Centre for Maintenance Optimization & Reliability Engineering

Director
Chi-Guhn Lee

Semi-annual report
June 9, 2020
Virtual meeting

Mechanical & Industrial Engineering
UNIVERSITY OF TORONTO

cmore.mie.utoronto.ca | 416·946·7867

C-MORE progress meeting agenda

Virtual meeting by Microsoft Teams
Tuesday June 9, 2020

Opening remarks, executive summary 9:00-9:20	Chi-Guhn Lee
Member presentation: Practices and Tools for Asset Management in the Mining Industry 9:20-9:50	Emilio Sarno Severi
5-minute break 9:50-9:55	

Collaborations with Consortium members

9:55-10:20	DND: Failure modal lifetime analysis of Propulsion diesel engines	Arun Shanmugam
10:20-10:45	Kinross: Analysis of maintenance KPIs effects on operational KPIs	Frank Fu

5-minute break
10:45-10:50

Collaborations with consortium members

10:50-11:15	UKMOD: An exploration of maintenance methods and their selection	Peter Gomez
11:15-11:40	UKMOD: Long-term project planning in the face of uncertainty	Janet Lam
11:45-12:05	TTC: track re-inspection frequency analysis with failure modes	

Table of contents

Executive summary
Chi-Guhn Lee, C-MORE Director

Introduction

Despite Covid-19 and its ability to touch all our lives, C-MORE has continued to work with its collaborating companies. More of our visits are virtual, perhaps, but the outcomes are no less important. The following report summarizes work undertaken since the meeting in December 2019.

The C-MORE team

Janet Lam, Assistant Director
In the first half of 2020, Janet continued to work on various projects with all member companies through direct research, as well as student supervision. Together with Chi-Guhn, Janet submitted an abstract for a half-day workshop at the MainTrain conference held by PEMAC for this coming fall in St. John, New Brunswick.

Andrew K. S. Jardine, Professor Emeritus
Professor Jardine has continued to teach aspects of Engineering Asset Management at both the graduate and post-experience levels. He also participated in the annual International Maintenance Conference (IMC) in Florida. In addition, Andrew served as external examiner for a Master's dissertation "An investigation into the benefits obtained by introducing Reliability Centred Maintenance in industrial organisations", at the Faculty of Engineering, Built Environment and Information Technology, University of Pretoria, Republic of South Africa.

Dragan Banjevic, C-MORE Consultant
In his work with C-MORE, Dragan collaborated mostly with Janet on projects with consortium members, notably with TTC and DND and to some extent with MOD and Kinross Gold. He also provided help in other projects with C-MORE students.

Sharareh Taghipour, Ryerson, External Collaborator
Sharareh is supervising/co-supervising two postdoctoral fellows and four PhD students; one of her PhD students and one MASc student recently completed their programs. Sharareh received an NSERC Discovery grant for "Decentralized Data Analytics and Optimization Methods for Physical Asset Management." She also received funding for two collaborative projects with industry: NSERC CRD, "Developing methods for measuring social, economic, and environmental impacts of maintenance activities for physical assets," with Fiix Inc.; NSERC

Alliance, “Real-time Optimization of Production Scheduling,” with Axiom Group. In addition, she received funding from the Ministry of Economic Development, Job Creation and Trade, to be matched with her John R. Evans Leaders Fund to purchase the “Industry 4.0 Smart Factory System.” The system will be used to develop predictive maintenance models and real-time optimization of production scheduling. Sharareh presented three papers at Reliability and Maintainability Symposium in January 2020 in Palm Springs. “Real-time optimization of maintenance and production scheduling for an Industry 4.0-based manufacturing system,” co-authored with PhD student, Mageed Ghaleb, won First Place for the Thomas L. Fagan, Jr., RAMS Student Paper Award.

Scott Sanner, University of Toronto

Scott has been involved in a range of applied projects covering network and power grid security, predictive modelling for residential HVAC, prediction of high resource health users with the Dalla Lana School of Public Health, uses of social media in financial applications, a new project for traffic signal control in large urban traffic networks, and a number of projects involving recommender systems for eCommerce applications. Scott recently published a paper on conversational recommender systems at the Web Conference (formerly WWW) 2020 and a journal article on optimal coordinated light rail and traffic signal control at IET Intelligent Transport Systems

Fae Azhari, University of Toronto

Fae’s research group now consists of 4 doctoral students, 3 MASc students, and 1 undergraduate student. Her projects include: complex naval asset management using sensor data, optimizing the fabrication and performance of multifunctional cementitious composites, fibre optic sensors for vibration monitoring, sensing system for gait analysis, bridge scour monitoring, condition-based maintenance of bridges, and compression creep behaviour of lead-free solder alloys. Her students Niloofar, Fredric, and Raymond presented their work at IWSHM 2019 at Stanford. Fae has been meeting with various people in the industry regarding research opportunities. Her lab was recently relocated, and she is in the process of obtaining more equipment.

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 especially busy working with a major utility company (Marafiq) to improve their Operation and Maintenance business processes and procedures. Ali has also been working with Don Barry and Steve Sinkoff to develop CMORE’s 5-day comprehensive spare parts management program.

Somayeh Alizadeh, Visiting Professor

Somayeh is an Associate Professor in the Industrial Engineering faculty at Khaje Nasir Toosi University in Iran. She joined C-MORE as a visiting scholar in July. Since her arrival, Somayeh has been researching Association Rule Mining and Sequence Pattern Mining. She has used Sequence Pattern Mining algorithms on CEA data produced through the maintenance of hydroelectric generating units. The patterns’ relevance to maintenance activities, especially sequenced activities, have been extracted using Sequence Pattern Mining algorithms. These patterns reduce downtime by determining the occurrence of the next maintenance activity or next component outage. The method helps to predict the future events of generating units and the next outage. There is a concern about a time gap among discovered events in patterns. For example, the sequence of two events (outages) with a one-month gap cannot be accepted as an interesting sequence. Therefore, she has focused on the time constraint problem in Sequence Pattern Mining. She is looking at Sequential Pattern Mining algorithms which could be able to consider the time gap problem.

C-MORE students and postdoctoral fellows

We have students at all levels working with us – from undergraduate to doctoral. C-MORE students have been extremely busy over the past six months, with many working on projects specifically related to Consortium members' concerns. Three students will be presenting at the June meeting: Arun Shanmugam, Frank Fu, and Peter Gomez. For more information on student research activities, see the section "Overall Project Direction."

C-MORE activities with consortium members

Defence Science and Technology Laboratory (DSTL)

We continued progress on a long-term procurement project; we looked at the effect of a limited budget in the success of a project, along with leveraging the projects already in progress. Peter Gomez, an MEng student, began work on exploring the maintenance taxonomy project first proposed at the December 2019 meeting. Both projects will be presented at the June meeting.

Department of National Defence (DND)

The DND team and C-MORE have been working diligently on the propulsion diesel engine health analysis project. Having identified four different failure modes, we proposed an age-based policy based on cost ratios of preventive versus corrective maintenance activities. As the costs are not available, we proposed a range of acceptable values based on existing common ratios.

Kinross Gold Corporation

Kinross proposed two projects for the first half of 2020. The first was a continuation of the KPI project. We have found a weak positive relationship between % cost from PMs and availability. Further work is being done for predictive analysis. This project will be presented at the June meeting. The second project was proving the business case for machine learning in maintenance, to show the amount of savings that could be had by using machine learning.

Toronto Transit Commission (TTC)

Since December, work has continued on the TTC reinspection project to incorporate track information into the prediction of the criticality of reinspections.

C-MORE educational programs

Andrew Jardine has continued to act as an unofficial C-MORE ambassador by teaching courses at the graduate and post-experience levels. The pandemic has changed the style of delivery; two courses were recently presented at the University of the West Indies, Trinidad, virtually by Zoom. Ali Zuashkiani is continuing to work with Don Barry and Steve Sinkoff to develop CMORE's 5-day comprehensive spare parts management program.

Conclusion

The semi-annual meeting is always a good opportunity for all of us to take stock and see exactly what we have accomplished in the past six months. I am gratified to see we are moving forward, and our work has continued relevance. At the very least, we can take pride in our adaptability as a research organization. I wish everyone good health as we move forward into the second half of 2020. Stay safe.

Chi-Guhn Lee
June 2020

Visits and interactions with consortium members and others

January 2020 – June 2020

January 7, 2020 **Teck**

Janet had a call with Gordon Kovaloff at the Trail location to give a tutorial on using SMS, spares management software.

January 10, 2020 **Kinross**

Janet and Dragan visited Emilio, Brian, and Theresa at the Kinross Toronto office to discuss strategies for this year. Two projects were proposed and refined.

January 17, 2020

C-MORE met with industry professionals Jim Reyes Picknell, Jim Jarvie, and Richard Beer to discuss possibilities in modelling reliability and maintenance using process control data, rather than maintenance data.

January 21, 2020 **Kinross**

Janet had a call with Theresa to discuss the details on some of the definitions in the KPI project.

February 5, 2020 **DND**

Janet and Arun had a call with Kulan, Ian, Jamie, and Nicolle on progress on the propulsion diesel engine project.

February 6, 2020 **FAO**

Chi-Guhn and Janet had a call with Edward Crummey from the Financial Accountability Office to discuss a potential project on predicting the maintenance costs of infrastructure under the jurisdiction of the Province of Ontario.

February 13, 2020 **FAO**

Chi-Guhn and Janet met with Edward Crummey and David West from FAO as follow up.

February 14, 2020 **Kinross**

Janet and two MEng students Syed and Aakash had a call with Emilio, Brian, Theresa, and Ethan from Kinross to discuss the objectives of the machine learning business case project.

February 24-28, 2020

Andrew participated in the PAM program, Ghana Institution of Engineering, Accra, Ghana.

February 27, 2020 **Kinross**
Janet and two MEng students, Syed and Aakash, had a call with Brian, Theresa, and Ethan from Kinross to discuss the challenges of getting sensor data from then haul trucks for the machine learning business case project.

March 3, 2020 **DND**
Janet and Arun had a call with Kulan, Jamie, and Nicolle on progress on the propulsion diesel engine project.

March 11, 2020 **Maple Leaf Foods**
Chi-Guhn, Janet, and Ali had a meeting with Faye Cooper, Mike Schultz and Mike Liu to discuss potential collaborations.

March 24, 2020 **DND**
Janet and Arun had a call with Kulan and Jamie on progress on the propulsion diesel engine project.

March 26, 2020 **Maple Leaf Foods**
Chi-Guhn and Janet had a conference call with Faye Cooper, Mike Liu, and Pat van de Bospoort to discuss C-MORE's main areas of research and potential areas of collaboration.

March 27, 2020 **Kinross**
Janet and Dragan had a conference call with Emilio, Brian, Gustavo, Theresa, and Ethan to present the progress on the KPI project.

March 28, 29, 2020
Andrew virtually presented a graduate course, "Maintenance Analysis and Optimization," for the Department of Mechanical & Manufacturing Engineering, University of The West Indies, Trinidad via Zoom.

April 2, 2020 **CEA**
Chi-Guhn and Janet had a call with Dan Gent from CEA to discuss a potential project on a data quality assessment project.

April 18, 19, 2020
Andrew virtually presented a graduate course "Maintenance Analysis and Optimization" for Department of Mechanical & Manufacturing Engineering, University of The West Indies, Trinidad via Zoom.

April 23, 2020 **DND**
Janet and Arun had a call with Kulan and Jamie on progress on the propulsion diesel engine project.

May 6, 2020 **TTC**
Chi-Guhn and Janet had a call with Luigi, Mostafa, Aleks, Tauqeer, Jennifer, and Hossein to discuss continued membership with C-MORE.

May 7, 2020
Chi-Guhn and Janet had a call with Professor Tamel El-Diraby to discuss potential projects with the Ministry of Infrastructure in costing the maintenance of Ontario's infrastructure.

C-MORE leadership activities

Chi-Guhn Lee, Director

Chi-Guhn continued to lead C-MORE in research projects and industry partnerships throughout this term. He gave several talks on integrating machine learning technologies in the area of physical asset management. In particular, he gave a talk on the general introduction to artificial intelligence to government officials, diplomats and other association leaders as part of The Korean Embassy Speakers Series in Ottawa in December 2019. He is working steadily on developing relationships with various organizations to increase C-MORE's presence in the area of asset management in the face of emerging technologies.

Janet Lam, Assistant Director

In the first half of 2020, Janet continued to work on various projects with all member companies through direct research, as well as student supervision. Together with Chi-Guhn, Janet submitted an abstract for a half-day workshop at the MainTrain conference held by PEMAC for this coming fall in St. John, New Brunswick.

Andrew K. S. Jardine, Professor Emeritus

Professor Jardine has continued to teach aspects of Engineering Asset Management at both the graduate and post-experience levels. He also participated in the annual International Maintenance Conference (IMC) in Florida. In addition, Andrew served as external examiner for a Master’s dissertation “An investigation into the benefits obtained by introducing Reliability Centred Maintenance in industrial organisations”, at the Faculty of Engineering, Built Environment and Information Technology, University of Pretoria, Republic of South Africa.

Dragan Banjevic, C-MORE Consultant

In his work with C-MORE, Dragan collaborated mostly with Janet on projects with consortium members, notably with TTC and DND and to some extent with MOD and Kinross Gold. He also provided help in other projects with C-MORE students.

Sharareh Taghipour, Ryerson University, External Collaborator

Sharareh is supervising/co-supervising two postdoctoral fellows and four PhD students; one of her PhD students and one MASc student recently completed their programs. Sharareh received an NSERC Discovery grant for "Decentralized Data Analytics and Optimization Methods for Physical Asset Management." She also received funding for two collaborative projects with industry: NSERC CRD, "Developing methods for measuring social, economic, and environmental impacts of maintenance activities for physical assets," with Fiix Inc.; NSERC Alliance, "Real-time Optimization of Production Scheduling," with Axiom Group. In addition, she received funding from the Ministry of Economic Development, Job Creation and Trade, to be matched with her John R. Evans Leaders Fund to purchase the "Industry 4.0 Smart Factory System." The system will be used to develop predictive maintenance models and real-time optimization of production scheduling. Sharareh presented three papers at Reliability and Maintainability Symposium in January 2020 in Palm Springs. "Real-time optimization of maintenance and production scheduling for an Industry 4.0-based manufacturing system," co-authored with PhD student, Mageed Ghaleb, won First Place for the Thomas L. Fagan, Jr., RAMS Student Paper Award.

Scott Sanner, University of Toronto

Scott has been involved in a range of applied projects covering network and power grid security, predictive modelling for residential HVAC, prediction of high resource health users with the Dalla Lana School of Public Health, uses of social media in financial applications, a new project for traffic signal control in large urban traffic networks, and a number of projects involving recommender systems for eCommerce applications. Scott recently published a paper on conversational recommender systems at the Web Conference (formerly WWW) 2020 and a journal article on optimal coordinated light rail and traffic signal control at IET Intelligent Transport Systems.

Fae Azhari, University of Toronto

Fae's research group now consists of 4 doctoral students, 3 MASc students, and 1 undergraduate student. Her projects include: complex naval asset management using sensor data, optimizing the fabrication and performance of multifunctional cementitious composites, fibre optic sensors for vibration monitoring, sensing system for gait analysis, bridge scour monitoring, condition-based maintenance of bridges, and compression creep behaviour of lead-free solder alloys. Her students Niloofar, Fredric, and Raymond presented their work at IWSHM 2019 at Stanford. Fae has been meeting with various people in the industry regarding research opportunities. Her lab was recently relocated, and she is in the process of obtaining more equipment.

Jue Wang, Affiliate Professor

For the first part of 2020, Jue continued his work in optimizing the operations of smart and connected products. One paper he co-authored was recently accepted for publication by *Operations Research*, the flagship journal of his field, and he just submitted a revision of another paper to *Operations Research* on the topic of quick fault diagnosis with sensors. A new paper about online fine-tuning is now under minor revision at *Production & Operations Management*.

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 especially busy working with a major utility company (Marafiq) to improve their Operation and Maintenance business processes and procedures. Ali has also been working with Don Barry and Steve Sinkoff to develop CMORE's 5-day comprehensive spare parts management program.

Somayeh Alizadeh, Visiting Professor

Somayeh is an Associate Professor in the Industrial Engineering faculty at Khaje Nasir Toosi University in Iran. She joined C-MORE as a visiting scholar in July. Since her arrival, Somayeh has been researching Association Rule Mining and Sequence Pattern Mining. She has used Sequence Pattern Mining algorithms on CEA data produced through the maintenance of hydroelectric generating units. The patterns' relevance to maintenance activities, especially sequenced activities, have been extracted using Sequence Pattern Mining algorithms. These patterns reduce downtime by determining the occurrence of the next maintenance activity or next component outage. The method helps to predict the future events of generating units and the next outage. There is a concern about a time gap among discovered events in patterns. For example, the sequence of two events (outages) with a one-month gap cannot be accepted as an interesting sequence. Therefore, she has focused on the time constraint problem in Sequence Pattern Mining. She is looking at Sequential Pattern Mining algorithms which could be able to consider the time gap problem.

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 January 2020 – June 2020. As with everyone else, C-MORE's activities have had to pivot in light of the COVID-19 crisis. In particular, some of the conferences were cancelled, leaving us to focus on member projects.

C-MORE has entered into an agreement with LEORON to offer a virtual Physical Asset Management 2 course in July. The original in-person course offered through the School of continuing studies have been rescheduled.

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	We continued progress on the long-term procurement project; we looked at the effect of a limited budget in the success of a project, along with leveraging the projects that are already in progress. Peter Gomez, an MEng student, began work on exploring the maintenance taxonomy project first proposed at the December 2019 meeting. Both projects will be presented today.
Department of National Defence	The DND team and C-MORE have been working diligently on the propulsion diesel engine health analysis project. Having identified four different failure modes, an age-based policy was proposed based on cost ratios of preventive versus corrective maintenance activities. Since the costs are not available, a range of acceptable values were proposed based on existing common ratios.

Member	Collaborations
Kinross	Kinross proposed two projects for the first half of 2020. The first was a continuation of the KPI project. We found a weak positive relationship between % cost from PMs and availability. Further work is being done for predictive analysis. This project will be presented today. The second project was proving the business case for machine learning in maintenance to show the savings that could be had by using machine learning.
Toronto Transit Commission	Work has continued on the TTC re-inspection project to incorporate track information into the prediction of the criticality of re-inspections.

Theoretical work

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

Name	Activity
Kuilin Chen, PhD student	Kuilin completed his qualifying exam in November 2019. He completed the work of developing digital twins using hybrid models of Transformer and Gaussian Processes, termed Attentive-GP. An abstract on probabilistic sequence-to-sequence learning based digital twin was accepted for the Reliability Conference 2020. In addition, he submitted the Attentive-GP paper to the Thirty-fourth Conference on Neural Information Processing Systems (NeurIPS), June 2020. Kuilin is working on Bayesian distributional reinforcement learning for efficient and risk-sensitive exploration.
Jiacheng (Frank) Fu, MEng student	Frank is working on the Kinross KPI project. The project started in January 2020 and involves determining whether an increase in planned maintenance time increases reliability/availability using KPI measures, as well as developing machine learning model(s) to predict reliability/availability in the future. In the past three months, Jiacheng has performed detailed exploratory data analysis on the dataset provided by Kinross Gold and organized by Dr. Lam. This included investigating the meaning of the data, searching and visualizing the patterns and trends, calculating correlations and conducting T-tests to find the most suitable features for linear regression models. He then trained and implemented linear regression models to predict each target (i.e. availability, utilization, etc.) using techniques such as grid search for hyperparameter tuning. At the current stage, each model is being evaluated by using hypothesis testing on the root mean squared error of the outputs.
Anmol Garg, MEng student	Anmol is a 1st year MEng Student in Mechanical and Industrial Engineering. In the pursuit of a career in data analytics, he has taken introductory courses. Now, in a real-world setting, he is involved in a project under Professor Chi-Guhn Lee which deals with detecting

Name	Activity
	banned items in baggage using a Convolutional Neural Network (CNN) which is meant to be employed in airport scanning systems.
Michael Gimelfarb, PhD candidate	Michael has continued his doctoral work on knowledge transfer in reinforcement learning using graph-structured data, Bayesian approaches and hierarchical RL. He is supervised by Scott Sanner and Chi-Guhn Lee.
Peter Gomez, MEng student	Peter began work on exploring the maintenance taxonomy project with DSTL. His work will be presented at the June meeting.
Aakash Iyer, MEng student	Aakash is a 2nd year graduate student pursuing a Master's of Engineering in Mechanical and Industrial Engineering with a technical specialization in Data Analytics, and an emphasis on ELITE – Finance & Management major. His main project under the supervision of Professor Chi-Guhn Lee deals with object detection and classification for baggage security scanning along with the development of an early warning system for hazardous object detection.
Scott Koshman, PhD student	Scott has continued his doctoral research on equipment health monitoring for Halifax Class frigates. He is supervised by Professor Fae Azhari.
Syed Hamdan Mustafa, MEng student	Syed Hamdan is pursuing his Master's of Engineering in Mechanical and Industrial Engineering with an emphasis in Data Analytics. He is working under the supervision of Dr. Chi-Guhn Lee on a project exploring a transfer learning approach to a security object detection scan task to be employed in airports. Starting in January 2020, he has also worked on the Kinross Machine Learning Business Case project which aims at optimizing the maintenance strategy of truck engines using predictive modelling.
Seyedvahid Najafi, PhD student	Vahid is a full-time PhD student who works on the maintenance modelling and optimization of multi-unit series systems. In his research, he is developing an opportunistic maintenance policy with general repair for a two-unit series system, in which the condition of one unit is monitored, and only the age information is available for the other unit. The problem is formulated in a semi-Markov decision process framework, and an algorithm is developed to find the optimal control limits and the long-run average cost per unit time. A paper entitled "An optimal maintenance policy for a two-unit series system with general repair" was accepted by IISE Annual Conference & Expo 2020, New Orleans, Louisiana, USA. He passed his qualifying exam in January 2020 and will be ready for his first annual progress review meeting in the beginning of 2021.
Arun Shanmugam, MEng student	Arun is an MEng student in Mechanical and Industrial Engineering. He has worked on the reliability modelling of propulsion diesel engines in Halifax Class frigates for the Department of National Defense (Navy) over the past eight months. Key activities include:

Name	Activity
	construction of event history from maintenance data; failure mode statistical analysis; development of decision models to determine optimal preventive replacement age for each failure mode.
Iris Shi and Runting Yang, Undergraduate students	Iris and Runting are both undergraduate students who will be starting on a project focusing on the COVID-19 pandemic. More specifically, they are looking into the impact of the lockdowns on the daily activities of video websites and if, or under what specific conditions, a website's policies (such as free membership) boost the profits.
Jia Qi (Cherry) Xi and Juntian (Mia) Zhang, MEng students	Cherry and Mia are both MEng Students in Department of Mechanical and Industrial Engineering. Their research project deals with the maintenance data from a hydroelectric power generator under CEA. By conducting data analysis and research on sequential pattern mining methodologies, they are aiming to identify maintenance patterns and sequential rules to support future maintenance decisions at the power plant. Major progress to date includes a survey on published algorithms and data explorations.
Songci Xu, PhD student	Songci started his first year PhD program in January 2020. He is working with the visiting PhD student, Bin Yang, and his research focuses on applying transfer learning to Intelligent Fault Diagnosis (IFD), one of the projects of LG. He is also working on the explanatory ability of deep domain adaptation to avoid the Negative Transfer; this will be part of his thesis.
Yushi Wang, MEng student	Yushi is an MEng student in Mechanical and Industrial Engineering working on the real asset procurement project of the generic real option problem under the swing option assumption using Reinforcement Learning. He is doing an investigation to perfect the formulation and solve it for reasonable numeric results.
Bin Yang, Visiting PhD student	Bin continues research on the applications of transfer learning to fault diagnosis of machines. In the past six months, he mainly worked on a project on partial transfer learning. This project focuses on the realistic assumption in engineering scenarios that the source domain contains sufficient diagnostic knowledge while the target domain just require partial. Bin has completed an article based on this work, which will be submitted to a journal in the field.
Ruiqi Yang, MEng student	Ruiqi Yang is a 2nd year graduate student pursing his Master's of Engineering in Mechanical and Industrial Engineering. He is now working on the MEng project "Applied Markov decision process in real option valuation" with his teammates and under Professor Lee's supervision.
Yige Zhang, MEng student	Yige is an MEng student who is working on a Real Option Pricing Project based on the UKMOD Project. Yige and his teammates are analysing the accessibility of the algorithm inspired by "Market self-

Name	Activity
	learning of signals, impact and optimal trading: invisible hand inference with free energy.”
Zihan Zhang, MASc student	During the past six months, Zihan has finished two courses (Markov Decision Process, Scientific Writing Course), continued a literature review in PHM application and proposed four project proposals: (a) hierarchical maintenance optimization considering replacement impact; (b) short-time-Fourier-transform-based fault diagnosis using Graph Convolution Network; (c) health-oriented group maintenance; and (d) prognosis management for rechargeable batteries integrating deep neural network and physical-stochastic processes. She will proceed with a battery project in summer session with undergraduates. She received a merit-based MIE Endowed Scholarship in the winter session and her two co-authored papers have been published in *Reliability Engineering & Safety Systems* and *IEEE Transactions on Sustainable Energy*.
Avi Sokol, PhD student	As a flex-time PhD student and a full-time Business Data Scientist and Inventory Specialist, Avi continues to research integration of Reinforcement Learning and Inventory Control to reduce waste in supply chains. In the past six months, Avi did a literature review on application of deep reinforcement learning in supply chains, reward function decomposition, and explainability & interpretability of the models to prepare for the qualifying exam scheduled for summer 2020.

CEA data mining project progress report

Jia Qi (Cherry) Xi, Juntian (Mia) Zhang

Project overview

Canadian Electricity Association (CEA) is a leading national electricity association in Canada founded in 1891. CEA provides value-added products and services to advance the strategic interest of the Canadian electricity industry. CEA currently uses an Equipment Reliability Information System (ERIS) to record continuous operating and outage data, along with timestamps.

This project is to generate algorithms and codes to find patterns and rules regarding the outage events in a sequence and make appropriate operation and maintenance decisions in the future. The algorithm output should contain the probability of high-related event occurrence in a specified time interval.

In the early stage of the project, we mainly focused on researching available algorithms and codes. We compared performance and efficiency of algorithms to find the most appropriate and efficient algorithms for this project and applied more new codes to provide algorithms with new functions and higher efficiency to fit better the requirement of CEA.

Based on the datasets provided by CEA, data pre-processing is required to filter out noise and outliers. Among the many open-source libraries available to apply sequential pattern mining algorithms, the SPMF library was chosen as the basic coding tool for our case study. While it has a specific input format requirement, data are transformed into a sequential database with the required input format.

Sequential pattern mining and sequential rule are the two main algorithm types used in this project to analyze datasets and discover frequent subsequences and rules among the transactions. The final outcome of the project will be a set of preventive maintenance policies based on the output of the selected data mining algorithms.

Basic concepts and algorithms

Pattern mining key terms

This section introduces some key terms about pattern mining and explains the definition of sequential pattern mining and sequential rule.

Algorithms for sequential pattern mining

There are five main types of algorithms for sequential pattern mining: horizontal database approach (Apriori, GSP), vertical database (SPADE, SPAM, CM-SPADE, CM-SPAM), pattern growth(PrefixSpan), closed sequential(CloSpan), constraint-based closed sequential(BIDE, Generalized Sequential Pattern Mining with Item Intervals), and hybrid(LASH) [1].

Algorithms for sequential rule mining

Several algorithms have been proposed for sequential rule mining; they can be divided into two groups, with time constraints and without time constraints. The typical algorithm with time constraints is TRuleGrowth; ERMiner is a sequential rule mining algorithm without time constraints [2].

Algorithm comparison

We have created a table comparing algorithms by their proposal year, category, availability for constraint, and big data support.

Available code resources

SPMF is an open-source data mining library written in Java, specialized in pattern mining (the discovery of patterns in data) [3].

MLlib is Apache Spark's scalable machine learning library usable in Java, Scala, Python and R. It offers high-quality algorithms and can be compiled on diverse platforms such as Hadoop, Apache Mesos [4].

Case study

Data pre-processing

The case study used the continuous operation and outage data of the hydroelectric and fossil generating units. Before implementing the pattern mining algorithms from the SPMF library, we conducted a series of procedures including data exploration, noise cleaning and data transformation to pre-process the raw data. Since the raw data only record a single item in each event, a key step in the pre-processing is the aggregation of multiple items within a chosen time-window into a single sequential transaction. The resulting transaction database is transformed into the required format of the SPMF library.

Algorithm implementation

Algorithms from both sequential pattern mining and sequential rule mining are selected for implementation. We used TRuleGrowth and ERMiner to discover sequential rules, whereas

HirateYamana, CloSpan, and CMSpade are applied to identify sequential patterns. Among these selected algorithms, TRuleGrowth and HirateYamana consider time-related factors in which constraints such as window size and time interval can be enforced. For each algorithm, three scenarios with different input parameters and constraints are designed to obtain patterns or rules under various settings.

Results and discussion

We organized the results obtained for each scenario and algorithm in charts and compared the performance of algorithms through graphics. We also derived the output sequential rules and sequential patterns of forced outage into preventive maintenance policies, which are valuable references for operating decisions.

Proposed model

While sequential pattern mining algorithms have used support as the principal measure or constraint, this is too general for a complex problem. For our type of case study with many types of components, outage events with specific time-stamp, and data stretched over a long time span, support is not reflective as it can be affected by the size of the dataset and it does neglect the effect of time in the calculation. Therefore, enhanced support is proposed to take into account the time, which is a critical element in the pattern mining outcome leading towards a useful maintenance plan or policy. In addition, a three-phase procedure is outlined as a modified implementation to obtain enhanced support:

Phase 1 - Break the dataset down into subsets of periods according to a set time window.
Phase 2 - Apply sequential pattern mining algorithms on each of the subsets.
Phase 3 - Calculate the support and enhanced support for each discovered sequential pattern.

References

[1] GAN, W. and LIN, J., 2019. A Survey of Parallel Sequential Pattern Mining. ACM Trans. Knowl. Discov. Data, [online] 0(1). Available at: <https://arxiv.org/pdf/1805.10515.pdf> [Accessed 21 May 2020].
[2] Fournier-Viger, P., Wu, C., Tseng, V., Cao, L. and Nkambou, R., 2015. Mining Partially-Ordered Sequential Rules Common To Multiple Sequences. [online] Philippe-fournier-viger.com. Available at: <https://www.philippe-fournier-viger.com/spmf/TKDE2015_sequential_rules.pdf> [Accessed 21 May 2020].
[3] Philippe-fournier-viger.com. 2020. SPMF: A Java Open-Source Data Mining Library. [online] Available at: <http://www.philippe-fournier-viger.com/spmf/index.php> [Accessed 21 May 2020].
[4] Spark.apache.org. 2020. Mllib | Apache Spark. [online] Available at: <https://spark.apache.org/mllib/> [Accessed 21 May 2020].

