Strategic Capacity Planning using Data Science, Optimization, and Machine Learning

by
Hans Nowak II
B.S., Applied Mathematics & Quantitative Economics, United States Naval Academy, 2012

Submitted to the MIT Department of Mechanical Engineering and MIT Sloan School of Management in partial fulfillment of the requirements for the degrees of

Master of Science in Mechanical Engineering
and
Master of Business Administration
in conjunction with the Leaders for Global Operations Program at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
May 2020
© 2020 Hans Nowak II. All rights reserved.

Signature of Author ....
MIT Department of Mechanical Engineering and Sloan School of Management
May 8, 2020

Certified by ....
Roy Welsch, Thesis Supervisor
Professor of Statistics and Management Science, MIT Sloan School of Management

Certified by ....
Daniel Frey, Thesis Supervisor
Professor of Mechanical Engineering, MIT Department of Mechanical Engineering

Accepted by..
Nicolas Hadjiconstantinou, Chair
Mechanical Engineering Committee on Graduate Students

Accepted by..
Maura Herson, Assistant Dean, MBA Program
MIT Sloan School of Management

This document does not contain technology or Technical Data controlled under either the U.S. International Traffic in Arms Regulations or the U.S. Export Administration Regulations.

THIS PAGE IS INTENTIONALLY LEFT BLANK

Strategic Capacity Planning using
Data Science, Optimization, and Machine Learning

by

Hans Nowak II
B.S., Applied Mathematics & Quantitative Economics, United States Naval Academy, 2012

Submitted to the MIT Department of Mechanical Engineering and MIT Sloan School of Management on May 8, 2020, in partial fulfillment of the requirements for the degrees of Master of Science in Mechanical Engineering and Master of Business Administration

Abstract

Raytheon's Circuit Card Assembly (CCA) factory in Andover, MA is Raytheon's largest factory and the largest Department of Defense (DOD) CCA manufacturer in the world. With over 500 operations, it manufactures over 7000 unique parts with a high degree of complexity and varying levels of demand. Recently, the factory has seen an increase in demand, making the ability to continuously analyze factory capacity and strategically plan for future operations much needed.

This study seeks to develop a sustainable strategic capacity optimization model and capacity visualization tool that integrates demand data with historical manufacturing data. Through automated data mining algorithms of factory data sources, capacity utilization and overall equipment effectiveness (OEE) for factory operations are evaluated. Machine learning methods are then assessed to gain an accurate estimate of cycle time (CT) throughout the factory. Finally, a mixed-integer nonlinear program (MINLP) integrates the capacity utilization framework and machine learning predictions to compute the optimal strategic capacity planning decisions.

Capacity utilization and OEE models are shown to be able to be generated through automated data mining algorithms. Machine learning models are shown to have a mean average error (MAE) of 1.55 on predictions for new data, which is 76.3% lower than the current CT prediction error. Finally, the MINLP is solved to optimality within a tolerance of 1.00e-04 and generates resource and production decisions that can be acted upon.

Thesis Supervisor: Roy Welsch
Title: Professor of Statistics and Management Science, MIT Sloan School of Management

Thesis Supervisor: Daniel Frey
Title: Professor of Mechanical Engineering, MIT Department of Mechanical Engineering

THIS PAGE IS INTENTIONALLY LEFT BLANK

Acknowledgments

This thesis would not have been possible without the help and guidance of many people. I would like to first thank my MIT academic advisors, Roy Welsch and Dan Frey, for their continuous mentorship and advice, and Raytheon for their continued support of the MIT LGO program. Specifically, I want to thank Mike Shimazu, Sarah Medina, and Steve Allen for their support and leadership throughout my time at Raytheon. I would also like to thank Paulo Cavalcanti, Dave Devietro, James Neil, Mike Kaczmarski, Robert Di Carlo, Brett Murphy, Steve Tribuna, and Steve Smith for teaching me various tools and skills during my research fellowship. Finally, I want to thank my family, who have supported me throughout my entire MIT journey. I would not be where I am without them.

THIS PAGE IS INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

List of Tables

List of Equations

Acronyms

ANN	Artificial Neural Network
BOM	Bill of Materials
CA	Capacity Available
CAPEX	Capital Expenditures
CART	Classification and Regression Tree
CCA	Circuit Card Assembly
CR	Capacity Required
CT	Cycle Time
CU	Capacity Utilization
DOD	Department of Defense
EN	Elastic-Net
ERP	Enterprise Resource Planning
FPY	First Pass Yield
FSA	First Starting Action
GAO	Government Accountability Office
GBT	Gradient Boosted Trees
GUI	Graphical User Interface
HMLV	High-Mix, Low-Volume
IIoT	Industrial Internet of Things
IoT	Internet of Things
KNN	K-Nearest Neighbor
LASSO	Least Absolute Shrinkage and Selection Operator
LEA	Last Ending Action
LP	Linear Program
LTAMDS	Lower Tier Air and Missile Defense Sensor
MAE	Mean Average Error
MDI	Mean Decrease in Impurity
MILP	Mixed-Integer Linear Program
MINLP	Mixed-Integer Nonlinear Program
MIP	Mixed Integer Program
MLP	Multi-Layer Perceptron
MSE	Mean Squared Error
OEE	Overall Equipment Effectiveness
OEM	Original Equipment Manufacturer
PCA	Principal Component Analysis
PCB	Printed Circuit Board
PT	Processing Time
QT	Queue Time
RF	Random Forest
ROI	Return on Investment
S&OP	Sales and Operations Planning

SAP	Systems, Applications and Products
SGD	Stochastic Gradient Descent
SMT	Surface Mount Technology
SQL	Structured Query Language
SVM	Support Vector Machine
TP	Throughput
TPM	Total Productive Maintenance
UTC	United Technologies Corporation
VIF	Variance Inflation Factor
WIP	Work-in-Progress

CH. 1 INTRODUCTION

This thesis seeks to develop a novel optimization framework for strategic capacity planning of a high-mix, low-volume (HMLV) factory. This chapter focuses on presenting the problem statement and objectives of the thesis. Additionally, it introduces the statements of hypotheses and research methodologies prior to providing an overview of the general thesis.

1.1 PROBLEM STATEMENT

Raytheon's Circuit Card Assembly (CCA) factory in Andover, Massachusetts is Raytheon's largest factory and the largest Department of Defense (DOD) CCA manufacturer in the world. It manufactures over 7000 unique parts within its Enterprise Resource Planning (ERP) system that encompass a high degree of complexity with varying levels of demand. Recently, two major factors have greatly increased the demand of the CCA factory in Andover:

1. A large-scale consolidation from three manufacturing factories across the United States into a single CCA factory in Andover
2. A rapidly growing market due to new Raytheon technology resulting in a high volume of new product introductions into the CCA factory

Together, these factors create an anticipated 416% increase in demand in the next five years as shown in Figure 1.

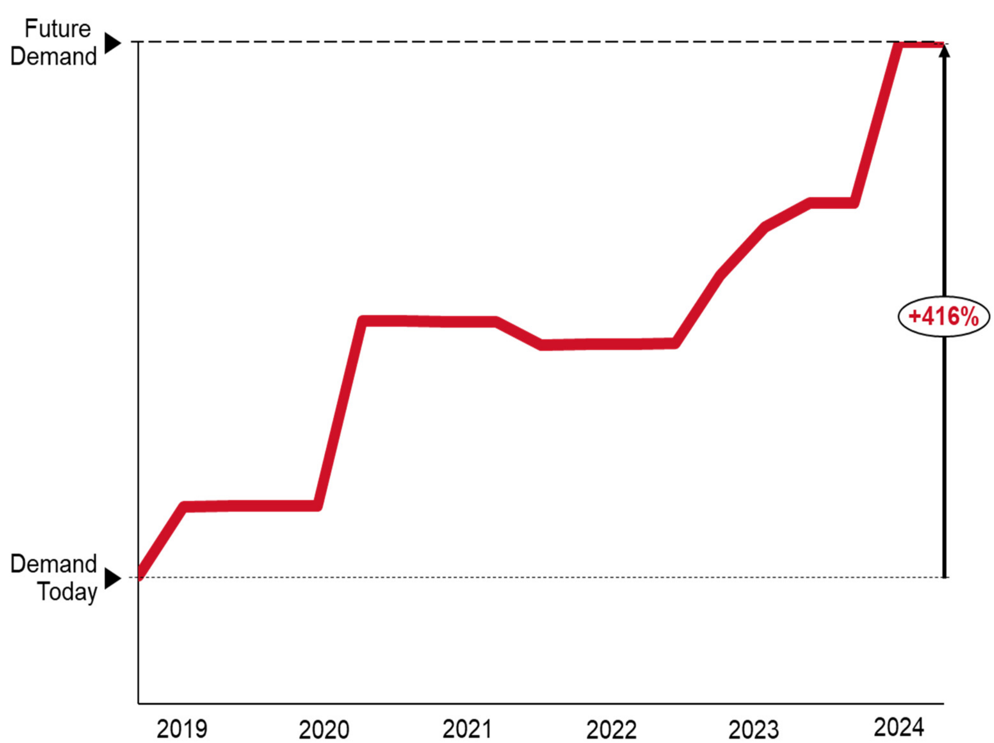

Figure 1: Anticipated increase in demand over the next five years

To ensure the CCA factory has the capital equipment required to absorb the increase in demand, a thorough and intensive capacity analysis was completed by an engineering team. To verify data fidelity and accurate results, this effort was extremely manually intensive and took the team over a year to complete. While this enabled a static snapshot of current capacity, it did not account for inevitable changes in the factory or look for ways to optimize strategic capacity planning.

With Raytheon recently receiving the contract to build the Lower Tier Air and Missile Defense Sensor (LTAMDS), which is the Patriot missile defense system's replacement, demand has already started to rapidly increase. Furthermore, with the potential for demand to continue to increase after Raytheon's recent merger with United Technologies Corporation (UTC), the ability to continuously analyze factory capacity for new products and variable demand is much needed. New operations, products, and other factors need to be seamlessly integrated into the analysis to enable continuous capacity planning.

Before the first capacity analysis effort, there was not a centralized way to look at overall factory capacity. Many operations locally estimated capacity; each one used different assumptions, different data, and had different goals for their analysis efforts. With hundreds of operations in the factory, this means that there are a lot of personal Excel spreadsheets that do not align. This is one of the most challenging aspects of capacity analysis at an extremely high-mix, relatively low-volume factory. CCA has over 500 distinctive operations and approximately 7000 unique parts with different cycle times. This equates to 3.5M cycle times. To make the analysis more difficult, there are more than 5000 routes that parts take through the factory due to needing a specialized, albeit shared, subset of operations.

1.2 OBJECTIVES

This thesis will involve creating a capacity analysis and visualization tool that can be used in strategic capacity planning. In order to be effective in this type of factory environment, the tool will need to automatically calculate static capacity and overall equipment effectiveness (OEE) to capture current state capacity and potential improvement areas in the factory. The complexity of the factory further makes accurate prediction of cycle time critical to assess intermediate due dates, schedule resources, and analyze future capacity utilization. From the capacity analysis of future demand, an optimization model can provide data-driven recommendations based on return on investment (ROI) for increasing the workforce vs. investing in capital equipment vs. shifting manufacturing schedule.

In sum the three primary objectives are:

1. Automate capacity utilization and overall equipment effectiveness analysis
2. Accurately predict cycle times of future demand
3. Provide data-driven recommendations for strategic capacity planning

1.3 STATEMENT OF HYPOTHESIS AND RESEARCH METHODOLOGY

We test three hypotheses in this thesis based on the three objectives mentioned in 1.2 OBJECTIVES. Our first hypothesis is that there is a more effective method to calculate current capacity than the current manual process. We aim to use current data sources at the factory to automatically calculate and provide useful estimates of capacity utilization and overall equipment effectiveness. Through scheduled executable Python scripts that use Structured Query Language (SQL) to automatically pull and calculate from Raytheon's current data warehouse, we aim to develop a tool that aligns with the most recent manual capacity analysis effort but relieves the company of the extensive time and resources required to calculate it.

Our second hypothesis is that cycle time can be more accurately predicted than the currently implemented method at Raytheon. Through supervised machine learning methods, we aim to develop a model that enables us to predict, with an acceptable accuracy, the cycle time of an item going through the factory. We measure model performance on out-of-sample data to quantify the predictive power of the model and compare it to Raytheon's current predictions.

Our final hypothesis is that mathematical programming methods can provide data-driven recommendations for strategic capacity planning. Through linear and nonlinear mixed-integer programs, we aim to develop a model that enables us to adjust multiple capacity planning factors to maximize ROI by minimizing cost. We measure performance by comparing model output versus the baseline capacity plan.

1.4 THESIS OVERVIEW

The remaining chapters of this thesis are structured in this manner:

CH. 2 PROBLEM BACKGROUND presents an overview of the defense industry followed by a deeper look at Raytheon. We then provide background knowledge on circuit card manufacturing, manufacturing capacity, strategic capacity planning, and OEE. We end with key challenges at Raytheon's CCA factory.

CH. 3 LITERATURE REVIEW looks into prior work on using mathematical optimization models for strategic capacity planning and machine learning for cycle time prediction.

CH. 4 CAPACITY UTILIZATION AND OEE ANALYSIS initially provides an overview of available data and features that can be calculated with it for capacity analysis. Capacity utilization and OEE are then calculated with these features. Results of the calculations are then presented.

CH. 5 OPTIMIZATION MODELS FOR STRATEGIC CAPACITY PLANNING lays out simple linear models for capacity planning and provides an example with generated data prior to adding constraints to mimic the complexity of the factory. Finally, a mixed-integer nonlinear program is developed to be used for strategic capacity planning.

CH. 6 MACHINE LEARNING TO PREDICT CYCLE TIME begins with a machine learning overview before features are generated and selected for modeling. Next, the ten models initially built are discussed and the machine learning framework used is laid out. We present the data ingestion and preparation done prior to building, training, and testing the models.

CH. 7 RESULTS AND DISCUSSION starts with an evaluation of the capacity planning model after it has been solved to an optimal solution. The business impact of enacting the model and model limitations follow.

CH. 8 CONCLUSION AND RECOMMENDATIONS summarizes the findings and recommendations for Raytheon, including recommendations for implementation of the model. Lastly, areas for future research and applications of the model are discussed.

CH. 2 PROBLEM BACKGROUND

This scope of this thesis covers a high-mix, low-volume manufacturing factory within the defense industry. This chapter provides an overview of the industry and one of the top companies within it. Additionally, we provide background knowledge on circuit card manufacturing and manufacturing principals used in this thesis before discussing some of the specific challenges in this type of factory.

2.1 THE DEFENSE INDUSTRY

The defense industry continues to remain strong due to a robust defense budget in the United States and is anticipated to continue to grow. As security threats have intensified and the defense budget continues to grow, defense expenditure is expected to grow at a CAGR of ~3% and reach $2.1 trillion by 2023. According to Deloitte, growth in defense spending and demand for military equipment will likely create increased opportunities for the industry. To meet the increased demand and improve yields, they recommend leveraging agile production and investing in independent research and commercial adaptability of digital technologies. [1]

Before WWII, the defense industry has historically relied on commercial technology innovation and adapted it for military use. Since WWII, however, the focus has shifted from commercial adaptability to independent research and development for the military. In fact, 51% of original equipment manufacturers (OEM) report that most innovation is done for military use and then adapted for commercial use. Nevertheless, this trend seems to be shifting back to our pre-WWII model with technologies such as artificial intelligence and the Internet of Things (IoT) emerging at a pace too fast for the defense industry. This does not mean that innovation is slowing in the defense industry. With almost $700B in the 2019 defense budget, investing in technology innovation is growing in the industry. [2]

Even though technology innovation is being invested in, implementation of the new technology has been challenging in the defense industry. According to a Jabil survey, 74% of respondents say that a lack of leadership to transform processes and mindsets is the biggest roadblock. This may be due to an older workforce in the industry and the generational gap that resulted from the defense budget cuts in the 1990s. Due to these cuts, hiring came to a standstill and now many defense companies have 25% of their workforce retiring within the next five years. [2] [3] To combat this, companies such as Raytheon and UTC each plan to hire 10,000 people in the next year.

While defense expenditure and defense companies such as Raytheon and UTC continue to grow, especially after their merger is complete in 2020, the defense industry in terms of companies is shrinking. According to a Government Accountability Office (GAO) analysis of DOD data, nearly half of defense contracts are awarded to five companies. [4] The pie is getting larger, but the number of pieces keeps shrinking. As a result to less competition, some believe this will lead to higher costs and less innovation. Others counter that efficiencies lead to lower costs for the Pentagon overall. [5]

As one of the top companies in the defense industry, Raytheon is a technology and innovation leader with core manufacturing concentration in weapons and military electronics. As suggested by Deloitte, Raytheon is looking to meet the increased demand and improve yields by investing in digital technologies and agile methodologies.

2.2 RAYTHEON COMPANY OVERVIEW

Raytheon Company (Raytheon) is a large multinational provider of technologically advanced products, services, and solutions to civilian and government organizations. Established in Cambridge, Massachusetts with strong ties to MIT and now headquartered in Waltham, Massachusetts, it is primarily known as a major defense contractor with manufacturing concentrations centered on military weapons and commercial electronics. With almost 100 years of innovation history, Raytheon continues to provide state-of-the-art electronics, defense technologies, and cybersecurity solutions for customers in over eighty countries. Its revenue of approximately $27B annually makes it the third-largest defense company in the world. [6] [7]

The company is currently divided into four primary business units including Integrated Defense Systems (IDS); Missile Systems (MS); Intelligence, Information, and Services (IIS); and Space and Airborne Systems (SAS). The largest of the business units by both revenue and profit is IDS, which specializes in air and missile defense, land-based radars, sea-based radars, and systems for command, control, communications, cybersecurity, and intelligence. It also produces sonar systems, torpedoes, and shipboard electronics. [8] It is headquartered in Tewksbury, Massachusetts with office and manufacturing sites spread throughout the greater Boston area. The Andover, Massachusetts site is home to the production of the Patriot missile defense system and the CCA center of excellence for the entire company.

Recently, Raytheon and UTC, a leading aerospace company, entered into an agreement to combine in an all-stock merger of equals. The combined company will be named Raytheon Technologies Corporation and will have approximately $74B in pro-forma revenue. Due to the merger, Raytheon plans to consolidate its four primary businesses into two businesses to be named Raytheon Intelligence & Space and Raytheon Missiles & Defense. These new businesses will join Collins Aerospace and Pratt & Whitney of UTC to form the four businesses of Raytheon Technologies Corporation. Altogether, the new company will be considered the second-largest defense company in the world. [9]

2.3 CIRCUIT CARD MANUFACTURING

Printed circuit boards (PCB) mechanically support and electrically connect components via deliberately placed conductive features. These components are typically soldered to the board with non-conductive materials. PCBs have been around for roughly one-hundred years and involve a conductive material such as copper inscribed or laminated onto and/or between non-conductive material. Until the mid-1980s, the primary method to create a PCB was with through-hole fabrication coupled with wave soldering and dip soldering techniques. Since then, surface mount technology (SMT) has become the fabrication technology of choice. Today, surface-mount components are typically soldered onto printed circuit cards with the use of reflow ovens. [10]

A circuit card assembly is comprised of a myriad of electronic components such as resistors, capacitors, chips, and diodes. Each component has multiple leads that are soldered onto specific points of a printed circuit card. While the through-hole technology method consists of fitting components with wire leads into holes in the PCB, SMT components have small leads and are placed directly onto the surface of the PCB. This method enables the use of smaller components, which subsequently enables smaller boards, and also enables much faster production times. Figure 2 shows the difference between the two technologies. While most of the circuit cards manufactured in Raytheon's CCA factory use SMT, some still use through-hole technology due to legacy design or large, specialized components. [11] [12]

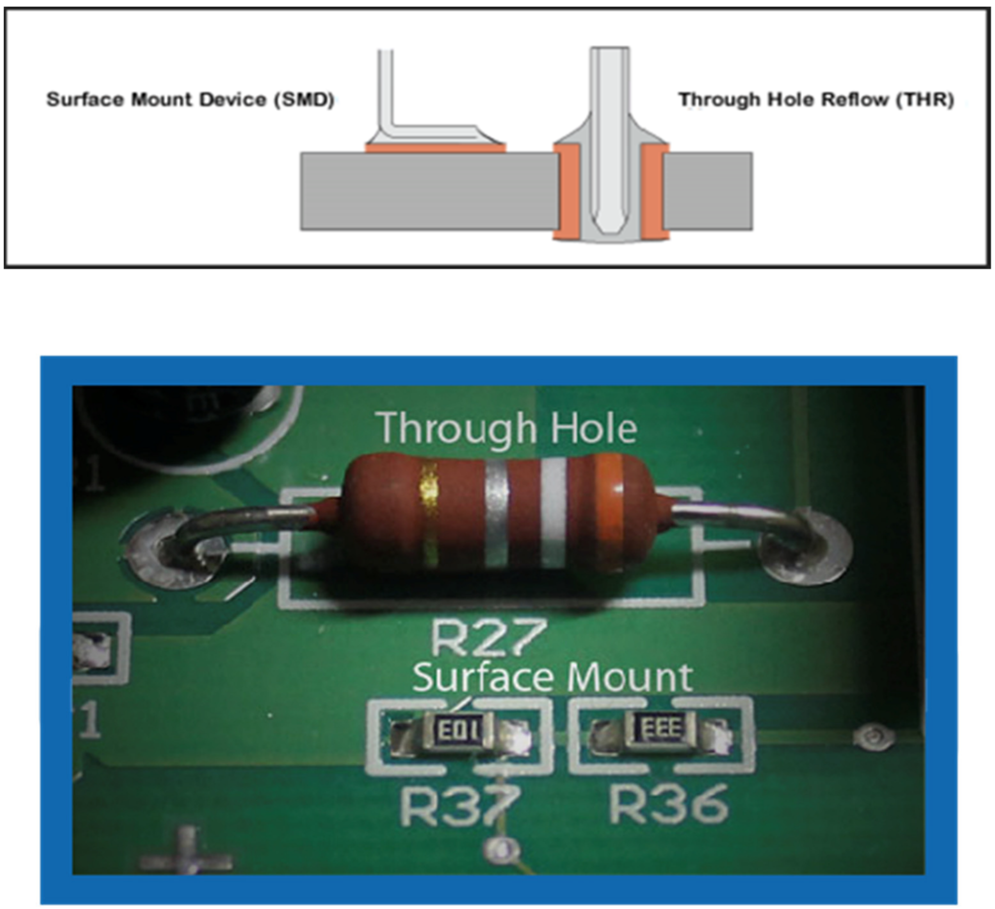

Figure 2: Pictures showing two components using different methods of placement [11] [12]

Circuit card manufacturing is highly complex and intricate due to the need to place hundreds of tiny, high-value components onto a small board. When these cards are being used in life-or-death equipment such as missile detection, the quality is of utmost importance. Raytheon's CCA center of excellence manufactures all of the circuit cards for Raytheon products at a high-mix, low-volume scale with over 7000 unique parts. A circuit card generally flows through the factory on a route such as the one shown in Figure 3. After a shop order is released, the new lot of cards is kitted, labeled, and baked. When the previous lot is finished with the SMT line, the component feeders and lines are set up for the new lot of cards. This setup typically takes hours to complete. Once complete, the lot then proceeds through the SMT line, which is comprised of a screen printer for solder paste, paste inspection, automated component placement, automated optical inspection, and finally a reflow oven. Next, the lot of cards goes to X-ray touch up to find and correct hidden assembly defects before manual assembly of certain components occurs. After assembly, the lot then goes through some form of coating to protect the board's components. Lastly, the lot is tested both mechanically and electrically before being marked as complete.

