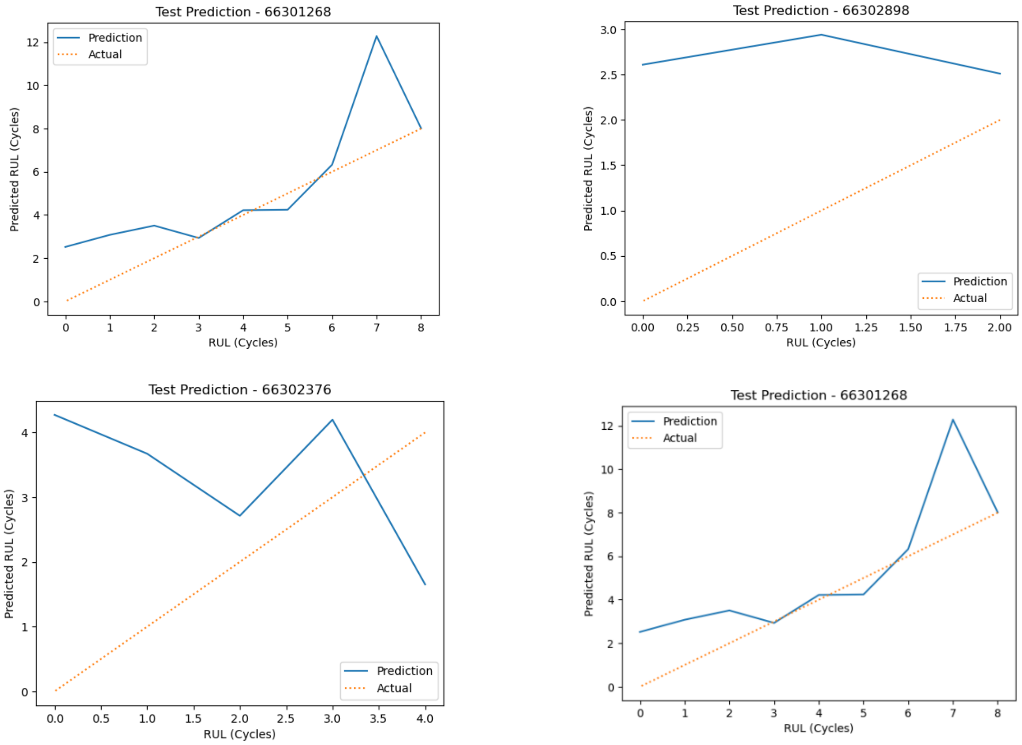


Figure 40 - RUL prediction in cycles for engines operated less than 8000 hours

The RMSE when predictions are converted to the number of cycles remaining are given in Table 7. The RMSE of the engine failures before 8000 hours show higher performance in comparison to the engine failures after 8000 hours.

Table 7 - RMSE of premature and standard failures when converted to cycles

Metric	RMSE (cycles)
Engine Failure Before 8000 Hours	2.935
Engine Failure After 8000 Hours	4.5605

The Figure 41 shows the precision, recall and F1 score at different threshold of all engines when used as a validation set. As seen from the sensor model, the oil model also shows poor performance for engines that prematurely failed and a good performance on engines that operated over 8000 hours.