Maintenance modelling and optimization of series systems considering general repair

Seyedvahid Najafi

Introduction

Modern manufacturing systems are composed of multiple components in which a high level of technology is applied. Machines wear out with age and usage, and if no appropriate plan is designed, unexpected failures can incur considerable costs. Failures of multi-unit systems can result in not only high maintenance costs but also industrial accidents. So, many companies try to apply appropriate maintenance policies to increase workplace safety and reduce maintenance costs.

A failure occurs when a system is unable to perform a required function. When a system fails, corrective maintenance (CM) returns it to operation. However, preventive maintenance (PM) can be performed to reduce the risk of failure when the system is still operational. When a unit fails in a system with economic dependency between units, opportunistic maintenance (OM) can be performed to improve the condition of other operational units.

Berg [1] applied a dynamic semi-Markov program to find an optimal policy for a two-unit system in which, opportunistic replacement is performed, if the age of the operational unit exceeds a control limit. Berrade et al. [2] considered interactions between the units of a two-unit system.

Repair actions can be classified into three categories, based on the degree of rejuvenation: perfect repair, imperfect repair (i.e., general repair), and minimal repair. Perfect repair completely restores a system to statistically as-new condition. General repair improves the condition of a system to a state between as-new and as-old [3]. Minimal repair is performed to return a failed unit to the same operational condition just before failure (as-old) [4]. Kijima et al. [3] studied a repairable system that is periodically replaced and received general repair upon failures. Kijima [5] represented two models to consider the effect of general repair on virtual age. Model Ⅰ, $V_n = V_{n-1} + A_n X_n$, assumes that each general repair reduces the deterioration incurred since the latest repair. In model Ⅱ, which is represented as $V_n = A_n(V_{n-1} + X_n)$, general repair reduces the deterioration of the system up to now, where $A_n \in [0,1]$ is a random variable showing the degree of nth repair. Extreme values 0 and 1 represent minimal and perfect repair, respectively.

The deterioration of a system is generally influenced by not only its age, but several factors referred to as covariates [6]. CM data can be utilized by a proportional hazards model (PHM), which takes into account both age and covariates and it is widely used to estimate the failure risk of units. Makis and Jardine [7] studied a system subjected to the PHM in which the values of a stochastic process Z (i.e., covariates) are known at discrete decision epochs. Optimality equations are derived, and the form of optimal replacement policies is examined. Jafari et al. [8] applied the SMDP to find an optimal opportunistic maintenance policy for a multi-unit system with two main units. CM data are used in a PHM to estimate the failure risk of unit 1, and age information is available for the other unit. More examples on the application of the PHM in CBM can be found in [9], [10] and [11].

System description

The system consists of two units, the more expensive unit (unit 1) is subject to CBM, and the age information for the second unit is available. When one of the units fails, the whole system fails, and failure is self-announcing. The deterioration of unit 1 is described by the gamma process $\{X_t | t \geq 0\}$. A Weibull PHM distribution is applied to estimate the hazard rate of unit 1.

To be able to formulate the problem in the SMDP framework, the joint age and deterioration process of unit 1, denoted by $Y_t = (t, X_t)$, is required to be discretized. A matrix-based approximation method, which was developed by Brook and Evans [6], is applied to calculate the conditional reliability of unit 1 over its lifetime.

Policy structure

The system is inspected at equidistant epochs $(\Delta, 2\Delta, \dots, N\Delta)$ and the hazard rate of unit 1 is calculated by Equation (1). Preventive maintenance (PM), corrective maintenance (CM) and opportunistic maintenance (OM) is performed on the system considering a hazard rate control limit L_1 for unit 1 and an age-based control limit L_2 for unit 2. If the age of the system reaches N, the whole system is replaced, otherwise, the following preventive actions are performed:

- PM for unit 1: At inspection epochs, if the hazard rate of unit 1 is less than L_1, no action is taken on unit 1; otherwise, general repair is performed.
- PM for unit 2: If the age of unit 2 is less than L_2, no action is taken on unit 2; otherwise, the unit is replaced.

If one of the units fails, and the age of the system is $N-1$, the whole system is replaced; otherwise, the following CM actions are performed:

- CM for unit 1: When unit 1 fails, if its hazard rate is less than L_1, minimal repair returns it to operation; otherwise, general repair is performed.
- CM for unit 2: When unit 2 fails, if its age is less than L_2, minimal repair is performed on unit 2; otherwise, it is replaced.
- OM for unit 1: When unit 2 fails, there is an opportunity to improve unit 1, and if unit 1 hazard rate is less than L_1, no action is taken; otherwise, general repair is performed on unit 1.
- OM for unit 2: When unit 1 fails, if the age of unit 2 is less than L_2, no action is taken on unit 2; otherwise, unit 2 is replaced.

State definition

In this section, the required elements are defined to solve the maintenance problem in the SMDP framework. The state space of the whole system can be expressed as:

$$\mathrm{S} = \{(\mathrm{z, i, j, g, f}) | \mathrm{z} \in \Omega, 0 \leq i, \mathrm{j} \leq \mathrm{N}, \mathrm{g} \in \{0,1\}, f \in \{0, F_1, F_2\}\} \tag{13}$$

Unit 1 and unit 2 age are represented by $i\Delta$ and $j\Delta$, and the maximum useful age of the system is N. If general repair is performed on unit 1, then element g equals 1, otherwise 0. The failure or operational status of the system is represented by the element f: if the system is operational, then $f = 0$, and F_1 and F_2 indicate the failure of unit 1 and unit 2, respectively. Actions (a_1, a_2) are performed on unit 1 and unit 2, respectively, where $a_1, a_2 \in A = \{0,1,2,3\}$, and actions consist of doing nothing (0) minimal repair (1), general repair (2), and replacement (3). If the deterioration of unit 1 is z, it will be reduced to max{0,(z- δ)} after general repair.

Considering the first step of the policy-iteration algorithm, the following system of linear equations are solved to obtain the optimal long-run expected average cost per unit time $g(L_1, L_2)$ and optimal control limits [9]:

$$\begin{aligned} & v_l = c_l(a_1, a_2) - g(L_1, L_2).\tau_l(a_1, a_2) + \sum_{m \in S} P_{l,m}(a_1, a_2) v_m,\ l \in S \\ & v_n = 0, \text{for an arbitrarily selected state } n \in S \end{aligned} \tag{14}$$

where $P_{l,m}(a_1, a_2)$ is the probability of transition from the current state l to next state m, when action (a_1, a_2) is performed. $\tau_l(a_1, a_2)$ and $C_l(a_1, a_2)$ represent the expected sojourn time and the expected cost of the system, respectively, given the action (a_1, a_2) is selected in the current state l.

Numerical example

We considered a feed subsystem of a boring machine as unit 1 whose data was published in Duan et al. [7], and the positioning accuracy of the subsystem is described by a gamma process, which indicates its degradation. The shape parameter and the scale parameter of the process are $\alpha(t) = 0.276t$ and $\beta = 4.86$. The hazard rate of unit 1 is described by a Weibull PHM with the scale parameter $\lambda = 85.42$, the shape parameter $k = 4.63$, and the covariate coefficient is $\theta = 0.281$. Unit 2 failure time follows Weibull distribution with the scale and shape parameters 85.42 and 4.63, respectively. Time dependent parameters are transformed from hours to days, and the maximum number of deterioration states is $D = 8$. Inspection interval is $\Delta = 10$ days, which has been partitioned into 10 intervals with the length of $d = 1$ day. The whole system is replaced when its age reaches $N\Delta = 50$ days.

Different combinations of control limits $L_1$and L_2 are considered to find the optimal limits which minimizes the long-run expected average cost per unit time, and some results are presented in Table 1. As can be seen from the table, the proposed policy (policy Ⅰ) is compared with the policy Ⅱ whose structure is the same as policy Ⅰ except that instead of general repair, replacement is performed. The results show that policy Ⅰ outperforms policy Ⅱ, and considering the optimal hazard rate control limit $L_1{}^* = 2.87\mathrm{E} - 02$ and the age control limit $L_2{}^* = 5\Delta$ results in the optimal long-run expected average cost rate $g^*\ (L_1{}^*, L_2{}^*) = \$126.38$.

Table 1. Long-run expected average cost per unit time under different control limits

	Policy Ⅰ	Policy Ⅱ	Policy Ⅰ	Policy Ⅱ	Policy Ⅰ	Policy Ⅱ
L_1 L_2	Δ	Δ	3Δ	3Δ	5Δ	5Δ
2.62E-05	$372.80	$529.96	$299.06	$451.52	$313.14	$437.45
3.24E-04	$311.89	$357.20	$230.15	$268.98	$226.55	$290.23
4.01E-03	$253.01	$266.11	$167.93	$170.50	$144.84	$209.08
2.87E-02	$247.78	$248.16	$156.70	$157.11	$126.38	$129.60

References

[1] M. Berg, “Optimal replacement policies for two-unit machines with increasing running costs 1,” Stoch. Process. their Appl., vol. 4, no. 1, pp. 89–106, 1976.

[2] M. D. Berrade, P. A. Scarf, and C. A. V Cavalcante, “Conditional inspection and maintenance of a system with two interacting components,” Eur. J. Oper. Res., vol. 268, no. 2, pp. 533–544, 2018.

[3] M. Kijima, H. Morimura, and Y. Suzuki, “Periodical replacement problem without assuming minimal repair,” Eur. J. Oper. Res., vol. 37, no. 2, pp. 194–203, 1988.

[4] R. E. Barlow and F. Proschan, “Mathematical theory of reliability” New York, 1965.

[5] M. Kijima, “Some results for repairable systems with general repair,” J. Appl. Probab., vol. 26, no. 1, pp. 89–102, 1989.

[6] D. R. Cox, “Regression models and life-tables,” J. R. Stat. Soc. Ser. B, vol. 34, no. 2, pp. 187–202, 1972.

[7] V. Makis and A. K. S. Jardine, “Optimal replacement in the proportional hazards model,” INFOR Inf. Syst. Oper. Res., vol. 30, no. 1, pp. 172–183, 1992.

[8] L. Jafari, F. Naderkhani, and V. Makis, “Joint optimization of maintenance policy and inspection interval for a multi-unit series system using proportional hazards model,” J. Oper. Res. Soc., vol. 69, no. 1, pp. 36–48, 2018.

[9] H. R. Golmakani, “Optimal age-based inspection scheme for condition-based maintenance using A* search algorithm,” Int. J. Prod. Res., vol. 50, no. 23, pp. 7068–7080, 2012.

[10] M.-Y. You, H. Li, and G. Meng, “Control-limit preventive maintenance policies for components subject to imperfect preventive maintenance and variable operational conditions,” Reliab. Eng. Syst. Saf., vol. 96, no. 5, pp. 590–598, 2011.

[11] X. Wu and S. M. Ryan, “Optimal replacement in the proportional hazards model with semi-Markovian covariate process and continuous monitoring,” IEEE Trans. Reliab., vol. 60, no. 3, pp. 580–589, 2011.

Kinross KPI project progress report: linear regression

Jiacheng Fu, MEng student

Introduction

In modern industry, as more human labour is replaced by machinery, maintenance starts to play an important role. Conducting regular maintenance can keep machines and equipment in good condition which can not only improve their efficiency of production but also reduces unnecessary costs due to failure. Thus, Kinross Gold, as one of the leading enterprises in the field of gold mining, proposed a project on investigating the pattern of maintenance time. The project was divided into two parts by objective. The first part aims to provide insights about the trend of truck/shovel availability with increasing planned maintenance time. It answers the questions such as does increasing planned maintenance time will increase availability of trucks and shovels. The second part of the project is to predict the availability using the given KPI measures as the inputs. Thus, this report aims to demonstrate the most recent progress of the project which uses linear regression model to reach the above project objectives.

Methodology

As mentioned in the previous section, the overall methodology uses linear regression model to explore the pattern between planned maintenance time and availability as well as making predictions. The entire process was done on Google Colab using Python programming language. The general idea is to build linear regression models between KPI measures(as features) and availability as well as other related variables(as the targets) to find which KPI measures have strong impacts on the targets by examining the coefficients of the linear model. In addition, the linear model with low percentage error can also be used for making predictions on the targets. The investigation was broken down into four steps: data cleaning, feature selection, linear model implementation and model evaluation.

Data cleaning

Data cleaning includes dropping null values, extracting and organizing relevant data. The purpose of this step is to create data frames and numpy arrays that contains KPI measures that will be used as features and variables that will be used as targets so that one can easily feed them into the linear algorithms later on. Thus, the KPI measures used as features (predictor variables) are:

- % of late PM (planned maintenance)
- % proxy for planned
- % of weekly schedule compliance
- % cost from PMs
- Cost

As mentioned, availability is the research target for this project. However, it is still wise to explore the variables that are highly related to availability in order to draw useful insights. Therefore, here is a full list of target variables:

- Availability
- Utilization
- Effective utilization (UofA)
- Hours
- Events (Evts)
- Mean time between failure (MTBF)

Feature selection

To increase the model accuracy and reducing the computational costs, it is very important to choose the highly related features and abandoning the ones that have less contributions to the targets. The features were selected based on two criteria: the correlation coefficient and p values of slope generated from the f and t tests.

So let’s first look at the correlation coefficients. The author examined the correlation between each pair of predictor variable and target variable. Correlation coefficient of each combination gives the information of the strength of the linear relationship between the variables. The threshold value of correlation coefficient was set to be 0.2. Thus, if the correlation coefficient is above 0.2, the predictor variable is linearly related to the target variable which indicates that the predictor variable can be used as the input of the linear regression model that predicts that particular target. Otherwise, the predictor variable should not be used since the relationship is not strong enough to be recognized as linear. Appendix I shows the correlation coefficients between targets and features with the ones that are above the threshold being highlighted. In addition, the correlation coefficients between features were calculated to find highly correlated features so the ones that were highly related to the features that were already selected as the inputs were dropped.

The next assessment performed on the predictor variables is the hypothesis testing including both f and t-test. In order to do so, ordinary least square (OLS) models were first implemented to each target variables with the original five features. As the name suggests, the ordinary least square model is the linear model that chooses the parameters of a linear function of a set of explanatory variables by the principle of least squares which is minimizing the sum of the squares of the differences between the observed dependent variable in the given dataset and those predicted by the linear function.[1] Then, f-tests were performed on each model to see whether the OLS model for each target variable was statistically reliable. A t-test was performed on each model to see if each predictor variable used by the model had sufficient contribution to the results.

The goal of the f-test is to evaluate the overall strength of the OLS model. It is calculated based on all of the model coefficients. Unlike the correlation coefficient, the f test aims tell the researcher about the magnitude of the linear model coefficients given that the relationship has already been identified as linear. Thus, the null hypothesis of the f-test is that all the model coefficients equal to zero whereas the alternative hypothesis is that at least one of the coefficients is a non-zero value.

Thus, the f-test determines whether the proposed linear relationship between the response variable and the set of predictors is statistically reliable and can be useful when the research objective is either prediction or explanation. The significance level of f-test was set to be 0.1. So if the OLS model has a p value below the significance level, the linear model will be considered as inappropriate and it should be completely abandoned.

The second hypothesis test was the t-test. Unlike the f-test, the t-test evaluates each individual regressor rather than the entire model. Simply speaking, the significance of each individual model coefficient is assessed. At first, each OLS model has 5 p-values corresponding to 5 features. So the null hypothesis here is that the individual coefficient inside the model is equal to zero and the alternative hypothesis says the opposite. As one can see, the goal of the t-test is to check whether certain feature has significant impact on the target variables based on its coefficient. The larger the coefficient, the more significant impact it has. The significance level for the t-test is the same as the f-test. For features with p value below the significance level, that feature can be considered as useless when predicting the target. In that case, we should drop that feature and rebuild the model for the next evaluation until all the features used in the model can reject the null hypothesis and that is exactly that the author did. In addition, every time the model was reconstructed with modified features, the f-test was performed again on the new model to ensure the model was still statistically reliable.

After conducting both correlation coefficients calculation and hypothesis testing, features that passed both test were selected. Appendix II shows the selected features for each target variable.

Model training and evaluation

A more complicated linear regression named net elastic model, was implemented to each target variable after feature selection. Just like OLS model, the net elastic model uses a linear function of predictor variables to predict the target. However, during the calculation of model loss function, it includes the L1 and L2 penalties to reduce the variance of the results which prevents the model from overfitting data. Therefore, two hyperparameters were introduced into the model: regularization parameter and elastic net parameter. The former, ranges from 0 to 1, is the coefficient that controls the overall strength of the L1-L2 penalty. With 0 being minimum and 1 being maximum. The latter is the mixing ratio of L1-L2 penalty with 0 being pure L2 penalty and 1 being pure L1 penalty.

In order to find the best values for the 2 hyperparameters for the prediction of each target, 10 fold cross validation was applied. In each case, the data was first split into training and test sets by a ratio of 7:3. For the first 70%, which is the training data, it was broken down into 10 subsets. Nine of them were used for training the model with specified hyperparameters and the last one, also called the validation set was used for testing and calculating the mean squared error. This process was repeated for 10 times by choosing different validation set from the 10 subsets. The 10 fold cross validation tested each combination of hyperparameters which includes regularization parameter and elastic net parameter by implementing grid search. For each pair of hyperparameters, the average value of the mean squared errors collected from the 10 trials was calculated. The algorithm then picked the combination of hyperparameters which produced the lowest average mean squared error as the optimal solution of hyperparameters for the elastic net model. In the meantime, the model coefficients and intercepts were also returned as part of the results. During the entire process, the rest 30% of the data which was used as test set was completely untouched and unseen by the model. After training, the models were applied on the test data and they were evaluated based on the mean absolute percentage error calculated from the predictions and true values.

Results and discussion

Table 1 shows the model results including all the model parameters, hyperparameters, and model mean absolute percentage error. Lamda refers to regularization parameter and L1 ratio refers to elastic net parameter.

Table1. Final details and results from model

Target Variables	**Final Selected Features**	**Hyperparameters**	**Parameters**	**Mean Absolute Percentage Error**

Availability	%cost from PMs, cost	Lamda = 0.1
		L1 Ratio = 0.05
			Coefficients = [0.0146, -9.08 X10^-7]
			Intercept = 0.8653
				8.34%

Utilization(Util)	%cost from PMs, cost	Lamda = 0.05
		L1 Ratio = 0.05
			Coefficients = [0.0382, -6.6 X 10^-7]
			Intercept = 0.678
				8.88%

Effective Utilization(UofA)	N/A	N/A	N/A	N/A

Hours	% cost from PMs , cost	Lamda = 0.8
		L1 Ratio = 0.05
			Coefficients = [-5.72, 7.83 X 10^-4]
			Intercept = 57.66
				189%

Events (Evts)	% of late PMs, % proxy for planned	Lamda = 0.1
		L1 Ratio = 0.05
			Coefficients = [-2.08, -2.09]
			Intercept = 16.64
				39.05%

Mean Time Between Failure (MTBF)	% of late PMs, % proxy for planned	Lamda = 0.5
		L1 Ratio = 0.05
			Coefficients = [0.85, 1.51]
			Intercept = 43.57
				33.30%

We can see that the errors for hours, events and mean time between failures are extremely high which implies that linear model may not be the best choices for them. One is even greater than 100% which is insanely large. However, for availability and utilization, it is pretty pleasant to see that model errors are relatively lower which are below 10% which indicates that these two linear models are highly reliable. As a result, further discussion will be based on the results from these two models.

So for the cases of availability and utilization, one can see that the coefficients of percentage cost from PMs are much higher than the other feature in the linear models which indicates its high importance. In addition, the coefficients of percentage cost from PMs are positive which suggests that increasing percentage cost from planned maintenance will increase both availability and utilization. Consequently, the hypothesis is that increasing costs from planned maintenance will increase availability and utilization. In order to prove its validity, the author created some scatter plots which are cost vs. availability, cost vs. utilization and cost vs. percentage cost from PM which are displayed in Appendices III and IV.

From the cost vs availability and cost vs utilization diagram in Appendix III, the total cost remains almost constant throughout the entire range of availability and utilization. If the cost from PM is also constant, it will not be in such cases since with increasing availability and utilization, the cost from scheduled and unscheduled maintenance will reduce due to less failure which makes the total cost reduce as well. So one can get the insight that the cost from PM for each month is not constant. Then, moving onto the cost vs. percentage cost from PM diagram in Appendix IV. The total cost remains almost constant (or slightly decreasing) throughout the entire range of percentage cost from PM. Since it is known that the PM cost is not constant for each month, it is highly likely that the increase of PM cost reduces the cost from scheduled and unscheduled maintenance(means less failure) so that the total cost remains unchanged or even decreases. Thus, it justifies the hypothesis that increasing the amount of cost on PM will improve the equipment condition which increases the availability and utilization.

Some other observations were made from the diagrams of availability and utilization vs. percentage cost from PMs in Appendix V. Both diagrams shows that increasing percentage cost from planned maintenance also reduces the variability of availability and utilization which can be recognized as better machine stability.

Conclusion

In conclusion, increasing the cost from planned maintenance can improve truck/shovel condition and functionality including its stability. Moreover, a linear model is very good at predicting availability and utilization, but it is poor at making predictions on the rest of the targets. In the future, more feature engineering will be done, and other machine learning tools will be used for this project.

Reference

[1]“Ordinary Least Squares regression (OLS),” *XLSTAT, Your data analysis solution*. [Online]. Available: https://www.xlstat.com/en/solutions/features/ordinary-least-squares-regression-ols. [Accessed: 22-May-2020].

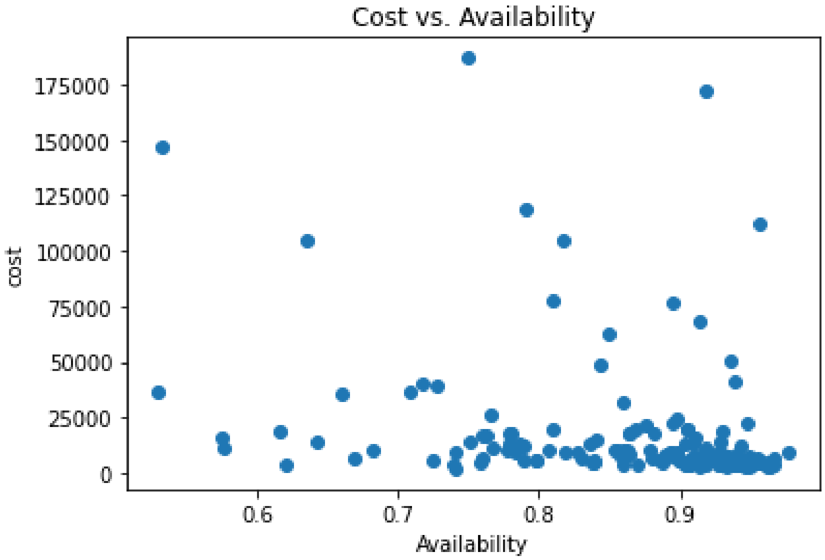
Appendix I: Correlation coefficients between targets and features

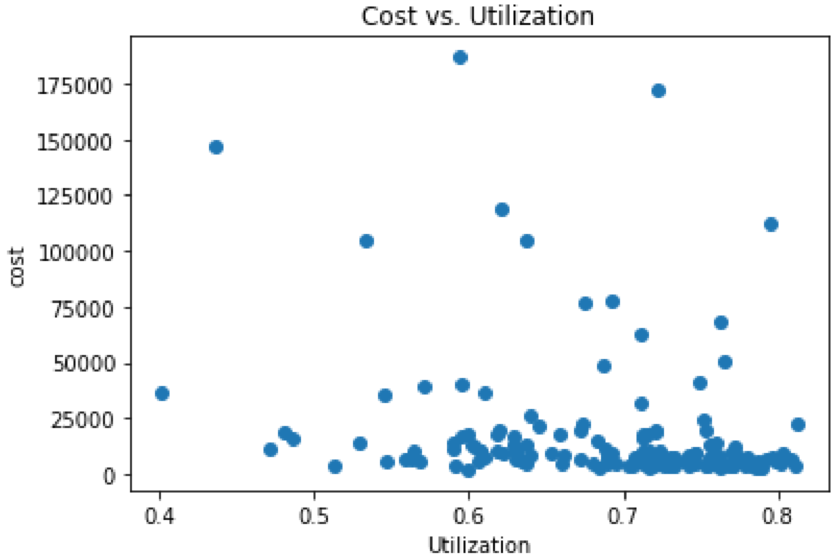
Target / Feature	% of Late PMs	% proxy for planned	% of weekly schedule compliance	% cost from PMs (Y)	Cost
Availability	0.0547	0.1203	0.1096	0.3429	0.2728
Utilization(Util)	0.0184	0.1269	0.1151	0.3740	0.2411
Effective Utilization (UofA)	0.0727	0.0519	0.0300	0.1311	0.0351
Hours	0.1728	0.1219	0.0705	0.3556	0.3486
Events(Evts)	0.3697	0.2726	0.1580	0.2459	0.0681
Mean Time Between Failure (MTBF)	0.2611	0.2297	0.0136	0.2520	0.0947

Appendix II: Final selected features from feature selection

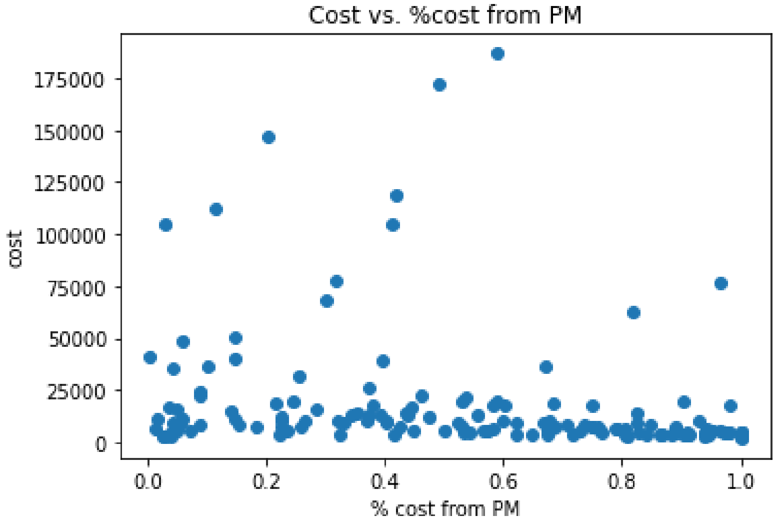
Target	**Selected Feature(s)**
Availability	%cost from PMs, Cost
Utilization(Util)	%cost from PMs, Cost
Effective Utilization (UofA)	Linear model is not appropriate
Hours	% cost from PMs , Cost
Events(Evts)	% of late PMs, % proxy for planned
Mean Time Between Failure(MTBF)	% of late PMs, % of cost from PM

Appendix III. Scatterplots of cost vs. availability and utilization

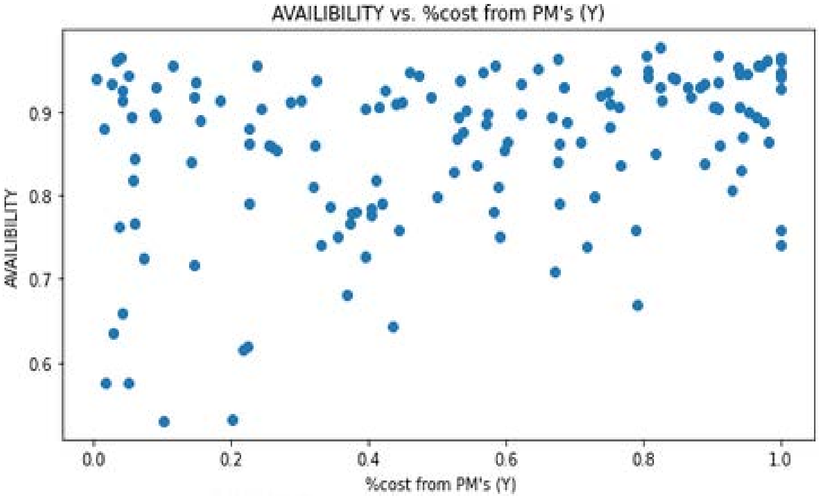


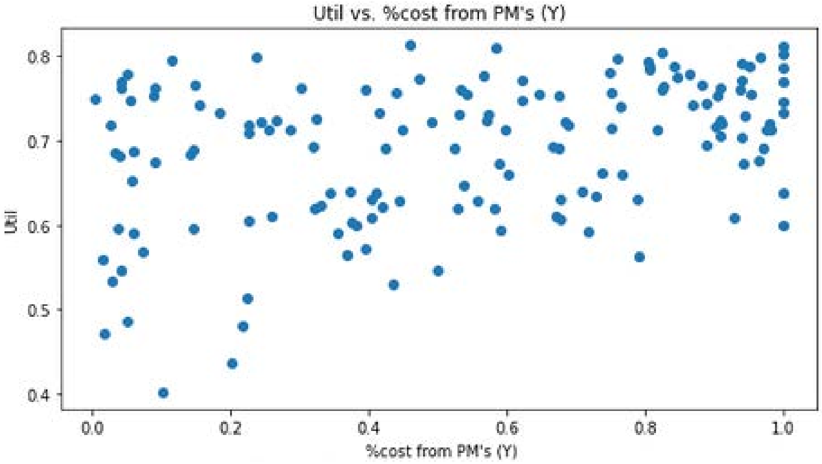


Appendix IV. Scatterplots of cost vs. percentage cost from PMs



Appendix V. Scatterplots of availability and utilization vs. percentage cost from PMs





TTC track re-inspection frequency analysis including failure modes

Janet Lam

Background

One of the responsibilities of the TTC's Non-destructive testing (NDT) team is to revisit known defects in the subway rail system according to a defined timetable. Depending on the severity, or priority of the defect, the schedule may be every 21 days, or annually, or something in between. When the defects are revisited, the NDT team notes the updated status, and this process is repeated until the defects are resolved

As this re-inspection process is known to consume a significant portion of the NDT team's limited time, analysis into the defects was performed. This report is an extension of the project last reported in December 2019.

Summary of previous work

In 2019, we focussed on investigating the MOWIS records to determine the provenance of the defect histories. A few major issues were resolved:

"Not Found" entries: Each defect can take one of six priority levels—red, yellow, purple, blue, brown, and gray, in decreasing order of priority. Though the records are otherwise indistinguishable within MOWIS, when records were upgraded from gray to a higher priority, they could not simply be updated, they needed to be re-entered as a new defect in the system.

For our analysis, it was necessary to identify these defects and connect them into one known defect history, as they are practically speaking the same defect. The process of connecting these histories was non-trivial. Matching locations, timing and failure modes was required.