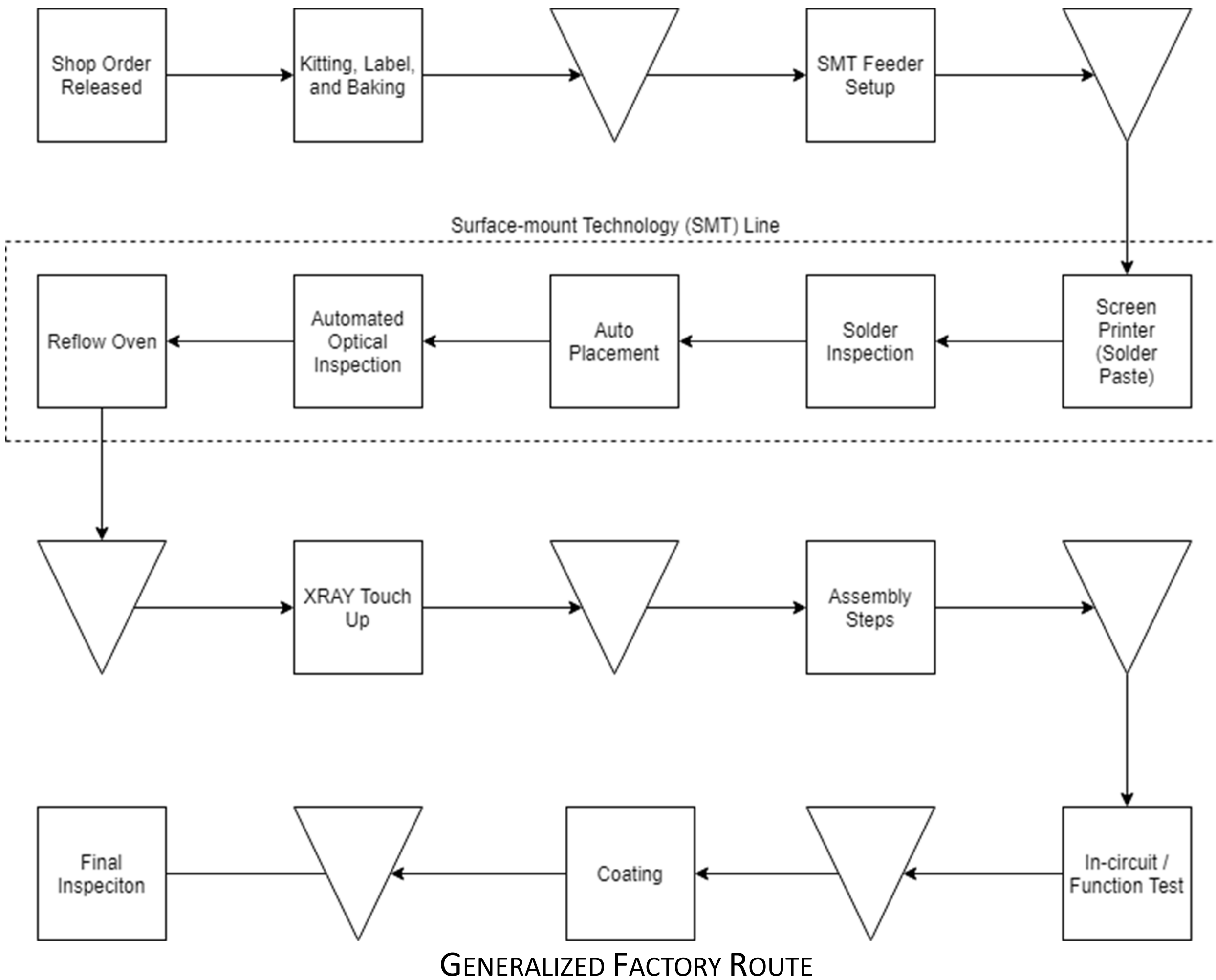

Figure 3: General route an item takes through Raytheon's CCA factory in the form of a process map

It is very important to note that this is just one route a lot of boards can take through CCA. Some boards do not go through SMT and some go through other soldering operations. There are also multiple assembly operations, several different types of coating operations, and many testing operations. Overall, there are over 500 operations in CCA and more than 5000 unique routes through the factory. This complexity is further explained in 2.7 KEY CHALLENGES AT RAYTHEON'S CCA FACTORY.

2.4 MANUFACTURING CAPACITY

First, we shall define capacity utilization (CU) for each resource within the factory. Willems describes it as a pie chart equal to the sum of two quantities: runtime and idle time. Runtime is the time the resource is processing an item, while idle time is the time the resource is not working on anything. Furthermore, capacity available (CA), which represents the entire pie, is all the resource can provide in a specified time interval. Capacity required (CR), on the other hand, is all the resource needs to provide in a specified time interval. Therefore, we can define capacity utilization as the ratio of capacity required to the capacity available.

$$capacity\ utilization = \frac{capacity\ required}{capacity\ avaiable}$$

Equation 1

We will discuss capacity in this thesis in terms of time. The time intervals we will primarily consider in this thesis are monthly hours. Therefore, if looking at a specific machine in the factory, capacity required will be the hours of processing time required in a given month to meet demand, and capacity available will be the hours available in the same month for processing.

We will define throughput (TP) as the average output of an item per unit time, work-in-progress (WIP) as the inventory of the item in the production system, and cycle time (CT) as the amount of time an item spends as WIP or, in other words, the total time to move an item through a process. Little's law connects these and states that on average:

$$CT = \frac{WIP}{TP}$$

Equation 2

For example, if there are on average 100 circuit cards of WIP in a lot at a specific operation and cycle time is 50 minutes per card then throughput would be one circuit card every 2 minutes or 30 cards per hour. [13] Note that this law assumes strictly stationary processes, which limits its ability to accurately predict in our models. [14]

We further breakdown cycle time of an operation into two components—queue time and production time—such that: $CT_{operation} = QT + PT$. Since CT of an operation is very sensitive to the item being processed, we can further breakdown cycle time by operation:

$$CT_{operation,item} = QT_{oi} + PT_{oi}\ \forall o \in \{Operations\}, i\ \in \{Items\}$$

Equation 3

Since lots sometimes have different sizes at this particular factory, we will look at the per unit level of production time to compare apples to apples, which we will call processing time:

$$pt_{oi} = \frac{PT_{oi}}{QTY_{oi}}$$

Equation 4

where QTY_{oi} is the quantity of items i processed through operation o.

While QT is technically occurring at an operation, it does not necessarily take time away from a machine or person. It just builds up WIP. Therefore, capacity required of an operation can then be defined as: $(\#\ Items\ Needing\ Processed)(Item\ Processing\ Time)$

$$CR_o = \sum_{i=1}^{I} d_{oi}pt_{oi}\ \forall o \in \{Operations\}$$

Equation 5

where d is the demand at operation o for item i. For example, if the monthly demand for Operation A is only 1000 of only Item 1, which has a processing time of 15 minutes in Operation A, the capacity required for Operation A is 15000 minutes or 250 hours.

Capacity available of an operation can then be defined as:
$(\#Shifts)(\#Working\ Days\ in\ Month)(Useful\ Shift\ Length)(\#Resources)$

$$CA_o = sh_o wd_o ls_o r_o\ \forall o \in \{Operations\}$$

Equation 6

where sh, wd, ls, and r are the number of shifts per day, working days, usable length of shifts, and resources for operation o respectively. This equation assumes that shifts do not overlap. For example, if there are 2 shifts that are 8 hours long with zero breaks in a working day and 20 working days in the month, the capacity available at an operation with a single resource is 320 hours. More simply put, capacity available is the sum of planned production time of each resource at an operation:

$$CA_o = ppt_o r_o\ \forall o \in \{Operations\}$$

Equation 7

Capacity utilization is, therefore:

$$CU_o = \frac{\sum_{i=1}^{I} d_{oi}pt_{oi}}{ppt_o r_o}\ \forall o \in \{Operations\}$$

Equation 8

$$= \frac{\sum_{i=1}^{I} d_{oi}pt_{oi}}{sh_o wd_o ls_o r_o}\ \forall o \in \{Operations\}$$
$$= \frac{250\ hours}{320\ hours}$$
$$= 78\%$$

With many companies considering around 85% to be an optimal utilization rate, this 78% would be considered good. [15] The optimal rate varies from company to company, but the basic thought process is that if it is too high then variability in demand or other factors can cause it to exceed 100%. If it is too low, then we are wasting a useful resource. Both factors will impact a company's bottom line. Nevertheless, in this scenario, if demand doubled to 2000, then capacity required would double to 500 hours, resulting in a capacity utilization of 156%. Since it is impossible to operate at 156%, demand will not be met in this current configuration. If we want to improve this operation's utilization, Willems points out that there are three available options. [16]

1. Increase resources at the operation (maintain speed but increase time available or resources)

    a. Add Shifts

    b. Add Machines if Operation is Machine Driven

    c. Add Workers if Operation is Labor Driven

2. Make the operation faster or reduce defects (in the same amount of time)

3. Shift demand at the operation

In this example, we now need to increase capacity available to at least 500 hours to meet demand or (500/85%)=588 hours if we want to maintain capacity utilization at 85%. Let's walk through the three available options:

1. Increasing resources

    a. Adding Shifts – Adding a third shift will increase CA to 480 hours, which is not enough. We could also look into adding one weekend day per week, increasing working days to 24. This would increase CA to 576 hours, making CU = 87%.

    b. Adding Machines – Adding a second machine will double CA to 640 hours, maintaining CU at 78%.

    c. Adding Workers – No change in a machine-driven operation; however, additional workers may be needed to operate any added machines.

2. Make the operation faster or reduce defects – If demand doubled and we wanted to maintain 78% utilization, we would need to cut PT in half. This seems highly

improbable. Additionally, defects were not factored into this scenario and therefore would not change CU.

3. Shift demand – Since we are only looking at a single time period, this option is not available. Nevertheless, if demand this month is 0 and next month it is 2000, we would be able to process 1000 each month and maintain CU at 78%. [16]

So what option(s) should we go with?

This is the question at the heart of strategic capacity planning discussed in 2.5 STRATEGIC CAPACITY PLANNING.

2.5 STRATEGIC CAPACITY PLANNING

Strategic capacity planning refers to the decision making of the sequence and the timing of machine purchases and workforce adjustments. It is a multi-criteria decision-making process involving trade-offs between finance, output, and risk. All manufacturing organizations are faced with difficult decisions surrounding these trade-offs; however, capital-intensive industries such as defense technologies make investment decision-making critically important. This is why extensive research has been done on capacity planning in industries with high capital investment costs. [17] [18] Geng and Jiang evaluated three methods for devising a strategic capacity plan: static capacity modeling, simulation-based search modeling, and mathematical programming modeling. [18]

The first method they evaluated was the static capacity model. Traditionally at many companies, including Raytheon, static capacity analysis via spreadsheets drove capacity planning due to its ease of use. While this method is easy to use and understand, its highly aggregated approach does not accurately account for different products needing different processing times, which is the case in the CCA factory. Therefore, different capacities are needed depending on the product mix. Moreover, when products do not follow similar routings through the factory, which is also the case in the CCA factory, a spreadsheet cannot sufficiently assess required capacity. [18]

The second method they evaluated was simulation-based modeling. Starting with the current state of the factory, small changes are iterated upon until performance improves to a better state. This method provides more accurate capacity analysis than static models; however, it requires an abundance of detailed information across the factory, making it a difficult method in a high-mix, low-volume factory with hundreds of operations and thousands of routings. It can also become overly complicated for long-term strategic planning purposes. [18]

The final method they evaluated was mathematical programming. such as linear programming (LP) models and mixed-integer linear programming (MILP) models. In this type of modeling, required constraints such as demand are formulated along with an objective function to find the optimal capacity planning decisions for machine purchases and workforce adjustments. With its ability to also provide an optimal production plan, this method has become the primary method used in capacity planning. [19] Additionally, with software such as Julia/Jump and solvers such as Gurobi quickly finding the optimal solutions to these models makes their adoption much easier. [20] Due to these reasons and the ability to easily adjust the model as strategically necessary, this is the method we implement in this thesis.

2.6 OVERALL EQUIPMENT EFFECTIVENESS (OEE)

Overall equipment effectiveness (OEE) is an effective way to measure productivity and efficiency in a manufacturing facility. Introduced by Nakajima in the Total Productive Maintenance (TPM) system, it is comprised of multiple metrics that focus on the capacity utilization of a specific manufacturing operation and come together in a generalized fashion to enable comparison between multiple manufacturing operations. [21] [22] This enables a company to identify manufacturing potential, locate production problems, and pinpoint improvement areas in order to increase productivity and decrease cost. Specifically, it was originally designed to reduce six losses:

1. Equipment Breakdown
2. Set-up and Adjustment Downtime
3. Minor Stoppage Downtime
4. Reduced Speed Losses
5. Quality Defects and Rework
6. Start-up Losses

OEE does this by breaking the productivity and efficiency of a manufacturing operation into three distinct parts—Availability, Performance, and Quality—and can be calculated my multiplying the three parts.

$$OEE = Availability * Performance * Quality$$

Equation 9

Availability (A) is the ratio of time an operation is actively producing to the time an operation is available to actively produce. In manufacturing, common terms that are used in this calculation are planned production time and run time, where planned production time is the shift length minus planned breaks and run time is planned production time minus stop time. Availability, in this setting, takes on this equation:

$$Availability = \frac{Run\ Time}{Planned\ Production\ Time}$$

Equation 10

For example, if a worker operates a machine during an eight-hour shift, but gets thirty minutes for lunch and two fifteen-minute breaks, the planned production time for that worker and machine is seven hours. If the machine has a problem for an hour while the worker is trying to operate it, then the run time is six hours. Therefore, the availability is 6/7 or 86%.

Performance (P) is the ratio of the minimum time to produce a number of outputs to the time the operation was actively producing. In manufacturing, common terms that are used in this calculation are ideal process time, total parts, and run time, where ideal process time is the fastest theoretical time required to produce one part and total parts is the total output in a given run time. It is important to note that total output includes defects in this calculation because defects are taken into account in the quality calculation. Performance, in this setting, takes on the equation:

$$Performance = \frac{Ideal\ Process\ Time * Total\ Parts}{Run\ Time}$$

Equation 11

For example, in seven hours of run time it is theoretically feasible to produce 28 parts since the ideal process time is fifteen minutes. If instead only 26 parts are made in that hour, the performance is 26/28 or 93%.

Quality (Q) is the ratio of good outputs to total outputs. In manufacturing, common terms that are used in this calculation are good parts and total parts, where good parts are equal to total parts minus defective parts. Quality, in this setting, takes on the equation:

$$Quality = \frac{Good\ Parts}{Total\ Parts}$$

Equation 12

For example, if one of the 26 parts created in the previous example were considered defective then only 25 of them would be considered good. Therefore, quality would be 25/26 or 96%. Note that parts that require rework are counted as rejects the first time they run through a manufacturing operation. This is similar to first pass yield. When a part runs through an operation a second time, the time required for rework is not factored into planned production time and, therefore, does not affect OEE.

When looked at individually, each of the three parts of OEE points to a potential process improvement area within a given operation. When looked at collectively as OEE, large scale comparison across operations can be tracked at a higher level and benchmarked. In manufacturing, a common benchmark for OEE is 85%, but this depends heavily on the industry.

In the previous examples where Availability=86%, Performance=93%, and Quality=96%, OEE would be 77%.

$$\begin{aligned} OEE &= A * P * Q \\ &= 86\% * 94\% * 96\% \\ &= 77\% \end{aligned}$$

Even though 77% is relatively good in most industries, it can be seen that this OEE score is primarily driven by availability. While we know that the machine had an unplanned breakdown for an hour from the example, other factors such as a long setup time or material shortage could have been the cause. Either way, the metric provides an indication to company leadership that something may be wrong and triggers further analysis to improve the productivity and efficiency of the operation.

2.7 KEY CHALLENGES AT RAYTHEON'S CCA FACTORY

The combination of four factors makes capacity analysis challenging at Raytheon's CCA factory. The high-mix, low-volume manufacturing coupled with the number of operations and routes, unique cycle times of operations and items, and factory size and operation type, make accurately analyzing for capacity required and capacity available difficult. In order to calculate capacity required, we must understand the demand of 7000 items in over 500 operations and the 3.5M associated cycle times. In order to calculate capacity available, we must also understand the schedules and resource levels of the 500+ operations. Due to the complexity of the factory, calculating capacity utilization to be used for strategic capacity planning is challenging and, doing so manually, is nearly impossible to sustain.

High-Mix, Low-Volume (HMLV) Manufacturing

Due to circuit cards needing to be specifically designed to their end-use, they must be highly customized and varied. Production in CCA is therefore designed to be HMLV in order to meet the circuit card demand. Unlike an assembly line which is designed for single-piece flow and highly efficient yet inflexible, a HMLV factory mimics a job shop where most items produced require unique setups and routing through the shop. [23] This makes capacity planning challenging in CCA.

CCA manufactures over 7000 unique parts with an average output of 1000 parts per day and growing. The highest volume part does 45k units per year, while over one hundred other parts are produced only once per year or even once every other year. In a single operation, setup times can vary by hours and cycle time can vary by days between items. The extremeness of HMLV at CCA makes manual methods of capacity analysis almost impossible to perform accurately.

Number of Operations and Routes

With over 500 operations in CCA shared between hundreds or even thousands of items, the high-mixed nature of the factory becomes even more complicated. While some items go through just a few operations within the factory, others go through more than fifty operations. We can see the complexity of the factory by plotting the network of random item routes through the factory. Figure 4 shows a progressing number of random items selected for network graphing from one to two with the number of operations and edges also indicated. One random item going through the factory is easy to follow with this particular item going through eleven operations. As this increases to random two items, it is still easy to follow with only twelve operations.