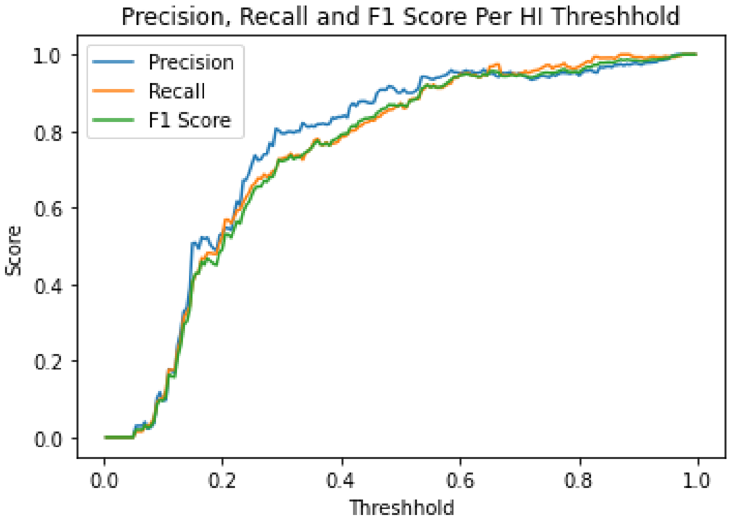


Figure 41 - Precision, recall and f1-score of all engines per threshold

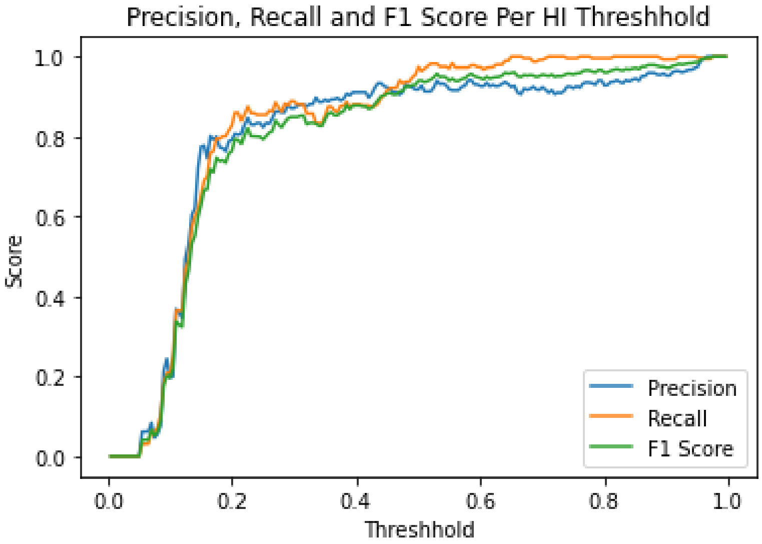


Figure 42 - Precision, recall and f1-score of engines above 8000 hours of operation

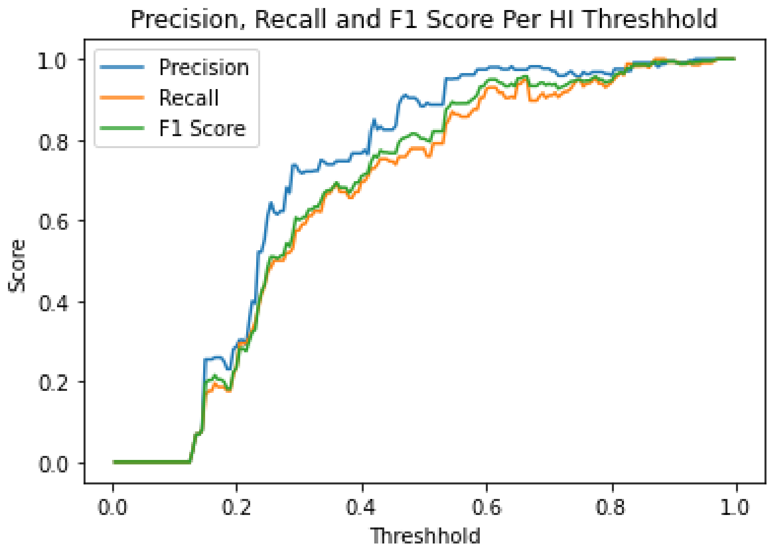


Figure 43 - Precision, recall and f1-score of engines under 8000 hours of operation

The figure below is the performance threshold of the prematurely failed engines with the health index (HI) prediction converted to cycle. We can observe a high performance despite having a low threshold on prematurely failed engines.

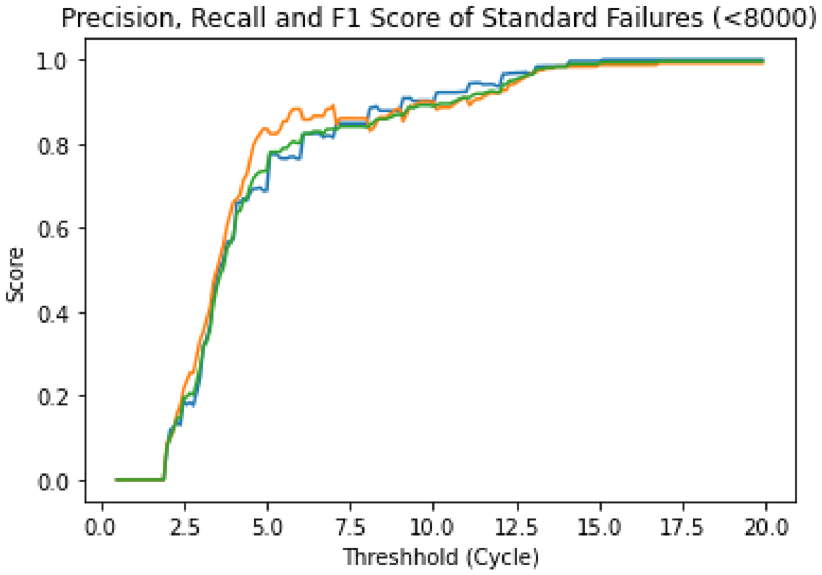


Figure 44 - Precision, recall and f1-score of engines under 8000 hours of operation per cycle threshold