Standards change: At some time in the mid 2010s, a standards change occurred wherein some defects previously identified as purple defects were reclassified as blue defects. The defects that were affected by the standards change needed to be identified and reverted, so that the change in priority on paper did not get analysed as a true change in defect severity.

In order to identify these defects, all defects that had a priority downgrade from purple to blue were manually inspected.

Analysis

For the analysis of defects, we defined a transition from one state to another as a failure fit the transition times to an exponential distribution per priority level.

Recommended re-inspection periods, non-modal

Based on data availability, it was possible to provide results for purple, blue and gray defects. To start, the following table summarizes the number of defects under each defect priority, along with their transition status. The definitions of the columns are as follows:

Upgrades – defects that transitioned to a higher priority
Stable – defects that did not transition to a higher priority
Other – defects that were not counted in the analysis because they are new (only have one record), or downgraded.

Table 1. Counts of defect histories by transition status

Priority	Total	Upgrades	Stable	Other
Purple	336	13	313	10
Blue	64	56	4	4
Gray	824	36	533	255

In order to analyze the defect transitions, we fit an exponential distribution and determined the failure rate, lambda for each priority level. In the following table, the mean times to failure (1/λ) are reported.

This was computed using a maximum likelihood estimator, computed as follows:

$$\hat{\lambda} = \frac{x}{\sum_{i=1}^{n} t_i},$$

where x is the number of upgrades, and t_i is the length of the defect history, regardless of whether it was upgraded or not.

Table 2 Transition rates for defect priorities

Priority	Mean time to transition (days)	Transition rate
Purple	3656.69	2.7345E-4
Blue	4195.75	2.3834E-4
Gray	9468.69	1.0561E-4

While these mean times to transition seem really high (10 or more years!), the reliability requirement is quite high, so very low probabilities of transition are experienced. The following table provides a comprehensive overview of reliabilities (probability of no transition) during a fixed re-inspection interval. The "Current" interval is the average interval performed by the NDT team. The "Standard" interval is the specified inspection interval according to the internal TTC requirements.

Note that for purple priority defects, the re-inspection interval can be increased to 60 days with only a one percent decrease in reliability, or a 0.984 probability that no transition will occur from the first identification. Similarly, for blue priority defects, the re-inspection interval can be increased to 80 days for a one percent decrease in reliability and a 0.981 probability that there will be no transition.

Table 3. Comprehensive reliability results for different re-inspection intervals

Priority Level	Time between Re-Insp.,	Interval Description	Reliability		Un-Reliability	
			R(t)	% decrease from current level	F(t)	% increase from current level
Purple	17.38	Current	0.995	0.00	0.00474	0.00
	21	Standard	0.994	0.10	0.00573	20.77
	40	1 / 40 days	0.989	0.62	0.01088	129.44
	60	1 / 60 days	0.984	1.16	0.01627	243.22
	80	1 / 80 days	0.978	1.70	0.02164	346.38
	100	1 / 100 days	0.973	2.23	0.02697	468.93
	120	1 / 120 days	0.968	2.77	0.03228	580.86
Blue	36.42	Current	0.991	0.00	0.00864	0.00
	45	Standard	0.989	0.20	0.01067	23.43
	60	1 / 60 days	0.985	0.56	0.01420	64.28
	80	1 / 80 days	0.981	1.03	0.01889	118.52
	100	1 / 100 days	0.976	1.50	0.02355	172.51
	120	1 / 120 days	0.971	1.97	0.02820	226.23
	140	1 / 140 days	0.967	2.44	0.03282	279.70

Breakdown by failure modes and priority

TTC has demonstrated interest in addressing different failure modes with different re-inspection intervals. For example, should a purple defect of a bolt-hole crack be treated differently from a purple defect of corrosion?

In order to perform this analysis, further breakdown of the number of defects that transitioned by mode and priority are required. For purple and blue defects, the number of defects per failure mode is summarized in the table below.

It is evident that by partitioning by failure mode, the number of transitions for each category is deeply reduced. Consequently, it’s not advisable to follow the same method of parameter estimation for many of these categories. We will perform the same analysis for bond web crack, corrosion, and weld failure mode for the purple priority defects only.

Table 4. Summary of transitions by failure mode

	Purple	Blue
Bolt hole crack	1	
Bond web crack	4	
Corrosion	3	1
Weld	3	1
Misc	2	2

By performing the same analysis on the three key failure modes, the following results were found.

Failure mode	Mean time to transition (days)	Transition rate
Bond web crack	1778.25	5.6235E-4
Corrosion	5553	1.8008E-04
Weld	3940	2.5381e-04

Using these results, similar reliability change tables were generated

Priority Level	Time between Re-Insp.,	Reliability		Un-Reliability	
		R(t)	% decrease from current level	F(t)	% increase from current level
Bond web crack	17.38	0.990	0.00	0.00973	0.00
	21	0.988	0.20	0.01174	20.71
	40	0.978	1.26	0.02224	128.69
	60	0.967	2.37	0.03318	241.13
	80	0.956	3.46	0.04399	352.30
	100	0.945	4.54	0.05468	462.23
	120	0.935	5.61	0.06526	570.94
Corrosion	17.38	0.997	0.00	0.00313	0.00
	21	0.996	0.07	0.00378	20.79
	40	0.993	0.41	0.00718	129.68
	60	0.989	0.76	0.01075	243.90
	80	0.986	1.12	0.01430	357.72
	100	0.982	1.48	0.01785	471.12
	120	0.979	1.83	0.02138	584.11
Weld	17.38	0.996	0.00	0.00440	0.00
	21	0.995	0.09	0.00532	20.77
	40	0.990	0.57	0.01010	129.49
	60	0.985	1.08	0.01511	243.37
	80	0.980	1.58	0.02010	356.66
	100	0.975	2.08	0.02506	469.39
	120	0.970	2.57	0.02300	581.54

Based on these results, it is evident that bond web crack defects are more likely to transition to a higher priority in a shorter period of time. Thus, increasing the re-inspection interval to 40 days will reduce the reliability by 1.26%, whereas corrosion and weld defects can be extended to 80 and 60 days respectively.

Failure modes with insufficient data

For the remainder of the failure modes listed in Table 4, it is perhaps more meaningful to consider a range of possible failure rates, given the number of defects that are on record. Then, we can consider the probability of having experienced this outcome if a failure rate were assumed.

For example, there are no defects that transitioned for blue bolt-hole cracks. In fact, there are four blue bolt-hole crack defects that did not transition. Using the information that each defect survived an average of 397 days without transitioning, we can reflect on the possible transition rates and their respective likelihoods. That is, in 1588 days, if the probability of no failures was 50%, what would the failure rate be? How about 70%, or 90%?

Table 5. Blue bolt-hole crack defects

Defect number	Days open	Status
166420	608	Blue – updated
168630	486	Blue – updated
171051	369	Blue – updated
175305	125	Blue – updated

As shown in the following table, given that there were no transitions over the observed lifetimes, we can be fairly confident that the mean time to transition for blue bolt-hole crack defects is above 2300 days, there's a probability of 85% or more that there will be no transition in one year.

Table 6. Potential mean times to transition and their resulting probabilities. Blue bolt-hole crack defects

	Chance of no transition over 1588 days?				
	10%	**30%**	**50%**	**70%**	**90%**
Implied mean time to transition	689.66	1318.97	2291	4452.233	15072.06
Implied transition rate	1.45E-03	7.58E-04	4.36E-04	2.25E-04	6.63E-05
Probability of no transition in one year	0.5890	0.7583	0.8527	0.9213	0.9761
Probability of no transition in two years	0.3470	0.5750	0.7271	0.8488	0.9527

This method was also applied for blue bond web crack defects, and the results are summarized in the following tables. For the cases with one or two transitions, we ran into some numerical difficulties and will be reserved for some future work.

Table 7. Potential mean times to transition and their resulting probabilities. Blue bond web crack defects

	Chance of no transition over 1135 days?				
	10%	**30%**	**50%**	**70%**	**90%**
Implied mean time to transition	492.92	942.71	1637.46	3182.17	10772.54
Implied transition rate	2.03E-03	1.06E-03	6.11E-04	3.14E-04	9.28E-05
Probability of no transition in one year	0.4769	0.6790	0.8002	0.8916	0.9667
Probability of no transition in two years	0.2274	0.4610	0.6403	0.7950	0.9345

Conclusions

This report broadly summarizes the probability of a defect transitioning from its current priority to a higher priority. This was done for purple, blue and grey priorities as a whole, as well as some of the specific failure modes.

Generally speaking, an extension of the re-inspection intervals can safely be made without a large risk in missing defects before they become high priority. This may result in saved resources of the NDT team.

Kinross KPI regression analysis report

Dragan Banjevic, Janet Lam

Introduction

Kinross Gold is interested in analysis of importance and predictability of key performance indices (KPIs) used to report monthly activity of their trucks. The main goal of the study was to show whether increase in planned maintenance time increases reliability/availability using provided KPI measures.

Kinross originally provided a data set that includes records on monthly maintenance activity for all units in the fleet for the period of Jan 2012 to April 2017 (2924 records). There are two unit types (classes), trucks, and shovels. There are 6 truck CAT models, and 8 shovel models (4 CAT and 4 Hitachi). Records are not given for all units for all months. Some records are incomplete as some indices are missing. For initial analysis 158 complete record of monthly activity of CAT 793D model, units 462-467, are selected. In later stage (April 2020), the monthly records from 2019 were included for 4 out of 6 trucks. Some KPIs are used to measure monthly maintenance activity, and some are used to report work performance. The question is how effective maintenance activity is (planned and unplanned) for work performance (availability, MTBF, etc.). Another question is whether predicting future work performance (e.g., in the next month) from current maintenance activity is possible.

The results of this initial analysis seems promising for establishing relationship of maintenance activity with work performance.

Data

The data include five KPIs used to measure maintenance activity and 20 variables that measure work performance. We selected all five maintenance KPIs and seven work performance KPIs for analysis. For simplicity, we slightly edited maintenance KPIs’ names and used short notation in analysis with software R.

Maintenance KPIs (input for regression analysis)

	KPI	New name	Description	Short name in R
1	Cost	Cost	Actual cost from all WO	in0
2	% OF Late PM'S	PerLate	# cancelled and 10 days late PMs/Total number of PMs	in1
3	% proxy for planned	PerProxy	# WO with 7 days or more lead time/Total number of WO	in2
4	% of weekly schedule compliance	PerComp	# WO scheduled and completed in that week/# WO scheduled for that week	in3
5	%cost from PM's (Y)	PerCost	Actual cost from PM WO/Actual cost from all WO	in4
6	proactive hours	PrHrs	Actual hours from PM WO	in5

Work performance KPIs (output for regression analysis)

	KPI	Description	Short name in R
1	Availability	Overall availability	out1
2	Utilization	Overall utilization	out2
3	UofA	Effective utilization	out3
4	Hours	Unscheduled Down Hours	out4
5	Events	Unscheduled Down Events	out5
6	MTBF	= (Oper. Time+Oper. Delay)/12	out6

Some output variables, such as availability, are presented as percentages. They are used in the analysis as their decimal values. Some work performance KPIs that are not included in the analysis have missing values, but this did not affect selected variables, so no record was deleted for that reason. One value for cost was reported as negative number. Four records were removed as outliers (with some extreme values). No other problems with the data were found.

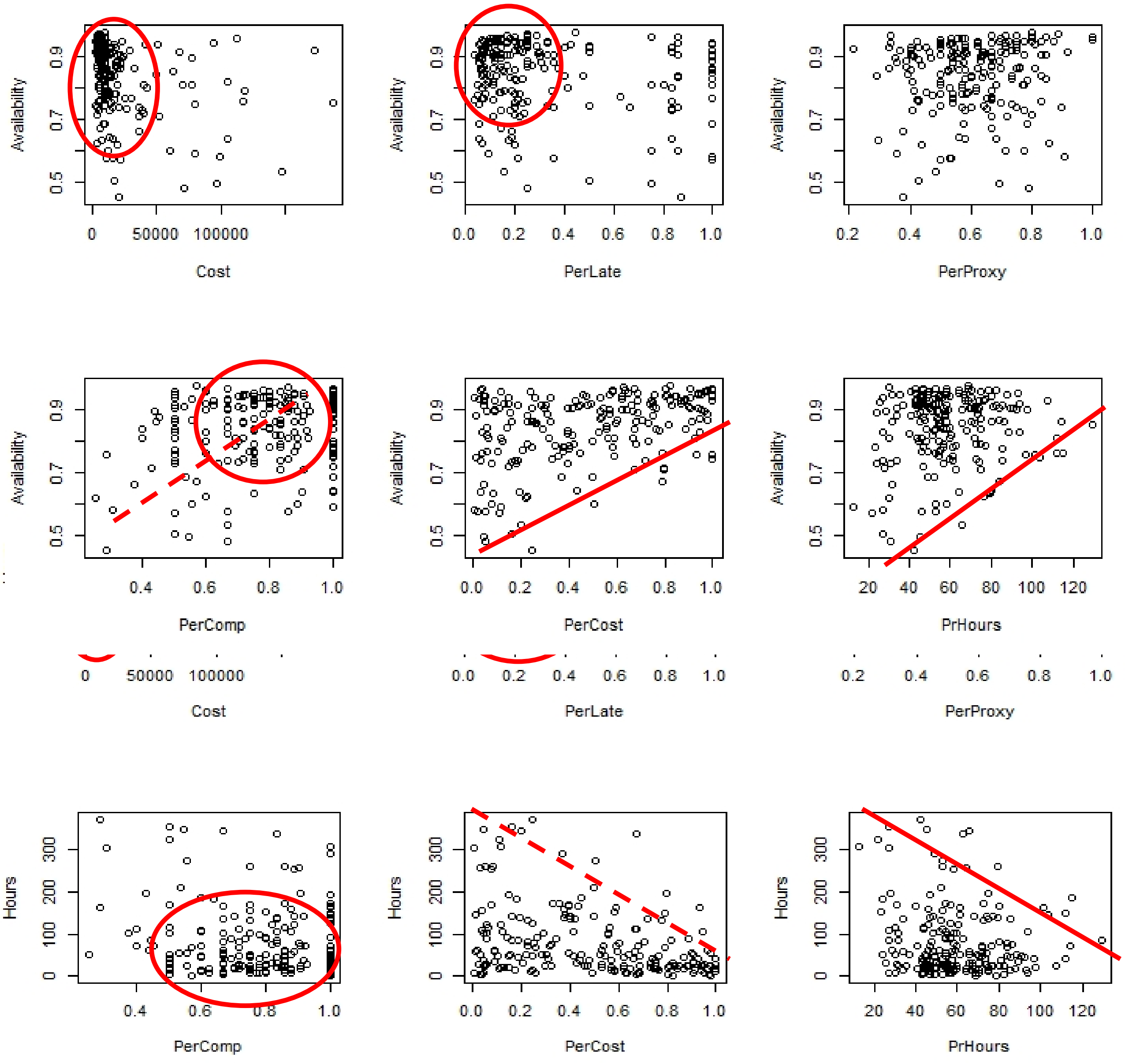
Analysis of correlations

We used multiple regression analysis as the main approach to analyse the data. As an initial step in the analysis we use scatter plots one output KPI against an input KPI for the same month, to see whether there is any visible relationship between them, such as linear. For example, we plot availability against % OF Late PMs. Here we present joint scatter plots of output KPIs vs all input KPIs (see next two pages). Scatter plots for Utilization and U of A are not presented, as they are similar to Availability scatter plots.

Strong relationships between input and output KPIs cannot be seen, but there are clear trends in several cases when an input KPI increases, then an output KPA improves (either increases, or decreases as it would be expected). For example, when PerComp, PerCost, or PrHours increase, then Availability tend to increase (Figure 1), or when PerCost, or PrHours increase, then Hours and Events tend to decrease (Figures 2 and 3). Notable clusters and trends are highlighted with red circles and straight lines.

The scatter-plots can be accompanied with paired correlation coefficients that quantify relationships of input and output variables. It should be noted that correlation coefficients measure strength of *linear relationship*, not any relationship. As we are interested in prediction of outputs from past inputs, we also look at relationships between current output KPIs, and input KPIs from the past. The correlation is a value between -1 and +1, where a value close to 0 indicates a week linear correlation between variables, and a correlation closer to +1 or -1 indicates stronger linear correlation between variables. Here we present correlations between output KPIs for a current month and input KPIs from one month before, and from two months before.

Figure 4. Joint scatter plots of MTBFs vs all input KPIs



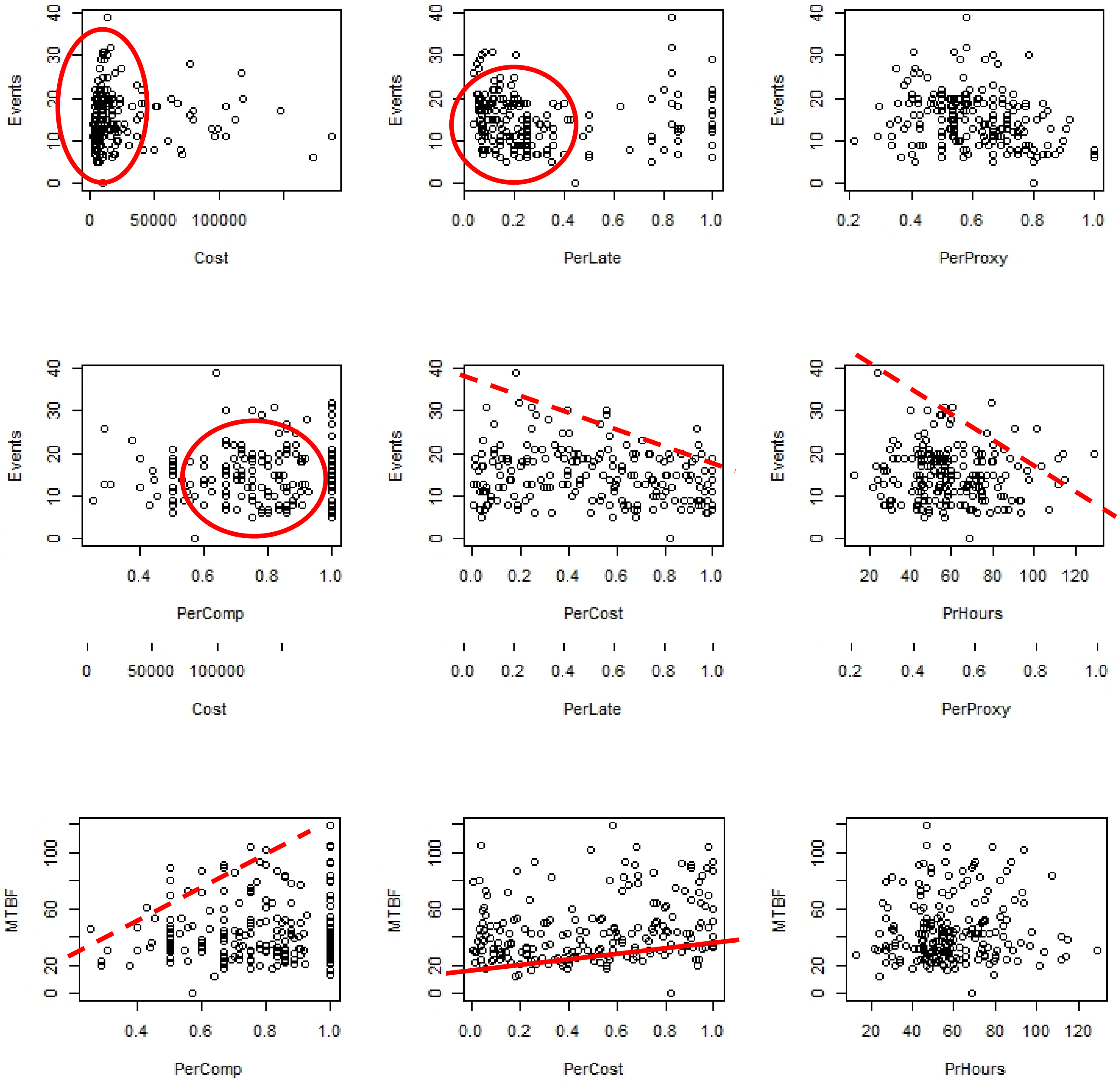

Events
40
30
20
10
0
0
50000
100000
Cost
PerLate
0.0
0.2
0.4
0.6
0.8
1.0
PerProxy
PerComp
PerCost
PrHours
20
40
60
80
100
120
MTBF
0
20
60
100

Table 1. Correlations between input and output KPIs for the same month

	Cost	PerLate	PerProxy	PerComp	PerCost	PrHrs
Avail	-0.298	-0.209	0.095	0.286	0.409	0.094
Util	-0.267	-0.162	0.128	0.283	0.411	0.093
UofA	0.083	0.136	0.124	0.011	0.035	0.014
Hours	0.348	0.158	-0.107	-0.193	-0.375	-0.125
Evts	0.036	-0.027	-0.261	0.109	-0.204	-0.060
MTBF	-0.115	-0.043	0.265	0.045	0.292	0.089

Table 2. Correlations between input and output KPIs for one-month back

	Cost	PerLate	PerProxy	PerComp	PerCost	PrHrs
Avail	0.001	-0.134	0.146	0.074	0.199	0.050
Util	0.019	-0.070	0.214	0.094	0.209	0.033
UofA	0.058	0.189	0.216	0.057	0.046	-0.048
Hours	-0.004	0.104	-0.149	-0.095	-0.212	-0.054
Evts	0.038	0.037	-0.211	0.118	-0.263	-0.006
MTBF	-0.030	-0.074	0.212	-0.074	0.322	0.042

Table 3. Correlations between input and output KPIs for two-months back

	Cost	PerLate	PerProxy	PerComp	PerCost	PrHrs
Avail	-0.108	-0.168	0.194	-0.004	0.204	0.063
Util	-0.073	-0.093	0.244	-0.030	0.204	0.068
UofA	0.112	0.225	0.164	-0.090	0.021	0.023
Hours	0.095	0.158	-0.166	-0.011	-0.204	-0.080
Evts	-0.058	0.147	-0.276	0.022	-0.226	-0.139
MTBF	0.019	-0.150	0.277	-0.009	0.282	0.170

No correlation is greater than 0.411, KPIs is stronger for the current month, but much weaker for one month back, or two months back. Correlation between output KPIs and PerProxy, or PerCost is consistent over current month, one-month and two-month periods. In addition to it, the same positive sign of the correlation means that, when PerProxy or PerCost increase, Availability, Utilization, U of A, and MTBF increase, but Hours and Events decrease, as it would be expected when proactive maintenance improves reliability. Another example: when Perlate increases, Availability, Utilization, and MTBF decrease, but U of A, Hours, and Events increase. How could we explain seemingly unexpected increase of U of A? even if availability decreases, the demand likely does not go down, so more of the available time will be used.

As a preliminary conclusion, we may say that PerProxy, and PerCost are the two input KPIs most indicative of improvements in maintenance and reliability in most of the cases. On the other hand, PrHours KPI is a week indicators of output KPIs, most of the time. How exactly those and other input KPIs affect the output KPIs we will see from the regression analysis results.

Regression analysis

We have started building regression analysis models for functional relationship of output KPIs with input KPIs, and, as expected from the correlation analysis, in all models **%cost from PM's** appears significant factor, and, to less extend, **% proxy for planned**. It does not mean that other input variables don't have contributions, but they are overshadowed with the first two. Due to not very strong correlations between input and output variables, the predictive power of the models is not strong, but is still useful.

We are interested in building extended models with including lags (not only the current month's input results, but also the past months' input results), and interactions. Other options are possible, and they may appear over the course of the analysis. We have also tried transforming some variables using log function, such as cost or MTBF, which somewhat improve regression models, but not significantly. Log transformation is used for the variables that can take large values in comparison with other input or output variables. Here we will report only models that include untransformed variables. All regression models are of the form

$$\hat{y} = b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_k x_k,$$

where $\hat{y}$ is predicted output KPI, $x_1, x_2, \ldots, x_k$ are input KPIs, and $b_0, b_1, \ldots, b_k$ are regression parameters. b_0 is called *intercept*. $b_i = 0$ means that input variable x_i is not statistically significant for prediction of output y, not necessarily that it is not somehow related to y, as indicated above.

The selection of significant predictor input variables is done by a statistical procedure, combined with some expert judgement, and is not straightforward. In the following we will report models for all output KPIs for three cases, when the current month input is used, when one-month back input is used, and when the both one month and two months' input is used. In the tables below we will report non-zero coefficients, and exclude intercept. Actual values of different coefficients are not comparable, in principle, as they are relative to the variable magnitude, not the variable importance. What is important for our analysis is the sign ("+" or "-") of a coefficient indicating direction of the change.

Table 4. Regressions for the same month: Output on the left side, input on the top

	Cost	PerLate	PerProxy	PerComp	PerCost	PrHrs
Avail	-8.440e-07			0.149	0.116	
Util	-5.796e-07			0.115	0.103	
UofA		No significant model				
Hours	7.172e-04			-61.63	-67.10	
Evts			-10.169		-2.442	
MTBF			41.356	15.631	14.810	

For example, $\widehat{MTBF} = Const + 41.356 \times PerProxy + 15.631 \times PerComp + 14.810 \times PerCost$ is a model for predicting MTBF. Positive coefficients are highlighted in green, and negative in light yellow. You can consider change in direction either for an output variable, or for an input variable, similarly as with correlations. For example, Availability will tend to go up if Cost go down, or PerComp, or PerCost go up. Availability Utilization and MTBF will tend to go up when PerCost go up, but Hours and Events will tend to go down when PerCost go up. It can be noted that PerLate does not contribute to prediction of any output KPI when accompanied with other input KPIs for the current month. The other two cases for prediction using past information are in a somewhat different situation.

Table 5. Regressions of the present using one-month back inputs: Output on the left side, input on the top

	Cost	PerLate	PerProxy	PerComp	PerCost	PrHrs
Avail		-0.050	0.094		0.057	
Util			0.110		0.051	
UofA	1.330e-07	0.021	0.055	0.029		-2.219e-04
Hours					-54.83	
Evts			-6.790		-4.594	
MTBF			21.973		20.302	

The regressions are somewhat different. PerProxy now appears to be significant in prediction of all outputs, except for Hours. There is a model for predicting U of A. The last month Cost is not informative, except for U of A.

Table 6. Regressions of the present using one and two months back inputs: Output on the left side, input on the top; Lag 1: one-month inputs, Lag 2: two-months inputs.

	Lag	Cost	PerLate	PerProxy	PerComp	PerCost	PrHrs
Avail	1						
	2	-4.41e-07	-0.059	0.137			6.57e-04
Hours	1			-69.30		-37.55	
	2	2.926e-04	35.89				-0.546
Evts	1				4.720	-3.577	
	2	-1.346e-05	3.712	-8.077		-2.757	
MTBF	1					18.802	
	2		-10.180	29.906			0.102

For simplicity, we did not include regression models for prediction of Utilization and U of A. In the case of inputs from the past two months, it seems the two-months back input appears more often than the one-month back input, but they are also combined, except in the case of Availability.

Effectiveness of prediction

Accuracy of prediction cannot be seen directly from regression models, but from comparison of actual outputs with predicted (fitted) outputs. It can be measured by the average prediction error, or a relative error, a percentage of error in comparison with actual value. In our cases, the accuracy, on average is not high, but has an interesting feature: it often depends on the magnitude of the prediction. We will explain it in some details by using scatter plots of actual values vs predicted/fitted values from regression models. If a model were accurate, the points on the graph would closely follow line $y = x$, otherwise they would be scattered around. We will show a couple of examples.

Figure 5. Scatter plot of availability vs fitted value for the same month prediction

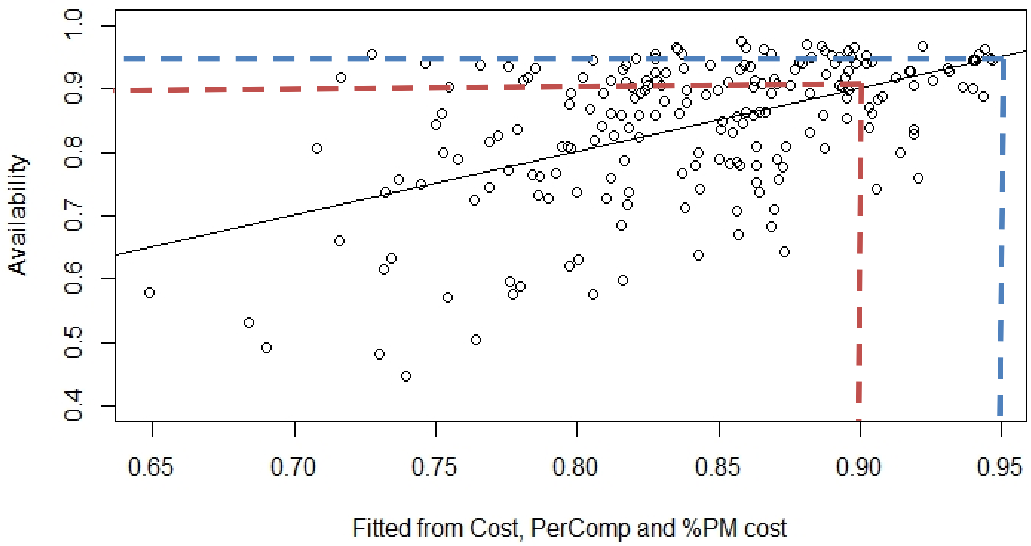
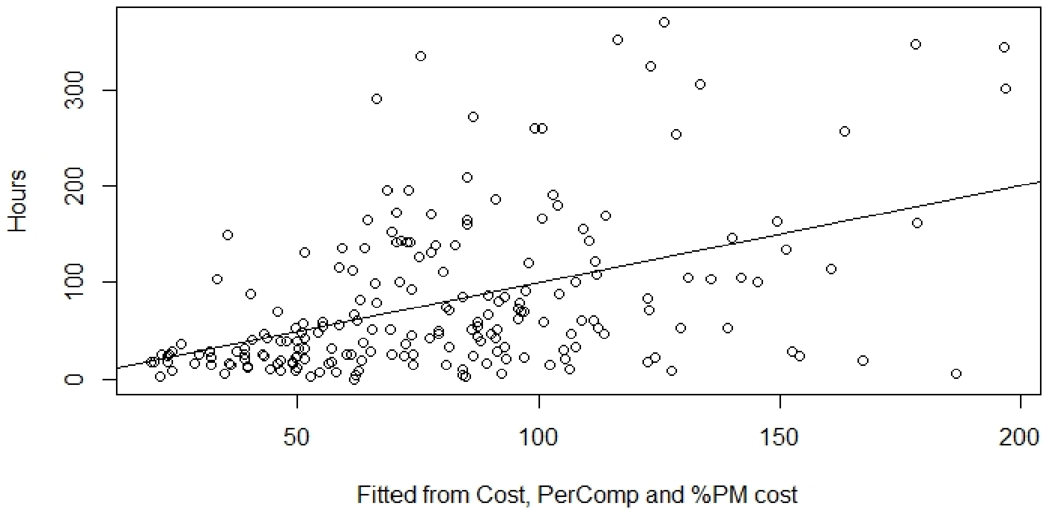


Figure 6. Scatter plot of hours vs fitted value for the same month prediction



Prediction of Availability from the same month input KPIs (Figure 5) is fairly accurate for high level Availability such as 95%, and less accurate for, e.g., 90%. Variation of *y* values for fixed *x* value increases when x decreases. The pattern is similar for cases when past information is used. The prediction of Hours from the same month input KPIs (Figure 6) is fairly accurate for low Hours such as below 50, and less accurate for values above 50. Variation of *y* values for fixed *x* value increases when x increases. Pattern is similar for cases when past information is used. Prediction for Events behaves similar to one for Hours. For MTBF similar behaviour as for Hours and Events is less visible. Accuracy of all models' predictions slightly increases with more information used from past.