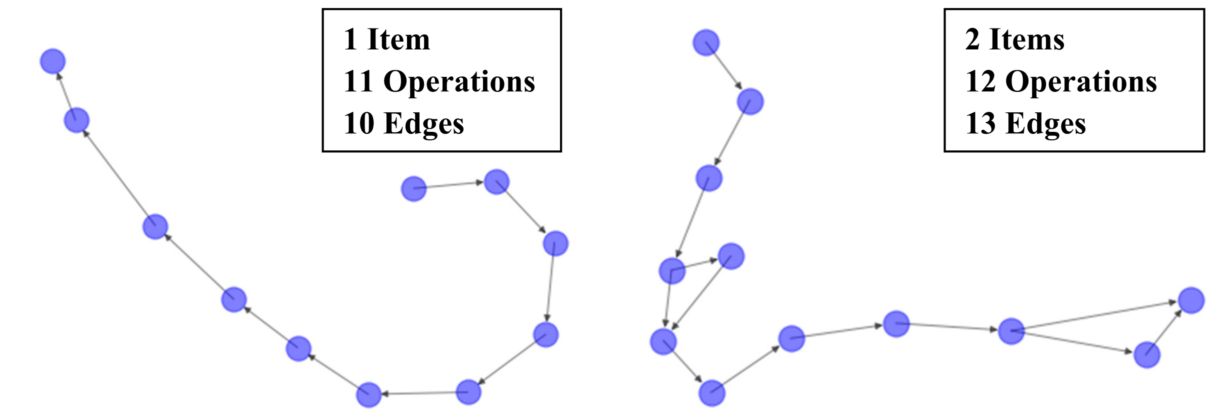

Figure 4: Network graph of 1 and 2 random items moving through the factory

Nevertheless, as we continue to add items to our network of routes, we quickly see how difficult it is to understand the flow of every item through the factory. As shown in Figure 5, 100 random items require 68 operations with 249 unique edges between them, and 1000 items require 252 operations with 1633 edges between them. Altogether, the 7000 items have over 5000 unique routes through the factory. Again, this complexity makes manual methods of capacity analysis almost impossible to perform accurately.

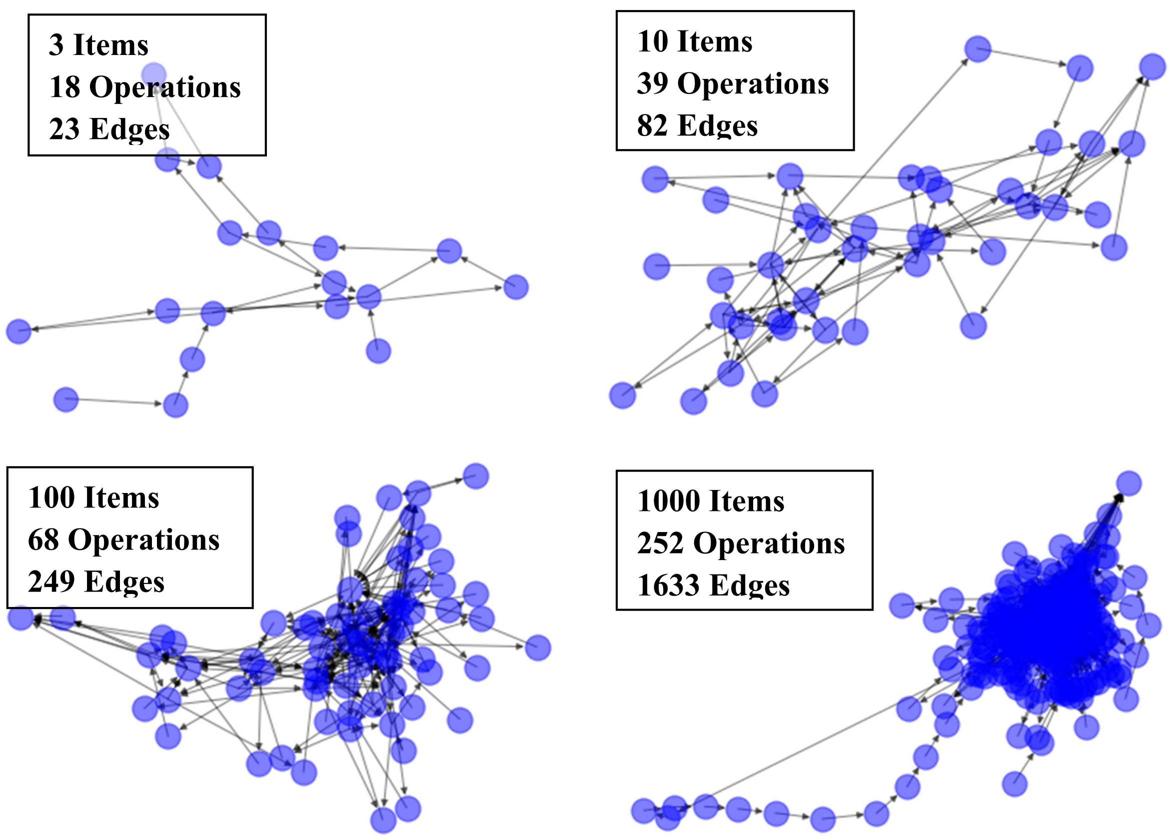

Figure 5: Network graph of 3 to 1000 items moving through the factory

Unique Cycle Times

With operations in the factory being drastically different from each other, it makes sense that their cycle times are also different. Furthermore, the high-mixed nature of circuit card sizes, component counts, and component makeups make their cycle times vary within a given operation. Since CCA has over 500 distinctive operations and approximately 7000 unique parts with different cycle times, it can be reasoned that there are approximately 3.5M cycle times to account for. Since CT is one of two factors necessary to calculate capacity required, having accurate CTs is important in capacity analysis. Accurate manual calculation of millions of cycle times would prove difficult.

Factory Size and Operation Types

Raytheon's CCA factory in Andover, Massachusetts is Raytheon's largest factory and the largest DOD CCA manufacturer in the world. With this size of factory, many workcenters and operations operate on different schedules and have different resource needs. For example, highly utilized operations such as SMT normally operate three shifts a day and weekends. On the other hand, less utilized operations may only operate two shifts a day with no weekends. Additionally, operations such as SMT are machine-driven while other operations such as assembly are labor driven. Furthermore, they all require a varying number of workers and machines. While SMT has seven assembly lines of machines with multiple workers on each line, some operations have a single worker at each station.

CH. 3 LITERATURE REVIEW

Many recent studies have shown the power of mathematical programs for capacity planning and the ability of machine learning methods to predict production metrics such as cycle time; however, embedding machine learning into mathematical programs for capacity planning has not been discussed. This chapter reviews relevant papers and topics associated with the two separately, while the remaining chapters bring the two together.

3.1 MATHEMATICAL MODELS FOR CAPACITY PLANNING

Mathematical programming such as LP models and MILP models are one of the primary methods used in capacity planning, especially now that solving them is relatively easy with today's technology. [19] Leachmen, Carmon, Bermon, and numerous others have studied the use of LPs for capacity planning. [18] Furthermore, Eickemeyer, Florensa, Zhou, and many others have studied the use of a MILP for capacity planning. [24] [25] [26] LPs can be solved quickly with solvers such as Clp and Gurobi, while MILPs can be solved with solvers such as Cbc and Gurobi. All of which are available for use with Julia/Jump. [27]

Mixed-integer nonlinear programs (MINLP) have also been studied by authors such as Zhou, Kristianto, and Chan. In order to solve these problems, various methods are implemented. Zhou and Li reformulated and approximated the MINLP as a MILP in order to solve. [28] Kristianto and Gunasekaran use branch and bound algorithms to solve. [29] While Chan *et al.* use a genetic algorithm to solve. [30] Other methods to solve these types of problems are the Jump Nonlinear Integer Program (Juniper) solver, which uses a nonlinear branch and bound heuristic technique, and the Global Optimization for Mixed Integer Programs with Nonlinear Equations (Alpine) solver, which uses an adaptive, piecewise convexification technique. [31] [32] Both packages are available for use with Julia/Jump. [27]

As discussed in 2.5 STRATEGIC CAPACITY PLANNING, static capacity planning models are inadequate for our strategic capacity planning problem and simulation models require a large amount of data and computing power. Due to these reasons, we use the commonly practiced mathematical programming models in this thesis.

3.2 MACHINE LEARNING FOR CYCLE TIME PREDICTION

Predicting cycle time for an item being produced in a factory has been long studied to improve factory performance. In complex manufacturing processes such as semiconductor manufacturing, hundreds of operations exist in a factory and multiple products are manufactured. This is similar to the complexity of Raytheon's CCA factory as described in 2.7 KEY CHALLENGES AT RAYTHEON'S CCA FACTORY. The flow of products in these types of factories follows in complexity as many products following different routes are utilizing the same resources. While Little's law works well in completely linear processes, its ability to accurately predict in these environments is limited.

Simulations are commonly studied and used for cycle time prediction as described in Chung, Wood, Kim, and Atherton; however, a simulation model for a highly complex factory such as CCA requires a vast amount of data and computing resources. [33] [34] [35] [36] [37] Simpler approaches such as statistical analysis and regression models have also been used for cycle time prediction as shown in Raddon and Enns; however, while these simpler methods provide great interpretability, they do not always provide the most accurate predictions in this type of factory. [37] [38] [39] A combination of these methods has also been studied by Kaplan and Liao. [34] [40] [41]

Backus *et al.* propose a machine learning approach to predict lot cycle times, stating that it provides a middle of the road approach between simpler models and full-scale simulations. In their paper, they show that historical data is able to be used to learn a predictive model for cycle time. From both measured and calculated metrics, data mining algorithms in conjunction with machine learning models such as k-nearest neighbor, regression trees, and neural networks were shown to provide useful CT predictions. It was shown that CT prediction for lots could be obtained from similar lots that had already completed production and that regression trees provided the best results due to their ability to handle both categorical and numerical features well.

Other authors have also studied the use of machine learning for prediction of production times. Wang *et al.* use machine learning and big data analytics to forecast cycle time due to having higher accuracy than linear regression techniques. [42] [43] Meidan *et al.* suggest data-driven methodologies to identify key features and predict CT using neural networks, decision trees, and other methods. [44] Tirkel also predicts CT of individual operations using decision trees and neural networks. [45] Furthermore, Rosalina compares various machine learning algorithms such as decisions trees, support vector machines, and neural networks in a manufacturing production process. [46] Lastly, Lingitz *et al.* tests linear regression, Ridge regression, LASSO regression, regression trees, boosted regression trees, random forests, support vector machines, k-nearest neighbor models, and artificial neural networks to predict lead time, finding that random forests and boosted regression trees performed the best. Lingitz *et al.* continue to point out that WIP appeared to be the most important feature followed by features such as historical lead times and arrival weekday. [47]

Following in the footsteps of this research and with the abundance of data available in Raytheon’s Data Warehouse, we believed that CT at Raytheon’s CCA factory could also be predicted using machine learning methods.

CH. 4 CAPACITY UTILIZATION AND OEE ANALYSIS

The first step in strategic capacity planning is developing capacity utilization framework to calculate utilization. In order for the capacity tool to have the greatest benefit to the company, it needs to be able to consistently calculate factory capacity without extensive manual effort. In order to minimize manual effort, the tool must rely primarily on continuously updating data and algorithms to interpret and analyze that data. This chapter first goes over the available data at Raytheon and calculates features needed for capacity analysis. These features are then used to calculate capacity utilization graphs and OEE for each operation. Together, they provide a high-level strategic tool to evaluate capacity in its current state.

4.1 OVERVIEW OF AVAILABLE DATA FOR CAPACITY ANALYSIS

Compared to many manufacturing companies, Raytheon does a phenomenal job at database maintenance and offers a data-rich environment. It utilizes a data warehouse structure to house all of its data from manufacturing transactions to the backend of its ERP data. This data is employed in a star schema model so that data can be accessed in an organized fashion through fact and dimension tables. Because of this, retrieval can be done remotely with an automated SQL pull for easy data analysis. Transactional data and industrial internet of things (IIoT) data are both stored in this warehouse along with Systems, Applications and Products in Data processing (SAP) ERP data. The only data used in this thesis that is not accessed through SQL and the Data Warehouse is projected demand data, which is calculated through a supply chain planning software called Kinaxis.

The following overview discusses further some of the data available—transactional, IIoT, known demand, and predicted demand.

4.1.1 Transactional Data

Manufacturing companies typically maintain transactional data for each lot that goes through production. A typical table of a company with multiple facilities, operations, and products contains information to identify the production facility, current operation, employee on duty, production route, and timestamps for each action taken. There are also other variables that describe the lot, such as the specific type of product and quantity. Each lot often contains thousands of rows of data in more than ten different categories, resulting in over 10000 variables to describe how that lot moved through the factory. A simplified example that displays a day's worth of transactional data for a specific operation and site is shown in Table 1.

SITE	ACTION	DATE_TIME	PRODUCT_TYPE	PRODUCT_REV	LOT_NUMBER	ROUTE	OPERATION_NUMBER	QUANTITY	EMPLOYEE_NUMBER
1	START	2/10/2014 23:30	AAAA	4	AAAA0001	AAAA-0004	1234	30	00135
1	SIGNOFF	2/11/2014 2:00	AAAA	4	AAAA0001	AAAA-0004	1234	30	00135
1	START	2/11/2014 3:00	AAAA	4	AAAA0001	AAAA-0004	1234	30	00135
1	COMPLETE	2/11/2014 4:00	AAAA	4	AAAA0001	AAAA-0004	1234	30	00135
1	START	2/11/2014 5:00	BBBB	1	BBBB0001	BBBB-0001	1234	50	00135
1	SIGNOFF	2/11/2014 6:00	BBBB	1	BBBB0001	BBBB-0001	1234	50	00135
1	START	2/11/2014 6:00	BBBB	1	BBBB0001	BBBB-0001	1234	50	02888
1	COMPLETE	2/11/2014 9:00	BBBB	1	BBBB0001	BBBB-0001	1234	50	02888
1	START	2/11/2014 9:30	BBBB	1	BBBB0002	BBBB-0001	1234	100	02888
1	SIGNOFF	2/11/2014 11:00	BBBB	1	BBBB0002	BBBB-0001	1234	100	02888
1	START	2/11/2014 12:00	BBBB	1	BBBB0002	BBBB-0001	1234	100	02888
1	SIGNOFF	2/11/2014 14:30	BBBB	1	BBBB0002	BBBB-0001	1234	100	02888
1	START	2/11/2014 15:00	BBBB	1	BBBB0002	BBBB-0001	1234	100	09999
1	COMPLETE	2/11/2014 19:00	BBBB	1	BBBB0002	BBBB-0001	1234	100	09999
1	START	2/11/2014 20:00	BBBB	1	BBBB0003	BBBB-0001	1234	100	09999
1	SIGNOFF	2/11/2014 21:30	BBBB	1	BBBB0003	BBBB-0001	1234	100	09999
1	START	2/11/2014 21:30	BBBB	1	BBBB0003	BBBB-0001	1234	100	00135

Table 1: An example of typical transactional data a manufacturing factory stores

4.1.2 Industrial Internet of Things (IIoT) Data

The Industrial Internet of Things (IIoT) refers to the vast amount of industrial connected devices filled with sensors collecting and sharing data. While not all machines have this type of functionality, some of the high volume lines at Raytheon offer these types of data. The IIoT data at Raytheon follows a similar structure to the transactional data; however, since each observation is automatically stored by a machine instead of manually by an operator, the quality of the data is much higher. This enables much more detailed information to be stored such as component placement and items left on a feeder. As a result of automated storage and more data, many more metrics can be calculated at with greater accuracy.

The following are a few of the additional pieces of information that can be pulled from these data:

- Components Remaining on Feeder
- Specific Components being Placed on Item
- Exact Time between Items
- Machine Error Codes
- Material Shortage Indication
- Item Recipe
- Specific Machine Start/Finish of a Process
- Timestamps for all

4.1.3 Confirmed Demand Data

Raytheon uses SAP as its ERP software. When looking at a transaction such as MD04 (Display Stock/Requirements Situation) in the software, one can see a listing of all planned consumption and all planned receipts of an item over time. This enables us to see when material is available and when it must be used to meet demand or in other orders. SAP calculates production dates (lead time scheduling) using routing and other master data inputted into the system. [48] These routings can be pulled from the data warehouse and used in conjunction with predicted CTs, which are discussed in CH. 6 MACHINE LEARNING TO PREDICT CYCLE TIME. To find the dates each operation needs to be complete in order to meet demand, we combine the two along with the demand due date. An example using the general route from Figure 3 is shown in Table 2 for operation start and end dates to meet demand just in time (with buffer).

Routing: Generic Lot	CT (days)	Start Date	End Date
Kitting, Label, and Baking	0	3-Aug	3-Aug
SMT Feeder Setup	0	3-Aug	3-Aug
SMT	1	3-Aug	4-Aug
XRAY Touch Up	1	4-Aug	5-Aug
Assembly	2	5-Aug	7-Aug
Test	3	7-Aug	10-Aug
Coating	1	10-Aug	11-Aug
Inspection	2	11-Aug	13-Aug
Due Date	20-Aug		
Buffer Time (days)	7		

Table 2: Routing paired with CT and demand due date to calculate operation start and end dates for just in time production

Like the transactional and IIoT data, this routing and demand data can be directly pulled from the data warehouse to enable the tool to remain automated. The CT data could also be pulled from this warehouse, but it was found to be not as accurate as the model we implement in CH. 6 MACHINE LEARNING TO PREDICT CYCLE TIME.

4.1.4 Predicted Demand Data

Predicted demand can be found in Kinaxis, a powerful supply chain planning software. Kinaxis predicts demand by conditioning sales data, collecting demand inputs, and then generating a statistical forecast for sales and operations planning (S&OP). [49] Due to this calculation being done within the software, a direct SQL query was not possible at the time of research. Nevertheless, this calculation can be automated within Kinaxis at specified intervals to output a report of predicted demand in the form of an Excel file. With this Excel file automatically updated and stored on the Raytheon shared server, a simple line of Python code can pull the data. For the purpose of this thesis, this predicted demand will be assumed to be correct; however, we acknowledge this limitation in 7.3 MODEL LIMITATIONS.

4.2 CAPACITY UTILIZATION CALCULATION

From Equation 8, we know that we need four variables to calculate capacity utilization of an operation—demand at that operation, processing time for that demand, planned production time of the operation, and the number of resources. In order to look at capacity utilization over time, we need to add in a time component:

$$CU_{ot} = \frac{\sum_{i=1}^{I} d_{iot} pt_{iot}}{ppt_{ot} r_{ot}}$$

The following section will describe where each of these variables come from and how they are calculated.

Operation Level Demand

As discussed in 4.1 OVERVIEW OF AVAILABLE DATA FOR CAPACITY ANALYSIS, demand is a combination of confirmed demand from SAP and predicted demand. This data can be directly pulled from the data. For finished item demand at the last operation O, we set it equal to confirmed demand (dc) plus predicted demand (dp) for that item i in time period t:

$$d_{iot} = dc_{it} + dp_{it} \; \forall i, t; o = O$$

Equation 13

Demand at the operational level, however, requires CT. We can use a simple CT metric stored in the warehouse or the one predicted in CH. 6 MACHINE LEARNING TO PREDICT CYCLE TIME. Either way, demand at a previous operation is defined by the CT of the current operation:

$$d_{i,o-1,t-CT_{iot}} = d_{iot} \; \forall i, o, t$$

For clarification, let us assume that we need 10 units of item one complete in time period 5. Complete means that o=O, the last operation. That is $d_{1,O,5} = 10$. If CT of item i at operation O in time period 5 is equal to 2 time periods, then $CT_{iot} = 2$ and $d_{1,O-1,3} = 10$. This means that we need to complete 10 units of item 1 in time period 3 at operation O-1. Note that these demand definitions assume that all demand is met just in time.

Cycle Time, Queue Time, and Processing Time

From transactional and IIoT data, numerous common manufacturing measures can be calculated at the lot level. One commonly known law is Little's law, which uses cycle time (CT), work-in-progress (WIP), and throughput (TP) as described in 2.4 MANUFACTURING CAPACITY. A lot's cycle time for an operation is equal to the time it waited in queue (QT) plus the production time (PT). Queue time is calculated by finding the timestamp of first starting action (FSA) of the current operation minus the timestamp of last ending action (LEA) of the previous operation. Production time is calculated as the last ending action of the current operation minus the first starting action of the current operation.

We define CT_{lot} as the cycle time of lot l at operation o in time period t. Because lots are only made up of a single item, item i is used instead of lot l when looking at the aggregate level. Lot is broken out because of the high-mix nature of the factory where cycle time is greatly affected by the item being processed. Operation is broken out because of the vast difference in operations and therefore cycle times. Lastly, time period is broken out because cycle time can improve over time as workers become more familiar with an operation and/or item. Cycle time can also change over time depending on how much WIP is on the factory floor, which is discussed further in 6.2 FEATURE GENERATION. We define QT_{lot} as the queue time and PT_{lot} as the production time of lot l at operation o in time period t.