The result indicates that the oil model cannot yield an acceptable performance for engines that failed before 8000 hours with the health index prediction, but if the health index is interpreted as the cycle remaining, the model provides a significant improvement in the performance. Hence, the oil model will use the cycles remaining threshold for the first 11 cycles of operation and use the health index threshold for the rest of its operation.

For the oil model, the proposed threshold is 0.165 and 5.1 cycles for engines above 8000 hours and under 8000 hours respectively. The average remaining useful life at the threshold is 1966 hours and 2392 hours for threshold of 0.165 and 5.1 cycles respectively. The performance at 0.165 HI threshold and 5.1 cycle threshold is shown in Table 5 and Table 6 respectively. The 0.165 is chosen due to high precision and recall with relatively low threshold and to keep consistent threshold with the sensor model. The 5.1 cycles are chosen as it provides moderate precision, but very high recall.

Table 8 - Performance at 0.165 HI threshold for engines operated over 8000 hours

Performance Metric	Result
Precision	0.802
Recall	0.758
F1-Score	0.718

Table 9 - Performance at 5.1 cycle threshold for engines operated over 8000 hours

Performance Metric	Result
Precision	0.774
Recall	0.835
F1-Score	0.735

6.0 Oil Model & Sensor Model Ensemble Result

This section outlines the results of the ensemble model with the defined threshold of 0.165 and 5.1 cycles. The results are presented in the figures below.

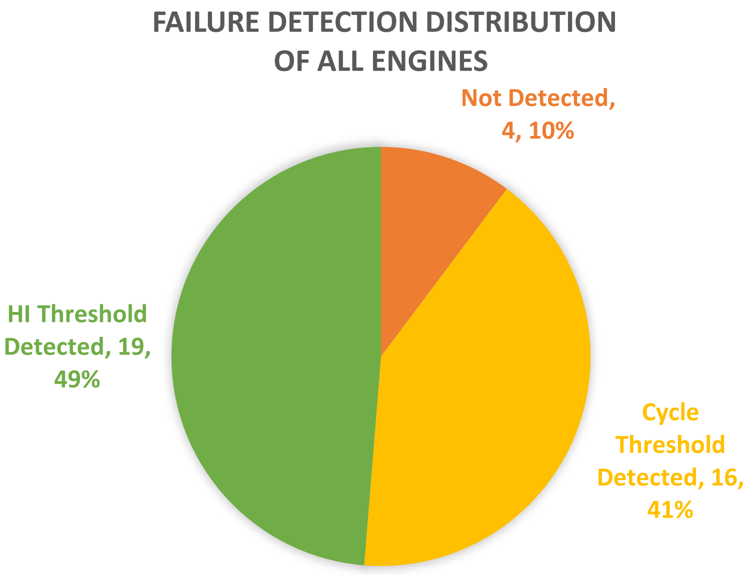


Figure 45 - Failure detection distribution of all engines

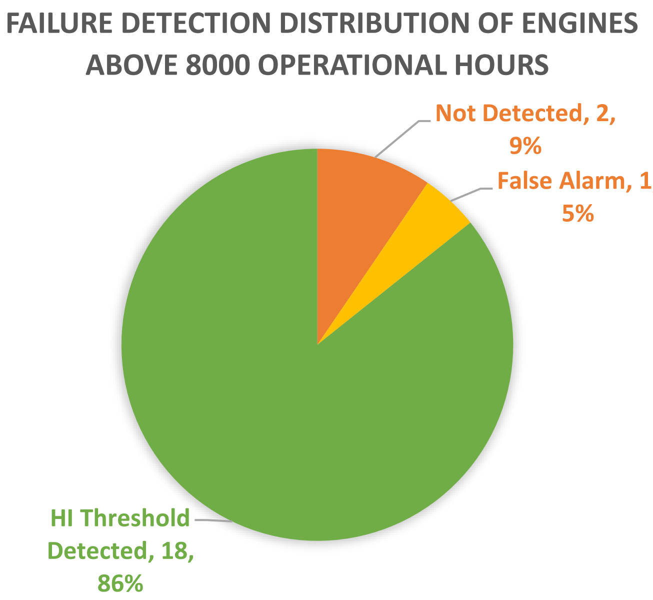


Figure 46 - Failure detection distribution of engines above 8000 operational hours