Conclusions

- There is no strong correlation between input and output KPI indicators.
- %Proxy and %PM Cost are more indicative of outputs KIs than the other variables.
- Regression models don't give precise estimates of the outputs but show tendency of increasing availability and MTBF, and decreasing hours and events when amount of proactive maintenance increases, for example, when %PM Cost increases.
- Models that include pasts two months' information tend to give somewhat better predictions.

Next steps

We want to include more data in the regression models, such as 2018 records, for CAT793D. We want to consider data for another model, CAT793F, either for a separate analysis, or in combination with CAT793D model. We want to do more detailed analysis of outliers that may appear from situations unrelated to regular proactive maintenance. A possibility that should be explored in modelling is to include all past information, *input and output KPI*s, not only input KPIs, to predict output KPIs. In addition, we can create models to predict cost of reactive maintenance from past information.

DND: Propulsion diesel engine reliability modelling

Arun Shanmugam

Background

The Canadian Department of National Defense has twelve Halifax-class (aka City-class) frigates in operation that have served the Navy since 1992. The propulsion system on these vessels consists of a Propulsion Diesel Engine (PDE) and two Gas Turbines that work in a Combined Diesel Or Gas (CODOG) arrangement, with the PDE functioning during cruising speeds and the turbines serving high speed dashes. The Propulsion Diesel Engine (PDE) is the object of this reliability modelling study.

Current procedures in maintenance activity & information management

The PDEs are subjected to an Oil & Coolant Conditioning Analysis Program (OCCAP) once every 30 days which involves offsite analysis of oil & coolant samples by a third party while maintenance decisions are made by the Fleet Maintenance Facility (FMF). Disparities have been observed between results of the OCCAP analysis and recommendations of the maintenance technicians and these disparities have proven challenging while making decisions regarding scheduling of maintenance activities.

The maintenance activities of the PDE are carried out in conjunction with the Defence Resource Management Information System (DRMIS) – a SAP-based integrated information system that supports maintenance activities and replacement part procurement among other business processes. The DRMIS generates work orders for scheduled preventive maintenance actions and keeps track of corrective maintenance orders. The integrated system generates purchase orders for replacement parts that need to be procured towards completing corrective maintenance. This introduces a significant challenge in analysis since orders are not indicative of failure or suspension events but associated part replacements that could be one or many depending on the maintenance activity.

Problem Statement

The objective of the project is to develop a model to predict engine failure with the use of a Weibull Proportional Hazards model. This will involve constructing an event history of failures and suspensions of the PDEs that will be used with OCCAP data to construct a Transitional Probability

Model on EXAKT Condition-Based Maintenance software. The analysis also has the scope to assess & refine current maintenance policy with the inclusion of costs involved during corrective and preventive maintenance activity.

Data cleaning

Mr. Jamie Dreyer in collaboration with Ms. Nicolle Kilfoyle prepared three files towards the analysis: the DRMIS work order dataset with 4678 orders from November 2012 to March 2019; OCCAP oil and coolant conditioning data from December 2012 to January 2019; and monthly PDE odometer reading data.

Since the DRMIS generates work orders that double as procurement orders, this introduces great ambiguity in determining the number of events. Further, the dataset of work orders also contains quite a few orders for replacement that are not critical to the functioning of the engine such as the replacement of bulbs. This necessitated a case-by-case analysis of work orders in close collaboration with DND, an ongoing process in the project.

The dataset contained orders for multiple parts associated with the same failure or suspension event which would have potentially resulted in an over-estimation of the number of failures/suspensions. This necessitated grouping of orders based on how close they are with each other chronologically, odometer running hour and most importantly, DNDs input so that the Weibull model would closely mirror reality.

The DRMIS generates orders codes for different classes of orders that link to preventive/corrective maintenance. However, this distinction is not always consistent since preventive maintenance orders generate “dummy” corrective maintenance orders to pull spare parts associated with the PM action. The characterization of a failure or suspension event is therefore, not a straightforward task due to the nature of the data and the complexity of the PDE as an asset.

Characterization of failures and suspensions

The costs associated with the procurement of replacement parts is a metric that we relied on to distinguish failure/suspension events critical to functioning. The hypothesis was that critical failures and/or suspensions would be characterized by a relatively higher total cost of replacement and hence, can be used to focus the analysis on the orders that link to engine functioning.

During the first phase of analysis, the cost threshold for a significant failure was determined to be 35000 CAD for corrective maintenance actions. It was also imperative to the analysis that all maintenance actions under consideration would bring the engine to a “good-as-new” working condition. Based on DND input, it was understood that all corrective actions resulted in restoration of PDE to good as new while not all preventive maintenance actions did not. 12K and 15K preventive maintenance actions and 24K overhauls brought the PDE to “as-new” condition and therefore, it was decided that orders linked to these PM actions would be considered significant and relevant for analysis. This resulted in the reduction of orders relevant to Weibull analysis from the initial number of 4678 to 80. This was followed by preparation of the data for EXAKT Weibull analysis.

Weibull analysis and further investigation

The Weibull analysis yielded a shape parameter that was not commensurate with assets such as the PDE. This was largely owing to the practical complications surrounding the data – the orders do not have a one-one correspondence with events as described earlier and multiple events with the same running hours in the reduced order set were considered to be one in the EXAKT analysis since the working age readings correspond to month-wise readings and not hour reading at the time of failure.

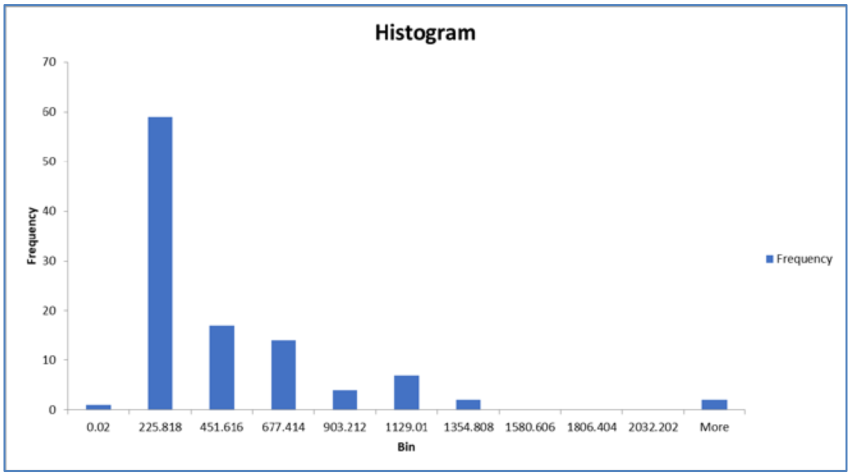
Consequently, we consulted with DND to revisit the list of events under consideration and this resulted in the second phase of analysis which involved revisiting the cost threshold and consolidation of duplicate/cancelled orders, in an attempt to refine the model.

It was determined that there did exist failure events below the 35000 CAD threshold and so, the cost threshold was moved to 10000CAD, and removal of duplicate/cancelled orders. This resulted in a new list of 209 orders which was subjected to the same EXAKT Weibull analysis as was done before. This resulted in only a marginal increase in shape factor but was still not reflective of the actual reliability behaviour of PDEs.

Inter-failure time analysis

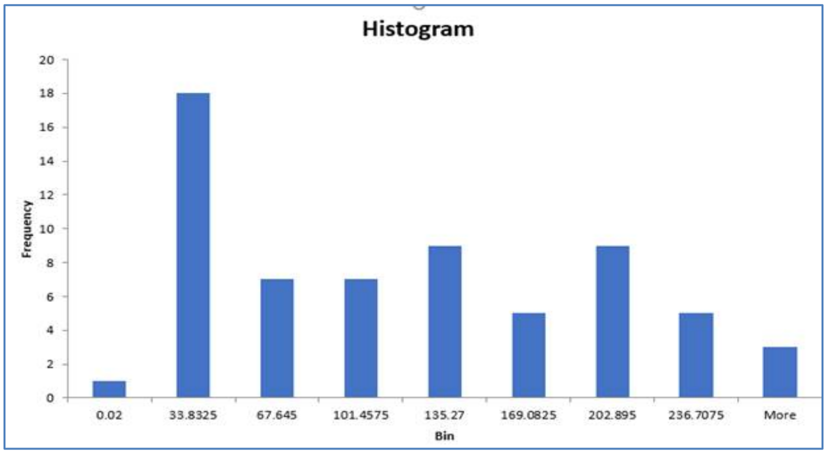
The continued repeat of unexpected outcomes with the Weibull analysis necessitated an Inter-failure time analysis. Time between successive events was analyzed using a histogram (see Figure 1) and it was observed that most IFTs could be found in the less than 250 running hours mark. This struck us as abnormal since typically failure events are farther apart, so we decided to take a closer look at these orders.

Figure 1. Inter-failure time analysis histogram



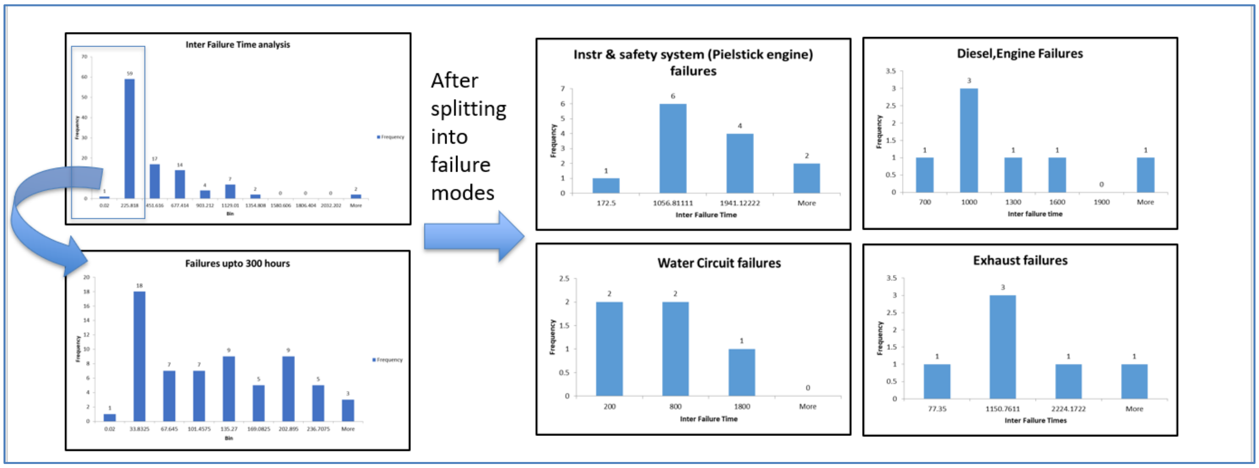
This ski-slope shape indicates a shape factor of less than one, which is not consistent with the behaviour of engines. When we increased the resolution of analysis of events less than 250 hours, we obtained the following histogram.

Figure 2. Analysis of orders less than 250 hours



Upon futher invetigation, we decided to conduct a failure mode wise analysis. This involved classification of each failure based on the sub-system the failure affected and this was followed by grouping failures of each sub-system together. As a consequence, the number of failure events were correspondingly reduced and since inter-failure time analysis requires at least two failure events, this resulted in only four failure mode being amenable to further analysi, namely: Instrumentation & Safety System failures, Diesel, Engine failures, Water Circuit failures and Exhaust failures. Figure 3 shows the split-up into IFT analysis for component failure modes.

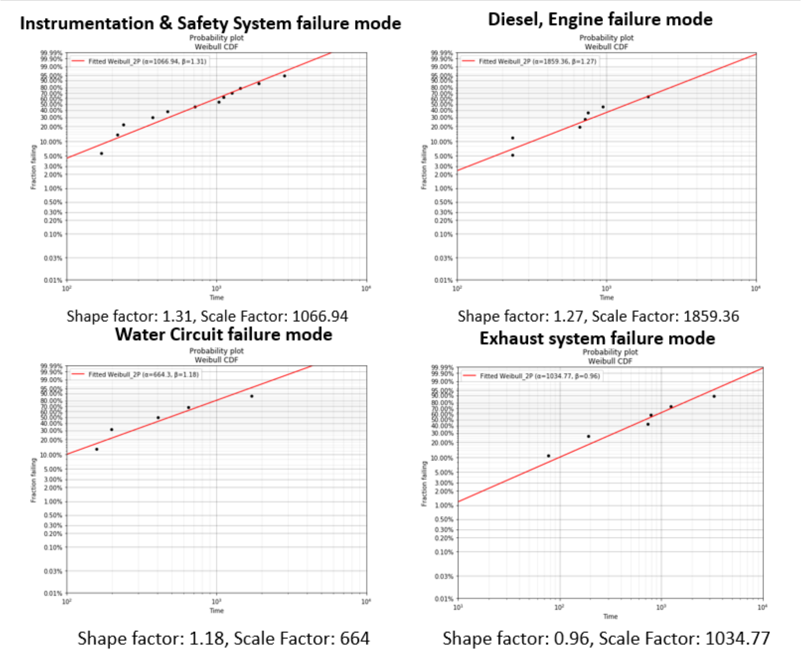
Figure 3. Split-up into failure modes



Weibull analysis

The Inter-failure times were plotted on the Weibull chart with a log-log scale as shown in Figure 4. From the figure, we observe that Weibull fit achieved and this proves the hypothesis that failure modes need to be investigated separately in a complex asset such as the PDE.

Figure 4. Plotting on Weibull paper



Instrumentation & Safety System failure mode

Shape factor: 1.31, Scale Factor: 1066.94

Diesel, Engine failure mode

Shape factor: 1.27, Scale Factor: 1859.36

Water Circuit failure mode

Shape factor: 1.18, Scale Factor: 664

Exhaust system failure mode

Shape factor: 0.96, Scale Factor: 1034.77

Shape factors for different failure modes are in the neighbourhood of 1, indicating the shape of the failure probability distribution is exponential. The scale factors provide an idea of the characteristic life of the PDE for each failure mode and what to expect from the behaviour of the PDE.

Preventive maintenance policy re-design

One of the key motivations of the project was to re-examine the current preventive maintenance policy through the lens of total cost of maintenance. Current maintenance policy involves preventive maintenance actions at 1500 hrs, 3 months, 6 months, 12K, 15K, 24K, etc. with maintenance actions at 12K/15K and 24K running hours bringing the engine to good-as-new working condition.

Preventive maintenance policies are broadly categorized as either interval (block) replacement or age-based replacement strategies. While block replacement strategies are easier to implement, age-based replacement strategies provide a lower total expected cost of replacement overall.

Three factors are critical when it comes to PM policy re-design: the underlying distribution must be Weibull, cost ratio of average failure replacement to average preventive replacement, and design must be optimized for minimization of total cost. Since labour or man hour costs & costs associated with downtime are not available, the cost associated with preventive replacement cannot be computed & so, cost ratio, k cannot be calculated directly. Typical values of k for assets such as PDE is in the neighbourhood of 6:1 and hence, decision models for each failure mode were generated in the neighbourhood of 6:1. Depending on failure criticality & expert knowledge, DND would proceed to choose from the given options to best fit engine operation.

Decision models

Decision models were generated by inputting inter-failure times on one hypothetical PDE for each failure mode on EXAKT software to come up with optimal preventive maintenance periods. Figure 5 shows the optimal preventive replacement age for the diesel engine failure mode as an illustration. It can be noted that 733.364 is the optimal replacement interval for this failure mode and this means that inspections for PM actions for diesel engine failures can be scheduled around this running hour mark.

Figure 5. Decision model for optimal preventive replacement age

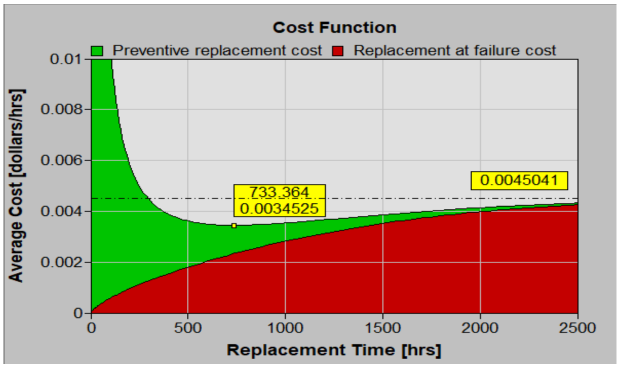
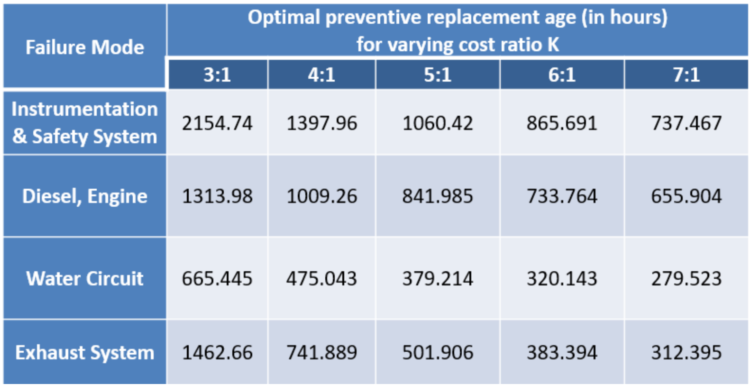


Figure 6 is a table that provides a good rule-of-thumb for optimal PM age in running hours of the engine for each failure mode. Depending on failure mode criticality and expert knowledge of the appropriate k value, the corresponding preventive replacement age can be picked. While it can be unrealistic to carry out PM at these age intervals for just one failure mode, this provides a good framework for designing a checklist for PM inspections. Once PM age intervals are decided, a comprehensive PM policy can correspondingly be designed around decided age intervals.

Figure 6. Optimal preventive replacement age for different values of cost ratio K



Failure Mode	Optimal preventive replacement age (in hours) for varying cost ratio K				
	3:1	4:1	5:1	6:1	7:1
Instrumentation & Safety System	2154.74	1397.96	1060.42	865.691	737.467
Diesel, Engine	1313.98	1009.26	841.985	733.764	655.904
Water Circuit	665.445	475.043	379.214	320.143	279.523
Exhaust System	1462.66	741.889	501.906	383.394	312.395

Conclusion and Future Work

The optimal preventive replacement age table provides a good platform to begin a more cost-centric approach to re-designing PDE maintenance policy. The next steps involve integrating the Oil & Coolant Conditioning Analysis (OCCAP) data to get a PM policy model that takes both age of the asset and condition monitoring information to provide a more holistic model for maintenance policy optimization.

Acknowledgements

I am grateful to Dr. Janet Lam and Prof. Dragan Banjevic for their guidance and advice. I would also like to thank Mr. Jamie Dreyer, Ms. Nicolle Killfoyle and Mr. Kulan Ambalavanar of Department of National Defense for their constant support and enthusiasm that proved crucial during many stages of this project.

UKMOD long-term project planning in the face of uncertainty

Janet Lam

Background

This project originally started as an analysis of the trade-off between achievements and flexibility when it comes to project planning. The overall idea was to consider the outcomes of various long-term projects and their duration, while the environment changed around us. While the original step-change events were paradigm-breaking technologies such as the Internet and smartphones, today's pandemic-driven lock down is another step-change that can waylay the best-laid plans.

The extension of this project is to include the consideration of budget limitations and building toolkits to be used at a later time.

Summary of previous work

In the last report, we discussed the varying payoffs of projects with different duration, which are in direct opposition to exposure resulting in loss of value. For example, we may plan to achieve a specific objective with a 10-year project or a 5-year project. Reasonably, the longer project may be "better" in a variety of ways due to more flexibility in scheduling and resources, whether it is a greater chance of achieving the objectives, or greater profit, or some other metric. However, the longer project has proportionately more exposure to external risk.

For example, once the project starts, a small percent of the potential value of the project is lost every year due to constant but minor updates in technology. Additionally, every year, there's a risk of a step-change event that results in a significant loss in value of the project. As a timely example, an investment into a downtown fine-dining restaurant may have lost significant value in 2020.

In the previous report, we reported back relative values of projects based on their project durations. The equivalence table answered the question: *what value must a project of duration x have to be worth as much or more than a project of duration y?*

Extensions – Budgets and technology cupboards

The work done to date gave us a view into the relationship of project duration and their risk, but did not account for the rest of the research and development realities, such as budgets and investments into existing capabilities.

Consider, for example, a fleet of existing aircraft that is at its mid-life. Then, a given budget can be used to update the fleet or used for a new project/aircraft altogether. Regardless of how it is used, the budget is fixed. The research question then becomes: *how might the budget best distributed to get the best value from our existing project and new project?* As we explore this question, we still maintain the gradual loss of value as the project is progressing, and the probability of a step loss in any given year.

The effect of mid-life refit on existing technology

When existing aircraft is given a refurbishment, there are three ways to categorize its outcome:

- good as new – original technology is restored and is working as if new, returned to 100% of original value
- better than new – the health of the equipment has been restored along with some new technology implemented. Restored to 120% of original value
- better than old – system health is partially restored, but not in as good health as new. Restored to 80% of original value

Depending on the amount of budget that is invested into the refurbishment, the value and results of the refurbishment may be different. For this study, we define *B* as the budget, and break it down into low, medium and high proportions of the budget invested into the refit. With a low proportion of the budget invested into refurbishment, let's say that the value of the existing technology following the refit is determined according to the following probability distribution:

Table 8 Probability distribution of restored value for low budget

% of original value	With probability
100	0.15
120	0.05
80	0.8

Similarly, for the medium and high budget scenarios, the resulting values were set as follows:

Table 9 Probability distribution of restored value for medium and high budget

% of original value	With probability (for med budget)	With probability (for high budget)
100	0.8	0.15
120	0.1	0.8
80	0.1	0.05

Value of new project with budget constraint

With the remainder of the budget, we will invest into our new long-term project, that can either be high, low or medium budget, depending on how much funds are left after refitting the existing technology. If the refit budget investment was low, then the new project can be high, low, or medium budget, since we can opt to choose less than the full budget. However, if the refit investment was high, then the new project budget can only be the low investment option.

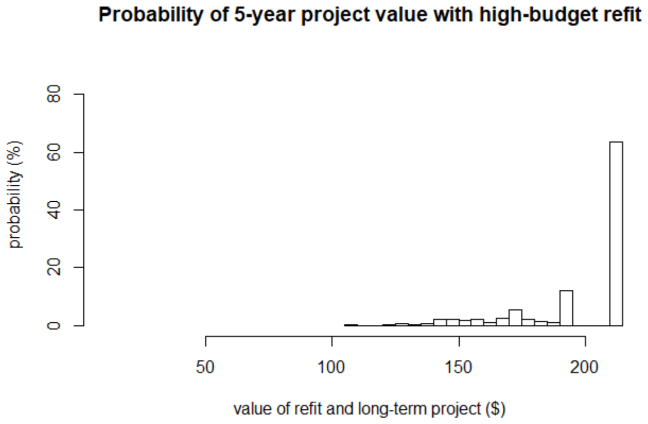
The treatment of the new projects will mirror the previous work that was done. That is, we'll have 5, 10 and 20 year projects representing the low, medium and high investment options respectively, and their probabilistic losses in value will remain the same. The parameters used previously were as follows:

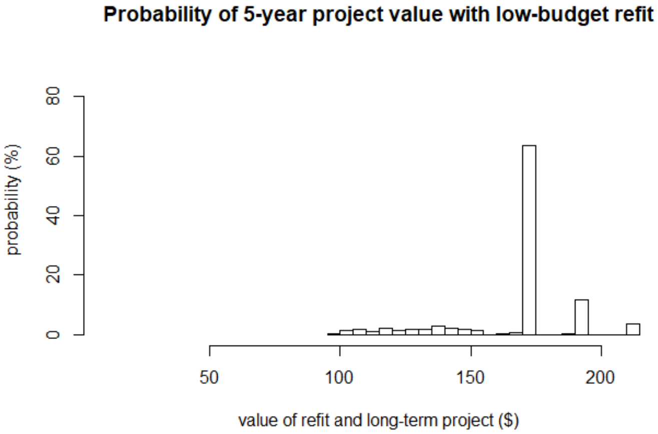
- Annual depreciation rate: U[0.01, 0.02]
- Annual probability of a catastrophic event: 0.05
- Percentage project value lost in the event of a catastrophic event: U[0.2, 0.8]

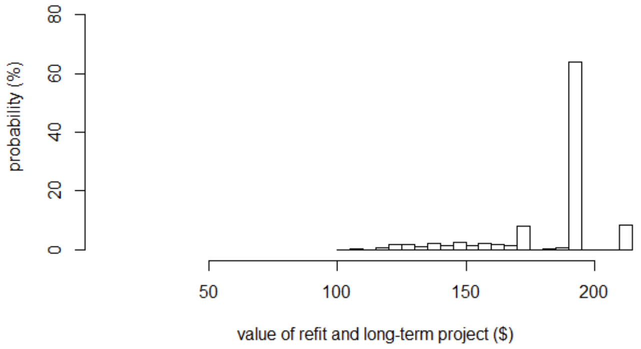
The total project value will be the sum of the two project values.

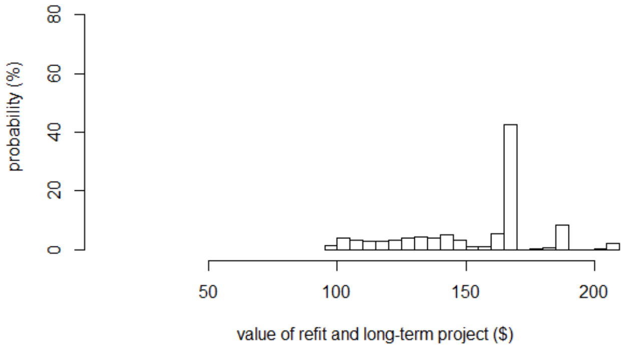
The resulting probability distributions can be seen in the graphs below. The graphs cannot be compared directly, because each project length is given the value "100" to start, but it's not reasonable that a 5-year project and a 20-year project have the same value. Rather, we can explore the shape of the graph and the patterns of the distributions.

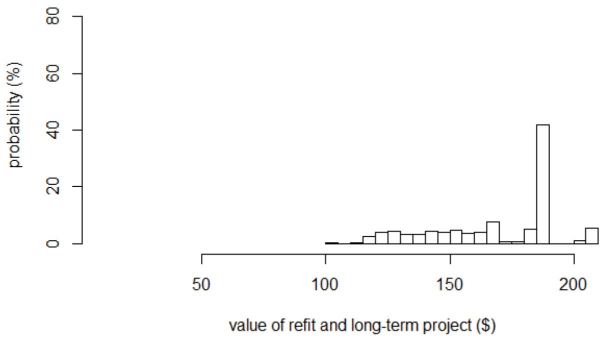
We can see that when a short project is considered (5-years), the contribution to existing technology is more impactful. This can be identified by the very tall peaks seen in the "5-year project" graphs. As the new project durations get longer, the impact of the refit is reduced, as the entropy over time smooths out the possible outcomes.

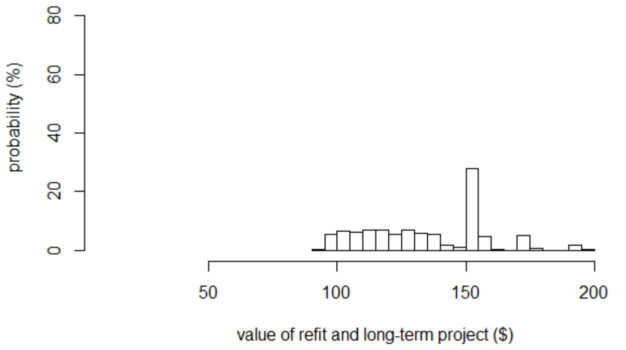





Probability of 5-year project value with medium-budget refit


Probability of 10-year project value with low-budget refit


Probability of 10-year project value with medium-budget refit


Probability of 20-year project value with low-budget refit

Next steps

Though these graphs may demonstrate the changes over time and probabilistic outcomes, the parameters were defined as fixed values based on general knowledge. It would be preferable to set up a system that allows users to experiment with different outcomes and probabilities, as well as relative values of short/long projects so that a more direct comparison may be made. A Shiny app may be developed to allow for user inputs of various parameters, including the value earned by refit investment levels and their associated probabilities, the regular and step losses on the new projects, and more.

Appendix 1. Maintenance scheduling integrating replacement impact

Zihan Zhang

Background

Maintenance scheduling for a complex system requires an accurate evaluation of the degradation states of its constituent components and a sufficient understanding of how these states evolve in the future. These challenges become more complicated when the components of a system are interdependent. There are three common forms of dependencies among components:

(1) Economic dependence: implies that either cost can be saved when several components are jointly maintained instead of separately;
(2) Structural dependence: applies if components structurally form a part, so that maintenance of a failed component implies maintenance of other components as well;
(3) Stochastic dependence: occurs if the state of a component influences the lifetime distribution of other components.

This project focuses on stochastic dependence, which refers to situations where the replacement of one component influences the degradation level of other components in the system. In particular, we assume that when a component is replaced, it affects the performance of other components in the system by accelerating their degradation process.

System characteristics

The system has N_{sa} subassemblies and N_c components. Assume that degradation of component follows Gamma process. Denote the degradation level of component i at time t by $X_i(t)$, we can express degradation process as $X_i(t):\ G(t;\alpha_i,\beta_i)$, where α_i and β_i are corresponding shape parameter and scale parameter of component i, and degradation increment in time interval (s,t) as $\Delta X_i(t-s)=X_i(t)-X_i(s), s<t$. Further, the probability density function (PDF) of component i in (s,t) can be given as

$$f_i\left(x;\alpha_i\left(t-s\right),\beta_i\right)=\frac{\beta_i^{\alpha_i(t-s)}}{\Gamma\left(\alpha_i\left(t-s\right)\right)}x^{\alpha_i(t-s)-1}e^{-\beta_i x}\mathrm{I}\left\{x>0\right\}, \tag{1}$$

where $\Gamma(\cdot)$ is the complete gamma function, and $\mathrm{I}\{\cdot\}$ is the indicator function, which equals 1 when the statement is true in the brace and 0 otherwise. When the system is brand new, the initial degradation levels of all the components are zero, i.e., $X_i(0)=0, i=1,\cdots,N_c$.