Together we get:

$$CT_{lot} = QT_{lot} + PT_{lot}$$

$$QT_{lot} = FSA_{lot} - LEA_{l,o-1,t}$$

$$PT_{lot} = LEA_{lot} - FSA_{lot}$$

Equation 14, Equation 15, Equation 16

where QT_{lot} for the first operation in a route is equal to zero. The averages for an item/operation pair can then be calculated at per item level since lots are made up of a single item:

$$CT_{io} = \frac{\sum_{t=1}^{T}\sum_{l=1}^{L} CT_{lot}}{L+T} \forall o; l \in I$$
$$QT_{io} = \frac{\sum_{t=1}^{T}\sum_{l=1}^{L} QT_{lot}}{L+T} \forall o; l \in I$$
$$PT_{io} = \frac{\sum_{t=1}^{T}\sum_{l=1}^{L} PT_{lot}}{L+T} \forall o; l \in I$$

Equation 17,Equation 18,Equation 19

From Equation 4, we know that $pt_{io} = \frac{PT_{io}}{QTY_{io}}$, but where $QTY_{io} = \frac{\sum_{t=1}^{T}\sum_{l=1}^{L} QTY_{lot}}{L+T} \forall o; l \in I$.

Capacity Available

We know that capacity available is equal to the planned production time of each resource multiplied by the number of resources. We, therefore, need both in order to calculate it. Planned production time can be calculated by gathering its three components—number of shifts, number of workdays in a month, and length of shifts. Raytheon's master schedule offers us information into the number of workdays in a given month, however, each operation works differently due primarily to capacity issues. Operations with high demand usually work three shifts and weekends, while operations with low demand may only work two shifts and no weekends.

Since all operations work on a similar three-shift schedule with starting and ending times defined, shifts can be inferred through data analysis for each operation based on the transactional data that happens during those time intervals. It can also more accurately be pulled from the data warehouse or by asking each operation manager their shift schedule. Length of shifts on the other hand is assumed to be constant and account for breaks. For example, an 8-hour shift that is known to give 30 minutes for lunch, and two 15 minute breaks, would result in defining shift length at a constant 7 hours.

Number of resources can also be inferred from the transactional data. While not shown in Table 1, a column of resources with unique names exists. For example, the SMT operation has seven unique resources in this column for the seven lines in the factory. While this is accurate for some resources, it is not accurate for all. Therefore, obtaining the information from process engineers yields better results.

4.3 CALCULATING OEE WITH CURRENT DATA SOURCES

OEE is a metric that includes Availability, Performance, and Quality as shown in Equation 9. In order to calculate it, we must first calculate its three parts. In this section, we will go through example calculations that use the transactional and IIoT data to calculate each part individually. We will then combine the three parts at the end to calculate OEE.

Availability

Availability at the operational level can be calculated using the transactional data from Table 1 Since the planned production time of the operation is known from other sources, we can use the transactional data to see when a lot is being worked on and when a lot is not being worked on. Planned production of a specific operation in this case is $ppt_{ot} = sh_{ot}wd_{ot}ls_{ot} = 3 * 1 * 7 = 21\ hours$ since we are looking at single resource operation 1234 in a single day, which we will assume is a single time period (1).

Run time is can be found through forming pairs (P) of transactional data, each with a starting action (START) and an ending action (SIGNOFF, COMPLETE, etc.). The summation of the difference of all pairs for a given operation provides the time a lot was being worked on in a given time period is the run time (rt).

$$rt_{ot} = \sum_{p=0}^{P} (ea_{pot} - sa_{pot})\ \forall o, t$$

Equation 20

where sa_{pot} is the datetime of a starting action of a start/end pair p where operation o is being run in time period t, and ea_{pot} is the datetime of an ending action of a start/end pair p where operation o is being run in time period t.

Continuing to use the case presented in Table 1, we are looking at operation 1234 in a single day. Through data transformation, we can make nine pairs of the starting and ending actions.

Pair	Starting Action	Ending Action	Time Difference (hours)
0	2/11/2014 0:00	2/11/2014 2:00	2
1	2/11/2014 3:00	2/11/2014 4:00	1
2	2/11/2014 5:00	2/11/2014 6:00	1
3	2/11/2014 6:00	2/11/2014 9:00	3
4	2/11/2014 9:30	2/11/2014 11:00	1.5
5	2/11/2014 12:00	2/11/2014 14:30	2.5
6	2/11/2014 15:00	2/11/2014 19:00	4
7	2/11/2014 20:00	2/11/2014 21:30	1.5
8	2/11/2014 21:30	2/12/2014 0:00	2.5
		Total	19

Table 3: Calculation of time difference for run time calculation

We can then calculate runtime for this operation and time period: $rt_{1234,1} = \sum_{p=0}^{8}(ea_{1234,1} - sa_{1234,1}) = 19\ hours$. Since we know availability is equal to the ratio of runtime to planned production time, we can see that availability in this example is equal to 19/21 or 90.5%. This method can be completed for each operation and time period so that availability can be known across the company. A visual depiction of this example is shown in Figure 6.

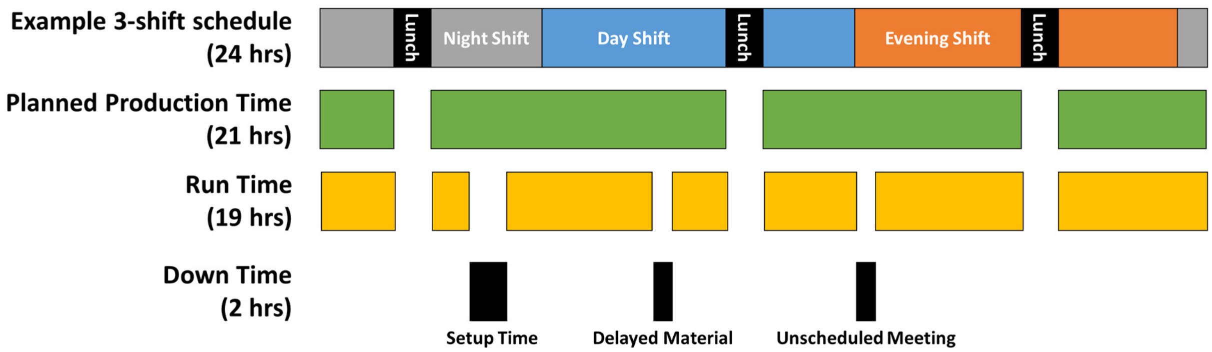

Figure 6: A visual depiction of Availability calculation for OEE

Performance

With CT for each lot, operation, and time period calculated, we can then start to calculate performance at the operational level. First, we need to differentiate runtime at a per lot level by defining $rt_{lot} = \sum_{p=0}^{P}(ea_{pot} - sa_{pot})\ \forall l, o, t$ so that,

$$rt_{ot} = \sum_{l=0}^{L} rt_{lot}\ \forall o, t$$

Equation 21

Total parts are already at a per lot level in the raw transactional data under Quantity. Lastly, ideal process time, which is the theoretical minimum time to process one item, can be defined as the minimum average process time for a given item, operation, time period combination. At the lot level, each lot ideal cycle time would be equal to the ideal cycle time for the item in the given lot.

For example, we can calculate the following table from the transactional data:

LOT_NUMBER	QUANTITY	OPERATION	ITEM	CURR_OP_START	PT (hrs)	PT_avg (hrs/item)
BBBB0001	50	1234	BBBB	2/11/2014 5:00	4	0.08
BBBB0002	100	1234	BBBB	2/11/2014 9:30	8	0.08
BBBB0003	100	1234	BBBB	2/11/2014 20:00	9	0.09
BBBB0004	100	1234	BBBB	2/12/2014 5:00	10	0.1

Table 4: Calculation of ideal processing time for use in Performance calculation of OEE

$$\begin{aligned} Ideal\ Process\ Time &= \min(PT_{avg_{lot}}) \\ &= \min\left(\left[PT_{avg_{1,1234,1}}, PT_{avg_{2,1234,1}}, PT_{avg_{3,1234,1}}, PT_{avg_{4,1234,1}}\right]\right) \\ &= \min([0.08,0.08,0.09,0.1]) = .08\ hrs/item \end{aligned}$$

Assuming the above four lots were processed in our defined time period, the minimum average process time for item BBBB in operation 1234 is .08 hrs/item. Therefore, we can assume the ideal process time for any lot of BBBB items in this time period is .08 hrs/item and plug this back into the above table to get:

LOT_NUMBER	QUANTITY	OPERATION	ITEM	CURR_OP_START	PT (hrs)	PT_avg (hrs/item)	Ideal PT
BBBB0001	50	1234	BBBB	2/11/2014 5:00	4	0.08	4
BBBB0002	100	1234	BBBB	2/11/2014 9:30	8	0.08	8
BBBB0003	100	1234	BBBB	2/11/2014 20:00	9	0.09	8
BBBB0004	100	1234	BBBB	2/12/2014 5:00	9	0.09	8
					30	Total	28

Table 5: Continued calculation of ideal processing time for calculation of Performance

Ideally, these four lots would be completed in 28 hours of processing time; however, it actually took 30 hours of processing time. Therefore, performance is 28/30 or 93%. We can follow this same process for all lots in a given operation and time period to get an overall performance for an operation.

Quality

Quality, as described by OEE, is the same as what many companies call first pass yield (FPY). Good parts are parts that successfully pass through an operation the first time without the need for rework and total parts are the parts that begin an operation. Many factories, like the one in this study, keep track of all of this information in a table such as the one in Table 6.

LOT_NUMBER	QUANTITY	OPERATION	ITEM	CURR_OP_START	REWORK	REJECT
BBBB0001	50	1234	BBBB	2/11/2014 5:00	1	0
BBBB0002	100	1234	BBBB	2/11/2014 9:30	0	0
BBBB0003	100	1234	BBBB	2/11/2014 20:00	0	0
BBBB0004	100	1234	BBBB	2/12/2014 5:00	0	1

Table 6: Quality data of different lots including rework and reject items

From this table, quality can be found by calculating removing reworked and rejected parts from the quantity to get good parts and using the original quantity as total parts.

LOT_NUMBER	QUANTITY	OPERATION	ITEM	CURR_OP_START	REWORK	REJECT	GOOD PARTS
BBBB0001	50	1234	BBBB	2/11/2014 5:00	1	0	49
BBBB0002	100	1234	BBBB	2/11/2014 9:30	0	0	100
BBBB0003	100	1234	BBBB	2/11/2014 20:00	0	0	100
BBBB0004	100	1234	BBBB	2/12/2014 5:00	0	1	99
	350			**Total**			348

Table 7: Calculation of good parts to compare against total parts for Quality calculation

We, therefore, see that 348 good parts were produced out of 350 total parts. Resulting in a quality of 348/350 or 99%

OEE

OEE is, therefore, the product of these three parts: $OEE = A * P * Q = 90.5\% * 93\% * 99\% =$ 83.32%. At 83.32% this operation would be considered to be operating at world-class OEE standards for this particular day. For us, we care to see how OEE is doing on average in a given operation and therefore look at it over multiple days. If desired, this metric could be tracked for performance monitoring.

4.4 STATIC CAPACITY UTILIZATION AND OEE RESULTS

After combining all data sources using SQL queries embedded in executable Python scripts that also conduct calculations, we are able to visually look at capacity utilization over time through programs such as Tableau. This overall process is depicted in Figure 7.

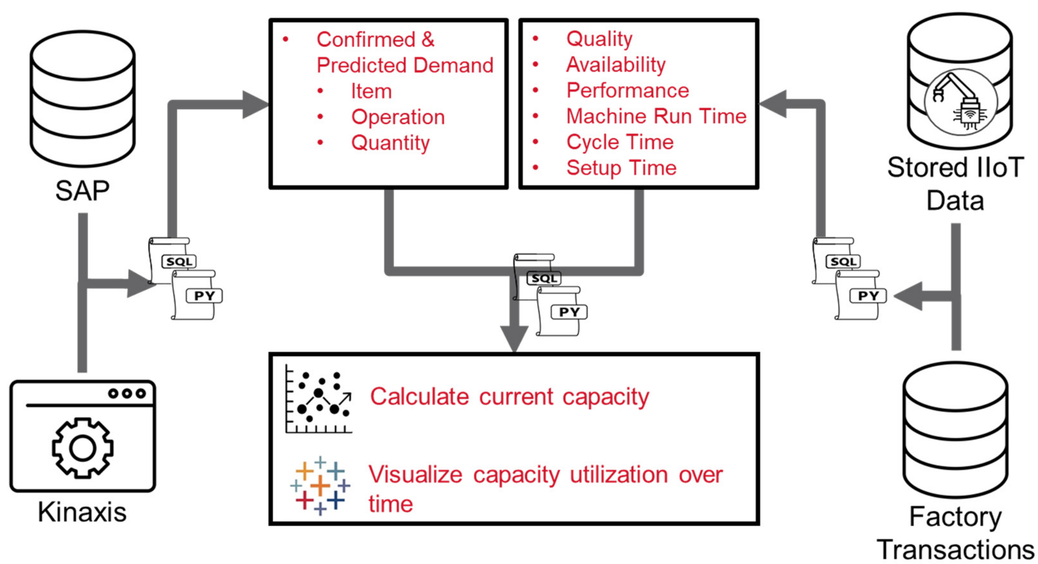

Figure 7: Visual representation of connecting data sources to calculate capacity utilization and OEE

Capacity utilization and OEE can then be easily viewed at the operational level as shown in Figure 8.

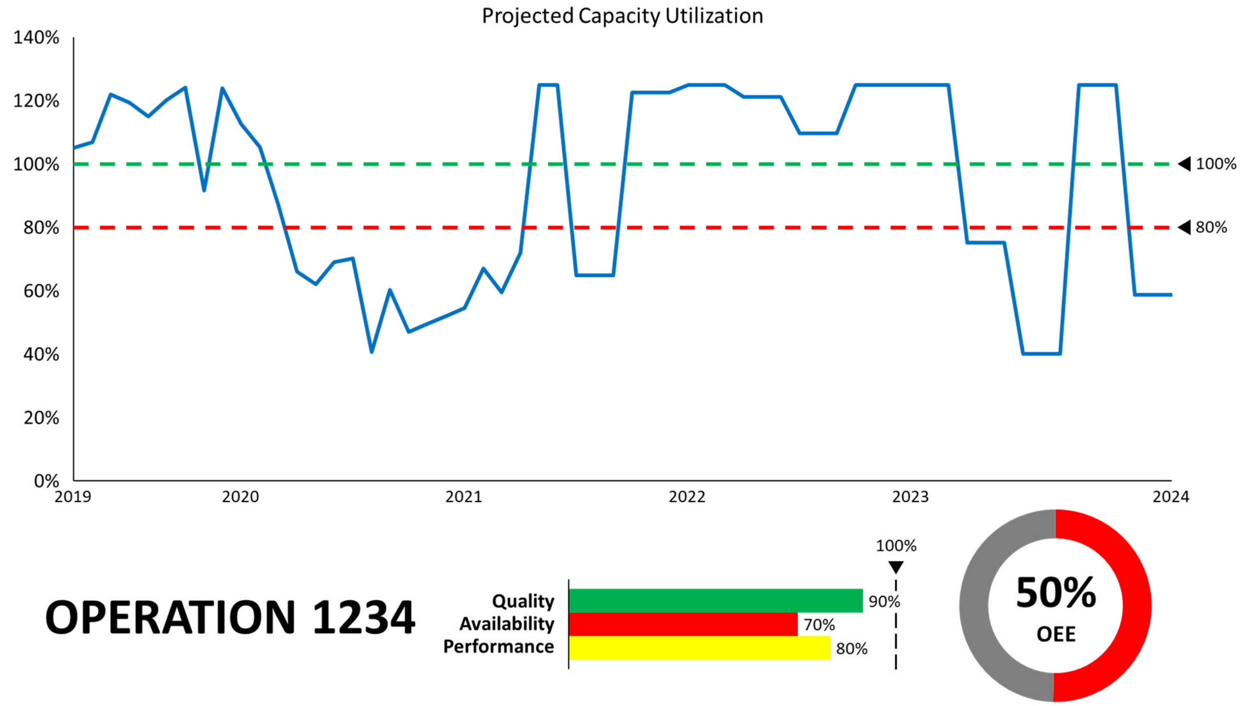

Figure 8: Capacity utilization and OEE visualization of a single operation

The question then becomes, so what? What can we do with this information? This is where strategic capacity planning comes in. As one can see from Figure 8, this operation will be over capacity if it maintains its current schedule of production as deemed by SAP with its current resource configuration. As discussed in 2.4 MANUFACTURING CAPACITY, we know that we have three options to increase capacity available or reduce capacity required.

1. Increase resources at the operation (maintain speed but increase time available or resources)

   a. Add Shift:

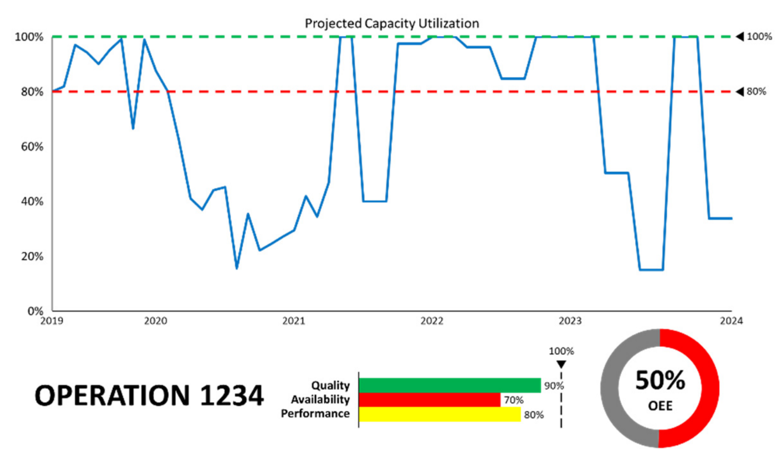

Figure 9: Capacity utilization result of adding a shift to operation 1234

b. Add Machines if Operation is Machine Driven:

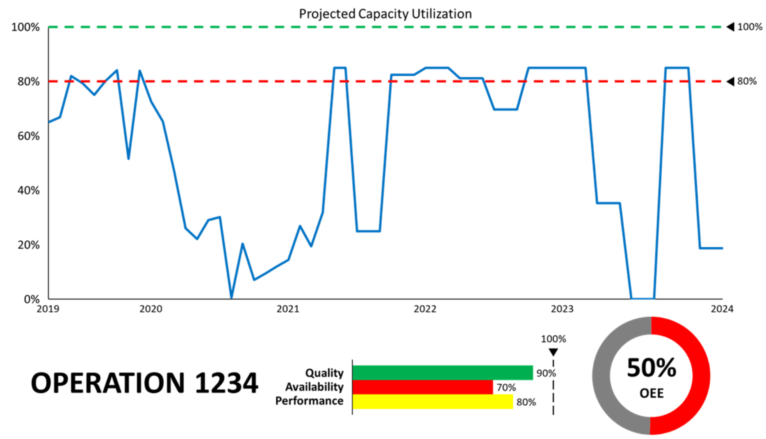

Figure 10: Capacity utilization result of adding a machine to operation 1234

c. Add Workers if Operation is Labor Driven: N/A

2. Invest in increasing OEE:

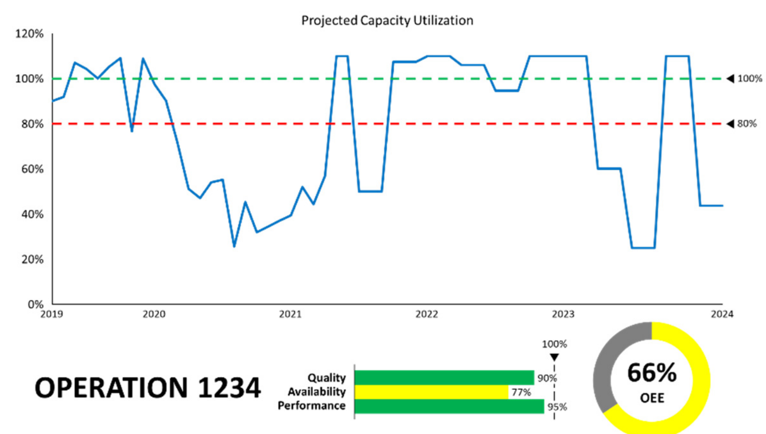

Figure 11: Capacity utilization result of increasing availability and performance

3. Shift demand at operation:

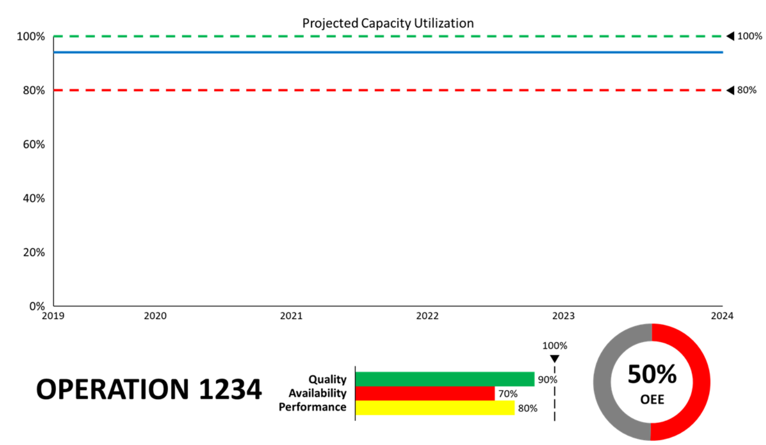

Figure 12: Capacity utilization result of completely flattening demand

As one can see, adding a shift, adding a machine, or shifting demand to be flat enables this specific operation to maintain itself below 100% capacity utilization. Adding a shift is relatively simple, but it requires hiring more workers and it barely keeps us below 100%. Adding a machine is extremely capital intensive, but it keeps our utilization near 80%. Investing in improving OEE does not do much to improve utilization at this operation, but may be useful to do in conjunction with another option. Lastly, shifting demand enables the operation to stay slightly below 100%, but shifting its demand shifts demand for many other operations. When your operations are as connected as they are in this factory as discussed in 2.7 KEY CHALLENGES AT RAYTHEON'S CCA FACTORY, we have to be sure to understand the effect we have across all operations and items. Most likely, the optimal solution requires a combination of the above options.