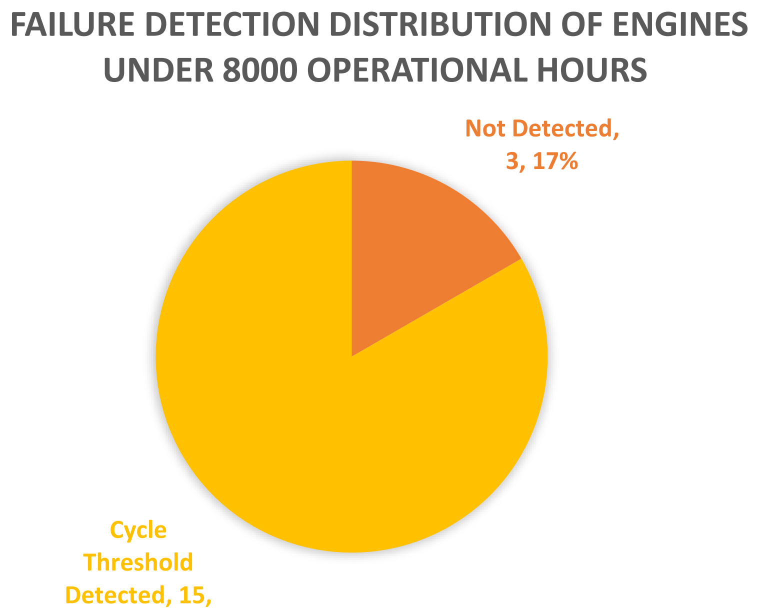


Figure 47 - Failure detection distribution of engines under 8000 operational hours

As shown in Figure 45, among the 39 engines, the threshold was not detected by the model for 4 engines However, among the 4 engines that were not detected, 2 engines do not contain both sensor data and oil data. We expect the percentage to decrease if both data were present. Another observation is that there was only one engine with an early false alarm or false positive that were classified as critical state significantly earlier than its actual failure.

Another metric that was analyzed is the residual between the actual health index at threshold prediction and the threshold. When the threshold is detected by the model, the actual health index or actual remaining useful life should be as close to the threshold as possible. The residual distribution for both HI threshold and cycle threshold is presented in the Figures below. Note that the engine that raised the false alarm is removed.