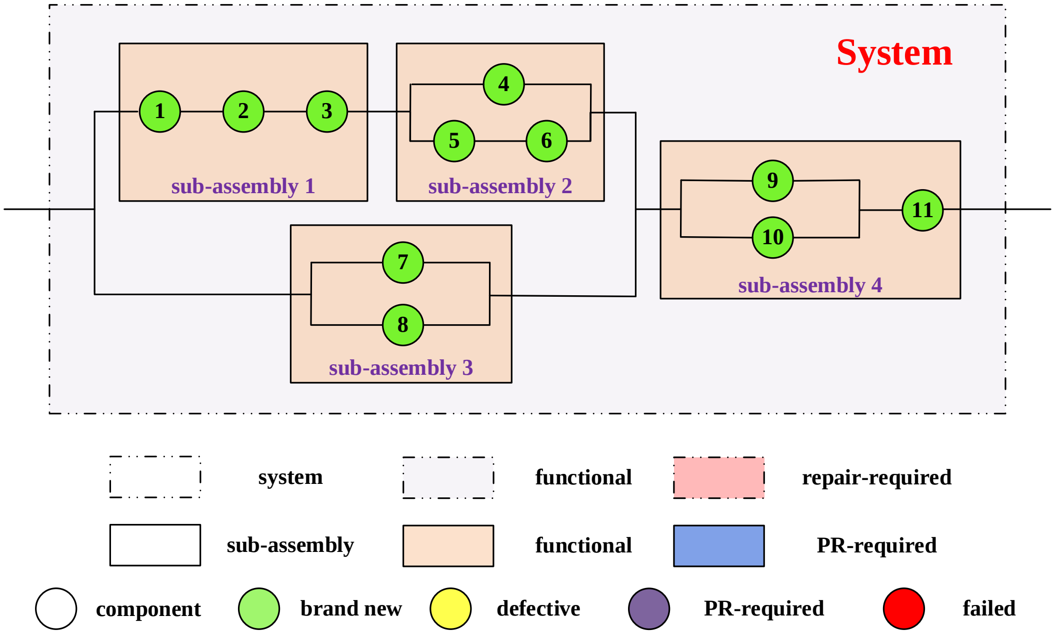


Figure 1. System structure

Reliability analysis

Based on the hierarchical structure of system, system reliability depends on reliability of individual subassemblies, and subassembly reliability is determined by its inside components. Therefore, we first analyze component reliability, then reliability of subassembly/system based on their structural dependence.

Component reliability

Considering component undergoes degradation process, where degradation level may contribute to the increment of hazard rate, we model the hazard rate as a function of degradation level to describe its deterioration mechanism. Notably, the degradation results in increasing the hazard rate instead of leading to malfunction. Let $\lambda\left(X_i\left(t\right)\right)$ be the hazard rate of component i with degradation level $X_i\left(t\right)$. Here, we assume that component degrades stationarily and $\lambda\left(X_i\left(t\right)\right)$ is a linear function of degradation level, i.e., $\lambda\left(X_i\left(t\right)\right)=\kappa X_i\left(t\right)$, then component reliability can be derived from Proposition 1.

- Proposition 1. When component i survives at time t_1 with degradation level $X_i(t_1)$, the reliability of component i at time $t_2\,(t_2 > t_1)$, $R_i^c(t_2;t_1,X_{i,t_1})$, can be expressed as

$$R_i^c\left(t_2;t_1,X_{i,t_1}\right)=\exp\left(-\kappa X_{i,t_1}-\int_0^{t_2-t_1}\alpha_i\log\left(1+\frac{s}{\beta_i/\kappa}\right)ds\right), \tag{2}$$

where $X_{i,t_1}=X_i(t_1)$.

Structure analysis

Assume all components degrades independently during operation. The reliability of system/subassembly is determined by individual components. Denote the structure function by $\phi(t)$, specially, $\phi_s(t)$ for system and $\phi_j^{sa}(t)$ for subassembly j, the reliability of subassembly/system can be derived once the component reliability is available. For instance, if two components are connected in series in a subassembly, the reliability of the subassembly is $\phi^{sa}\left(R_1^c(t),R_2^c(t)\right)=\min\left\{R_1^c(t),R_2^c(t)\right\}$, whereas $\phi^{sa}\left(R_1^c(t),R_2^c(t)\right)=\max\left\{R_1^c(t),R_2^c(t)\right\}$ is for a parallel connection.

Subassembly/system reliability

Because subassembly/system reliability is determined by their inside components, if components inside subassembly j survives at time t_1 and their degradation levels are known, the reliability of subassembly j at time $t_2\,(t_2 > t_1)$, $R_j^{sa}\left(t_2;t_1,\tilde{X}_{t_1}^{c}\right)$, can be given as

$$\begin{aligned}&R_j^{sa}\left(t_2;t_1,\tilde{X}_{t_1}^{c,j}\right)\\&=\phi_j^{sa}\left(R_1^c\left(t_2;t_1,X_{1,t_1}\right),\ldots,R_i^c\left(t_2;t_1,X_{i,t_1}\right),\ldots,R_{|S^j|}^c\left(t_2;t_1,X_{|S^j|,t_1}\right)\right),i\in S^j,\end{aligned} \tag{3}$$

where $\tilde{X}_{t_1}^{c,j}=\left\{X_{1,t_1},\ldots,X_{i,t_1},\ldots,X_{|S^j|,t_1}\right\},i\in S^j$. Analogously, the system reliability, $R_s\left(t_2;t_1,\tilde{X}_{t_1}^{sa}\right)$, can be expressed as

$$\begin{aligned}&R_s\left(t_2;t_1,\tilde{X}_{t_1}^{sa}\right)\\&=\phi_s\left(R_1^{sa}\left(t_2;t_1,\tilde{X}_{t_1}^{c,1}\right),\ldots,R_j^{sa}\left(t_2;t_1,\tilde{X}_{t_1}^{c,j}\right),\ldots,R_{N_{sa}}^{sa}\left(t_2;t_1,\tilde{X}_{t_1}^{c,N_{sa}}\right)\right),\ j=1,\ldots,N_{sa}\end{aligned} \tag{4}$$

where $\tilde{X}_{t_1}^{sa}=\left\{\tilde{X}_{t_1}^{c,1},\ldots,\tilde{X}_{t_1}^{c,j},\ldots,\tilde{X}_{t_1}^{c,N_{sa}}\right\},\ j=1,\ldots,N_{sa}$.

Problem statement

Assumptions

1. At each preventive replacement opportunity, only one component or subassembly can be replaced.
2. The degradation processes of all components are stationary (the premise of proposition 1).
3. If one component has failed, then there could be an opportunity to check the other components within the subassembly, whiling conducting preventive replacement.

Production scenario

All the components start to operate from brand new states, where $X_{i,0}=0$. A fixed mission is allocated to the system to finish. Per mission consumes time T_m and receives c^R production benefits after successful accomplishment. But as time goes by, the system will deteriorate shown in the increasing degradation level of components, and finally break down when the system reliability exceeds the predetermined threshold δ_s^F . In order to ensure the long-term benefits, maintenance activities need scheduling to restore the performance of the system. After restoration, the system restarts its production and repeat the procedure until the planning horizon exceeds T_m.

Maintenance policy

If the system fails during the operation, a repair need executing to return the system to a 'as-good-as-new' state. If the system successfully finishes a mission, preventive replacement (PR) is conducted if the system threshold is lower than the acceptable healthy threshold δ_s^H . Here, two levels of preventive replacement can be scheduled: (a) subassembly level preventive replacement (sPR), and component level preventive replacement (cPR).

Although PR can restore the performance of the replaced component/subassembly, this disassembling action will increase degradation levels of other components, which may result in system malfunction in the next mission. To avoid the underlying malfunction risk, a corrective replacement (CR) is needed. Note that, all the downtime due to maintenance actions will have a penalty of c^D per unit time. Implementation details of maintenance activities are outlined below:

1. **Repair.**

If the system breaks down during the mission, it will be repaired with a cost c^{CR}. Besides, all the repairs take τ_{CR} before starting a new mission. If a failure occurs, then the entire system will be repaired to "as-good-as-new" condition.

2. **Preventive replacement (PR).**

After each mission, if PR satisfies the execution threshold, component/subassembly will be replaced. PR includes a process of dissembling, replacing and reassembling. Here, we regard the cost/time for disassembling/assembling are the same, so we use 'disassembly time/cost' to represent the time/cost related to both processes of disassembling and reassembling. The discrepancies between two levels of PR exist in:

(1) sPR: If it is determined to replace a subassembly, we only need to disassemble the system and subassembly, then replace it without dissembling components inside;
(2) cPR: If it is determined to replace a component, we need to disassemble system, its outside subassembly, and other components in the subassembly; then replace the component.

Denote PR cost/time of component i in subassembly j by c_i^{cPR} / τ_i^{cPR}. They can be expressed as

$$c_i^{cPR} = c^{sd} + c_j^{d} + c_{S_i^j}^{d} + c_i^{cr},$$
$$\tau_i^{cPR} = \tau^{sd} + \tau_j^{d} + \tau_{S_i^j}^{d} + \tau_i^{cr}, \qquad (5)$$

where c^{sd} / τ^{sd} are cost/time of disassembling system, c_j^d / τ_j^d are cost/time of disassembling subassembly j, $c_{S_i^j}^d / \tau_{S_i^j}^d$ are cost/time of disassembling components in subassembly j, c_i^{cr} / τ_i^{cr} are cost/time of replacing component i.

If c_j^{sPR} / τ_j^{sPR} are PR cost/time of subassembly j' by $c_{j'}^{sPR} / \tau_{j'}^{sPR}$, we can obtain

$$c_{j'}^{sPR} = c^{sd} + c_{j'}^{d} + c_{j'}^{sr},$$
$$\tau_{j'}^{sPR} = \tau^{sd} + \tau_{j'}^{d} + \tau_{j'}^{sr}, \qquad (6)$$

where $c_{j'}^{sr} / \tau_{j'}^{sr}$ are cost/time of replacing subassembly j'.

Although PR can restore state of replaced component/subassembly, the disassembling actions will increase degradation levels of other components.

Specially, denote $X_{i,p}^{-}$ as the degradation level of component i after the p-th mission and $X_{i,p}^{+}$ as the degradation level of component i after PR of the p-th mission. Let Y_i be the increment of degradation level of component i if it is affected by disassembling actions. It is assumed that Y_i is independent of the degradation level $X_{i,p}^{-}$ and follows a Gaussian distribution with mean μ_{Y_i} and variance $\sigma_{Y_i}^2$, $Y_i :\ N\left(\mu_{Y_i}, \sigma_{Y_i}^2\right)$, where μ_{Y_i} ? $\sigma_{Y_i}^2$. Based on the above definition, we have

$$X_{i,p}^{+} = X_{i,p}^{-} + Y_i \qquad (7)$$

It is also assumed that when disassembling the system, all the degradation levels of components will not be affected. It is common in real application, such as opening rear hatch to repair a car. Then the degradation impact also has two corresponding levels:

Subassembly level: When disassembling a subassembly, the degradation levels of components inside the sub-assemblies connected with the disassembled assembly will increase εY_i, where $\varepsilon \in (0,1)$. For instance, the disassembly of subassembly 1 will affect subassembly 2 and 3; the disassembly of subassembly 4 will affect subassembly 2 and 3; and the disassembly of subassembly 2/3 will affect all other sub-assemblies except themselves.

Component level: When disassembling components inside a specific subassembly, all the degradation levels of un-replaced components will increase Y_i. Note that because the subassembly has been removed from the system, the component level disassembling actions will not have impact on other components in other sub-assemblies. For example, when replacing component 4 after taking down subassembly 2, the degradation levels of component 5 and 6 will both increase.

3. Opportunistic corrective replacement (OCR).

As PR will influence degradation levels of other components, it may lead to system malfunction in the next mission. Thus, corrective replacement (CR) can be scheduled when replacing components. Note that CR is only for component level PR rather than subassembly level PR. It is because when we have removed the subassembly from the system, we can use the opportunity to replace its inside components which are not replaced by PR, but their degradation levels after PR cannot be accepted. This replacement has no secondary effects on degradation levels of other components, because all the components inside this subassembly have been separated in PR. Therefore, this type of replacement is also an opportunistic corrective replacement (OCR), whose function is two-tier:

(1) Reduce malfunction risk in the following mission;
(2) Share the setup cost.

For notational convenience, we don't make difference between OCR and CR in this work. Denote π_p^j as the decision rule of CR of disassembled subassembly j after PR of p-th mission, that is

$$\pi_p^j = \begin{cases} 1, & \text{CR is executed,} \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

Because CR should be executed when the degradation level of component l inside the disassembled subassembly reaches a pre-specified threshold, defined by

$$P\left(X_{l,p}^{-} + Y_l < \delta_c^l\right) > \xi, \tag{9}$$

where δ_c^l is the CR threshold of component l inside the subassembly j and ξ is the acceptable risk threshold. Since Y_l follows a Gaussian distribution, it follows that

$$\Phi\left(\frac{\delta_c^l - \mu_{Y_l} - X_{l,p}^{-}}{\sigma_{Y_l}}\right) > \xi. \tag{10}$$

Therefore, CR of disassembled subassembly j after PR of p-th mission can be given as

$$\pi_p^j = \begin{cases} 0 & ,\text{if } \sum_{l \neq i}^{S^j} \mathbf{I}\left\{Y_l \geq \mu_{Y_l} + \sigma_{Y_l} \Phi^{-1}(\xi)\right\} = 0, \\ 1 & ,\text{otherwise}. \end{cases} \tag{11}$$

Remark. Because OCR leverages the opportunity of PR, we can take these two types of replacements into consideration and regard them as a preventive maintenance (PM). According to different levels of PR, we analogously divide PM into two levels: PM with sPR (sPM) and PM with cPR (cPM).

As mentioned above, sPM will not provide opportunities to schedule OCR, so the cost/time of sPM is the same as sPR. We can easily obtain cost/time of sPM of subassembly j', $c_{j'}^{sPM}$ / $\tau_{j'}^{sPM}$ by

$$\begin{aligned} c_{j'}^{sPM} &= c^{sd} + c_{j'}^{d} + c_{j'}^{sr}, \\ \tau_{j'}^{sPM} &= \tau^{sd} + \tau_{j'}^{d} + \tau_{j'}^{sr}, \end{aligned} \tag{12}$$

For cPM of component i in subassembly j, its cPM cost/time c_j^{cPM} / τ_j^{cPM} can be expressed as

$$\begin{aligned} c_i^{cPM} &= c^{sd} + c_j^{d} + c_{S_i^j}^{d} + c_i^{cr} + \pi_p^j \cdot \sum_{l \neq i}^{S^j} c_l^{cr}, \\ \tau_i^{cPM} &= \tau^{sd} + \tau_j^{d} + \tau_{S_i^j}^{d} + \tau_i^{cr} + \pi_p^j \cdot \sum_{l \neq i}^{S^j} \tau_l^{cr}. \end{aligned} \tag{13}$$

Appendix 2. Bayesian distributional reinforcement learning

Kuilin Chen

Introduction

Deep reinforcement learning (DRL) has achieved great success in some sequential decision making problems, such as Atari games [7]. Although researchers are trying to expand DRL in practical applications, it is not straightforward to apply DRL in safety critical tasks, such as equipment maintenance, robot manipulation and manufacturing process. When the agent is learning, it must take non-optimal actions to learn more about the environment. Classic exploration approaches like ε-greedy rely on random actions for exploration, which could lead to catastrophic consequences. The purpose of this research is to develop a risk-sensitive agent that visits dangerous states as rarely as possible during the learning phase.

A promising approach to develop a risk-sensitive agent is through distributional RL [1] since distributions over Q-values allow risk-sensitive decision making. In classic value-based reinforcement learning, the value functions $\mathrm{Q(s,a)}$ for state-action (s,a) pairs are learned, which describe the expectation of total discounted rewards following a prescribed policy. An optimal policy can be derived according to Bellman's optimality equation by finding the action a in state s that leads to the optimal value function $\mathrm{Q^*(s,a)}$ [13]. Note that the total discounted rewards are random due to the stochasticity in environment and policy. Instead of learning a point estimation, distributional reinforcement learning algorithms are developed to maintain the full distribution of future return [5, 11, 14]. Distributional reinforcement learning is able to mitigate the chattering in learning caused by noisy samples combined with function approximation through effectively averaging the distributions [1]. In addition, the uncertainty information in the distribution can be used for efficient exploration and risk-sensitive learning.

Existing distributional DRL methods approximate the continuous distribution of Q-values with finite support [15]. Frequent visits to dangerous states cannot be avoided in learning phase because ε-greedy method is used for exploration, though certain quantile of the Q-value distribution is used as the objective function to mitigate the risk in normal operation [3, 2]. In this article, we develop a Bayesian distributional DRL method, which models the distribution of Q-values by a Gaussian Process (GP) regression layer. In addition, posterior sampling is utilized to explore the environment efficiently, with less visits to dangerous states. The effectiveness of the proposed method is demonstrated by numerical studies and can be applied to a wide range of safety critical tasks.

Background

We consider a Markov decision process (MDP) defined by a tuple $(\mathcal{S}, \mathcal{A}, \mathrm{p}, \mathrm{R})$, where $\mathcal{S}$ is the state space , and $\mathcal{A}$ is the action space. The unknown state transition probability $p\colon \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, +\infty)$ represents the probability density of the next state $\mathrm{s}' \in \mathcal{S}$ given the current state $\mathrm{s} \in \mathcal{S}$ and action $\mathrm{a} \in \mathcal{A}$. A bounded reward $\mathrm{R}\colon \mathcal{S} \times \mathcal{A} \rightarrow [\mathrm{R}_{\min}, \mathrm{R}_{\max}]$ is emitted by the environment after each state transition. Under a given policy π, the expectation of future return of an action a in a state s is

$$\mathrm{Q}_\pi(\mathrm{s}, \mathrm{a}) \equiv \mathrm{E}[\mathrm{R}_1 + \gamma \mathrm{R}_2 + \cdots | \mathrm{S}_0 = \mathrm{s}, \mathrm{A}_0 = \mathrm{a}, \pi]$$

where $\gamma \in [0,1]$ is a discount factor to trade off the importance of immediate and future rewards. An optimal policy is derived from the optimal value $\mathrm{Q}^*(\mathrm{s}, \mathrm{a}) = \max_\pi \mathrm{Q}_\pi(\mathrm{s}, \mathrm{a})$ by selecting the highest valued action in each state [13].

Bayesian deep Q-learning

According to Central Limit Theorem, $\sum_{t=0}^{+\infty} \gamma^t \mathrm{R}_t$ follows Gaussian distribution even R_t does not follow Gaussian distribution. Therefore, it is reasonable to place a Gaussian process (GP) prior $\mathrm{p}(\mathrm{q}) = \mathcal{N}(0, \mathrm{K}_{\mathrm{S},\mathrm{S}})$ on $\mathrm{Q}(\mathrm{s}, \mathrm{a})$. Each element in the kernel matrix $(\mathrm{K}_{\mathrm{S},\mathrm{S}})_{\mathrm{i},\mathrm{j}}$ is computed by the squared exponential kernel function as follows [10]

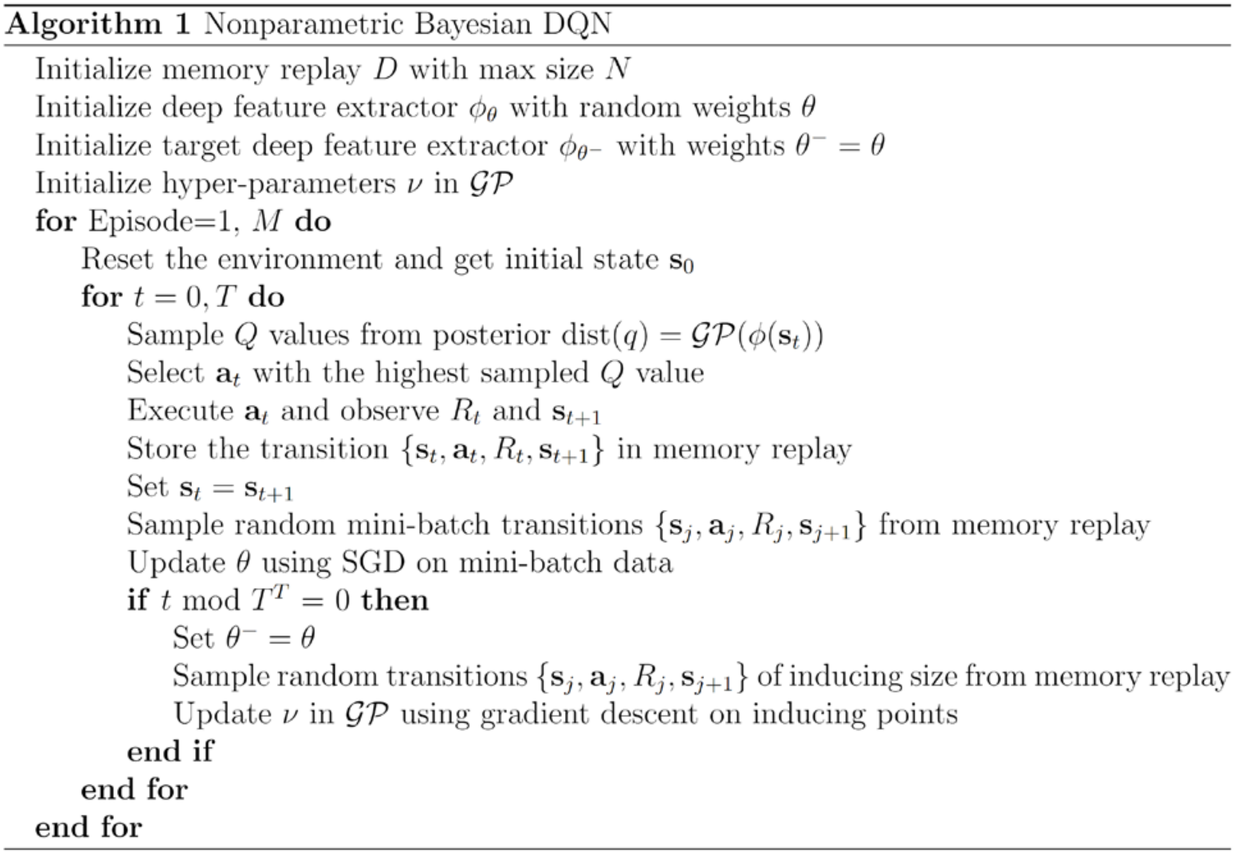
$$\kappa(\mathrm{s}_\mathrm{i}, \mathrm{s}_\mathrm{j}) = \exp\left(\frac{\left(\phi_\theta(\mathrm{s}_\mathrm{i}) - \phi_\theta(\mathrm{s}_\mathrm{j})\right)^2}{-2\lambda}\right)$$

where ϕ_θ is a deep neural network based feature extractor parameterized by θ and λ is the length-scale in the squared exponential kernel that controls the smoothness of the function. In addition, a target neural network parameterized by θ^- is maintained to provide noisy target values y. It is assumed that y is obtained by adding Gaussian noise $\mathcal{N}(0, \sigma^2)$ to the unobserved true value function $\mathrm{Q}(\mathrm{s}, \mathrm{a})$. The mean and covariance in posterior distribution of Q-value at a new state s^* can be derived as follows

$$\mathrm{E}[\mathrm{q}_*] = \mathrm{K}_{\mathrm{s}_*,\mathrm{S}}\left[\mathrm{K}_{\mathrm{S},\mathrm{S}} + \sigma^2 \mathrm{I}\right]^{-1} \mathrm{y}$$
$$\mathrm{cov}(\mathrm{q}_*) = \mathrm{K}_{\mathrm{s}_*,\mathrm{s}_*} - \mathrm{K}_{\mathrm{s}_*,\mathrm{S}}\left[\mathrm{K}_{\mathrm{S},\mathrm{S}} + \sigma^2 \mathrm{I}\right]^{-1} \mathrm{K}_{\mathrm{S},\mathrm{s}_*}$$

Computing $\mathrm{p}(\mathrm{q}_* | \mathrm{s}_*, \mathrm{S}, \mathrm{y})$ requires access to all previous observed states and values, which is not possible in modern DRL settings. Sparse GP can be developed by selecting a subset data from memory replay as inducing points for GPs [4, 6, 8, 12].

Instead of selecting a random action with a small probability to do exploration, action selection is done through posterior sampling, which naturally balances the exploitation and exploitation. We sample the values for state-actions pairs from their posterior distribution and select the action with the highest sampled values. The variance of values for visited state-action pairs are usually smaller. If such state-action pairs lead to dangerous next states, the means of the values are also lower. Therefore, it is less likely to select an action that leads to undesirable states in posterior sampling. Meanwhile, the variance for unvisited state-action pairs are larger, which could be selected posterior sampling to encourage exploration.

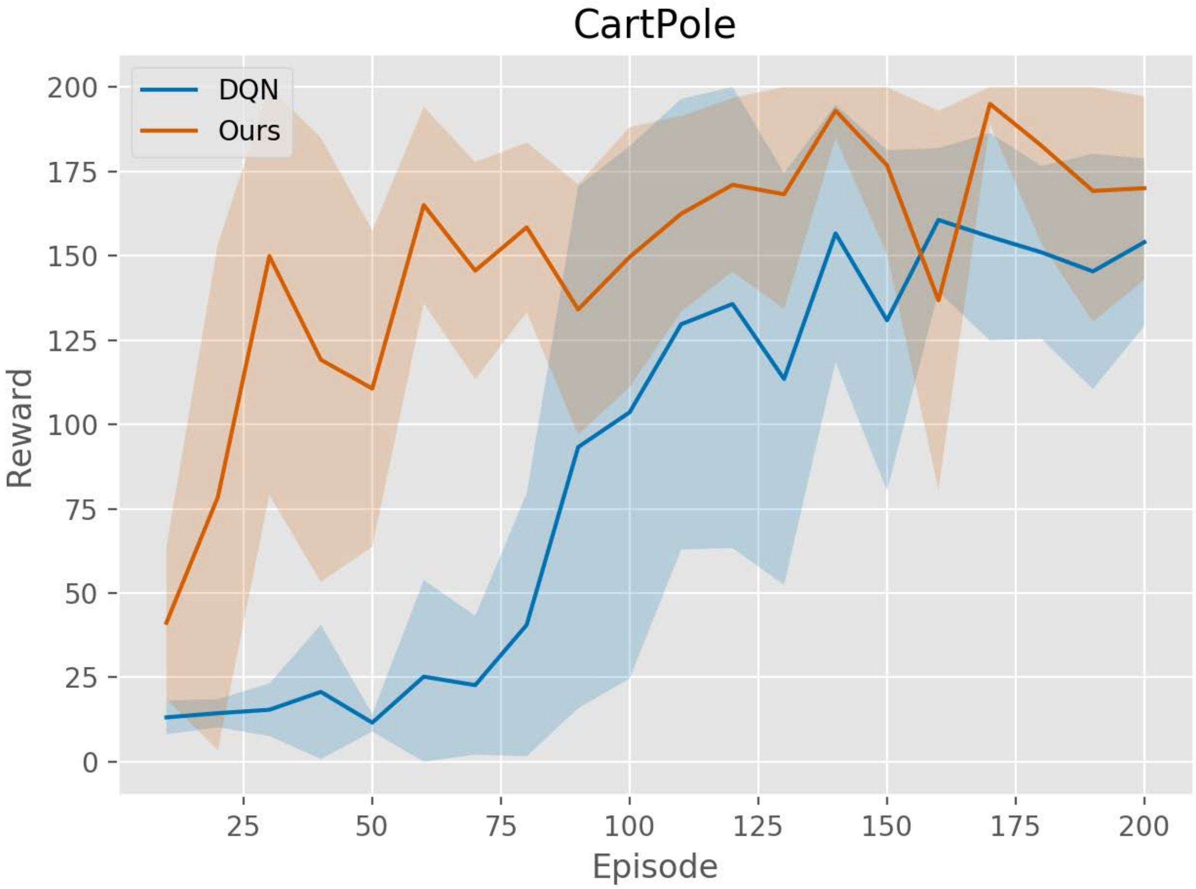


Algorithm 1 Nonparametric Bayesian DQN

```
Initialize memory replay $D$ with max size $N$
Initialize deep feature extractor $\phi_\theta$ with random weights $\theta$
Initialize target deep feature extractor $\phi_{\theta^-}$ with weights $\theta^- = \theta$
Initialize hyper-parameters $\nu$ in $\mathcal{GP}$
for Episode=1, $M$ do
    Reset the environment and get initial state $\mathbf{s}_0$
    for $t = 0, T$ do
        Sample $Q$ values from posterior dist$(q) = \mathcal{GP}(\phi(\mathbf{s}_t))$
        Select $\mathbf{a}_t$ with the highest sampled $Q$ value
        Execute $\mathbf{a}_t$ and observe $R_t$ and $\mathbf{s}_{t+1}$
        Store the transition $\{\mathbf{s}_t, \mathbf{a}_t, R_t, \mathbf{s}_{t+1}\}$ in memory replay
        Set $\mathbf{s}_t = \mathbf{s}_{t+1}$
        Sample random mini-batch transitions $\{\mathbf{s}_j, \mathbf{a}_j, R_j, \mathbf{s}_{j+1}\}$ from memory replay
        Update $\theta$ using SGD on mini-batch data
        if $t$ mod $T^T = 0$ then
            Set $\theta^- = \theta$
            Sample random transitions $\{\mathbf{s}_j, \mathbf{a}_j, R_j, \mathbf{s}_{j+1}\}$ of inducing size from memory replay
            Update $\nu$ in $\mathcal{GP}$ using gradient descent on inducing points
        end if
    end for
end for
```

Results

The proposed method is applied to a toy benchmark task to demonstrate its effectiveness. In this task, a pole is attached by an un-actuated joint to a cart, which moves along a frictionless track. The agent controls the system by applying force towards right or left to the cart. The task fails when the pole is more than 15 degrees from vertical, or the cart moves more than 2.4 units from the center. The goal of the agent is to prevent the pole from falling. The same neural network architecture is used in both the proposed method and the standard DQN, except that the last layer in the proposed method is a Gaussian Process regression layer. The experiment is repeated 100 times with random initial conditions. The results indicate that the proposed method is superior in learning efficiency in safety sensitivity. The high rewards by the proposed method is achieved by avoiding dangerous states (such as pole falling).



Conclusions

A Bayesian distributional RL algorithm is proposed to do efficient and safe exploration. The proposed algorithm outperforms the standard DQN algorithm in terms of sample efficiency. The total discounted reward is modelled by a Gaussian Process regression layer in the proposed method. It is natural to consider Bayesian optimization techniques to further improve efficiency and safety in the learning phase.