Each option to improve capacity utilization comes with certain costs and risks that must be weighed and decided upon as discussed in 2.5 STRATEGIC CAPACITY PLANNING. In order to do this, we have implemented an optimization model that reduces the cost associated with maintaining capacity utilization below 100% while maintaining demand in order to maximize ROI of our strategic capacity planning decisions. The models used are discussed in CH. 5 OPTIMIZATION MODELS FOR STRATEGIC CAPACITY PLANNING.

CH. 5 OPTIMIZATION MODELS FOR STRATEGIC CAPACITY PLANNING

Our methodology for developing an optimization model that works well for strategic capacity planning is to start simple and add complexity to match key decisions Raytheon needs in capacity planning. We initially start with a simple LP and eventually work our way to a mixed-integer program (MIP) due to not being able to produce partial items. An example problem is shown with this MIP to ensure functionality prior to continuing to add complexity to our model. As we continue to add variables necessary for optimizing capacity planning, our model becomes nonlinear. Our final model is a mixed-integer nonlinear program (MINLP) with a machine learning constraint. After our final model is formulated, we then dive into the mathematics and programs used to solve it.

5.1 LINEAR OPTIMIZATION MODELS

5.1.1 Simple Linear Model

Let us first look at a simplified linear programming model developed by Stephen Graves for production planning where we want to meet demand with our current capacity for the lowest production and inventory costs possible. We assume that all items (I) have independent demand, resources (K) are shared, time periods (T) are large such as monthly, and costs are linear. We define a_{ik} as the amount of resource k required per unit of production of item i; b_{kt} as the amount of resource k available in period t; d_{it} as the demand for item i in period t; cpr_{it} as the unit cost of production for item i in time period t; and cin_{it} as the unit cost for holding item i in time period t in inventory. [50] Our model then looks like this:

Decision Variables

pr_{it} production of item i in time period t

in_{it} inventory of item i in time period t

Objective Function

$$\min \sum_{t=1}^{T} \sum_{i=1}^{I} cpr_{it} pr_{it} + cin_{it} in_{it}$$

Constraints

Demand:

$$in_{i,t-1} + pr_{it} - in_{it} \geq d_{it} \quad \forall i, t$$

Capacity:

$$\sum_{i=1}^{I} a_{ik} pr_{it} \leq b_{kt} \quad \forall k, t$$

Non-negativity:

$$pr_{it}, in_{it} \geq 0 \quad \forall i, t$$

Equation 22 [50]

The objective function looks to minimize the production costs and inventory holding costs. The demand constraint balances production and inventory to meet demand. In any time period, the supply for an item is the production in the current period plus the inventory from the previous period. This supply is equal to the demand and the leftover inventory for the item in that period. Since inventory also shows up in the non-negativy constraint, demand is required to be satisfied for each item in each period. Backorders and lost sales are not allowed. The capacity constraint limits the production of items by the availability of shared resources, where production of one unit of item i requires a_{ik} units of resource k. Typical resources that are constrained are labor or equipment related. [50]

5.1.2 Simple Linear Model with Workforce Constraint

In the simplified model, we assume two things that are most likely not true in a company—linear production cost and fixed resource levels. While we will continue to assume a linear production cost, we will look into adjusting resource levels to change capacity and meet demand. Initially, let us look at a problem where work force is the only resource. We will define a_i as the level of work force required per unit of production of item i; cwf_t as the cost per employee in time period t; ch_t as the hiring cost in time period t; and cf_t as the firing cost in time period t. [50] Now our model looks like this:

Decision Variables

pr_{it} production of item i in time period t

in_{it} inventory of item i in time period t

wf_t work force level in time period t

h_t change to work force level by hiring in time period t

f_t change to work force level by firing in time period t

Objective Function

$$\min \sum_{t=1}^{T} \sum_{i=1}^{I} (cpr_{it} pr_{it} + cin_{it} in_{it}) + \sum_{t=1}^{T} (cwf_t wf_t + ch_t h_t + cf_t f_t)$$

Constraints

Demand:

$$in_{i,t-1} + pr_{it} - in_{it} \geq d_{it} \quad \forall i, t$$

Workforce Capacity:

$$\sum_{i=1}^{I} a_i pr_{it} - wf_t \leq 0 \quad \forall t$$

Workforce Balance:

$$wf_{t-1} + h_t - f_t - wf_t = 0 \quad \forall t$$

Non-negativity:

$$pf_{it}, in_{it}, wf_t, h_t, f_t \geq 0 \quad \forall i, t$$

Equation 23 [50]

As one can see, the variable, hiring, and firing costs of the workforce were added to the objective function. Variable costs correlate to costs such as wages; hiring costs correlate to costs such as finding workers and marketing; and firing costs correlate to costs such as outplacement and severance. The demand constraint remains the same; however, the capacity constraint reflects that workforce is the only resource constraint. Additionally, a workforce balance constraint was added for continuity of workforce across time periods. [50]

5.1.3 Simple Linear Model with Workforce and Machine Constraints

While labor is one constrained resource at factories, machines are another we need to incorporate into our model. We will define z_i as the amount of machines required per unit of production of item i; cm_t as the unit cost of a machine in time period t; and cb_t as the cost to buy or build a new machine in time period t. [50] Now our model looks like this:

Decision Variables

pr_{it} production of item i in time period t

in_{it} inventory of item i in time period t

wf_t work force level in time period t

h_t change to work force level by hiring in time period t

f_t change to work force level by firing in time period t

m_t machine level in time period t

b_t change in machine level by buying/building in time period t

Objective Function

$$\min \sum_{t=1}^{T} \sum_{i=1}^{I} (cpr_{it} pr_{it} + cin_{it} in_{it}) + \sum_{t=1}^{T} (cwf_t wf_t + ch_t h_t + cf_t f_t) + \sum_{t=1}^{T} (cm_t m_t + cb_t b_t)$$

Constraints

Demand:

$$in_{i,t-1} + pr_{it} - in_{it} \geq d_{it} \quad \forall i, t$$

Workforce Capacity:

$$\sum_{i=1}^{I} a_i pr_{it} - wf_t \leq 0 \quad \forall t$$

Machine Capacity:

$$\sum_{i=1}^{I} z_i pr_{it} - m_t \leq 0 \quad \forall t$$

Workforce Balance:

$$wf_{t-1} + h_t - f_t - wf_t = 0 \quad \forall t$$

Machine ‘Balance’:

$$m_{t-1} + b_t - m_t = 0 \quad \forall t$$

Non-negativity:

$$pr_{it}, in_{it}, wf_t, h_t, f_t, m_t, b_t \geq 0 \quad \forall i, t$$

As one can see, the variable and building costs of the machines were added to the objective function. Variable costs correlate to costs to run the machines and building costs correlate to the costs to buy or build, and install a new machine. The demand and workforce balance constraints remain the same; however, the machine capacity constraint reflects that machines are an additional constraint. Additionally, a machine ‘balance’ constraint was added for continuity of machines across time periods where it is assumed that machines can be added but not removed. [50]

5.2 WALKTHROUGH OF A SIMPLE CAPACITY PLANNING PROBLEM

The following section enables us to show the type of output the modeling in 5.1 LINEAR OPTIMIZATION MODELS can provide through a walkthrough without using proprietary company data. It is done at a small scale with all data provided to the reader for reproduction purposes. As a result of the walkthrough, the reader should gain a better understanding of the models involved in strategic capacity planning.

5.2.1 Gathering Data for the Model

Let us look at a simplified example of our larger problem. The time period we will consider is three days, which we will look at daily. First, we need to understand what items need to be made in the timeframe we are looking at—our demand. By looking at demand, we see that only three items need to be manufactured in the next three days—Item 1, Item 2, and Item 3. Our final product demand matrix can then be formed to look like this:

Actual Demand	Item 1	Item 2	Item 3
Month 1	20	0	10
Month 2	20	0	0
Month 3	20	60	10

This shows that 20 units of Item 1 need to be completed each day and 60 units of Item 2 need to be completed in only Day 3.

Next, we can get the path an item takes through the factory by looking at each item's router. By doing this, we can make a router matrix like this:

Factory Routes	Item 1	Item 2	Item 3
Operation A	1	1	0
Operation B	1	0	1
Operation C	1	1	1
Operation D	1	0	0
Operation E	1	1	1

This shows us that there are only five operations we need to be concerned about to manufacture these three items. It also shows that Item 1 must go through five operations to be complete, while Item 2 only has to go through every other operation.

Next, we can look further into these operations to see how many machines are being used in each. We make the assumption in this simple problem that all operations are machine constrained and not workforce constrained. If we want to increase capacity, we must add an additional machine. While adding a machine will require adding additional workforce, increasing workforce alone will not add capacity. We also make the assumption that all operations are operating on a single shift with seven hours of planned production and that no additional shifts or production time can be added. This assumption enables us to maintain linearity of our model for the time being. Our simplified machine resource matrix looks like this:

	Number of Shifts	Prod Hours	#Machines	Current Capacity (hrs)
Operation A	1	7	5	35
Operation B	1	7	2	14
Operation C	1	7	3	21
Operation D	1	7	1	7
Operation E	1	7	2	14

This shows that Operation A has seven hours of planned production time for each of its five machines for a total of 35 hours of available capacity for production.

Next, we begin to look into the processing time required per item. This can be inferred from historical data through normal and advanced analytical methods, which are discussed in 4.2 CAPACITY UTILIZATION CALCULATION. For now, we will assume that the processing time required per item at each operation is:

	Processing Time Required per Item (hrs)		
	Item 1	Item 2	Item 3
Operation A	0.5	0.75	0
Operation B	0.25	0	0.5
Operation C	0.5	0.75	0.5
Operation D	0.25	0	0
Operation E	0.25	0.25	0.5

This shows that one unit of Item 1 can be processed through Operation A in 0.5 hours. Not accounting for queue time, which we will assume is zero in this simplified scenario, Item 1 can go through the entire factory in 1.75 hours.

Next, we need to gather information on how much various things cost the company. We need to look into what it costs to manufacture each item at each operation—the variable cost of production; the cost to hold work-in-progress inventory of each item at each operation—inventory holding costs; the cost of running a machine—variable unit cost of machine resources; and the cost to add an additional machine at each operation. For variable cost of production, we can look at things such as the bill of materials and spread the cost over each operation based on processing time. If we do this, we might see a variable production cost matrix that looks like this:

	Variable Cost of Production ($)		
	Item 1	Item 2	Item 3
Operation A	28.57	42.86	0.00
Operation B	14.29	0.00	28.57
Operation C	28.57	42.86	28.57
Operation D	14.29	0.00	0.00
Operation E	14.29	14.29	28.57

Holding costs depend on the space in the factory, labor, prices of damaged goods, and other factors, but we will simply assume that holding costs are 1% of the cumulative cost of the item. [51] Our inventory holding cost matrix then looks like this:

Holding Costs ($)	Item 1	Item 2	Item 3
Operation A	0.29	0.43	0.00
Operation B	0.43	0.43	0.29
Operation C	0.71	0.86	0.57
Operation D	0.86	0.86	0.57
Operation E	1.00	1.00	0.86

The cost to run a machine depends primarily on the energy the machine uses, the labor required to run the machine, and the maintenance required for upkeep. We will assume that each machine requires one person to run it at a wage of $50 per hour and that it costs $25 per day, on average, to run the machine. Since employees will be working one eight-hour shift, the cost to run a machine is $425 per day it is in operation. Our machine resource matrix then looks like this:

	Machine Costs ($)
Operation A	425.00
Operation B	425.00
Operation C	425.00
Operation D	425.00
Operation E	425.00

Lastly, the cost to buy, build, and install a machine is very expensive relative to the aforementioned costs. One major assumption we make here, however, is that buying, building, and installing is instantaneous. Upon researching the costs to add an additional machine at each operation, we come up with the following additional machine cost matrix:

	Cost to Add a Machine ($)
Operation A	250000.00
Operation B	200000.00
Operation C	150000.00
Operation D	100000.00
Operation E	50000.00

With this information, we can begin to formulate a linear program like the one in 5.1 LINEAR OPTIMIZATION MODELS. Our decision variables are production, inventory, and the number of machines broken down into the ones purchased and total.

5.2.2 Developing the Mixed Integer Linear Model

Decision Variables

pr_{iot} production of item i at operation o in time period t

in_{iot} inventory of item i at operation o in time period t

m_{ot} machine level at operation o in time period t

b_{ot} increase to machine force level at operation o in time period t

cap_{ot} capacity of operation o in time period t

Constraints

The initial number of machines is known:

$$m_{o0} = [5\ 2\ 3\ 1\ 2]$$

The initial inventory of each item and operation is known to be zero:

$$in_{o,i,0} = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0]$$

The demand that must be met is known:

$$d_{it} = [20\ 20\ 20; 0\ 0\ 60; 10\ 0\ 10]$$

Processing time for each item to go through an operation is known:

$$pt_{oi} = [.5\ .75\ 0; .25\ 0\ .5; .5\ .75\ .5; .25\ 0\ 0; .25\ .25\ .5]$$

Variable machine resource costs are known:

$$cm_o = [425\ 425\ 425\ 425\ 425]$$

Adding machine costs are known:

$$cb_o = [250000\ 200000\ 150000\ 100000\ 50000]$$

Variable costs of production are known:

$$cpr_{oi} = [28.57\ 42.86\ 0; 14.29\ 0\ 28.57; 28.57\ 42.86\ 28.57; 14.29\ 0\ 0; 14.29\ 14.29\ 28.57]$$

Inventory costs of work-in-progress are known:

$$cin_{oi} = [.29\ .43\ 0; .43\ .43\ .29; .71\ .86\ .57; .86\ .86\ .57; 1\ 1\ .86]$$

The set of operations, items, and time period is known:

$$o \in Operations = \{1,2,3,4,5\}$$

$$i \in Items = \{1,2,3\}$$

$$t \in TimePeriod = \{0,1,2,3\}$$

Demand needs to be met and inventory tracked:

$$in_{io,t-1} + pr_{iot} - in_{iot} \geq d_{it} \quad \forall\, i \in \{1,2,3\}, o \in \{1,2,3,4,5\}, t \in \{1,2,3\}$$

Machines need to be tracked and balanced to enable buying machines but not getting rid of machines:

$$m_{o,t-1} + b_{ot} - m_{ot} = 0 \quad \forall o \in \{1,2,3,4,5\}, t \in \{1,2,3\}$$

We know that the planned production time per machine at each operation is:

$$ppt_o = [7\ 7\ 7\ 7\ 7]$$

Therefore, the capacity of each operation in a given time is:

$$cap_{ot} = ppt_o m_{ot}\ \forall o \in \{1,2,3,4,5\}, t \in \{1,2,3\}$$

We know that our production must not exceed our capacity:

$$\sum_{i=1}^{3} pr_{iot} pt_{oi} - cap_{ot} \leq 0\ \forall o \in \{1,2,3,4,5\}, t \in \{1,2,3\}$$

Lastly, we know that Item 1 cannot go through Operation B without going through Operation A. The production through an operation is dependent:

$$pr_{i,o-1,t} + in_{i,o-1,t-1} - pr_{i,o,t} - in_{i,o-1,t} \geq 0\ \forall i \in \{1,2,3\}, o \in \{2,3,4,5\}, t \in \{1,2,3\}$$

This constraint says that we cannot produce at a later operation unless we have already produced it at the previous operation in the current time period or have it in inventory at the previous operation in the previous time period. We also have to choose to store it in inventory at the previous operation in the current time period or allow it to go forward the later operation. It cannot do both.

We also add some constraints on our decision variables to be both non-negative and integers. Non-negativity forces our other constraints to behave properly, such as demand being met. The integer constraint makes our optimization problem more difficult; however, it also makes it more realistic due to the fact that we cannot add half a machine or make half an item.

These constraints can be written like this:

$$\begin{aligned} pr_{iot}, in_{iot} &\geq 0, \quad \in \mathbb{Z}, \quad \forall\, i \in \{1,2,3\}, o \in \{1,2,3,4,5\}, t \in \{0,1,2,3\} \\ m_{ot}, b_{ot} &\geq 0, \quad \in \mathbb{Z}, \quad \forall o \in \{1,2,3,4,5\}, t \in \{0,1,2,3\} \\ cap_{ot} &\geq 0, \quad \forall o \in \{1,2,3,4,5\}, t \in \{0,1,2,3\} \end{aligned}$$

Lastly, our objective is to meet our constraints while minimizing our cost:

Objective Function:

$$\min \sum_{t=1}^{3}\sum_{o=1}^{5}\sum_{i=1}^{3}(cpr_{oi}pr_{iot} + cin_{oi}in_{iot}) + \sum_{t=1}^{3}\sum_{o=1}^{5}(cm_o m_{ot} + cb_o b_{ot})$$

Equation 24

Altogether we have a MILP model with 186 rows, 180 columns, and 462 nonzeros. We also have 20 continuous variables and 160 integer variables we are solving for. Using software such as Gurobi and Julia/Jump, we can quickly solve this problem in less than 0.01 seconds with 60 simplex iterations to find the optimal solution within a tolerance of 1.00e-4. [20] The best objective is $384360.05. While this objective value is great to know for financial planning, what is more important to know is the optimal production and inventory plan, and when to add additional machines.

From the data in Table 8, we see that we will need to add two machines to Operation C on Day 2 and one machine to Operation E on Day 3. We also see that we need to complete eight of the ten units needed for Item 3 on Day 3 a day in advance on Day 2. We then hold that in inventory until we use it to meet demand on Day 3. This production shifting helps us tremendously in terms of cost-saving due to inventory costs being so much less than the cost to add an entirely new machine. In fact, if we do not shift production and hold work-in-progress/finished inventory and instead opt to add enough machines to meet demand on our current schedule, our best objective increases to $1784965.40, which is more than 4x more expensive due to requiring us to add 11 total machines.

From the walkthrough, it can be seen that capacity planning decisions can be made through mathematical programming and that their results can save a tremendous amount of money, resulting in an optimal or near-optimal ROI.