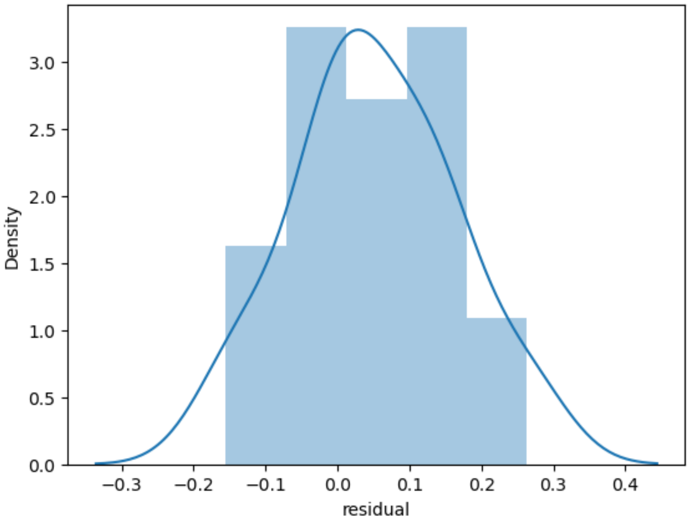


Figure 48 - Residual distribution at HI threshold detection

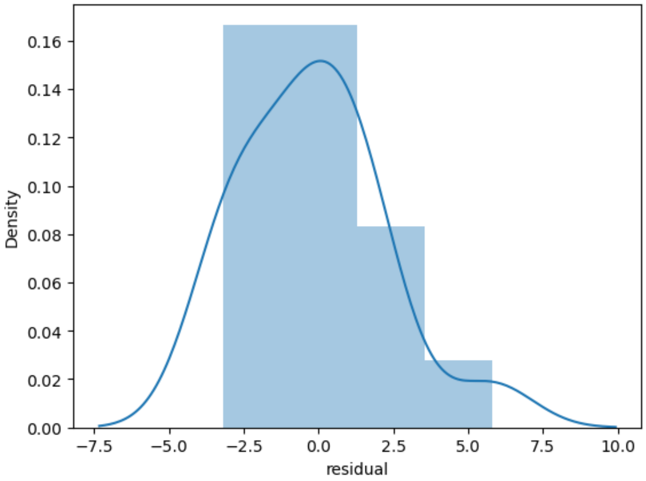


Figure 49 - Residual distribution at cycle threshold detection with outlier removed

Table 10 - RMSE and MAE at HI and cycle threshold prediction

Metric	**RMSE**	**MAE**
Residual at HI Threshold Prediction	0.12	0.09
Residual at Cycle Threshold Prediction	2.33 cycles	1.80 cycles

7.0 Conclusion and Next Step

In this project, the ensemble of sensor and oil model is proposed to predict for the remaining useful life of the engines and provide the state of the engines using the decision boundaries. The final model was able to identify critical state for 34/39 engines and failed to identify for 4/39 engines. However, amongst the 4 engines that were not predicted as critical, 2 engines did not contain both the sensor and the oil data. If both dataset exists, we expect the recall rate to improve. Additionally, only one engine amongst 35 that were identified as critical raised a false alarm or had type 1 error.

In future work, the cost analysis requires to be performed and determine the threshold based on the cost analysis. The current threshold of 0.165 is selected based on the performance of the model. Once we have further information on the cost of Type 1 Error and Type 2 Error, the model and the threshold will be optimized to reduce the overall cost.

The failure modes analyzed in the project were just wear failures on piston and crankshaft. The engines on-site are subjected to various failure modes such as coolant leakage which could show a different responses in the measurement data. To build a generalized model, we would require a larger dataset with all the failure modes for further development.

References

[1] Abhinav Saxena, Kai Goebel, Don Simon, and Neil Eklund. *Damage propagation modeling for aircraft engine run-to-failure simulation*. In 2008 international conference on prognostics and health management, pages 1–9. IEEE, 2008

[2] R. Huang, L. Xi, X. Li, C. R. Liu, H. Qiu, J. Lee, *Residual life predictions for ball bearings basedon self-organizing map and back propagation neural network methods, Mechanical Systems and Signal Processin*g 21 (1) (2007) 193–207

[3] Shuai Zheng, Kosta Ristovski, Ahmed Farahat, and Chetan Gupta. *Long short-term memory network for remaining useful life estimation*. In 2017 IEEE international conference on prognostics and health management (ICPHM), pages 88–95. IEEE, 2017

[4] Chaoub, Alaaeddine & Voisin, Alexandre & Cerisara, Christophe & Iung, Benoît. (2021). *Learning representations with end-to-end models for improved remaining useful life prognostics*.

[5] Li, Xiang & Ding, Qian & Sun, Jian-Qiao, 2018. "*Remaining useful life estimation in prognostics using deep convolution neural networks*," *Reliability Engineering and System Safety*, Elsevier, vol. 172(C), pages 1-11

[6] Han Li, Wei Zhao, Yuxi Zhang, and Enrico Zio. *Remaining useful life prediction using multi-scale deep convolutional neural network. Applied Soft Computing,* 89:106113, 2020.

[7] Wen L, Dong Y, Gao L. *A new ensemble residual convolutional neural network for remaining useful life estimation*. Math Biosci Eng. 2019 Jan 28;16(2):862-880. doi: 10.3934/mbe.2019040. PMID: 30861669.

[8] Maas, Andrew & Le, Quoc & neil, Tyler & Vinyals, Oriol & Nguyen, Patrick & Ng, Andrew. (2012). *Recurrent Neural Networks for Noise Reduction in Robust ASR.* 13th

Annual Conference of the International Speech Communication Association 2012, INTERSPEECH 2012. 1.

[9] Olah, C. (2015) *Understanding LSTM Networks*,
http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[10] Amazon (n.d.) *How XGBoost Works* Amazon.com
https://docs.aws.amazon.com/sagemaker/latest/dg/xgboost-HowItWorks.htm

Files

The files can be access through the Google Drive link below:

https://drive.google.com/drive/folders/1ANQPZCoFCNvGEHEJvfeOgyxXmezlKZO2?usp=sharing