References

[1] Marc G Bellemare, Will Dabney, and Remi Munos. A distributional perspective on reinforcement learning. In Proceedings of the 34th International Conference on Machine Learning-Volume 70, pages 449–458. JMLR. org, 2017.

[2] Will Dabney, Georg Ostrovski, David Silver, and Remi Munos. Implicit quantile net-works for distributional reinforcement learning. In International Conference on Machine Learning, pages 1096–1105, 2018.

[3] Will Dabney, Mark Rowland, Marc G Bellemare, and Remi Munos. Distributional reinforcement learning with quantile regression. In Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[4] James Hensman, Alexander Matthews, and Zoubin Ghahramani. Scalable variational gaussian process classification. In Artificial Intelligence and Statistics, pages 351–360,2015.

[5] Stratton C Jaquette. Markov decision processes with a new optimality criterion: Discrete time. The Annals of Statistics, pages 496–505, 1973.

[6] Alexander G de G Matthews, James Hensman, Richard Turner, and Zoubin Ghahra-mani. On sparse variational methods and the kullback-leibler divergence between stochastic processes. In Artificial Intelligence and Statistics, pages 231–239, 2016.8

[7] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. Nature, 518(7540):529–533, 2015.

[8] Joaquin Quinonero-Candela and Carl Edward Rasmussen. A unifying view of sparse approximate gaussian process regression. Journal of Machine Learning Research,6(Dec):1939–1959, 2005.

[9] Carl Edward Rasmussen. Gaussian processes in machine learning. In Summer School on Machine Learning, pages 63–71. Springer, 2003.

[10] Bernhard Scholkopf, Alexander J Smola, Francis Bach, et al. Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press, 2002.

[11] Matthew J Sobel. The variance of discounted markov decision processes. Journal of Applied Probability, 19(4):794–802, 1982.

[12] Michalis Titsias. Variational learning of inducing variables in sparse gaussian processes.InArtificial Intelligence and Statistics, pages 567–574, 2009.

[13] Christopher JCH Watkins and Peter Dayan. Q-learning. Machine learning, 8(3-4):279–292, 1992.

[14] DJ White. Mean, variance, and probabilistic criteria in finite Markov decision processes: A review. Journal of Optimization Theory and Applications, 56(1):1–29, 1988.

[15] Derek Yang, Li Zhao, Zichuan Lin, Tao Qin, Jiang Bian, and Tie-Yan Liu. Fully parameterized quantile function for distributional reinforcement learning. In Advances in Neural Information Processing Systems, pages 6190–6199, 2019.9

Appendix 3. Deep partial transfer learning network: a method to selectively transfer diagnostic knowledge across related machines

Bin Yang, Chi-Guhn Lee, Yaguo Lei, Naipeng Li

Introduction

Many of the recent successes of machine learning have been driven by deep learning, and intelligent fault diagnosis is no exception. Deep learning-based diagnostic models achieve superior performance to traditional approaches. However, the reliable deep learning-based models are obtained based on training with a large amount of labeled data, which is often unavailable in real-world engineering applications. As a result, many diagnostic models that perform strongly in laboratory settings struggle in real-world situations.

Transfer learning has become a promising solution to close the gap between performance in the laboratory and in the real world by allowing diagnostic knowledge gained in one or more diagnostic tasks to be transferred to another. The tasks from which the knowledge is transferred out of is called the source domain, whereas the other (which the knowledge is transferred to) refers to the target domain. The former is where the diagnostic models can be fully trained with the large number of labeled data, and the latter is where the models are difficult to be trained due to the lack of labeled data. There are three commonly-used transfer learning approaches in intelligent fault diagnosis: instance-based, feature-based, and model-based approaches. Feature-based approaches have shown the best transfer performance among the three especially for the tasks subject to serious discrepancy across domains, such as the scenarios to transfer diagnostic knowledge across machines of different yet similar types. They map data into a common feature space and then extract transferrable features that follow similar distributions across domains. Through the extracted features, diagnostic models trained with the source domain data can be reused on the target domain.

Deep learning has been used to construct nonlinear mappings from the original data space to a common feature space. The learned domain-invariant features often follow different distributions depending on whether they are from the source domain or the target domain, but they are beneficial to improving performance of transfer learning-based diagnostic models. According to the recent literature review, the maximum mean discrepancy (MMD) serves as an effective way to estimate the distribution discrepancy. It has been widely concerned to help adapt the distributions of learned transferrable features so as to improve the performance of diagnostic models. In addition, the generative adversarial network (GAN) is also considered to solve the transfer learning tasks about machinery fault diagnosis.

A common assumption among the existing studies on deep transfer learning for diagnostic models is domain symmetry, which involves two requirements on the source and target domains. First, the target domain samples are balanced across all health states of machines. Second, the diagnostic knowledge required by the target domain is consistent with the source domain. In reality, however, such constraints are unrealistic. For the first requirement, during the operation of machines in engineering scenarios (the target domain), the healthy phase typically accounts for the majority of the entire useful life cycle, while faults are less frequently experienced. As a result, the collected dataset consists of massive healthy samples and insufficient faulty samples. As for the second requirement, some typical faults observed in the source domain may never be experienced in the target domain so that only partial diagnostic knowledge in the source domain is useful to the target domain. Such asymmetry will likely result in significant reduction in diagnostic performance of models. Therefore, we propose to address issues arising from the asymmetry across domains in the course of transferring diagnostic knowledge.

We assume that the source and target domains are asymmetric with respect to their health state set and the degree of imbalance among available health states. Specifically, the source domain has sufficient labeled data with possibly balanced samples across all health states, whereas the target domain has unlabeled data, and its set of health states is strictly included in those of the source domain and the samples are possibly imbalanced across every health states. This is a realistic assumption since we can easily obtain desired data in the source domain to provide diagnostic knowledge for the target domain. For example, it is convenient to simulate the operations of an equipment and generate possibly complete faults exhaustively in a laboratory, and the collected data could follow balanced samples across all health states. Affected by the domain asymmetry, the diagnostic models trained in a source domain may not correctly recognize the health states of target domain samples.

In order to achieve the aforementioned partial transfer learning task, we propose an adversarial network architecture-based diagnostic model with transfer capability in presence of domain asymmetry, which we call the deep partial transfer learning network (DPTL-Net). In DPTL-Net, domain-asymmetry factors are automatically learned to weight MMD-based distribution adaptation so that partial diagnostic knowledge can be selectively transferred.

Deep partial transfer learning network

The architecture of the proposed DPTL-Net is shown in Fig. 1, which includes a domain-shared ResNet, a domain discriminator, and an importance weighted distribution adaptation module. Each part is detailed as follows.

Domain-shared ResNet

The domain-shared RestNet has two identically configured networks for the source and target domains but they share the training parameters. The cross-domain samples $\boldsymbol{x}_i^{\mathcal{D}} = \{\boldsymbol{x}_i^{\mathrm{s}}, \boldsymbol{x}_i^{\mathrm{t}}\} \in \Re^N$ are first handled by a convolutional layer through

$$\boldsymbol{x}_i^{\mathcal{D},\mathrm{inp}} = f_{\mathrm{inp}}(\boldsymbol{x}_i^{\mathcal{D}}; \boldsymbol{\theta}^{\mathrm{inp}}) = \sigma_{\mathrm{r}}(\boldsymbol{x}_i^{\mathcal{D}} * \boldsymbol{k}^{\mathrm{inp}} + \boldsymbol{b}^{\mathrm{inp}}), \tag{1}$$

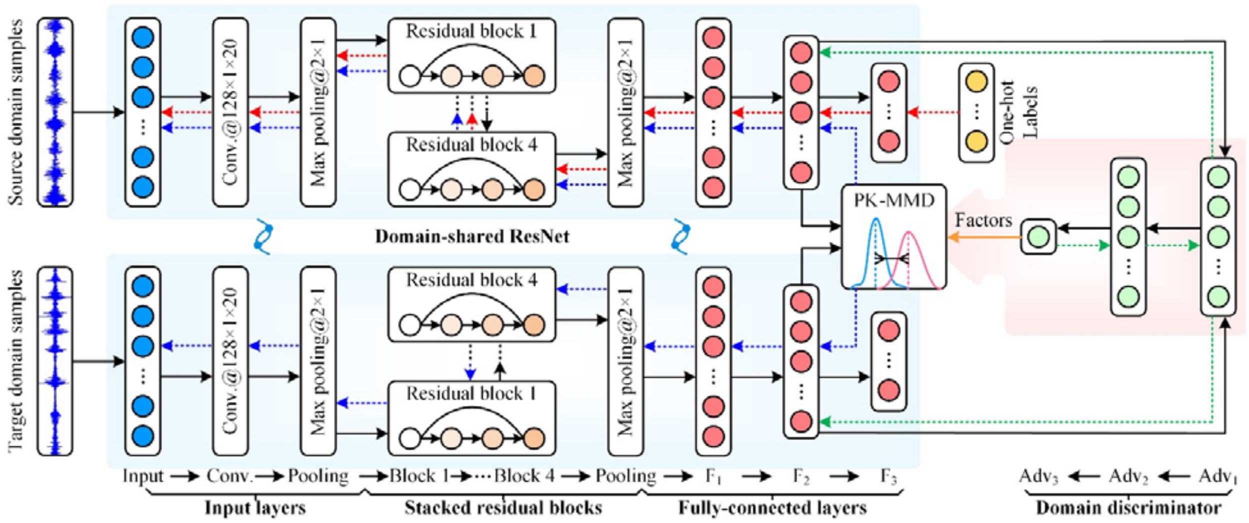


Figure 1. Architecture of proposed DPTL-Net (dotted lines show error back propagation).

where $\boldsymbol{\theta}^{\text{inp}} = \{\boldsymbol{k}^{\text{inp}}, \boldsymbol{b}^{\text{inp}}\}$ is the training parameters of the input layer, and $\sigma_{\text{r}}(\cdot)$ is the activation function of rectified linear unit (ReLU). After that, the learned transferrable features are put forward into a max-pooling layer and become

$$\boldsymbol{v}_i^{\mathcal{D},\text{inp}} = \text{down}(\boldsymbol{x}_i^{\mathcal{D},\text{inp}}, s) \tag{2}$$

where s is the coefficient to control the dimension of the output features.

In order to get high-level transferrable features, the resulting features are put forward to the stacked residual blocks. Each residual block consists of two convolutional layers, and its output is calculated by the sum of features processed before and after the two convolutional layers. Such process is called the short connection as follows:

$$\boldsymbol{x}_i^{\mathcal{D},l} = \sigma_{\text{r}}\left[f_{\text{RB}}\left(\boldsymbol{x}_i^{\mathcal{D},l-1}; \boldsymbol{\theta}^l\right) + \boldsymbol{x}_i^{\mathcal{D},l-1}\right], l = 1,2, \cdots L, \tag{3}$$

where $\boldsymbol{x}_i^{\mathcal{D},0} = \boldsymbol{v}_i^{\mathcal{D},\text{inp}}$, f_{RB} is the nonlinear mapping constructed by each residual block, $\boldsymbol{\theta}^l$ is the training parameters in the lth residual block.

The features output from stacked residual blocks are processed by a max-pooling layer again, and then they are mapped into the source domain label space by the fully-connected network including three layers. The layer F_1 directly flattens the high-level features as one dimensional vectors. The output of the hidden layer F_2 represents the highest-level features before classification. The output layer F_3 includes $|\mathcal{Y}^s|$ neurons, which predict the health states of transferrable features by using the Softmax function. The nonlinear mapping of the fully-connected layers can be expressed as follows:

$$P(\tilde{y}_i^{\mathcal{D}} = j) = f_{\text{FC}}\left(\boldsymbol{v}_i^{\mathcal{D},L}; \boldsymbol{\theta}^{\text{FC}}\right), \tag{4}$$

where $P(\tilde{y}_i^{\mathcal{D}} = j)$ is the probability of the input ith features belonging to the jth health state, $\boldsymbol{v}_i^{\mathcal{D},L}$ is the flattened features in the layer F_1, and $\boldsymbol{\theta}^{\text{FC}}$ represents the training parameters of the fully-connected layers. Due to the lack of sample labels in the target domain, the domain-shared ResNet is trained by using the cross-entropy loss of the source domain samples, which is shown as follows:

$$\mathcal{L}_{\mathrm{c}} = -\frac{1}{n_{\mathrm{s}}}\sum_{i=1}^{n_{\mathrm{s}}}\sum_{j=1}^{k}\boldsymbol{I}(y_i^{\mathrm{s}} = j)\log P(\tilde{y}_i^{\mathrm{s}}), \tag{5}$$

where $\boldsymbol{I}(\cdot)$ is the indicator function.

Domain discriminator

The domain discriminator is performed by a three-layer perceptron. It maps the high-level features of the layer F_2 into the domain label space. The output of the domain discriminator can be expressed as

$$x_i^{\mathcal{D},\mathrm{Adv}_3} = f_{\mathrm{adv}}(\boldsymbol{x}_i^{\mathcal{D},\mathrm{F}_2};\boldsymbol{\theta}^{\mathrm{adv}}), \tag{6}$$

where $f_{\mathrm{adv}}\colon \boldsymbol{x}_i^{\mathcal{D},\mathrm{F}_2} \mapsto \Omega$ is a nonlinear mapping from feature space to the domain label space Ω, and $\boldsymbol{\theta}^{\mathrm{adv}}$ is the training parameters. The discriminator includes two hidden layers, i.e., Adv_1 and Adv_2, and the output layer Adv_3 lacks the activation function. The Wasserstein loss below is maximized to train the domain discriminator.

$$\mathcal{L}_{\mathrm{adv}} = \frac{1}{n_{\mathrm{s}}}\sum_{i=1}^{n_{\mathrm{s}}} x_i^{\mathrm{s},\mathrm{Adv}_3} - \frac{1}{n_{\mathrm{t}}}\sum_{j=1}^{n_{\mathrm{t}}} x_j^{\mathrm{t},\mathrm{Adv}_3}. \tag{7}$$

After each iteration to train the domain discriminator, the domain-asymmetry factors are generated for the source domain samples, and they are calculated as follows:

$$\mathcal{W}_i^s = \frac{1-\sigma_{\mathrm{s}}\left(x_i^{\mathrm{s},\mathrm{Adv}_3}\right)}{1/n_{\mathrm{s}}\cdot\sum_{i=1}^{n_{\mathrm{s}}}\left[1-\sigma_{\mathrm{s}}\left(x_i^{\mathrm{s},\mathrm{Adv}_3}\right)\right]}, \tag{8}$$

where $\sigma_{\mathrm{s}}(\cdot)$ is the Sigmoid function. The factors are further put forward to weight PK-MMD in the distribution adaptation module.

Importance weighted distribution adaptation module

The domain-asymmetry factor weighted PK-MMD is used to estimate the distribution discrepancy of the learned transferrable features in the layer F_2 (the highest feature layer apart from the layer F_3 for classification) when the transfer learning tasks are subject to the domain asymmetry. The distribution discrepancy of learned transferrable features are corrected by using the following loss function:

$$\mathcal{L}_{\mathrm{DA}} = D_{\mathrm{P}\mathcal{H}}^2\left(\boldsymbol{\mathcal{W}}^s X^{\mathrm{s},\mathrm{F}_2}, X^{\mathrm{t},\mathrm{F}_2}\right). \tag{9}$$

where $X^{\mathrm{s},\mathrm{F}_2}$ and $X^{\mathrm{t},\mathrm{F}_2}$ are respectively the learned transferrable features of the batch-size samples in the layer F_2, and $\boldsymbol{\mathcal{W}}^s = \{\mathcal{W}_i^s | i = 1,2,\cdots n_{\mathrm{s}}\}$ is the domain-asymmetry factors.

Training process

Different from the training of the standard WGAN, DPTL-Net individually maximizes the Wasserstein loss to train the domain discriminator, which generates factors to estimate the domain asymmetry of transferrable features during the distribution adaptation process. WGAN essentially corrects the Wasserstein distance of the transferrable features when the gradients of the Wasserstein loss are reversed to propagate toward the domain-shared ResNet In such distribution adaptation process, it is noted that the source domain samples are not weighted, which will result in contradictory effects with the importance weighted distribution adaptation module. Therefore, the gradients of the Wasserstein loss do not propagate to update the training parameters of the domain-shared ResNet in this article. The training of DPTL-Net adopts the commonly-used min-max strategy (two steps) in the standard GAN. In the first step, the discriminator is trained to generate domain-asymmetry factors for source domain samples.

$$\max_{\boldsymbol{\theta}^{\mathrm{adv}}} \ \mathcal{L}_{\mathrm{adv}}. \tag{10}$$

It is noted that Eq. (10) trains the domain discriminator by maximizing the Wasserstein loss of the outputs. According to the basic principle of WGAN, the training parameters $\boldsymbol{\theta}^{\mathrm{adv}}$ must be subject to the K-Lipschitz continuity, otherwise the exploding gradients will result in an instable discriminator and further forward incorrect factors to the distribution adaptation module. Therefore, the training parameters are truncated into the range of $[-\xi, \xi]$ after each iteration to update parameters.

In the second step, the domain-shared ResNet is trained by simultaneously minimizing the loss functions shown in Eq. (5) and Eq. (9).

$$\min_{\boldsymbol{\theta}} \ \mathcal{L}_{\mathrm{c}} + \lambda \cdot \mathcal{L}_{\mathrm{DA}}, \tag{11}$$

where $\boldsymbol{\theta} = \{\boldsymbol{\theta}^{\mathrm{inp}}, \boldsymbol{\theta}^{l=1:L}, \boldsymbol{\theta}^{\mathrm{FC}}\}$ is the training parameters of the domain-shared ResNet, and λ is the tradeoff parameter. The RMSProp optimization algorithm (non-momentum based algorithm) is used to respectively implement Eqs. (10)~(11) due to its capability in solving the instable optimization objective, and the training parameters of DPTL-Net are updated in turn as

$$\begin{aligned} &\boldsymbol{\theta}^{\mathrm{adv}} \leftarrow \boldsymbol{\theta}^{\mathrm{adv}} + \eta \cdot \nabla_{\boldsymbol{\theta}^{\mathrm{adv}}}(\mathcal{L}_{\mathrm{adv}}) \\ &\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \cdot \nabla_{\boldsymbol{\theta}}(\mathcal{L}_{\mathrm{c}} + \lambda \cdot \mathcal{L}_{\mathrm{DA}}) \end{aligned}, \tag{12}$$

where η is the learning rate.

Case studies

Planet gearbox dataset description

The effectiveness of the proposed DPTL-Net is demonstrated on planet gearbox datasets, which are detailed in Table 1.

The datasets A and B are collected from a drivetrain dynamic simulator rig, which includes a motor, a two-stage planet gearbox, a fixed-shaft gearbox, and a magnetic power brake. The motor power is transmitted to drive the magnetic power brake by passing through the planet gearbox and the fixed-shaft gearbox in turn. We respectively simulated five health states in the first stage of the planet gearbox including the normal (N) state, the tooth crack on the sun gear (CS), the wear on the sun gear (WS), the tooth crack on the planet gear (CP), and the bearing fault of the planet gear (BFP). During the experiment, the datasets A and B are respectively collected under the motor rotating speeds of 2100 r/min and 3000 r/min with the sampling frequency of 5120 Hz. The two datasets both contain 1000 samples, and each sample contain 2560 sampling points. In the dataset A, all the samples are balanced across every health states. As for the dataset B, we remove samples from partial faulty types and randomly select $m\% \in (0,1)$ faulty samples to create imbalanced dataset.

Table 1. Details of planet gearbox datasets

Datasets	Health states	The number of samples	Working conditions
A	N CS WS CP BFP	1000 (200×5)	2100 r/min
B	N	200	3000 r/min
	CS WS CP BFP	800 (Partial faulty types are missing and m% ∈ (0,1) faulty samples are randomly available.)	

According to Table 1, the diagnostic knowledge contained in the dataset A (the source domain) is expected to be performed on the dataset B (the target domain). Affected by the different rotating speed of the motor, samples in the two datasets are subject to distribution discrepancy. We create partial transfer learning tasks shown in Table 2. The label space of the target domain is inconsistent with the source domain, and we randomly select 30% faulty samples to make the target domain dataset imbalanced. The partial transfer learning tasks T1, T2, and T3 are expected to demonstrate the performance of DPTL-Net as the health states of the target domain dataset are gradually removed.

Table 2. Partial transfer learning tasks

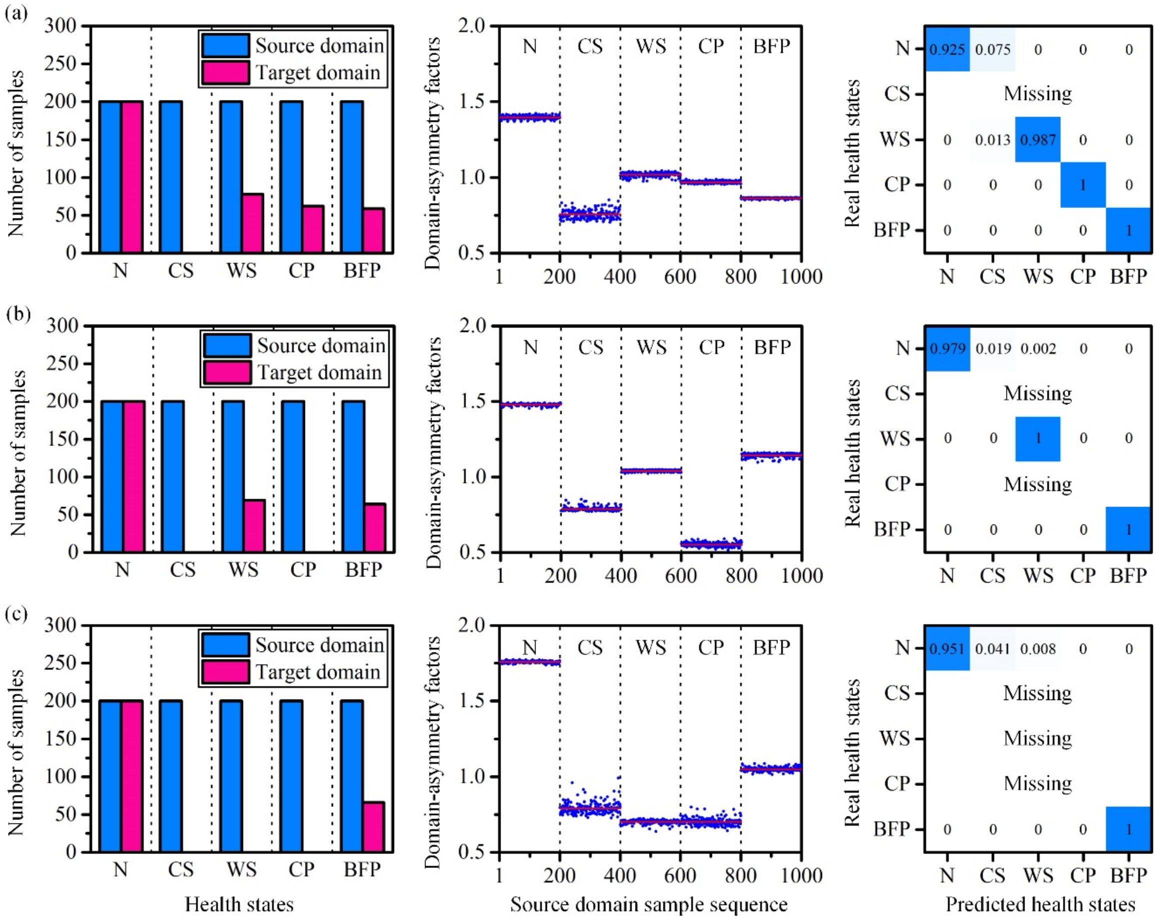
Tasks	Target domain (dataset B)	Source domain (dataset A)
T1	N, WS, CP, BFP	N, CS, WS, CP, BFP
T2	N, WS, BFP	
T3	N, BFP	
Note: Randomly select $m\% \in (0,1)$ faulty samples to create imbalanced target domain dataset.		

Diagnostic results

We analyze the learned domain-asymmetry factors on the given partial transfer learning tasks T1, T2, and T3. The learned domain-asymmetry factors for source domain samples are plotted in Figure 2. We also presents the number of the source and target domain samples on every health states, and the confusion matrix of diagnostic results.

As shown in Figure 2(a), the target domain samples from the CS state belong to the outlier health states. Thus, the source domain samples from CS get the lowest domain-asymmetry factors. The average values of domain-asymmetry factors for source domain samples in the shared health states (N, WS, CP, BFP) follow the number of target domain samples on each health state. For example, there are massive N samples in the target domain, and thus the source domain samples from the normal condition are weighted by the larger asymmetry factors than other states. By weighting source domain samples with the automatically learned domain-asymmetry factors, the proposed DPTL-Net is able to correctly recognize the health states of target domain samples. Similar results can be observed in Figure 2(b) and Figure 2(c). Comparing the learned domain-asymmetry factors respectively on the tasks T1, T2, and T3, we can find the adaptively adjusted values on each health state. The factors for the source domain samples in N are increased when the health states in the target domain are gradually removed because of the changing ratio of the number of target domain samples across the shared health states.

Figure 2. Domain-asymmetry factors and confusion matrix on partial transfer learning tasks: (a) T1, (b) T2, (c) T3.



Conclusions

We proposed an adversarial network-based diagnostic model named DPTL-Net to overcome the domain asymmetry issues in transfer learning-based fault diagnosis of machines. At the core of the DPTL-Net, domain-asymmetry factors are automatically learned by individually training a domain discriminator with Wasserstein loss, and then they are used to weight the PK-MMD-based distribution adaptation module. The factors are able to distinguish transferrable features from outlier health states and block their contributions from influencing distribution adaptation. Based on the theoretical analysis and experiments, the factors for source domain samples can be adaptively adjusted following the change of imbalanced degrees of target domain samples. The PK-MMD-based distribution adaptation module is weighted with such factors to selectively enhance similarity between target domain samples and partial source domain samples from the shared health states. Based on the comparison results, DPTL-Net shows superior performance to other transfer learning-based diagnostic models in the presence of domain asymmetry.

Appendix 4. Applying object detection to security scanned images

Shashank Saurav, Anmol Garg

Objective

The objective in this project is to automate the security scan procedure of baggage items. Current systems are designed to manually check for each scanned item using X-ray scanning equipment. This baggage check is usually required at airports or other transportation hubs such as bus or train stations where security of passengers and property is of high importance. The key idea here is to eliminate this manual check and replace it with an automated algorithm through Machine Learning that would detect any banned/restricted items on the fly. This will help save costs directed towards the manual checking and minimize any human mistakes.

Introduction

One of the first Object Detection methods on images used sliding window (J. Lee, 2017) which involved taking multiple crops of the input image and then each crop is fed to a deep Convolutional Neural Network (CNN) to make a classification decision. The crop dimensions, shape would be an issue as the object may appear at any location with any shape and size. This limitation was alleviated by applying several crop sizes and dimensions and running them all through the CNN but this posed a time and computation cost issue.

Another approach implemented is Regional – Convolutional Neural Network (R-CNN), wherein region proposal networks to capture blobby (feature-rich) regions and then spits out a finite number of boxes where an object can be potentially present. These limited region proposals are then fed to a CNN for object classification. (Malik, 2013) However, this still needed that finite number of boxes to be fed to CNN for processing rendering redundant computations. Variants of R-CNN such as Fast R-CNN & Faster R-CNN (Sun, 2015) (Farhadi, 2018)have come up since then. Now current state-of-the-art strategies such as You Only Look Once (Yolo) have entered the domain and dominated the Object Detection section of Computer Vision with incremental improvements in different versions. Instead of doing multiple forward propagations of proposed regions on deep CNN, Yolo makes grid cells of the image and then makes proposed predictions on each grid cell. This way, only a single forward propagation of the image through a deep Neural Network is needed which saves computation cost and time. Thus, the name, You Only Look Once.

For now, we plan to implement Yolov3 (3^{rd} incremental version of Yolo) (Farhadi, 2018)on the baggage scanning images to see if the results match the requirements.

Dataset

We were given a dataset with images of banned items. The current subset has the following classes and count:

Class Description	Class ID	Original Count of Images
Handgun	0	550 + 500 (dataset 3)
Kitchen Knife	1	50 + 500 (dataset 3)
Cutter Knife	2	50
Hard Disk	3	50
USB	4	50
Hard Disk – 2	5	50
Kitchen Knife – 2	6	50
Shuriken	7	50
Battery	8	50
Battery – 2	9	50
Phone	10	50

Against every image, there is a true label and boundary box attributes in the form of a text file. The true label denotes the ID of that class and the attributes denote the dimensions of boundary box (which are the dimensions at which the object is present). The dimensions are normalized to the size of the image and the centre coordinate is calculated from the top-left corner of the image. The first two dimensions denote the centre (x & y coordinate) of the boundary box. The other two attributes are the height & width of the bounding box.

Data augmentation

As the image count for some classes is 50 which is a bit on the low end, we applied data augmentation techniques to increase the image count. This would further avoid overfitting. Also, augmentation techniques are limited as the dimensions of the boundary box of the image should be in line with the approach of augmentation.

1. Image Flipping:

The images are flipped each horizontally, vertically, and both horizontally and vertically. The centre coordinates are calculated according to the type of flipping. This technique provides 3 times the number of images. Thus, now we have 200 count of images for 50 count of original images.

2. Image Rotation:

Each of the 200 count of flipped images is rotated clockwise by 90, 180 & 270 degrees. The centre coordinate as well as height and width are adjusted according to the degree of rotation. This again gives us 3 times the number of images. Thus, we get 800 images for 200 flipped images.

The final count of images is now observed below:

Class Description	Class ID	Original Count of Images	Original & Augmented Count of Images
Handgun	0	550 + 500 (dataset 3)	8800 + 500(dataset 3)
Kitchen Knife	1	50 + 500 (dataset 3)	800 + 500 (dataset 3)
Cutter Knife	2	50	800
Hard Disk	3	50	800
USB	4	50	800
Hard Disk – 2	5	50	800
Kitchen Knife – 2	6	50	800
Shuriken	7	50	800
Battery	8	50	800
Battery – 2	9	50	800
Phone	10	50	800

Yolov3 architecture

Applying Yolo, it is known that it uses features learned by a deep CNN to detect any object. It uses only convolutional layers thus making it a fully connected network (FCN). The architecture has 75 convolutional layers apart from skip connection & upsampling layers. Pooling is not implemented and a stride of 2 is applied to downsample the feature maps.