	Production Day 1			Inventory Day 1			Day 1 Additional Machines
	Item 1	Item 2	Item 3	Item 1	Item 2	Item 3	
Operation A	21	21	10	1	15	0	0
Operation B	20	6	10	0	0	0	0
Operation C	20	6	10	0	6	0	0
Operation D	20	0	10	0	0	0	0
Operation E	20	0	10	0	0	0	0

	Production Day 2			Inventory Day 2			Day 2 Additional Machines
	Item 1	Item 2	Item 3	Item 1	Item 2	Item 3	
Operation A	20	33	17	1	26	0	0
Operation B	20	22	16	0	0	0	0
Operation C	20	22	16	0	28	8	2
Operation D	20	0	8	0	0	0	0
Operation E	20	0	8	0	0	8	0

	Production Day 3			Inventory Day 3			Day 3 Additional Machines
	Item 1	Item 2	Item 3	Item 1	Item 2	Item 3	
Operation A	19	34	10	0	0	0	0
Operation B	20	60	10	0	0	0	0
Operation C	20	32	2	0	0	0	0
Operation D	20	60	10	0	0	0	0
Operation E	20	60	2	0	0	0	1

Table 8: Resulting table of decision variables from walkthrough optimization problem

5.3 ADDING CONSTRAINTS TO MIMIC FACTORY

5.3.1 Operational Level Constraints

Now we will begin adding more complexity to our model to more accurately model our factory. First, we will add the factor that different operations use different machines or no machines at all. Some of these machines are extremely complex and expensive to add, while others are simply a bench with tools for manual workers to use. One can see that the term machine is used broadly here as the primary inanimate object used in an operation. We will define z_{io} as the number of machines required per unit of production of item i in operation o; m_{ot} as the number of machines available at operation o in time period t; cm_{ot} as the variable unit cost of operating a machine for operation o in time period t; and cb_{ot} as the variable building cost of a machine for operation o in time period t.

$$\sum_{i=1}^{I} z_{io}pr_{iot} - m_{ot} \leq 0 \quad \forall o, t$$

This modification of adding in the operation level can be added into the previous parts of the model:

Objective Function

$$\min \sum_{t=1}^{T}\sum_{i=1}^{O}\sum_{i=1}^{I}(cpr_{iot}pr_{iot} + cin_{iot}in_{iot}) + \sum_{t=1}^{T}\sum_{o=1}^{O}(cwf_{ot}wf_{ot} + ch_{ot}h_{ot} + cf_{ot}f_{ot}) + \sum_{t=1}^{T}\sum_{o=1}^{O}(cm_{ot}m_{ot} + cb_{ot}b_{ot})$$

Constraints

Demand:

$$in_{i,o,t-1} + pr_{iot} - in_{iot} \geq d_{iot} \quad \forall i, o, t$$

Workforce Capacity:

$$\sum_{i=1}^{I} a_{io}pr_{iot} - wf_{ot} \leq 0 \quad \forall o, t$$

Machine Capacity:

$$\sum_{i=1}^{I} z_{io}pr_{iot} - m_{ot} \leq 0 \quad \forall o, t$$

Workforce Balance:

$$wf_{o,t-1} + h_{ot} - f_{ot} - wf_{ot} = 0 \quad \forall o, t$$

Machine 'Balance':

$$m_{o,t-1} + b_{ot} - m_{ot} = 0 \quad \forall o, t$$

Non-negativity:

$$pr_{iot}, in_{iot}, wf_{ot}, h_{ot}, f_{ot}, m_{ot}, b_{ot,}d_{iot} \geq 0 \quad \forall i, o, t$$

Equation 25

5.3.2 Introducing Capacity Utilization to our Model

While our model already involves capacity utilization, let us begin to combine the capacity terminology and equations we defined in 2.4 MANUFACTURING CAPACITY to our model. Starting with Equation 8, the capacity utilization equation, we can see that this constraint shows up in our model under workforce capacity and machine capacity. Adding a time period dimension to this equation, for a general resource we get:

$$CU_{ot} = \frac{\sum_{i=1}^{I} d_{iot} pt_{io}}{sh_{ot} wd_{ot} ls_{ot} r_{ot}} = \frac{\sum_{i=1}^{I} d_{iot} pt_{io}}{ppt_{ot} r_{ot}} \forall o \in \{Operations\}, t \in \{Time\ Periods\}$$

As a constraint we would never want CU_{ot} to be >1 and we want capacity required to be in terms of decision variables, so we can reformulate the equation into the following constraint:

$$\sum_{i=1}^{I} pr_{iot} pt_{io} - ppt_{ot} r_{ot} \leq 0$$

Equation 26

If we desire to look at it in terms of labor capacity or machine capacity, we can replace r_{ot} with wf_{ot} or m_{ot}. We could also add in another dimension making it r_{jot} indicating that there are different types of resources j.

5.3.3 Implementing Dependent Production

In our original models, we are only looking at a single operation serving demand for all items. Nevertheless, we know that there are multiple operations working together to complete an item by a specific date so that subsequent operations can continue to work on that item. We also know that production at an operation is dependent on previous operations:

$$pr_{o-1,i,t} + in_{o-1,i,t-1} - pr_{o,i,t} - in_{o-1,i,t} \geq 0 \ \forall o, i, t$$

Equation 27

This constraint says that we cannot produce at a later operation unless we have already produced it at the previous operation in the current time period or have it in inventory at the previous operation in the previous time period. We also have to choose to store it in inventory at the previous operation in the current time period or allow it to go forward to the later operation. It cannot do both.

5.3.4 Workforce Needed to Run Machines

Since CCA is not a fully automated factory, it follows that a specific ratio of workers is required for operation and that this number would depend on the number of machines being operated. We define this ratio as wfr_o.

$$m_{ot} wfr_o sh_{ot} - wf_{ot} \leq 0 \quad \forall o, t$$

Equation 28

For example, if Operation A has two machines that operate for one shift and a workforce ratio of 3 workers per machine then Operation A needs at least six workers to operate correctly. If two shifts, then 12 workers would be required. While there are various types of workers in the factory, we assume that there is only one type for simplicity and make up for this assumption by weighting the variable worker cost.

5.3.5 Operation Demand

While finished item demand is well known, demand of an operation depends on the CT associated with each operation:

$$d_{i,o-1,t-CT_{iot}} \geq d_{iot} \; \forall i, o, t$$

Equation 29

For clarification, let us assume that we need 10 units of item one complete in time period 5. Complete means that o=O, the last operation. That is $d_{1,O,5} = 10$. If CT of item i at operation O in time period 5 is equal to 2 time periods, then $CT_{iot} = 2$ and $d_{1,O-1,3} = 10$. This means that we need to complete 10 units of item 1 in time period 3 at operation $O - 1$.

For finished item demand at the last operation O, we set it equal to confirmed demand (dk) plus predicted demand (dp) for that item i in time period t:

$$d_{iot} = dc_{it} + dp_{it} \; \forall i, t; o = O$$

Equation 30

5.3.6 Introducing OEE to our Model

While our planned production time of an operation is $ppt_{ot} = sh_{ot}wd_{ot}ls_{ot}$, we rarely ever actually produce for that amount of time. While ppt_{ot} account for planned breaks through ls_{ot}, it does not account for unplanned stops such as equipment failures and material shortages or planned stops such as setup time and change over time. Using Equation 26, if we multiply this time by our calculated Availability factor A_{ot}, we get the actual expected available runtime of the operation—capacity available at the operation. Our updated general resource constraint then becomes:

$$\sum_{i=1}^{I} pr_{iot}pt_{io} - ppt_{ot}r_{ot}A_{ot} \leq 0$$

Equation 31

If we plan to produce for 10 hours and have two resources, we would expect to have a capacity available of 20 hours; however, if our Availability metric was calculated to be 90% then our capacity available is only 18 hours.

Performance follows this same logic, but we would need to change our constraint to use ideal process time instead of process time. The general resource constraint would then become:

$$\sum_{i=1}^{I} \frac{pr_{iot} IPT_{io}}{P_{ot}} - ppt_{ot} r_{ot} A_{ot} \leq 0$$

Equation 32

If we produce 60 units all with an ideal process time (IPT) of 10 minutes, then we expect our capacity required to be 600 minutes or 10 hours; however, if our Performance metric was calculated to be 80% then our capacity required is 750 minutes or 12.5 hours. Nevertheless, since the processing time we implement in our model is the average processing time of an item at a given operation, the Performance metric is already accounted for at the item. We, therefore, assume that $\frac{IPT_{io}}{P_{ot}} = pt_{io}$ and do not explicitly include the metric in our model.

Quality, on the other hand, does not get factored into the capacity utilization constraint of the model. Instead, it is added to the demand constraint:

$$in_{i,o,t-1} + pr_{iot} - in_{iot} \geq \frac{d_{iot}}{Q_{ot}} \quad \forall i, o, t$$

Equation 33

If we had a demand of 99 items and our Quality metric is 99%, then we now need to account for the production of at least 100 items in order to ensure we meet demand.

We solve for OEE metrics in 4.3 CALCULATING OEE WITH CURRENT DATA SOURCES and initially assume that they remain constant when solving our model.

5.3.7 Machine Learning for CT Prediction

While processing time is relatively constant for a given operation/item combination, CT is not. Instead, it depends greatly on item, operation, time period, number of items in a lot, WIP, etc. and is not necessarily a static number. While we could assume that is was constant for a given item, operation combination, we purposefully do not due to many lot characteristics and WIP levels affecting CT. CT is therefore dependent on many predictor variables:

$$CT_{iot} = f\left(i, o, t, pr_{iot}, in_{i,o-1,t}, \dots, d_{iot}\right) \forall i, o, t$$

Equation 34

where $f: \left(i, o, t, pr_{iot}, in_{i,o-1,t}, \dots, d_{iot}\right) \rightarrow CT_{iot}$ by machine learning techniques discussed in CH. 6 MACHINE LEARNING TO PREDICT CYCLE TIME.

Since a random forest was considered to have the best performance for CT prediction as shown in 6.7 MODEL BUILDING, TRAINING, AND TESTING, CT becomes:

$$CT_{iot} = \psi_{\mathrm{L},\theta_1,\ldots,\theta_m}(\boldsymbol{x}) = \frac{1}{M}\sum_{m=1}^{M} \varphi_{L,\theta_m}\left(i, o, t, pr_{iot}, in_{i,o-1,t}, \ldots, d_{iot}\right)$$

where a set of M randomized models $\{\varphi_{L,\theta_m} | m = 1, \ldots, M\}$ have been trained on the same data L but each built from an independent random seed θ_m. The combination of the predictions of these models are averaged into a new ensemble model, denoted $\psi_{\mathrm{L},\theta_1,\ldots,\theta_m}$. [52]

5.4 FINAL NONLINEAR OPTIMIZATION MODEL

Our final model incorporates the complexity of the Raytheon CCA factory into our original simple models and is no longer a linear model due to nonlinear constraints. This section ties our model's pieces together and then discusses the solving method used.

5.4.1 Final Model Decision Variables

Looking back at 2.4 MANUFACTURING CAPACITY, we can see that our decision variables align with the three available options to change capacity utilization.

1. Increasing Resources
    a. Adding Shifts - $sh_{ot}, wd_{ot}, ls_{ot}$
    b. Adding Machines - m_{ot}, b_{ot}
    c. Adding Workers - wf_{ot}, h_{ot}, f_{ot}
2. Make operation faster or reduce defects - A_{ot}, Q_{ot}
3. Shift Demand -pr_{iot}, in_{iot}

5.4.2 Final Model Objective Function

Our objective function looks to minimize the cost to the factory while the constraints ensure that demand is being met on time—two of the tradeoffs associated with capacity planning, which is discussed in 2.5 STRATEGIC CAPACITY PLANNING. While it is known that constraints can be combined to make the problem easier to solve, they were kept separate for interpretability purposes.

$$\min \sum_{t=1}^{T}\sum_{i=1}^{O}\sum_{i=1}^{I}\left(cpr_{iot}pr_{iot} + cin_{iot}in_{iot}\right) + \sum_{t=1}^{T}\sum_{o=1}^{O}\left(cwf_{ot}wf_{ot} + ch_{ot}h_{ot} + cf_{ot}f_{ot}\right) + \sum_{t=1}^{T}\sum_{o=1}^{O}\left(cm_{ot}m_{ot} + cb_{ot}b_{ot}\right)$$

5.4.3 Final Model Constraints

Production Demand:

$$in_{i,o,t-1} + pr_{iot} - in_{iot} - \frac{d_{iot}}{Q_{ot}} \geq 0 \;\; \forall i, o, t$$

Workforce Capacity:

$$\sum_{i=1}^{I} pt_{io} pr_{iot} - ppt_{ot} wf_{ot} A_{ot} \leq 0 \quad \forall o, t$$

Machine Capacity:

$$\sum_{i=1}^{I} pt_{io} pr_{iot} - ppt_{ot} m_{ot} A_{ot} \leq 0 \quad \forall o, t$$

Workforce Balance:

$$wf_{o,t-1} + h_{ot} - f_{ot} - wf_{ot} = 0 \quad \forall o, t$$

Machine ‘Balance’:

$$m_{o,t-1} + b_{ot} - m_{ot} = 0 \quad \forall o, t$$

Dependent Production:

$$pr_{o-1,i,t} + in_{o-1,i,t-1} - pr_{o,i,t} - in_{o-1,i,t} \geq 0 \; \forall i, o, t$$

CT Random Forest Prediction:

$$CT_{iot} = \psi_{\mathrm{L},\theta_1,\dots,\theta_m}(\boldsymbol{x}) = \frac{1}{M} \sum_{m=1}^{M} \varphi_{L,\theta_m}\big(i, o, t, pr_{iot}, in_{i,o-1,t}, \dots, d_{iot}\big)$$

Operation Demand:

$$d_{i,o-1,t-CT_{iot}} \geq d_{iot} \; \forall i, o, t$$

Final Operation Demand:

$$d_{iot} = dk_{it} + dp_{it} \; \forall i, t; o = O$$

PPT Equation:

$$ppt_{ot} = sh_{ot} wd_{ot} ls_{ot}$$

Maintain OEE Constant:

$$A_{ot}, Q_{ot} = A_{o,t=0}, Q_{o,t=0}$$

Non-negativity:

$$pr_{iot}, in_{iot}, wf_{ot}, h_{ot}, f_{ot}, m_{ot}, b_{ot}, sh_{ot}, wd_{ot}, ls_{ot}, A_{ot}, Q_{ot} \geq 0 \quad \forall i, o, t$$

Integer constraints:

$$pr_{iot}, in_{iot}, wf_{ot}, h_{ot}, f_{ot}, m_{ot}, b_{ot}, sh_{ot}, wd_{ot} \in \mathbb{Z}$$

5.5 MIXED-INTEGER NONLINEAR OPTIMIZATION

A mixed-integer nonlinear program (MINLP) follows similar conventions to LPs. We are looking to minimize a $f(x, y)$ subject to $g_j(x, y) \leq b, where\ j \in J$, $Ax + By \leq b$, $x \in \mathbb{R}$, and $y \in \mathbb{Z}$. These types of problems are typically solved using pre-existing solvers that use algorithms such as the branch-and-bound algorithm or outer approximation. Nevertheless, solving this type of model requires either a vast amount of computing power or a manageable model with respect to the number of constraints and variables.

5.5.1 Common Methods to Solve MINLPs

The branch and bound algorithm was first introduced by Land and Doig to solve MILPs, but it has since then been expanded to MINLPs. Simply put, the branch and bound algorithm solves the MINLP problem by relaxing the integer constraints and solving the continuous problem first to get the lower bound. If all decision variables are already integer values, the MINLP is also solved. If not, the continuous relaxation is branched into sub-problems and constrained until integer solutions are found. While this works relatively quickly for MILPs, MINLPs are computationally more demanding and require techniques such as those proposed by Leyffer to solve. On the other hand, the outer approximation method is a decomposition technique first introduced by Duran and Grossmann. It works by constructing a polyhedral outer approximation of the nonlinear feasible region and iteratively improving. More details can be found in Kronqvist's 2018 paper. Typically, the outer approximation solves a continuous relaxation problem to find the initial lower bound before solving the MINLP. [53]

5.5.2 Solving the MINLP

In order to more effectively solve our MINLP, we must first preprocess the model. First, we remove redundant constraints. For instance, we can combine the PPT Equation and the Capacity Constraints to lower our number of constraints. We can also remove the OEE constraint since, in its current form, our model restricts OEE metrics to a constant. We can also look into bound tightening and reformulations as described in Belotti *et al.* [53]

Next, we make conscious decisions to reduce the size of the model to make it more manageable. As an example: In the next 60 months, there is demand for 2072 unique items across 326 operations. If we include all of these items and operations across the 60 months, our model would have millions of variables and constraints. This is entirely too large for the limited computing power we have available. In order to reduce this, we can use the expertise of process engineers and operations managers to understand what operations are key to look at. From these experts, we concluded that 41 key operations should be primarily focused on. This reduces both our model variables and constraints by 90.5%.

Lastly, we can use primal heuristics to get an initial feasible solution to help our solver with our original problem. After preprocessing, we then use a solver such as Juniper to solve our model. Juniper is an open-source MINLP solver developed by Kroger *et al.* that utilizes a branch and bound method with primal heuristics for quicker solving. It is easy to install and use in Julia/Jump and is the primary solver implemented in this thesis for the MINLP. [31] [53]

CH. 6 MACHINE LEARNING TO PREDICT CYCLE TIME

Due to the inaccuracy of simply using a single number such as the average for cycle time, capacity utilization would not be accurately modeled in the long run. Therefore, a model to predict cycle time of current and future items is needed. Because CT varies so greatly depending on things such as WIP, operations and item characteristics such as type, dimensions, number of components, and component size/type, it is believed that using historical data will enable us to better predict future cycle times with similar characteristics. This chapter dives into our use of machine learning to predict cycle time at the operation level by building upon concepts such as feature generation and selection before ultimately building, training, and testing the models.

6.1 MACHINE LEARNING OVERVIEW

Our hypothesis is that CT can be more accurately predicted via supervised machine learning methods than the currently implemented method at Raytheon. In order to test our hypothesis, we must predict a response variable based on a set of predictor variables. Our primary aim is to make the most accurate predictions of the response variable. The ability to identify which predictor variables have the largest impact on the response variable is secondary. Because of this, we look into implementing ten different types of machine learning models, including linear regression models, regression trees, nearest neighbor models, clustering algorithms, and an artificial neural network.

In order to develop the best model, we first develop a very large data set with hundreds of predictor variables that align to our response variable. Nevertheless, many of these variables are redundant or irrelevant in prediction of our response variable and, due to having limited computing power available, feature selection is necessary before training the models. Once predictor variables are decided upon, various machine learning models can be trained and evaluated.

6.2 FEATURE GENERATION

Feature generation involves creating potentially useful predictor variables from available data for use in the machine learning model. While we do not discuss all of the features we generated in this thesis, we do discuss a few that could be recreated by most manufacturing factories.

6.2.1 Date Timestamps

Date timestamps can also be derived from the transactional and IIoT data by defining the following:

$$Completion\ Date_{lot} = LEA_{lot}$$
$$Start\ Date_{lot} = FSA_{lot}$$
$$Queue\ Start\ Date_{lot} = LEA_{l,o-1,t}$$

From these timestamps, we can pull out further data such as day of the week, week in the year, etc. in order to utilize trends for better prediction. A description of the above terms can be found in 4.2 CAPACITY UTILIZATION CALCULATION.

Other useful features can be generated from these timestamps for prediction of CT. For example, cycle time, on average, tends to get better through the week and increase on the weekends with Monday being the highest as seen in Figure 13. Certain months also look to be more productive than others as seen in Figure 14. CT for certain items and operations may also improve over time as seen in Figure 15.

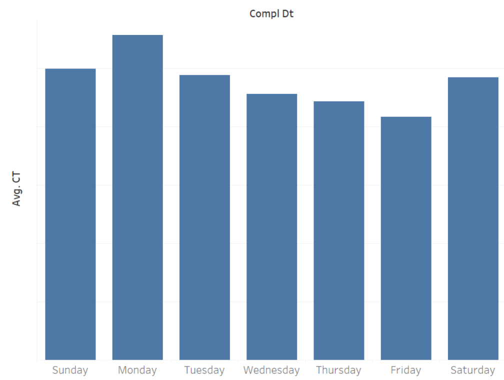

Figure 13: Average cycle time throughout the factory depending on the day of week

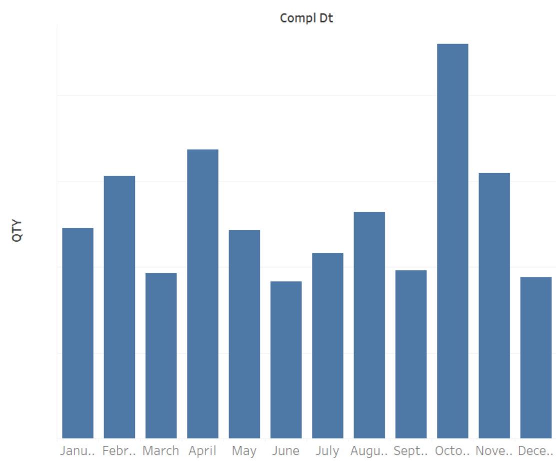

Figure 14: Number of finished items per month in 2018

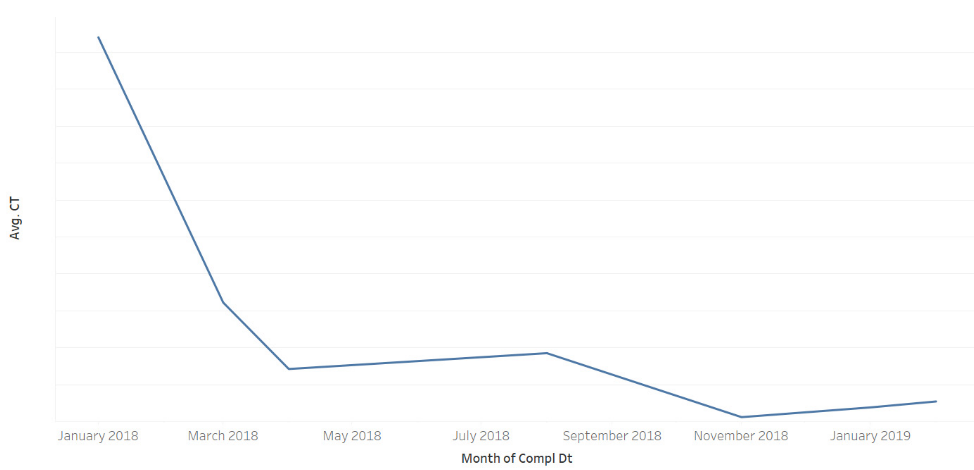

Figure 15: Cycle time for some items improve over time

6.2.2 Work-in-progress/Queue Length

The time a lot sits in an operation's queue and therefore the magnitude of its CT depends highly on the queue length, making estimates of WIP and the queue length useful for CT prediction. The queue length of an operation at a given datetime can be estimated by summing up all lot quantities at that operation with Queue Start Dates prior to the given time and Start Dates after the given time. Looking at the average queue quantity and average cycle time of an operation over time, one can see in Figure 16 that as the queues build up, average CT also increases and lags behind.

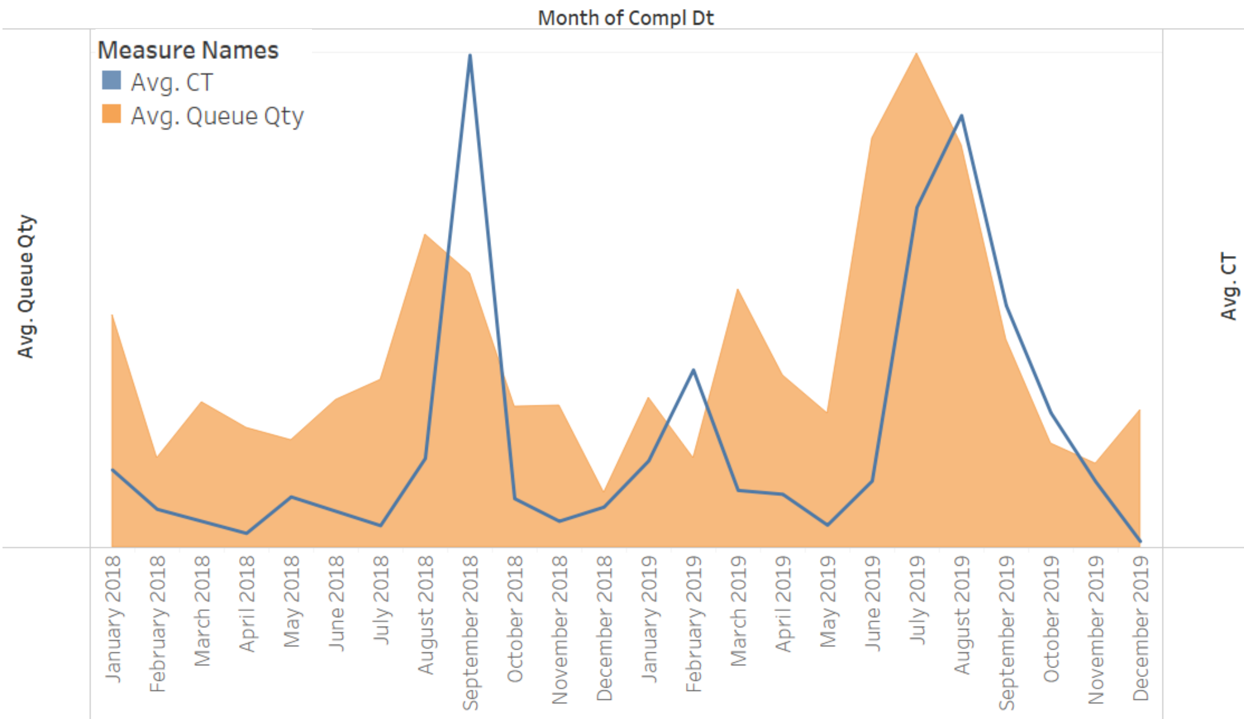

Figure 16: On average, increasing queue length increases the cycle time of an item

6.2.3 Bill of Materials Data

Manufacturing companies also have bill of materials (BOM) data centrally stored on enterprise software such as SAP. BOM data consists of a list of raw materials, components, and sub-assemblies with their respective quantities needed to manufacture the end product. A typical bill of materials has data that looks like it does in Table 9.

Material	Circuit Card 99
Plant	2000

Item	Component	Component Description	Quantity	Unit
0010	1234-ABC-567	axial carbon resist0r 4.7kOhm	10	EA
0020	1234-ABC-568	Resistor, Heat Sinkable, 7.5	1	EA
0030	1234-ABC-569	Resistr, .25Kilohms	10	EA
0040	1234-ABC-570	capaciter, 47uf	10	EA
0050	1234-ABC-571	1000uF 35 Vdc Aluminum Electrolytic Capacitor	1	EA
0060	1234-ABC-572	Integrated Circuit	5	EA
0070	1234-ABC-573	Inductor	1	EA
0080	1234-ABC-574	Diode	1	EA
0090	1234-ABC-575	Transistor	1	EA

Table 9: An example of what a BOM might look like for a circuit card

The problem with this data, however, is that component names are commonly not systemized to match component types. Furthermore, component descriptions are often manually entered by many different users over the years, resulting in different formatting, different spelling, typos, etc. An example to demonstrate some of the problems with this is in the component description column of Table 9. You can see that resistor is spelled incorrectly in Item 0010 and 0030, and capacitor is spelled incorrectly in Item 0040. Additionally, the formatting is different throughout.

In order to use this type of information for predictive purposes, we need to be able to compare component makeup across materials. Circuit Card 99, a made-up card, for example, should be known to have 21 resistors, 11 capacitors, 5 integrated circuits, 1 inductor, 1 diode, and 1 transistor. In order to do this, we will use the Levenshtein distance. [54]

The Levenshtein distance is a metric that measures how close or different two sequences of words are. It can be written as:

$$lev_{ab(i,j)} = \begin{cases} min\begin{cases} lev_{ab}(i-1,j)+1 \\ lev_{ab}(i,j-1)+1 \\ lev_{ab}(i-1,j-1)+1_{(a_i \neq b_i)} \end{cases}, if \min(i,j)=0 \\ \max(i,j)\,, if \min(i,j) \neq 0 \end{cases}$$

Equation 35

where a is one string and b is another string, $1_{(a_i \neq b_i)} = 0$ when $a = b$ and 1 otherwise, and $lev_{ab(i,j)}$ is the distance between the first i characters of string a and the first j characters of string b. The Levenshtein distance between "resister" and "resistor" is equal to 2 due to needing two actions to go from one to the other—deletion of the 'e' and addition of the "o". [54]

We can then calculate the Levenshtein similarity ratio as:

$$ratiolev_{ab(i,j)} = \frac{length(a) + length(b) - lev_{ab(i,j)}}{length(a) + length(b)} * 100\%$$

Equation 36

The ratio for "resistor" and "resister" is then equal to 87.5%.

We can use the FuzzyWuzzy package in Python to quickly calculate these ratios to identify commonalities across components. More specifically, it can look at partial ratios to deal with more complex strings. If the $length(a) < length(b)$ then the algorithm looks for the best resulting ratio from a $length(a)$ substring in b. [55] From Equation 35 and Equation 36, we can calculate a matrix that looks like Table 10.

	Resistor	Capacitor	Integrated Circuit	Inductor	Diode	Transistor
axial carbon resist0r 4.7kOhm	**88%**	44%	33%	38%	40%	60%
Resistor, Heat Sinkable, 7.5	**100%**	44%	27%	50%	40%	70%
Resistr, .25Kilohms	**88%**	33%	29%	38%	40%	60%
capaciter, 47uf	13%	**89%**	29%	38%	40%	21%
1000uF 35 Vdc Aluminum Electrolytic Capacitor	50%	**100%**	33%	50%	40%	53%
Integrated Circuit	43%	44%	**100%**	50%	40%	40%
Inductor	62%	67%	50%	**100%**	40%	62%
Diode	40%	62%	40%	44%	**100%**	40%
Transistor	75%	56%	40%	62%	40%	**100%**

Table 10: A Levenshtein Ratio matrix for an example BOM

Using a proper cutoff, such as 80%, we can then classify each component and output a usable matrix for prediction such as the one shown in Table 11.

	Resistor	Capacitor	Integrated Circuit	Inductor	Diode	Transistor	…	Amplifier	Total Components
Material 99	21	11	5	1	1	1	…	0	40
…	…	…	…	…	…	…	…	…	…

Table 11: An example of features generated from BOM analysis and the Levenshtein Ratios

6.2.4 Other Features Generated

Additional analytics can be completed on the transactional and IIoT data to gain potentially useful information for predicting CT. For example, manufacturing averages at the operation level can also be calculated such as:

$$CT_{avg_{ot}} = \frac{\sum_{l=0}^{L} CT_{lot}}{L};\ PT_{avg_{ot}} = \frac{\sum_{l=0}^{L} PT_{lot}}{L};\ QT_{avg_{ot}} = \frac{\sum_{l=0}^{L} QT_{lot}}{L};\ QTY_{avg_{ot}} = \frac{\sum_{l=0}^{L} QTY_{lot}}{L}\ \forall o,t$$

Other manufacturing measures specific to Raytheon were also calculated from the transactional data. While all of these data points are not necessarily useful for prediction of CT, they are included until filtered out during feature selection in 6.3 FEATURE SELECTION.

6.3 FEATURE SELECTION

Feature selection enables us to choose a subset of the available response variables to reduce the dimensionality of the problem without losing significant amounts of information. Given our set of response and predictor variable pairs, we are tasked to create a model of the predictor variables that predicts our response variable well; however, it is often the case that only a subset of these features are needed to achieve low errors. For example, consider a housing cost problem in which the response variables are a long list telling us every detail of a house listed for sale all the way down to the size of each wall, capacity of the hot water heater, and years since the roof was replaced, and where we are tasked to predict the sale price of the house from these predictor variables. Then it is possible that knowing only a subset of these variables, such as square footage, number of bedrooms and bathrooms, and neighborhood, will be sufficient in predicting the sale price of the house. If we know what subset of predictor variables is required, then our models are much better at prediction of our response variable. While we are primarily interested in feature selection to reduce the dimensionality of our problem and enable our computing resources to be effective, feature selection has also been known to increase model performance. [56]

One way to perform feature selection is through feature extraction, where new independent predictor variables are created from the old predictor variables. The least important of the variables are then dropped in order to achieve dimensionality reduction. Principal component analysis (PCA) is one technique that can be utilized by using an orthogonal transformation to convert our initial, potentially correlated variables into new independent, uncorrelated variables. Further details can be found in Principal Component Analysis by Jolliffe. [57] [58] After creating these new independent, uncorrelated variables, dimensionality can be reduced by keeping only the most import features as described below. While this method is good for reducing the number of dimensions and ensuring independence, it does make predictor variables much less interpretable. In order to maintain interpretability, at least through feature selection, feature elimination is implemented.

Feature elimination reduces the number of features being trained on by removing them from consideration. One type of feature that is often removed are features that are highly correlated with one another. These features are called collinear features and tend to lower model performance due to high variance and less interpretability. [59] While collinearity can be picked up with something as simple as Pearson's pairwise correlation, multicollinearity between many variables can be reduced by looking at the variance inflation factor (VIF). With this VIF indicator, we can use methods such as mixed-integer optimization or mixed-integer quadratic optimization to select an uncorrelated subset of the features. [60]

Features that are deemed irrelevant by importance measures are also typically removed. One such measure is called the Mean Decrease Impurity importance (MDI) and refers to random forests. [61] Breiman proposed that the importance of a predictor variable for predicting a response variable was equal to the sum of the weighted impurity decreases for all nodes the predictor variable is used, averaged over all the trees in the forest. [62] Louppe went on to further prove that importance measurements of irrelevant predictor variables equal zero when computed by randomized trees. While the theorems described in Louppe 2013 refer to infinite sample sizes and ensembles, Loupe points out that real-world finite problems such as ours can still follow this logic even though importance measurements may be biased. [61] Because of this, we primarily utilize feature elimination through feature importance measurements computed by Random Forests and Gradient Boosted Trees.

6.4 OVERVIEW OF TESTED MODELS

6.4.1 Model Selection Introduction

Many different machine learning models exist today with numerous applications to both regression and classification. In this thesis, we discuss seven types of models and ten total models. The seven types are linear models, support vector machines, stochastic gradient descent, decision trees, nearest neighbors, ensembles, and neural networks. The ten models are Ridge, Least Absolute Shrinkage and Selection Operator (LASSO), Elastic-Net (EN), Support Vector Machine (SVM), Stochastic Gradient Descent (SGD), Classification and Regression Tree (CART), K-Nearest Neighbor (KNN), Random Forest (RF), Gradient Boosted Trees (GBT), and an Artificial Neural Network (ANN). These models were selected due to prior research showing their validity in this space as discussed in 3.2 MACHINE LEARNING FOR CYCLE TIME PREDICTION, their known ability to have good predictive power, and, for some, their ability for interpretation. Each model we implemented offers different advantages and disadvantages as shown in Table 12.

Type of Model	Actual Model	Advantages	Disadvantages
Linear Regression Models	Ridge	Good with small datasets	Does not adaptively capture nonlinear structures
	LASSO/EN	Easy to interpret	
Support Vector Machine	SVM	Low generalization error	Sensitive to tuning parameters
		Effective in high dimensions	Slow training on large datasets
Stochastic Gradient Descent	SGD	Easy to implement	Good number of hyperparameters
		Efficiency	Sensitive to feature scaling
K-Nearest Neighbor	KNN	Intuitive algorithm	Number of neighbors needs defines
		Robust to outliers	High relative computational complexity
			Moderately hard to interpret
Decision Trees	CART	Easiest model to interpret	Can lead to overfitting
		Adaptively captures nonlinearity	Not as powerful for prediction
Ensembles	RF	Handles categorical features well	Hard to interpret
	GBT	Adaptively captures nonlinearity	Can be slow
		Few parameters to tune	
		Performs well with many features	
Neural Networks	ANN	Known to have the greatest predictive capability with numerical variables	Many parameters to tune
			Large number of samples required for good performance
			Not robust to outliers
			Hard to interpret

Table 12: Comparison of machine learning models [63]

With our primary goal being predictive power, we expected the last four models to provide us with the best results; however, we included the first six to investigate whether similar predictive power could be achieved while maintaining interpretability. Since they are relatively simple machine learning models, they are quick to build, train, and test. This section will provide a brief overview of each model implemented.

6.4.2 Linear Models

Linear models are a set of regression methods in which the response value is expected to be a linear combination of predictor variables. Mathematically, if $\hat{y}$ is the response variable then $\hat{y}(w, x) = w_0 + w_1 x_1 + \cdots + w_p x_P$. Basic linear regression fits the model with coefficients $w = (\omega_1, \ldots, w_p)$ to minimize the residual sum of squares between predictor variables (X) and response variables (y): $\underset{w}{min}||Xw - y||_2^2$. Since noisy data may lead to overfitting in this method, regularization comes into play. [64] [65]

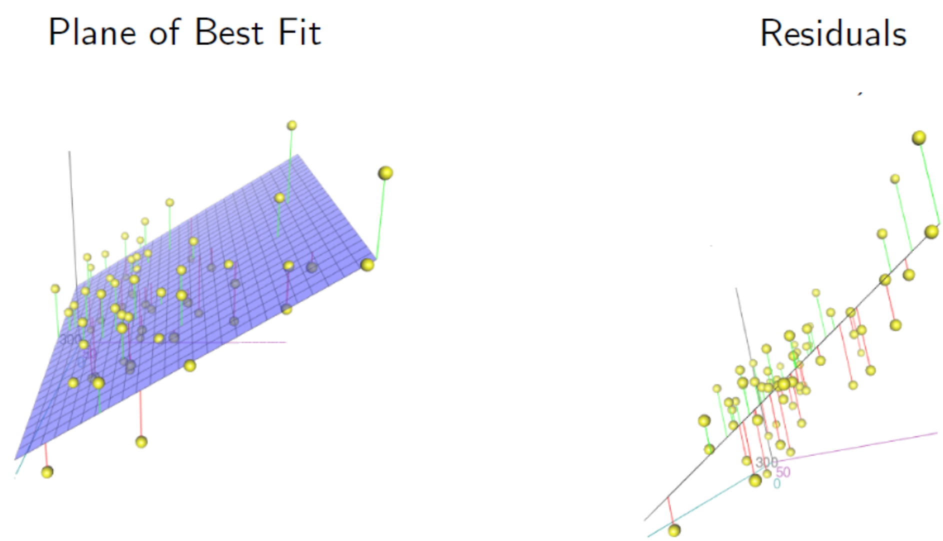

Figure 17: Graphical depiction of linear regression and minimization of the residuals [65]

Ridge

Regularization incorporates a complexity penalty directly into the minimize problem. Ridge regression imposes a penalty on the size of coefficients by minimizing the residual sum of squares with an added penalty: $\underset{w}{min}||Xw - y||_2^2 + \alpha||w||_2^2$, where $\alpha \geq 0$ is the constant penalty term that corrects against overfitting and can be optimized through cross-validation techniques., and $||w||_2$ is the ℓ_2-norm (Euclidean norm) of the coefficient vector. While Ridge regression results in simple and interpretable models, one common problem with it is the large number of coefficients that are extremely small yet not zero. [64] [65]

LASSO

LASSO regression is another penalty adding regression we look into that sets more coefficients equal to zero. It stands for Least Absolute Shrinkage and Selection Operator and is even easier to interpret. Like Ridge Regression, it mathematically is reducing the residual sum of squares with an added penalty: $\min_{w} \frac{1}{2n_{\text{samples}}} ||Xw - y||_2^2 + \alpha||w||_1$, where $||w||_1$ is the ℓ_1-norm of the coefficient vector. Again, the hyper-parameters of LASSO regression can be tuned through cross-validation. [64] [65]

Elastic-Net

Elastic-net is the third regression model that combines the properties of Ridge and LASSO by the use of two penalty terms. The minimization of the residual sum of squares in this case is: $\min_{w} \frac{1}{2n_{\text{samples}}} ||Xw - y||_2^2 + \alpha\rho||w||_1 + \frac{\alpha(1-\rho)}{2} ||w||_2^2$, where ρ is a constant that sets the ratio between the ℓ_1-norm and ℓ_2-norm of the coefficient vector. [64] [65]

6.4.3 Support Vector Machines

Support vector machines are effective in both high dimensional spaces and lopsided dimensionality. Additionally, it is effective when computing resources are limited due to using a subset of training points. Its mathematical formulation can be found in Smola and Scholkopf's 2003 paper, but, in summary, it is looking for a function of the predictor variables, with at most ε-deviation from the response variable. [64] [66]

6.4.4 Stochastic Gradient Descent

Stochastic Gradient Descent is another effective approach to fit linear regression models through various loss functions and penalties. While one of the three linear models above is commonly recommended, SGD is well suited for problems with a large number of samples. Its mathematical formulation can be found in Zhang's 2004 paper, but, in summary, it is solving the linear regression problem by using an iterative algorithm that starts from a random point and descends to find the lowest gradient or slope of that function. [64] [67]

6.4.5 Nearest Neighbors (KNN)

The K-Nearest Neighbor method looks to find a number (k) of training samples of predictor variables most similar to them, and predict the response variable from these. In short, this method finds the k observations in the training data which are closest to a set of predictor variables and averages their response variables. Closeness is defined through distance metrics such as Euclidean distance, Manhattan distance, and Minkowski distance. The average is commonly used because it is assumed that each neighbor uniformly contributes to the prediction of the response variable; however, weights can be added to closest neighbors and a weighted average can be used instead. [64] KNN is considered interpretable by some due to being able to locally interpret from neighbors; however, interpretation is not intuitive or easy to do. [68]

6.4.6 Decision Trees (CART)

The most easily interpretable model is a Classification and Regression Tree (CART) model as it more closely mirrors human thought. Another benefit of CART is that it can discover predictor variable interactions without incorporating interaction terms. Additionally, it can adaptively discover nonlinearities as seen in Figure 18.