The features learned from the CNN are passed onto a classifier/regressor that will make the detection and a final prediction. Each prediction will have an 8x1 dimension with information on image ID boundary box location & dimensions, objectness score, predicted class and its probability.

$$Final\ Prediction = [img\_id \quad x1 \quad y1 \quad x2 \quad y2 \quad P0 \quad id \quad P_{id}]$$

Img_id : the id of the image where the prediction is made.
X1, y1 : top left coordinate of the boundary box measured from top-left corner of image
X2, y2 : bottom right coordinate of the boundary box measured from top-left corner of image
P0 : Probability of detecting an object (Object Probability)
Id : Id of the class detected
P_id : Confidence level of the class detected

The prediction methodology is described below. Yolov3 makes a prediction on three different scales i.e. feature maps of three different sizes. For an input of 416x416 image, the strides of 32, 16 & 8 are implemented to have detections on feature maps of size 13x13, 26x26 & 52x52 respectively.

The CNN will downsample the input image until a yolo layer i.e. a detection layer is reached where it will make detections using a stride of 32 (13x13 feature map). The layers are then upsampled by a factor of 2 and concatenated with feature maps of a previous layer with the same size. Now, a detection is made with stride of 16 (26x26 feature map) and the upsampling procedure is implemented again to make detection with a stride of 8 (52x52 feature map)

The detection in yolo layer is made using anchors. The purpose of having anchors is to make multiple detections on a single image that may have overlapping objects at or around the same location. Typically, 3 anchors are used so that we do not expect more than 3 objects overlapping in any image.

The 3 anchor boxes will thus make three boundary box predictions for each grid in the feature map. Hence, for a feature map of size 13x13, the yolo layer will generate 13x13x3 boundary boxes (bbox) i.e. 507 bbox. Therefore, for all three scales of feature maps, we get (13x13 + 26x26 + 52x52) x 3 = 10647 bbox.

Corresponding to each bbox is an output vector of size (5+C) where C represents the number of classes. In our case, we have 11 classes and so C will be 11. The first two values (tx & ty) in the vector suggest the centre of the bbox measured from the top-left corner of the grid cell. Next two values suggest the height and width (th & tw) of the bbox. Then, we have the objectness score which makes up the first five values of the vector. The next C values represent the score that the detected object belongs to each of the C classes.

To get the actual bbox dimensions pertaining to the image, the above dimensional values are scaled & normalized.

$$b_x = \sigma(t_x) + c_x$$
$$b_y = \sigma(t_y) + c_y$$
$$b_w = p_w * e^{t_w}$$
$$b_h = p_h * e^{t_h}$$

where c_x & c_y are the coordinates of the top-left corner of the grid cell from the top-left corner of the image. $\sigma(x)$ is the sigmoid operation performed to normalize the coordinates from the top-left corner of the grid cell. p_w & p_h are the predefined width & height of the anchor box pertaining to that scale. The width & height are thus log-transformed to get width and height of the bbox relative to the image.

Using the above four parameters, the top-left and bottom-right coordinates of the bbox are calculated which is described in the final prediction matrix above. The sigmoid operation is applied on objectness score and class scores to get probability of having an object and confidence level of each class.

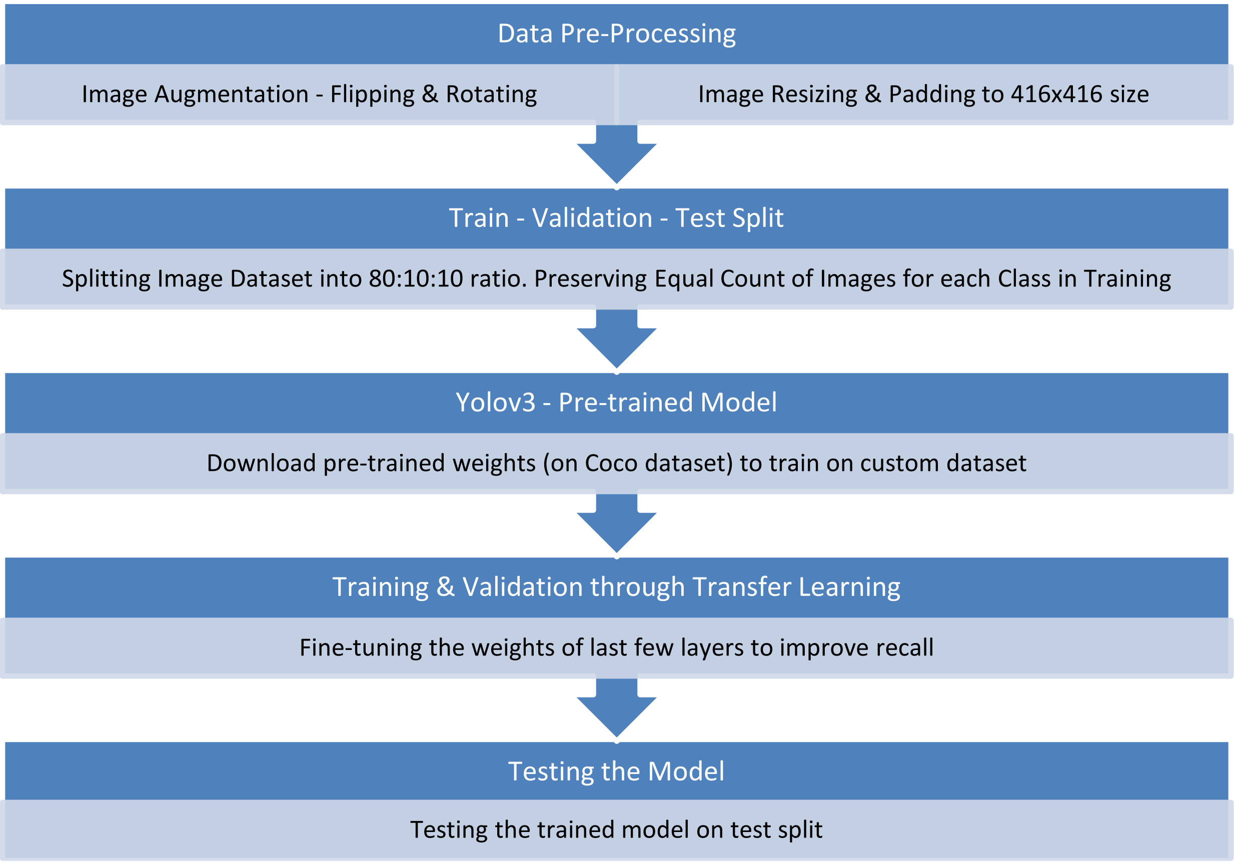
Now, we have an output matrix of size (B x 10647 x (5+c)) where B represents the batch size. It is now wished to have a single bbox for each object in the image i.e. the number of detections must be reduced from 10647 to 1. This is done using Object Confidence Filtration and Non-maximum suppression techniques.

For every object in an image, we now have 1x(5+C) i.e. 1x16 vector which will be further reduced to 1x8. The maximum confidence level out of the C classes & its id is appended to the first 5 attributes which are the bbox dimensions & objectness probability. Then, the image-id that is being worked on is appended to get the final prediction as shown above.

This concludes the working architecture of Yolov3. Let us see the implementation & methodology in the dataset.

Proposed methodology

The following methodology is proposed for the custom dataset.



Data Pre-Processing

Image Augmentation - Flipping & Rotating

Image Resizing & Padding to 416x416 size

Train - Validation - Test Split

Splitting Image Dataset into 80:10:10 ratio. Preserving Equal Count of Images for each Class in Training

Yolov3 - Pre-trained Model

Download pre-trained weights (on Coco dataset) to train on custom dataset

Training & Validation through Transfer Learning

Fine-tuning the weights of last few layers to improve recall

Testing the Model

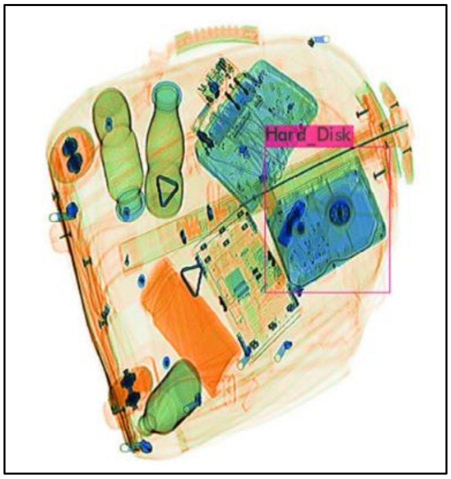
Testing the trained model on test split

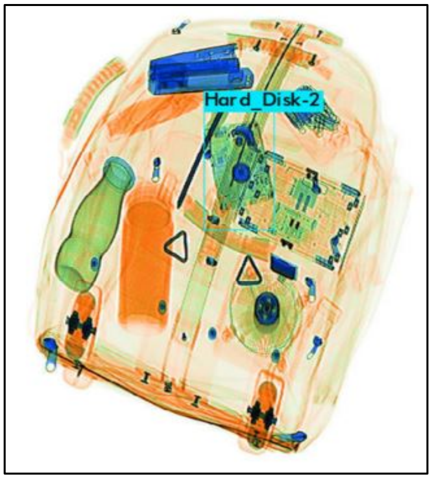
The testing will be based on the Recall metric denoting the percentage of items detected out of how many should have been detected. This metric is chosen as it is not advisable for a banned item to not be detected.

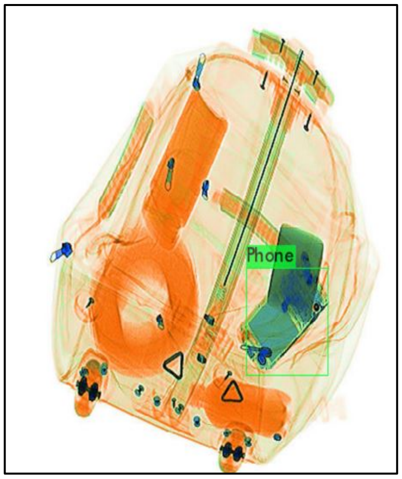
Current progress

To check Yolov3 accuracy, the training dataset (700 images of each class) was fed to a pre-built Yolov3 model with the option to train on a customized dataset. The results were exceedingly good, with high overall accuracy and recall values (average precision 95%; average recall 97%). Results for test set (100 images of each class (except Handgun) + 8100 images of Handgun) are shown below.

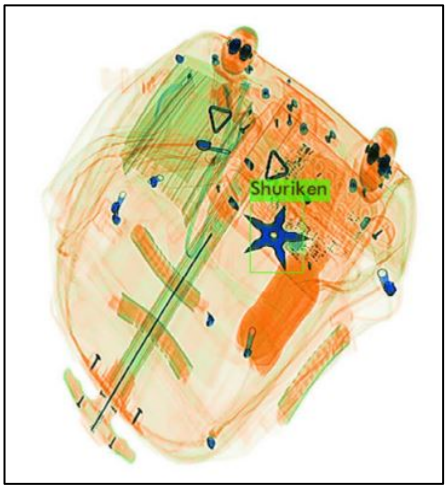
Class	**Precision**
Handgun	98.17
Kitchen Knife	92.78
Cutter Knife	92.64
Hard Disk	97.29
USB	71.57
Hard Disk 2	99.97
Kitchen Knife 2	97.23
Shuriken	98.10
Battery	99.96
Battery 2	99.57
Phone	88.45

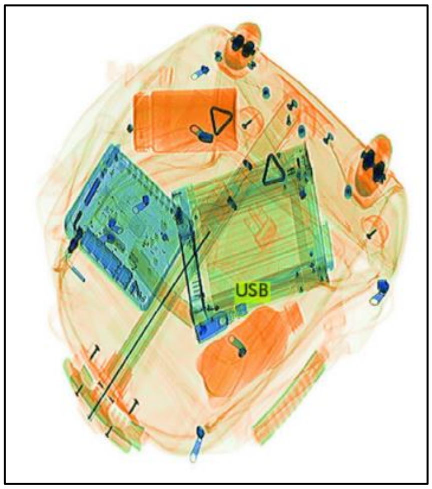

Hard_Disk


Hard_Disk-2


Phone

:


Shuriken


USB

Since the pre-build model was a bit complex with limited flexibility in modification of code, we decided to reproduce the Yolov3 architecture model that could give out the needed metrics (such as recall score for each class) as well as have the flexibility of deployment as per user requirements. Until now, we have reproduced a working Yolov3 model using pre-trained weights that can predict all classes in the Coco dataset as well as give out recall for a particular class.

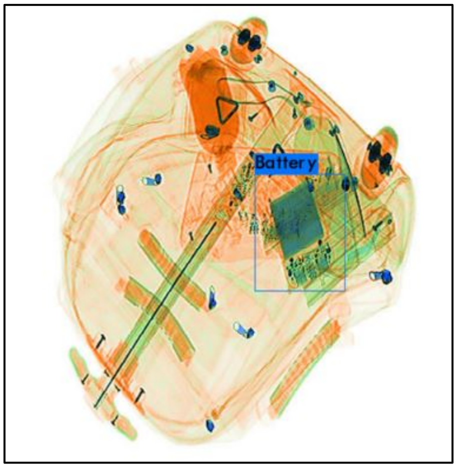
The following are our findings on test images of 5 classes of the Coco dataset:

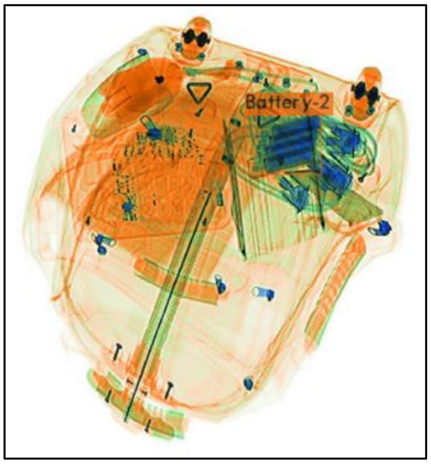
Recall Score for class cow = 0.9
Recall Score for class elephant = 0.93
Recall Score for class aeroplane = 0.934
Recall Score for class bird = 1.0
Recall Score for class bear = 0.976

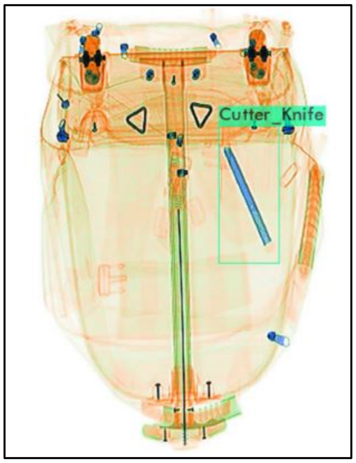
Way forward

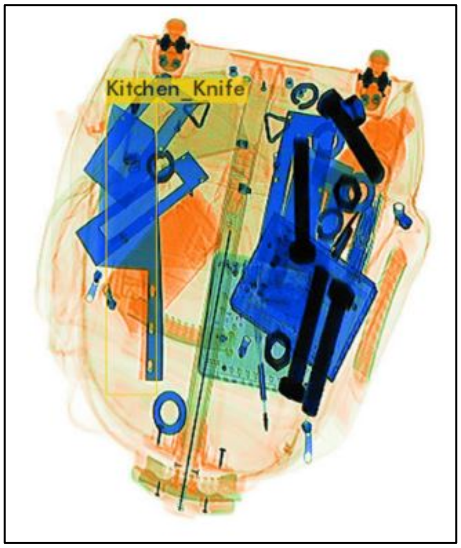
Now, we plan to train the built model with the scanned images of baggage dataset and fine tune it to get maximum recall scores.

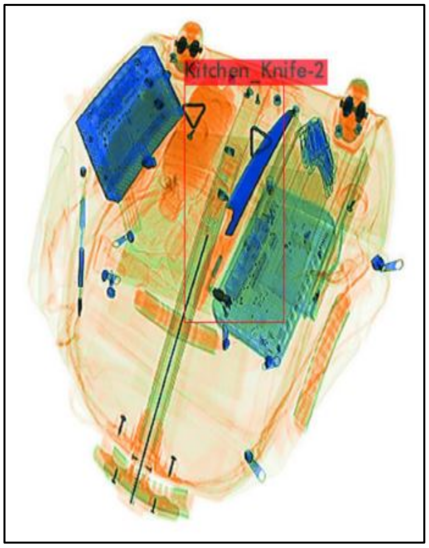
For deployment in real-world scenario, we propose a live feed of the scanning machine being fed to the model that would detect all banned items as the baggage moves on the conveyor belt.

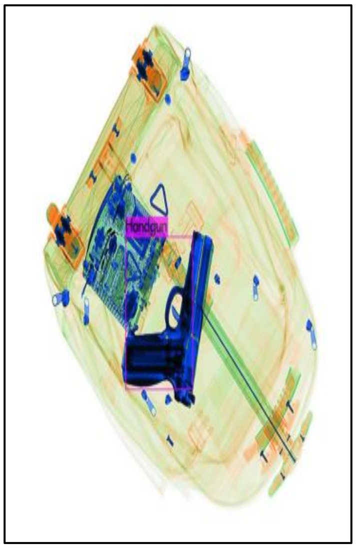












The current subset of dataset contains images with only one banned item per image; therefore to tackle the case of overlapping and/or multiple banned items in baggage, we advise training the model on a dataset with these cases.

References

1. Farhadi, J. R. (2018). YOLOv3: An Incremental Improvement.

2. J. Lee, J. B. (2017). Object detection with sliding window in images including multiple similar objects.
3. Malik, R. G. (2013). Rich feature hierarchies for accurate object detection and semantic segmentation.
4. Ross B. Girshick, J. D. (2013). Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*.
5. Sun, S. R. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.

Appendix 5. Object detection security scan

Syed Hamdan Mustafa, Aakash Iyer

Problem statement

Security concerns have made air travel a pretty invasive affair, but the security process is actually designed to protect the passengers. It is critical to the safety of all who fly that hold luggage, hand luggage and each passenger's physical person are all checked by scanners to ensure compliance with restrictions. Hence, X-ray scanning and analysis of baggage is an essential aspect of airport security. Most carry-on baggage is scanned manually which increases the security risk of human error and exacerbates the already packed flight schedules. Inclusion of automation is one of the ways to optimize the current scenario. The project aims to utilize deep learning methodologies to automate baggage scanning and to develop features to improve the security condition. This approach is focussed on object classification rather than object detection.

Literature review

a) Very Deep Convolutional Networks for Large-scale Image Recognition (Karen Simonyan & Andrew Zisserman, 2015)

   The paper discusses the effect of convolutional network depth on its accuracy in the large-scale image recognition setting. Here, the authors discuss the evaluation of networks of increasing depth using an architecture consisting of very small convolutional filters showed significant improvement on prior-art configurations achieved by pushing the depth to 16-19 weight layers.

b) Using Deep Convolutional Neural Network Architectures for Object Classification and Detection Within X-Ray Baggage Security Imagery (Samet Akcay, Mikolaj E. Kundegorski, Chris G. Willcocks, and Toby P. Breckon, 2018)

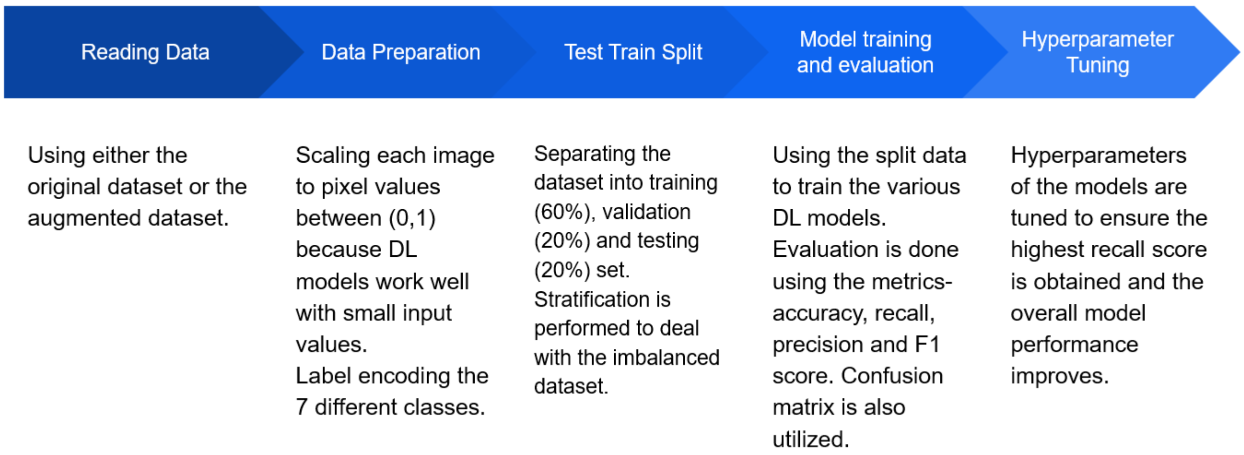
   The authors perform a comparative analysis of Deep Neural Network Architectures and techniques to show that object localization strategies cope well with cluttered X-ray security imagery to overcome redundancies faced by classification techniques. The authors consider various CNN approaches including traditional CNN, transfer learning, CNN+SVM, and CNN driven detection paradigms such as SW-CNN, F-RCNNs, R-CNN, and YOLOv2.

Other valuable references are noted in the final section of the report.

Proposed methodology

The methodology followed for the project is discussed using the different aspects. The overall approach is summarized below:

Figure 1. Project methodology pipeline



Using either the original dataset or the augmented dataset.

Scaling each image to pixel values between (0,1) because DL models work well with small input values.
Label encoding the 7 different classes.

Separating the dataset into training (60%), validation (20%) and testing (20%) set. Stratification is performed to deal with the imbalanced dataset.

Using the split data to train the various DL models. Evaluation is done using the metrics- accuracy, recall, precision and F1 score. Confusion matrix is also utilized.

Hyperparameters of the models are tuned to ensure the highest recall score is obtained and the overall model performance improves.

Dataset

To implement automation in object detection in X-ray scans of baggage, it is proposed to utilize an image classification approach using neural networks for identifying malicious items. In the current work, a dataset containing **2,050 images** is used containing different items for identification. The breakdown of the dataset is as follows:

Table 10. Dataset breakdown

Description	Count
Handgun	500
Airgun	50
Other firearms	500
Kitchen Knife	100
Cutter Knife	50
Other Knife	500
Battery	100
Hard Disk	100
Phone	50
Shuriken	50

To ease the classification, all gun images were combined as Handgun class (1,050 images) and all knife images were also combined as Knife class (650 images), giving a total of 7 classes for the multi-class classification problem. Stratification was used to deal with the imbalanced dataset; this ensures that the ratio of each class in the training, validation and test set is the same for non-biased classification.

Role of augmentation

Due to the low number of images, using a neural network may lead to poor generalization and underfitting. In addition, the robustness of the model could not be validated with low data. The idea behind image augmentation is that altered versions of existing images are generated using the existing image set via image transformation (crop, resize, rotation, etc.) that aid in alleviating the issue of small dataset by producing extra images. It is important to note that the number of images produced is a hyperparameter and can be tuned for the best results.

Transfer learning

In transfer learning, the knowledge of an already trained machine learning model is applied to a different but related problem. The general idea is to use the knowledge a model has learned from a task with a lot of available labeled training data in a new task that does not have much data. Inductive transfer techniques utilize the inductive biases of the source task to assist the target task. There are two types of models utilized:

1. Feature extraction: The key idea here is to just leverage the pre-trained model's weighted layers to extract features but not to update the weights of the model's layers during training with new data for the new task.

2. Fine tuning: This is a more involved technique, where the final layer is not replaced, and selectively previous layers are also retrained.

Importance of recall

Recall is important when ensuring the capture of positive cases as the cost of missing a positive is more problematic than the cost of including a negative. In this project, identification of hazardous item is of prime importance for ensuring security and safety.

Comparative benchmark and early warning system

The results of the models are compared with professional models widely used in industrial solutions. To that end, Microsoft Azure's Custom Vision Model is chosen as a benchmark.

The second objective is to develop an early warning system to the user indicating the identification of malicious objects in baggage and thus simulating an alarm system embedded with the solution. This would ideally be done using a dataset containing non-malicious items (i.e. safe baggage); the model could then be put to test to understand if it can directly detect malicious items without the need for further classification into the aforementioned 7 classes.

Model summary

There are two types of models used in the project. The professional model serves as a benchmark.

Public models

Basic CNN: CNN model with three convolutional layers, coupled with max pooling for auto-extraction of features from the images and down sampling the output convolution feature maps.

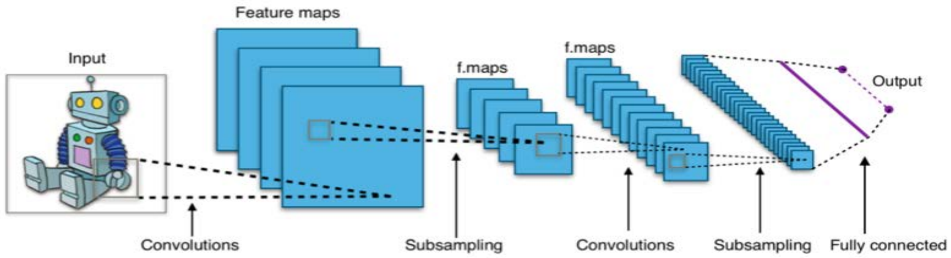


Figure 2: Basic CNN architecture

CNN with regularization: Here, the base CNN model is improved by adding in one more convolution layer, another dense hidden layer. Besides this, dropout of 0.3 is added after each hidden dense layer to enable regularization. Basically, dropout is a powerful method of regularizing in deep neural nets. It can be applied separately to both the input layers and the hidden layers. Dropout randomly masks the outputs of a fraction of units from a layer by setting their output to zero (in this case, it is 30% of the units in the dense layers).

CNN with image augmentation: Image augmentation is the process of taking in existing images from the training dataset and applying image transformation operations to them, such as rotation, shearing, translation, zooming, and so on, to produce new, altered versions of existing images. In this case, there are different approaches utilized for pursuing data augmentation:

i. **Providing altered image in each epoch**

- Here, in each epoch, the number of images entering the model were kept the same, but each epoch had the altered images for training the model. Keras' ImageDataGenerator is used for this approach.
- The drawbacks of this approach are in increased processing time due to dependency on increasing the number of epochs for efficient image augmentation and existence of imbalance in the dataset.

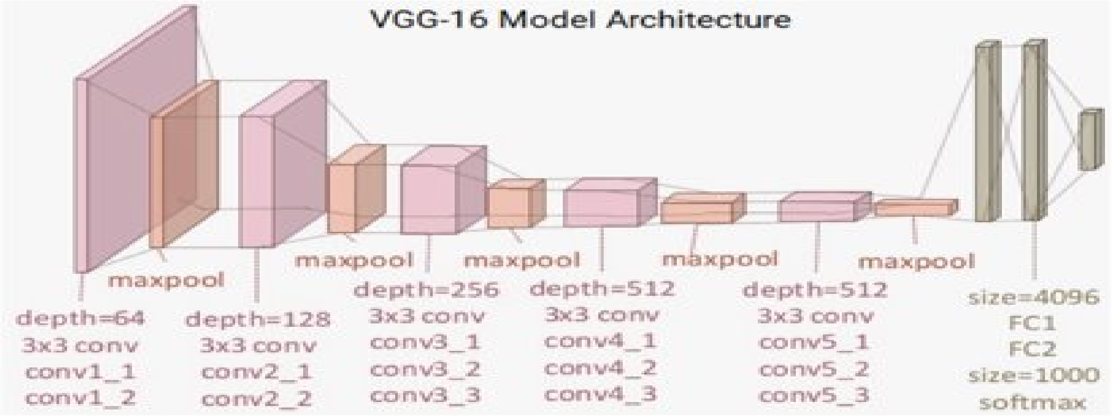
ii. **Increasing image dataset by creating augmented images**

- Here, the images are first augmented and then the new larger dataset is fed to the model. For this approach, skimage from sklearn is used.
- The benefit of this approach is in the increased control over what data are fed. As a result, images can be augmented per class and oversampling can be done for overcoming imbalanced dataset.
- The drawback is in determining the right number of augmented images as high number of images can lead to duplicate images.

The second approach produced better results for this project.

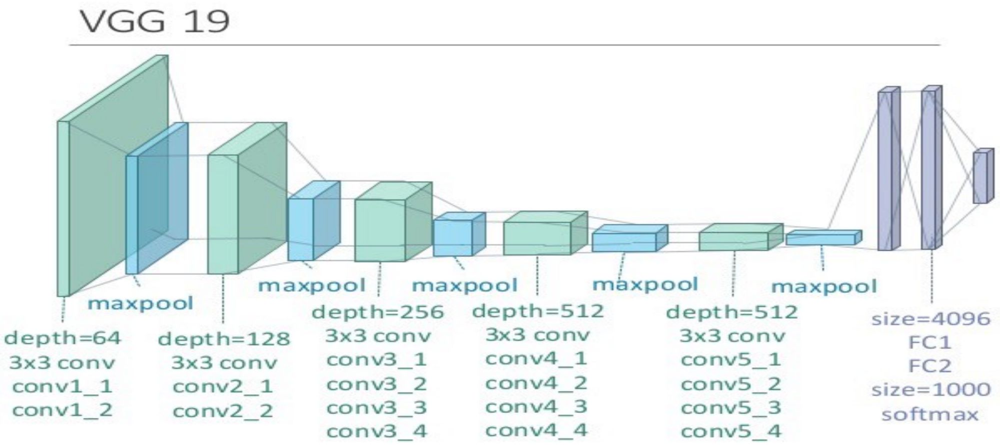
CNN + transfer learning with VGG-16: VGG-16 model is a 16-layer network built on the ImageNet database, which is built for the purpose of image recognition and classification. Key idea here in Transfer Learning is to leverage the pre-trained model's weighted layers to extract features but not to update the weights of the model's layers during training with new data for the new task. Leveraging the convolution blocks of the VGG-16 model and then flattening the final output so that it can feed into the own dense layers for the classifier.

Figure 3. VGG-16 architecture



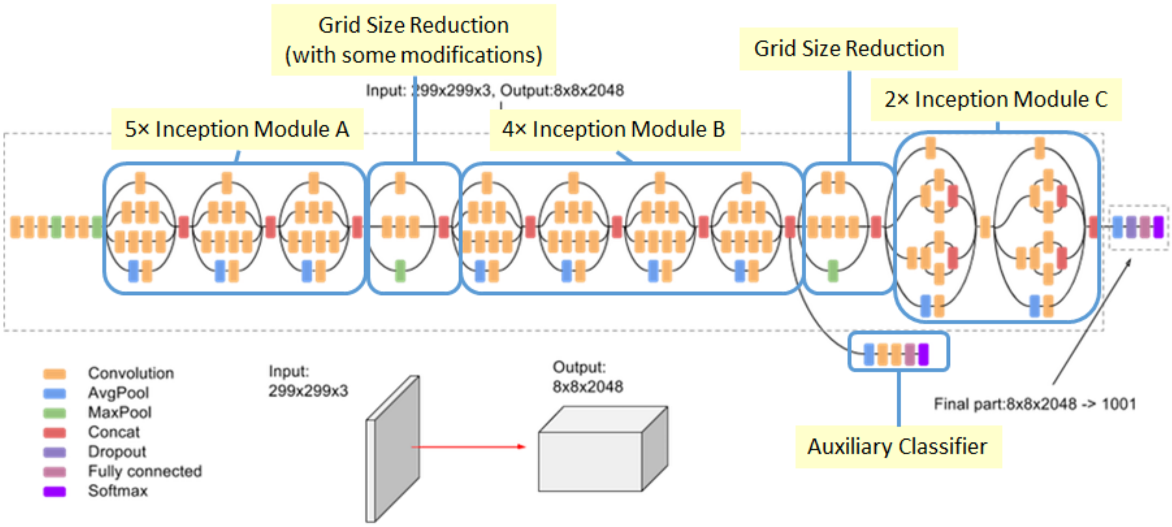
CNN + transfer learning with VGG-19: Just like VGG-16, VGG-19 is a trained Convolutional Neural Network, from Visual Geometry Group, Department of Engineering Science, University of Oxford. The number 19 stands for the number of layers with trainable weights. 16 Convolutional layers and 3 Fully Connected layer. There are various variants tried here by training different number of layers, the best results for this model is presented here.

Figure 4. VGG-19 architecture



CNN + transfer learning with inception V3: The pre-trained Inception-v3 model achieves state-of-the-art accuracy for recognizing general objects with 1000 classes, like "Zebra", "Dalmatian", and "Dishwasher". The model extracts general features from input images in the first part and classifies them based on those features in the second part.

Figure 5. Google Inception v3 architecture



Professional Model

Microsoft Azure Custom Vision: Microsoft Azure's Custom Vision is a customizable embedded computed vision technology for specific domains. As one of the leading Data Science solution, Custom Vision was chosen as a benchmark in order to have an anchor to perform accurate comparitive analysis to industrial private model.

Model results and discussion

This section discusses the results for the models. Models were evaluated using the following metrics:

i. Accuracy
ii. Precision (macro-average)[1]
iii. Recall (macro-average)
iv. F1 Score (macro-average)

They are all important measures; however, recall is understood to be the vital one since a major shortcoming of this model would be the inability to classify the malicious items. It could lead to a catastrophic incident.

It is important to note that there are different variants of the models. This report comprises only the best variant of each model. The different variants are created by training different layers of the DL model, for example. The results are summarized in the tables below.

The second part of the project focussed on increasing the recall by modifying the architecture of the DL model and the augmentation code.

[1] Macro-average will compute the metric independently for each class and then take the average (hence treating all classes equally).

The results are presented in two ways. The first are the models trained with the raw dataset (no augmentation). There is a total of 2,050 images as mentioned earlier.

Table 11. Model results without augmentation

Sr. No.	Approach	Test Accuracy	Precision	Recall	F1 Score
1	Basic CNN	0.91	0.81	0.74	0.76
2	CNN with Regularization	0.94	0.88	0.81	0.82
3	VGG-16 (as a feature extractor)	0.94	0.90	0.89	0.89
4	VGG-16 with Fine-Tuning	0.97	0.93	0.94	0.93
5	VGG-19	0.95	0.90	0.91	0.90
6	Google Inception v3	0.98	0.94	0.94	0.93

Google Inception v3 seems to be the best model with the non-augmented dataset. It takes ~1 hour to train and evaluate these models.

The second part includes training the models with an augmented dataset. The images are resized, rotated (clockwise and anti-clockwise), flipped (horizontally and vertically) and noise is added. There is a total of 10,500 images (1,500 images per class).

Table 12. Model results with augmentation (no blurred images)

Sr. No.	Approach	Test Accuracy	Precision	Recall	F1 Score
1	Basic CNN with Augmented data	0.92	0.92	0.92	0.92
2	CNN with regularization and augmented data	0.95	0.95	0.95	0.95
3	VGG-16 (as a feature extractor) with augmented data	0.95	0.96	0.95	0.95
Sr. No.	**Approach**	**Test Accuracy**	**Precision**	**Recall**	**F1 Score**
4	VGG-16 with Fine-Tuning with Augmented data	0.99	0.99	0.99	0.99
5	VGG-19 with augmented data	0.99	0.99	0.99	0.99
6	Google Inception v3 with augmented data	0.99	0.99	0.99	0.99

There are a lot of candidates for the best models: VGG-16, VGG-19, and Google Inception v3. However, it takes ~5 hours to train the VGG-16 and Google Inception v3 models whereas it takes ~8 hours to train the VGG-19 model.

The improved results with the augmented dataset were noted primarily after removing the blurred images. This could be a possible shortcoming of the model which could require the baggage to be stopped during the scan. This could lead to an issue since it would take a considerably larger amount of time to scan the bags. The results with adding blurred images to the dataset are as follows (for reference).

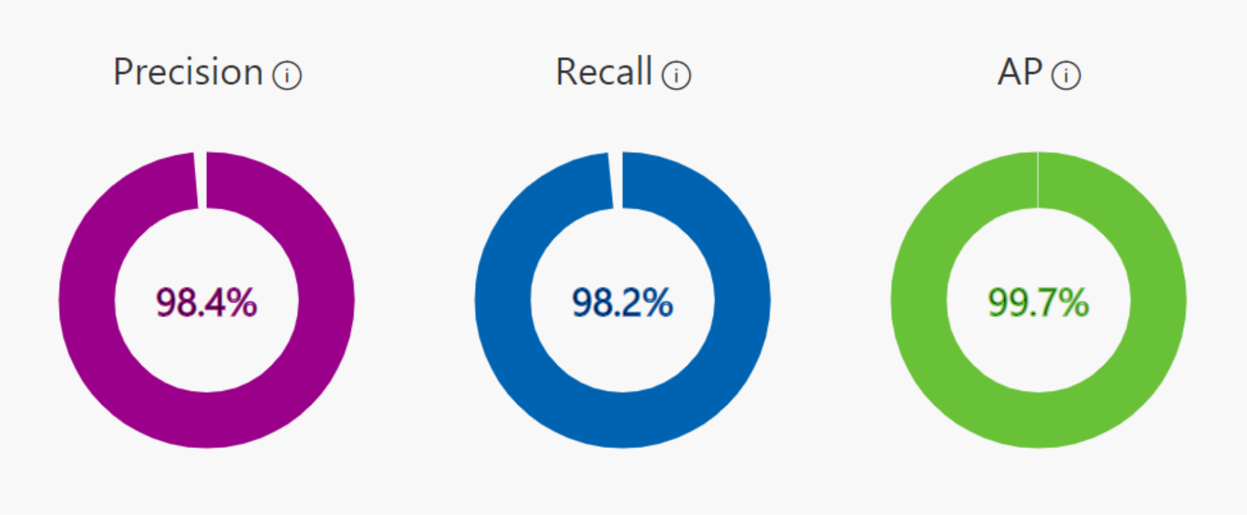
Table 13. Model results with augmentation (blurred images included)

Sr. No.	Approach	Test Accuracy	Precision	Recall	F1 Score
1	CNN with Augmented data	0.88	0.93	0.87	0.89
2	CNN with regularization on augmented data	0.90	0.95	0.89	0.91
3	VGG-16 (as a feature extractor) with augmented data	0.89	0.94	0.88	0.90
4	VGG-16 with Fine-Tuning with Augmented data	0.91	0.96	0.90	0.92
5	VGG-19 with augmented data	0.91	0.96	0.89	0.92
6	Google Inception v3 with augmented data	0.91	0.96	0.90	0.92

The evaluated metrics decrease when the blurred images are included.

As mentioned in Sections 3 and 4, the model results are also compared to a professional model, i.e., Microsoft Azure's Custom Vision. This is a proprietary software and hence there is no information available regarding the models used. The comparison is done with our best model (which is using the augmented dataset). Custom Vision gives the following results when it is done using the augmented dataset.

Figure 6. Custom vision results with augmented dataset (including blurred images)



It can be noted that our models perform relatively better when the blurred images are removed. However, there is room for improvement when it comes to the blurred images.

Future work

There is a considerable amount of work to be done in this project since there are other teams who are working on it using different approaches. Another team is using an object detection approach and using models such as YOLO and R-CNN. The different approaches/teams are supposed to collaborate and produce a consistent format of model results for comparison and records.

As for the project itself, the following steps are ensuring that the models are more robust especially when it comes to blurred images. It was found to be a particular issue during the model testing. The Custom Vision (benchmark) results indicate that better results are possible when the size of dataset is large and contains blurred images (i.e. using augmentation). There are various other transfer learning models that could be utilized such as AlexNet and ResNet. There is also an approach utilizing Support Vector Machine (SVM) classifier, combined with CNN. The aim for the best model is to have the highest possible recall without compromising significantly on the other metrics and have a relatively lower training/evaluation runtime.

Another particular focus of the project going forward is introducing an alarm system that would alert the user if the particular object being scanned is malicious. Since there are no images available for non-malicious baggage; this will be done by considering some of the classes as malicious whereas the other classes as non-malicious. This is not an ideal solution, but it will be used as a way to develop the alarm system.

References

Models: CNN, CNN+Reg, CNN + ImageDataGenerator, VGG-16, Google Inception V3
Primary Reference: https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a

Model: VGG-19
Primary Reference: https://towardsdatascience.com/transfer-learning-in-tensorflow-9e4f7eae3bb4
https://medium.com/@14prakash/transfer-learning-using-keras-d804b2e04ef8

Model: ResNet50
Primary Reference: https://towardsdatascience.com/deep-learning-using-transfer-learning-python-code-for-resnet50-8acdfb3a2d38

Appendix 6. Generic real option value analysis under swing option assumption

Yushi Wang, Ruiqi Yang, Yige Zhang

Problem statement

[1] Real options are decisions a company's manager can make to expand, curtail, change, delay, or terminate projects based on the economic circumstance, technology changes, or market conditions. It is referred to as "real" because it typically references projects involving a physical asset or tangible asset such as machinery and buildings instead of a financial instrument. Conventional ways of decision making such as return on investment (ROI), net present value (NPV) and cost-benefit analysis. Such methods ignore the outcome uncertainty, the choice of investment timing and irreversibility of resource commitment. By applying real options value analysis (ROV), managers can estimate the opportunity cost of continuing or abandoning a project or other strategies and make decisions accordingly, which gives companies flexibility on time and capital and potential benefits when they make future decisions.

[2] A swing option is a type of contract used by investors mostly in energy markets that lets the option holder buy a predetermined quantity of energy at a predetermined price while retaining a certain degree of flexibility in the amount purchased and the price paid. In this project, we references the [9] structure of the swing option contract and formulate our project as a procurement problem under swing option framework, to be more specific, we will consider: (1) Local Constraints: the least and most number of tangible asset an option holder can purchase per unit time period; (2) Global Constraints: (i) the least number of tangible assets should purchase, (ii) the most number of tangible asset could purchase during total time horizon, and (iii) budget; (3) Penalty: if the global constraints are not fulfilled, a terminal penalty will be added up.

Hypothesis

Since there are no data available for this project, the work depends on simulations. We assume that the asset's value follows the Merton's Jump Diffusion process (MJD), with constant depreciation rate and negative log jump size. The constant depreciation rate indicates that the general trend of asset's value is decreasing as time goes on. The jump size captures the situation when there are crucial events happen, such as technology change or war, which will change the asset's value significantly.

Problem setup

Merton's Jump Diffusion Model

Following Zhang, J., & Zhao, H. (2006)[3], we assume the object that one is interested to purchase follows a jump diffusion process and we use the same notation except write P_t as the price.

$$dP_t = (\mu - D - \lambda E(e^x - 1))P_t dt + \sigma P_t dB_t + (e^x - 1)P_t dN_t$$

where μ and σ represent the drift and volatility of the process. λ is defined as a constant intensity for the jump process. Besides that, random variable x is the jump size and it follows a normal distribution. B_t and N_t are Brownie motion and Poisson process respectively.

Amin (1993) [4] introduced a methodology which is analogous to CRR model, to discretize the MJD process with guaranteed week convergence to it's continuous time formulation that suggested by Merton (1976). The author firstly defined the drift of the logarithm of the asset's price as $\alpha = (\mu - D - 0.5\sigma^2 - \lambda E(e^x - 1))$. Then for every fixed positive integer n, partition the trading interval $[0, \tau]$ into n subintervals of length $h_n = \tau/n$. Furthermore, the author assume that $X^c(t) = ln[P^c(t)/P^c(0)]$.

Then the transition probability for $X_n(t + h_n) - X_n(t) = \alpha h_n + l\sigma\sqrt{h_n}\ \forall\ l \in Z$ will be:

$$\begin{aligned} Prob[X_n(t + h_n) - X_n(t) = \alpha h_n + \sigma\sqrt{h_n}] &= q_n(1 - \lambda h_n) \\ Prob[X_n(t + h_n) - X_n(t) = \alpha h_n - \sigma\sqrt{h_n}] &= (1 - q_n)(1 - \lambda h_n) \\ Prob[X_n(t + h_n) - X_n(t) = \alpha h_n + l\sigma\sqrt{h_n};\ l\,] &= \lambda h_n\, dN_n(l) \end{aligned}$$

where

$dN_n(l) = N(\alpha h_n + (l + 1/2)\sigma\sqrt{h_n}) - N(\alpha h_n + (l - 1/2)\sigma\sqrt{h_n}), N(\cdot)$ is the normal CDF and $dN(\pm 1) = 0$

$\hat{q}_n = 1/2 + o(h_n)$ and $q_n = \hat{q}_n + ((\mu - r)/\sigma)\sqrt{h_n}$

The author states that from a practical perspective, we can assume that $\hat{q}_n = 1/2$ if n is sufficiently large.

Major formulation

In this section, we will consider building a model by using tools we have. Our problem is to value a real option with multiple exercise time. Rather than to formulate the option model itself, we choose to build a portfolio that imitate the profit during the life of the option, and by Law of one price in a complete market, the total sum of rewards for each transaction will be our option value. The setting of our portfolio is similar to a problem of optimal execution in risky asset trading; if the underlying asset is the stock, then it is the one of the classical problems of Quantitative Finance. According to Igor Halperin and Feldsheyn [7], the goal of optimal execution problem for stock is to design trading strategy, for partitioning a large trade order to buy or sell a large block of a stock of some company into smaller chunks, and buy these chunks sequentially so that a potential market impact would be minimized, and at the meantime, the trader would like to maximize his profit from the optimal trading strategy. However, in our problem, the risk asset is a real asset, which is that it has intrinsic value in and of itself does not rely on monetization and exchange in order to provide value for its owner.

Another innate character we consider the real asset would have is that a real asset's will continuous to generate value or utility that depends more than the asset's value itself, this character is like the dividends that a stock will continuous to generate but dividends for stock highly depends on the stock value. We consider this as the return of the real assets on hand, and we will revisit this idea later when we build our model.

One-period reward

Immediate reward

Similar to Igor Halperin and Feldsheyn [7], we adopt the notation and assumption of the portfolio model suggested by Boyd et. al. [8]. In this model, we have only one underlying asset, the total dollar value of assets on hand at beginning of time t is $x_t = A_t P_t$, where A_t is the total number of assets owned and Pt is the price for underlying asset at time t. An investment portfolio includes a yield risk-free interest rater_f and a risk-free bank cash account b_t which also serves as our budget. We ignore the case when selling asset is allowed, which means we don't have a short position in the asset, $A_t \geq 0$. The total portfolio value at the beginning of time t is:

$$\Pi_t = x_t + b_t$$

Trade $u_t = a_t P_t$ at the beginning of the time period t, where $a_t \geq 0$ is the amount of assets purchased, therefore the asset value x_t^+ immediately after trade is deterministic, which is:

$$x_t^+ = x_t + u_t$$

Assuming that all rebalancing of asset positions are financed from the bank account, this implies:

$$b_t^+ = b_t - u_t$$

And we require $b_t^+ \geq 0$ serves as our budget. Furthermore, this serves as our self-finance constraints, since the portfolio value remains instantaneously upon a trade, $\Pi_t^+ = \Pi_t$.

$$\begin{aligned} \Pi_t^+ &= x_t^+ + b_t^+ \\ &= (x_t + u_t) + b_t^+ + b_t - b_t \\ &= \Pi_t + u_t + b_t^+ - b_t \end{aligned}$$

We assume that our underlying assets with dollar amount x_t andx_t^+ will generate one period value r_t and r_t^+ accordingly. The change of portfolio value when action a_t is taken in period t is $\Delta\Pi$ with the form:

$$\begin{aligned} \Delta\Pi &= \left[(1 + r_t^+)x_t^+ + \left(1 + r_f\right)b_t^+\right] - \left[(1 + r_t)x_t + \left(1 + r_f\right)b\right] + r_t x_t \\ &= (r_t^+ - r_t)x_t + \left(r_t^+ - r_f\right)u_t + r_t x_t \\ &= r_t^+ x_t + \left(r_t^+ - r_f\right)u_t + r_t x_t \end{aligned}$$

The plus $r_t x_t$ is because we assume that the assets we have already owned; i.e. $A_t P_t$, will still generate value in current period. And, one thing should be noticed that, if no action has been taken in current period, then $r_t^+ = r_t$, then we substitute this relationship into our model, the change of portfolio value $\Delta\Pi$ will become:

$$\Delta \Pi = r_t x_t$$

Which is consistent with our assumption that the underlying assets we owned will continuously generate value.

According to our methodology, we are pricing the option according to a complete market, and this concludes our first part of reward function in period t.

$$\begin{aligned} R^{(0)}(P_t, A_t, a_t\ ) &= r_t A_t P_t + (r_t - r_f) a_t P_t \\ &= -r_t P_t a_t + (a_t + A_t) P_t r_t \end{aligned}$$

Simple example

To present our idea more clearly, let's consider a simple three period example. In this example, we will purchase a_1 and a_2 assets at the start of period 1 and 2 respectively, and the detailed roadmap in Figure 1.

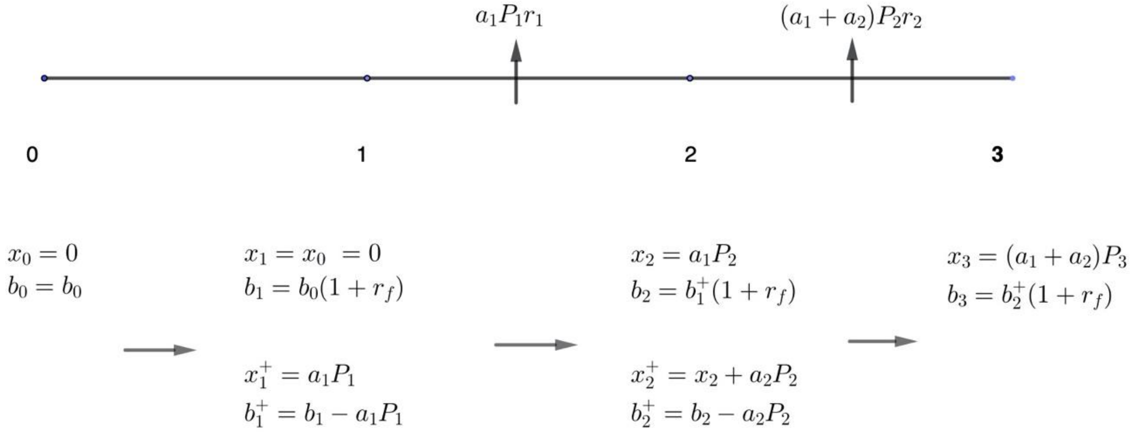


Figure 1: 3 periods example

After purchasing a_1 units, there will be real asset in our portfolio. The real assets and bank account generate different return period, r_t and r_f respectively. During period 1, the profit generated by the real assets will be $a_1P_1(1+r_1) - a_1P_1 = a_1P_1r_1$, and we subtract a_1P_1 is because the real assets at the beginning of time 2 will worth a_1P_2 and be added back. And this idea can be generated through all time periods.

We assume that no action can be taken at the terminal, which yields the portfolio value at terminal are known once we have all information at time three, i.e., the total number of assets we have bought so far and how much per unit of real asset worth at the terminal.

$$\begin{aligned} \Pi_3 &= x_3 + b_3 + a_1P_1r_1(1+r_f) + (a_1+a_2)P_2r_2 \\ &= (a_1+a_2)P_3 + b_0(1+r_f)^3 - a_1P_1(1+r_f)^2 - a_2P_2(1+r_f) \\ &\quad + a_1P_1r_1(1+r_f) + (a_1+a_2)P_2r_2 \end{aligned}$$

After re-grouping, we can get:

$$\begin{aligned}\Pi_3 &= (a_1 + a_2)P_3 + \left(a_1P_1r_1 - a_1P_1r_f\right)\left(1 + r_f\right) + \left((a_1 + a_2)P_2r_2 - a_2P_2r_f\right) \\ &\quad + \left(b_1\left(1 + r_f\right)^2 - a_1P_1\left(1 + r_f\right) - a_2P_2\right) \\ &= (a_1 + a_2)P_3 + R_1^{(0)}\left(1 + r_f\right) + R_2^{(0)} + \left(b_1\left(1 + r_f\right)^2 - a_1P_1\left(1 + r_f\right) - a_2P_2\right)\end{aligned}$$

Which means, if we set terminal value as

$$(a_1 + a_2)P_3 + \left(b_1\left(1 + r_f\right)^2 - a_1P_1\left(1 + r_f\right) - a_2P_2\right)$$

we will replicate the portfolio exactly. However, we are pricing the real option. Mainly, we are pricing the time flexibility of this option.

And to evaluate an option, we decide to use a negative penalty if it exists, otherwise 0, as the terminal value.

Market impact and risk penalty

We assume that every period we will face a random unit dollar value of assets' negative market impact to historical decisions, which yields a cost.

$$R_t{}^M = -(c_1(A_t + a_t) + c_2t + \epsilon_t)(A_t + a_t)P_t$$

Furthermore, we consider use $E(\epsilon_t) = 0$, $var(\epsilon_t) = \sigma_M{}^2$. And $\sigma_M{}^2$ is high enough so that we should consider the risk when making decision. From there, we introduce the last piece of our one period reward, the negative risk reward with risk-aversion λ_r.

$$R_t{}^{risk} = -\lambda_r P_t{}^2 \sigma^2{}_M (A_t + a_t)^2$$

One period rate of return

To evaluate the return rate of owned assets, we consider define the return rate as a function of existing information. For simplicity, we consider r_t as a function of P_t , i.e,$r_t = F(P_t)$. Then, in terms of given information at time t, $r_t$is a fixed number with respect to action a_t, hence there will be no difference before taking an action, denoted as r_t and after$r_t{}^+$. Furthermore, we assume that r_t is a martingale, then when using Itô's lemma and set drift term to 0, we can have an ODE(ordinary differential equation) of function $r_t = F(P_t)$ as below:

$$F_t + (\alpha - \lambda k)P_tF_{P_t} + \frac{1}{2}\sigma^2 P_t{}^2 F_{P_tP_t} = 0$$

As r_t is assumed to be independent to time$t$, the ODE above can be solved as:

$$F(P_t) = \frac{m}{m-1}P_t{}^{\frac{m-1}{m}} \times C \text{ , where} m = \frac{\sigma^2}{\alpha - \lambda k}.$$

Since function Frepresents the rate of return of holding assets, constant$C$ should be sufficiently small because the first part of multiplication is almost a portion of price.

Multi-period convex optimization

The formulation for MDP above can also solved as a risk- and cost-adjusted reward maximization problem reads:

$$
\begin{aligned}
& max_{a_t \forall t} E[\sum_{t=0}^{T-1} \gamma^t R((A_t, P_t, Y_t, t), a_t)] \\
& R_t = R^{(0)}{}_t + R^M{}_t + R^{risk}{}_t \\
& R^{(0)}{}_t (a_t) = -a_t P_t r_f + (A_t + a_t) P_t r_t \\
& R^M{}_t (a_t) = -(c_1(A_t + a_t) + c_2 t + \epsilon_t)(A_t + a_t) P_t \\
& R_t{}^{risk} = -\lambda P_t{}^2 \sigma^2{}_M (A_t + a_t)^2 \\
\text{w.r.t.} \quad & a_{min} \leq a_t \leq a_{max} \; \forall\, t \\
& b_t - a_t P_t \geq 0 \;\; \forall t \\
& A_{min} \leq \sum_{t=0}^{T-1} a_t \leq A_{max} \\
& a_t \in \{0, 1, \ldots, a_{max}\}
\end{aligned}
$$

MDP summary

We summarize our MDP model as follows:

1. State space: $S_t = P \times A_t \times T \times B_t$, where P is the price, $A_t$is the historical decision, T is the current time, B_t is the total budget we have at time t
2. Decision epochs: $[0, T]$
3. Actions $\{0, \ldots, a_{max}\}$
4. One period reward, $R_t = R^{(0)}{}_t + R^M{}_t + R^{risk}{}_t$
5. In terms of model dynamic, we choose to use the market dynamic exactly.

Theoretical results

Without penalty

Under the assumption that we remove all the constraints, including budget constraint and local total number of unit can purchase, and consider the entire problem simply as a real option pricing with the global constraint, and take consideration of market impact and risk penalty, we will have a nice and consistence analytic solution, which are:

$$R(P_{T-1}, A_{T-1}, T-1) = P_1[r_{T-1} - r_f - c_2(T-1)]/2[c_1 P_{T-1} + \lambda_r \sigma^2{}_M P_{T-1}{}^2] - A_{T-1}$$

and

$$R(P_t, A_t, t) = \frac{P_t\left[r_t + r_f(\gamma m - 1) - c_2 t\right]}{2\left[c_1 P_t + \lambda_r \sigma^2{}_M P_t{}^2\right]} - A_t \; \forall t \neq T - 1$$

where $E(P_{t+1} | P_t) = P_t m$

With penalty

Under swing option framework, we assume that there will be few more constraints on trading. For example, there will be a minimum amount of purchase A^* so that a penalty P^* will be added on the unit left to meet A^*. Besides that, there will be limited purchasable number at decision epoch, such as, $0 \leq a_t \leq 5$ indicates that one can only buy at most 5 units.

However, under these constraints there will be no closed form optimal because optimal actions may not lying into the feasible action set at each time epoch. So, the best actions derived by Backward Induction will not hold anymore.

Since a closed form optimal solution is not available, but we can still go further discussion under these constraints. Consider an agent makes a decision at time $T-1$ without considering penalty and the agent gets an optimal action such that the agent does not meet the minimum purchase requirement. Then from result above:

$$a_{T-1}{}^{*} = \frac{r_t - r_f - c_2(T-1)}{2(c_1 P_{T-1} + \lambda \sigma_M{}^2 P_{T-1}{}^2)} - A_{T-1}$$

After the agent realizes that there may be a penalty, the optimal action will be:

$$a_{T-1}{}^{**} = \frac{r_t - r_f - c_2(T-1) + P^*/P_{T-1}}{2(c_1 + \lambda \sigma_M{}^2 P_{T-1} \ )} - A_{T-1}$$

As we can see, $a_{T-1}{}^{**} > a_{T-1}{}^{*}$. But what if $a_{T-1}{}^{**} > A^* - A_{T-1}$? In this case, the agent will meet the minimum purchase requirement and there will be no penalty at all. Hence, $a_{T-1}{}^{**}$will not be the optimal action anymore. So, $a_{T-1}{}^{**}$ is optimal only when $a_{T-1}{}^{**} < A^* - A_{T-1}$. Consider the last situation that $a_{T-1}{}^{**} > A^* - A_{T-1}$. Known that V_{T-1}has a quadratic framework with respect to a_{T-1}, so if the agent reduces the amount of purchase, the value of V_{T-1}will increase. But once it is lower than$A^* - A_{T-1}$, V_{T-1} will be imposed to a penalty, which means keep reducing the amount of purchase will lead to a decrease in V_{T-1}. Hence, if $a_{T-1}{}^{**} > A^* - A_{T-1}$, the best action is $A^* - A_{T-1}$. Moreover, we can see that under the penalty assumption on the minimum amount of purchase, if the agent does not purchase enough amount to meet A^*, then the holding amount will not exceed A^*.

Furthermore, we can find a threshold for P^*so that we can determine the best action by setting $a_{T-1}{}^{**} = A^* - A_{T-1}$, then we will get:

$$P_{threshold}{}^{*} = P_{T-1}[2(c_1 + \lambda \sigma_M{}^2 P_{T-1} \ )A^* - r_t + r_f + c_2(T-1)]$$

To be more specific, once at time$T-1$, the right-hand side is smaller than left hand side, the agent should take action $A^* - A_{T-1}$, otherwise $a_{T-1}{}^{**}$.

Proposed method and current issues

We tried to solve a simplified version of the above MDP formulation using backward induction and Monte Carlo method.

For backward induction, since the state space is excessively large. The backward induction unable to solve daily decision problem. Furthermore, since we don't have any data available, which causes the result is highly depend on the input parameter, and a proper methodology to tune parameters is also one of the main problems we are facing.

The Monte Carlo method has the same issues; in addition, as time is one dimension in our state space, this means each state can only be visited at most once in each episode. And in practice, after we run 2,000,000 episodes, there are still many states not be visited.

Moving forward

By summarizing the issues, we have so far, we consider using function approximation approach to solve the Dynamic Programming problem.

References

[1] Chen, J. (2020, January 29). How a Swing Option Works. Retrieved from https://www.investopedia.com/terms/s/swing-option.asp
[2] Kenton, W. (2020, February 5). Real Options: Exploring the Various Types. Retrieved from https://www.investopedia.com/terms/r/realoption.asp
[3] Zhang, J., & Zhao, H. (2006). *Asset pricing under jump diffusion*. Hong Kong: School of Economics & Finance, University of Hong Kong.
[4] Amin, K. I. (1993). Jump Diffusion Option Valuation in Discrete Time. *The Journal of Finance*, *48*(5), 1833–1863. doi: 10.1111/j.1540-6261.1993.tb05130.x
[5] Kazuhisa Matsuda. (2004, December). *Introduction to Merton Jump Diffusion Model.* New York: Department of Economics, The Graduate Center, The City University of New York.
[6] Merton, R. C. (1975). *Option pricing when underlying stock returns are discontinuous*. Cambridge, MA: M.I.T. Alfred P. Sloan School of Management.
[7] Halperin, I., & Feldshteyn, I. (2018). Market Self-Learning of Signals, Impact and Optimal Trading: Invisible Hand Inference with Free Energy. *ArXiv*, *abs/1805.06126*.
[8] S. Boyd, E, Busetti, S. Diamond, R.N. Kahn, K. Koh, P. Nystrup, and J. Speth, ”MultiPeriod Trading via Convex Optimization”, Foundations and Trends in Optimization. Vol. XX, no. XX, 1-74 (2017)
[9] Kai-Siong Leow. (2013, May). *Pricing of swing options: A Monte Carlo Simulation Approach.*
[10] Sutton, Richard S., and Andrew G. Barto. *Reinforcement Learning an Introduction*. The MIT Press, 2018.