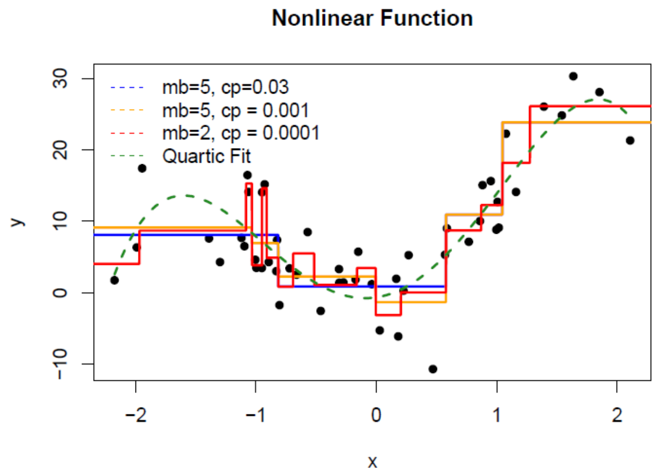

Figure 18: Graphical depiction of CART overcoming a nonlinear function [65]

CART uses a smart heuristic solution for the optimization problem. Let T denote the resulting decision tree, such that $\hat{y} = T(x_i)$; $Leaves(T)$ denote leaves on the tree (terminal nodes); N_l denote how many data points are contained in the l^{th} leaf; $Error(T)$ denotes the error associated with tree T for inaccurately predicting y on the training data; and $Error(T_{base})$ is the error from a tree trained without covariates. These errors are calculated using the Gini Index. The mathematical formulation is then:

$$\min_{T} Error(T) + cp + Leaves(T) + Error(T_{base})$$
$$s.t. N_l \geq minbucket$$

where $minbucket$ is the lower limit on the number of observations in each leaf and cp is the complexity parameter that prunes splits of the tree that do not improve the model fit. As such, the smaller $minbucket$ is the more splits are possible, and the smaller cp is the more splits are likely in the final tree. Both of which can either be set or derived optimally through cross-validation. [64] [65]

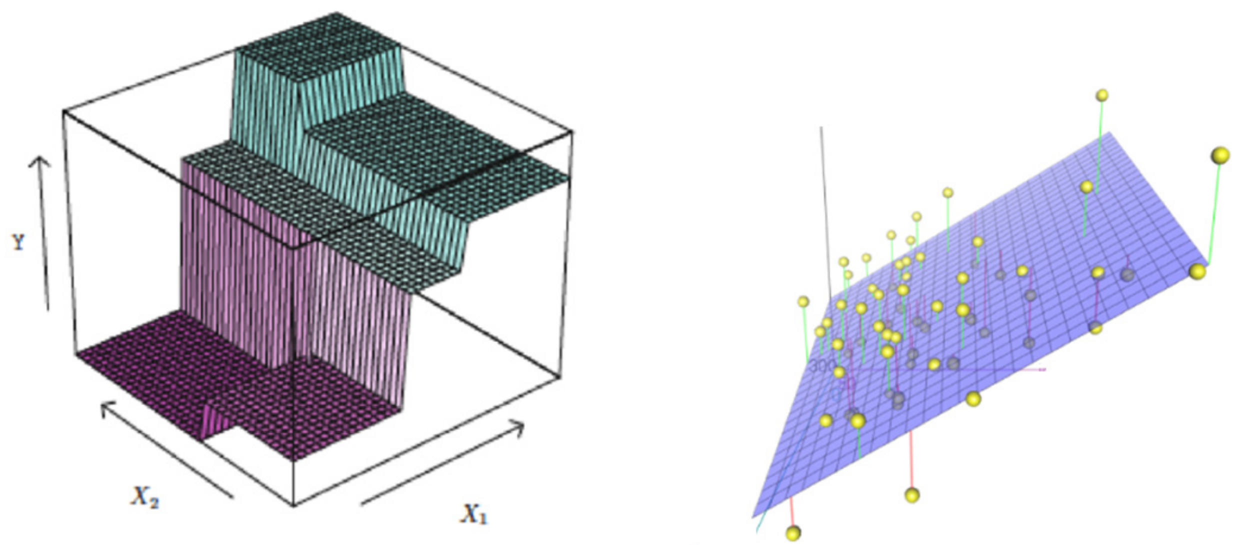

Figure 19: The difference between CART and multiple regression graphically [65]

6.4.7 Ensemble Methods

Ensemble methods combine predictions of several estimators in order to improve robustness and accuracy over a single estimator. Averaging methods such as Random Forests build several single CARTs and then averages their predictions. On average, this prediction is generally better than any of the individual CART models due to a reduction in variance. Boosting methods such as Gradient Boosted Trees use sequentially built estimators and attempt to reduce the bias of the combined estimator. Together this combined estimator is generally better at prediction than individual estimators.

Random Forest

A Random Forest model is an ensemble of a group of different CART models. Each CART tree makes a prediction of the response variables given the values of the predictor variables. The prediction of the random forest is the average of each tree's prediction. Each CART model in the forest is trained using a random sample from the training set. This is commonly referred to as bagging, and it mitigates overfitting. Each split of each bootstrapped tree is then made by considering only a random subset of predictor variables. The power of this model comes from each CART model finding patterns in its subset of data. When combined, the forest is able to then identify complicated patterns. Nevertheless, Random Forests become very large and complex and, as a result, are not interpretable. [64] [65]

Gradient Boosted Trees

A Gradient Boosted Tree model works like a Random Forest by using a combination of trees as a weak learner. A loss function is first defined and then trees are added one at a time to the model. Before a new tree is added, the loss function is optimized by modifying the new tree. Existing trees remaining unchanged. This continues until a set number of trees is reached or the loss function reaches a specified level. The mathematics behind GBTs can be found in Breiman's 2001 article. [62]

6.4.8 Neural Networks

Artificial Neural Networks (ANN) are commonly called feed-forward neural networks or Multi-Layer Perceptrons (MLP) and are made up of three primary layers—input, hidden, and output. The input layer consists of all input features. Each neuron in each hidden layer transforms the values from the previous layer with a weighted linear summation and outputs a nonlinear activation function. The output layer transforms the values from the last hidden layer and transforms them into a prediction.

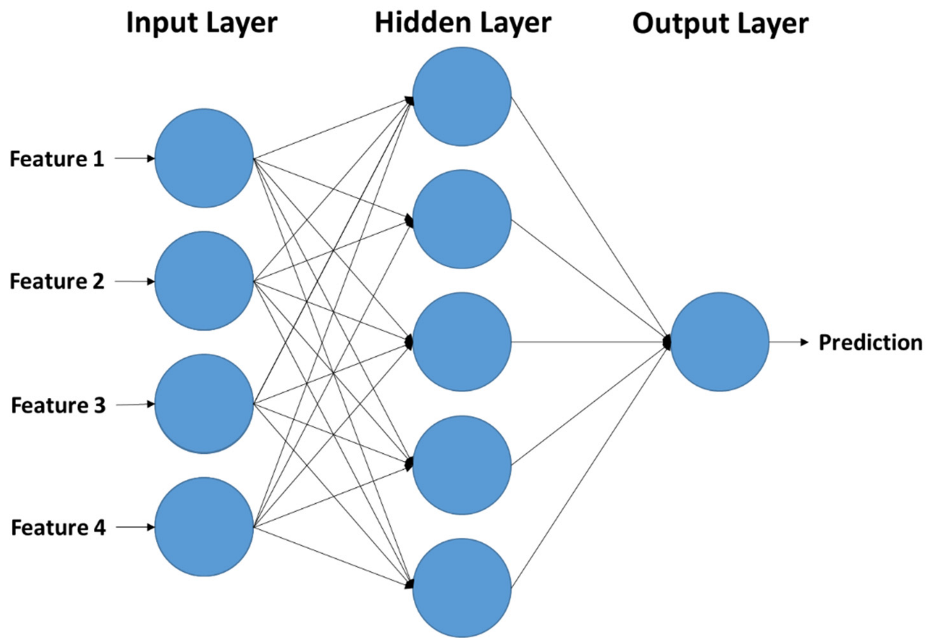

Figure 20: Graphical representation of a neural network model with four features, one hidden layer, and one response variable

Neural networks are trained by using backpropagation coupled with the previously mentioned gradient descent algorithm. The specific mathematics can be found in the book Practical Machine Learning. [69] The model learns a function $f(\cdot): R^m \rightarrow R^o$ by training on a dataset, where m is the number of features and o is the number of predictions. Given a set of predictor and response variables, the network can learn a nonlinear estimator for regression. While neural networks have been known to provide great predictive power and have the capability to learn nonlinear models, they also require a lot of tuning since there are so many hyperparameters. [69]

6.5 MACHINE LEARNING FRAMEWORK OVERVIEW

In order to accurately predict CT of a given lot, we require accurate models that can be computed on with limited computing power. As a result, we developed a four-step machine learning framework.

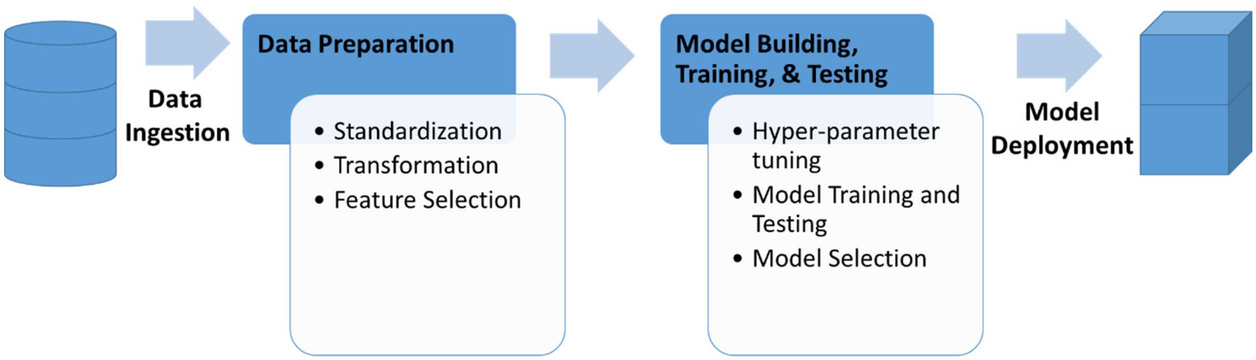

Figure 21: Four-step machine learning framework used in this thesis

1. DATA INGESTION

- **Data Ingestion**

  In order to have a chance at predicting CT of a lot, multiple pieces of numerical and descriptive information related to each lot needs to be ingested from various sources. This is discussed in 6.6 DATA INGESTION.

2. DATA PREPARATION

- **Data Pre-Processing**

  The second step consisted of standardizing and transforming data into a useful format for predictive models.

- **Feature Selection**

  The next step consisted of eliminating variables that are not useful in predictive modeling. Two primary criteria were used to select the predictor variables—features with higher importance in random trees were kept and features that were highly correlated with each other were removed.

3. MODEL BUILDING, TRAINING, TESTING

- **Predictive Modeling Analysis**

  After selecting our set of predictor variables and reducing the dimensionality of the problem, predictive models were built. Ten total predictive models without tuning were used for the prediction of cycle time: Ridge, Lasso, Elastic-Net, SVM, SGD, CART, Random Forest, GBT, KNN, and an Artificial Neural Network.

- **Down-Selection of Models**

  A total of ten models were built for each of the operations with anticipated demand in the next five years, and the top tier of models as determined by prediction metrics were selected for further analysis. This step, while not technically necessary, was done to save computing time by not tuning every model.

- **Model Tuning and Final Selection**

  After the top tier models were selected, their specific hyperparameters were tuned and the models were built again, tested, and validated. The top model as determined by the same prediction metrics was ultimately selected.

4. MODEL DEPLOYMENT

- **Final Predictive Modeling Testing**

  This model was then used to test our hypothesis that CT can be more accurately predicted via supervised machine learning methods than the currently implemented method at Raytheon.

- **Model Deployment**

  After successfully verifying our hypothesis, the model was automated and deployed to be used as an input into the strategic planning model.

6.6 DATA INGESTION AND PREPARATION

6.6.1 Data Ingestion

In order to have a chance at predicting CT of a lot, multiple pieces of numerical and descriptive information related to each lot need to be ingested from various sources. To enable the sustainability and scalability of the framework developed, a tool for automatic data ingestion was required to be implemented. The tool uses Python script embedded with SQL queries to automatically connect to the data warehouse and other network documents to ingest all available data for CT prediction. More specifically, transactional data and IIoT can be aggregated with material and component data at a lot level. From this ingested data, useful metrics as discussed in 6.2 FEATURE GENERATION can be automatically calculated and concatenated within the same tool. This is sometimes referred to as feature generation and can then be compiled as an executable file that automatically runs at specified times.

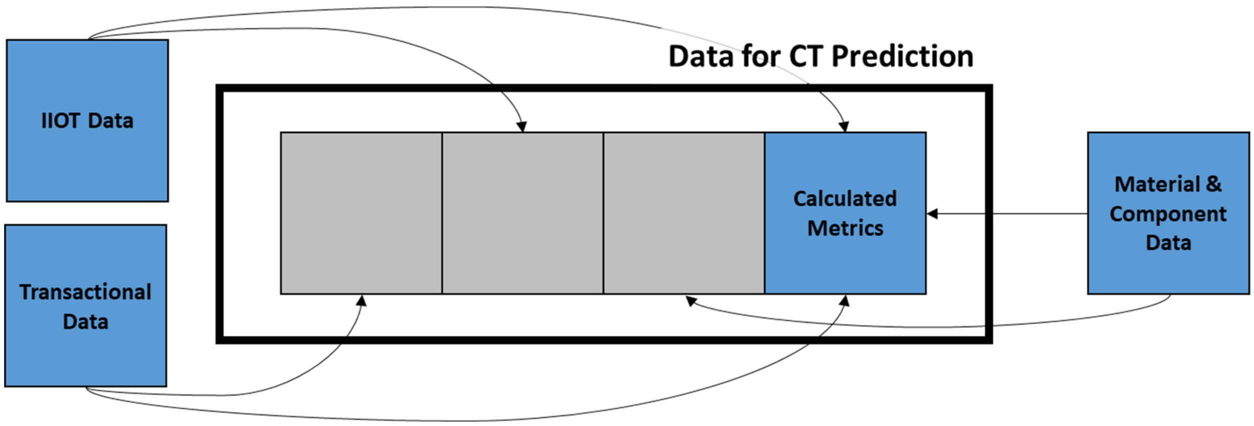

Figure 22: The ingestion of data from various sources

6.6.2 Data Preparation

Prior to building, training, and testing models, we had to first prepare the data we ingested through standardizing the data, transforming the data, and eliminating irrelevant data.

Data standardization is one of the foundations of machine learning. Both dimension and value of a predictor variable would make a great difference when evaluating the variable especially if predictor variables have highly varying magnitudes. Standardization prevents this by eliminating the effect of units and the variation of all numerical variables. Standardization of the data in this thesis refers to standard deviation standardization. This method makes each numerical feature have a mean and standard deviation of 0 and 1 respectively.

The standardization formula can be expressed as:

$$x^* = \frac{x - \bar{x}}{\sigma}$$

where x denotes the original data, $\bar{x}$ represents the mean, and σ is the standard deviation. [70]

Once numerical variables are standardized, categorical variables must be transformed into useful numeric variables. While a string representing an item is not compatible with model building, each unique item can become its own predictor variable. If the lot is comprised of that item, then the variable is equal to 1. Otherwise, it is equal to 0. These new variables are commonly referred to as dummy variables and are typically created for each categorical variable. Once concatenated onto the data, the original categorical variable is dropped.

Now that all of our predictor variables are standardized and numerical in nature, we begin eliminating variables that are not useful in predictive modeling. This enables us to reduce dimensionality and improve our prediction metrics. We do this by selecting statistically significant features, where importance is greater than zero in an ensemble of random trees. Packages such as scikit-learn in Python enable us to use both Random Forest and Gradient Boosting Trees to quickly calculate the importance of each feature. A graphical representation of the relative importance of 25 feature variables is shown in Figure 23.

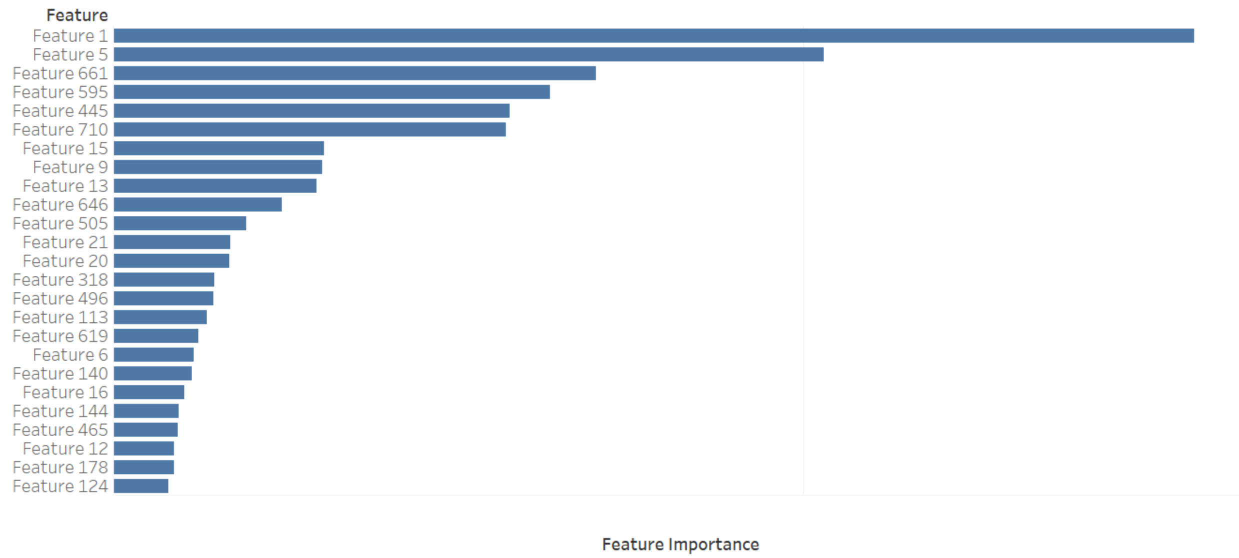

Figure 23: Example graph of 25 features and their relative importance to prediction

We also calculated Pearson's pairwise correlation coefficients and VIFs between each of the remaining variables and removed collinear features. When deciding between features to remove, the features with lower feature importance were chosen to be removed. A graphical representation of the correlation matrix is shown in Figure 24.

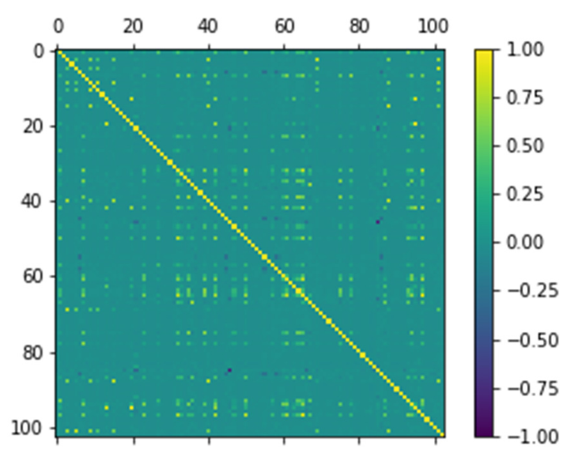

Figure 24: Example correlation matrix of approximately 100 of the feature variables

6.7 MODEL BUILDING, TRAINING, AND TESTING

As a result of data preparation, we were left with a list of approximately 100 relevant predictor variables for use in our machine learning models. Ten total predictive models without tuning were initially used for the prediction of cycle time: Ridge, Lasso, Elastic-Net, Support Vector Machine, Stochastic Gradient Descent, CART, KNN, Random Forest, GBT, and an Artificial Neural Network. Performance metrics were then used to select the top tier models. After the top tier models were selected, their specific hyperparameters were tuned and the models were built again, tested, and validated. The top model as determined by the same performance metrics was ultimately selected.

6.7.1 Performance Metrics and Down-Selection of Models

Performance of a model relates to its prediction capability on independent test data. We can use our given data set to mimic predictive performance on new data by sample splitting. We take our data set with n observations and split it into two mutually exclusive, disjoint, and exhaustive sets—a training set of size n_{train} and a testing set of size n_{test}. A 70%/30% split is often recommended and seen as reasonable. [71] After creating this partition, we build the ten models using only the training set and then evaluate its performance by how well is can predict the observations in the test set. We refer to performance metrics calculated using the training set as in-sample and metrics calculated on the test set as out-of-sample.

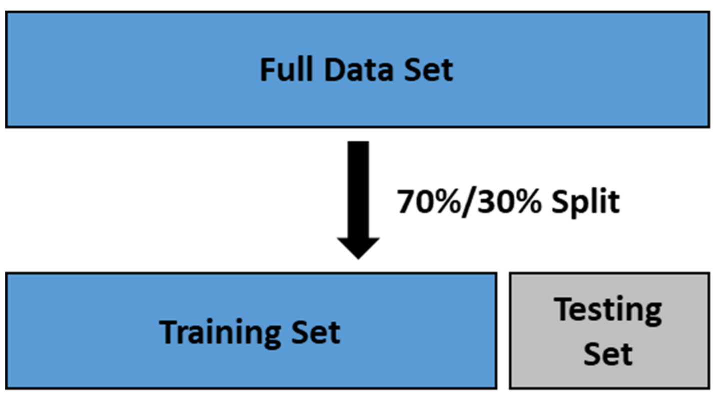

Figure 25: Splitting the full data set into a training set and testing set

Mean Squared Error

The primary performance metric we look at is the Mean Squared Error (MSE). Suppose that based only on our training set, we build a model and $\hat{f}(x)$ is the resulting prediction equation. For each n_{test} individuals, we have response variables y_i and predictor variables x_i and can generate predicted values $\hat{y}_1, \dots, \hat{y}_{n_{test}}$. [65] Out-of-sample MSE can, therefore, be defined as:

$$MSE = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} (y_i - \hat{y}_i)^2$$

Equation 37

Looking at the MSE of all models across all operations, we can begin to see what models are performing the best out of the ten. This metric ranges from 0 to ∞ with 0 being the best and is indifferent to the direction of errors. The MSE of a semi-random 15 operation subset of test sets is shown in Table 13. The subset is semi-random to ensure that operations related to machine-driven operations, labor driven operations, testing operations, and troubleshooting/rework operations were included. As expected, the two troubleshooting and rework operations have the worst error by a long shot.

MSE	Ridge	Lasso	EN	SVM	SGD	CART	RF	GBT	KNN	ANN
Operation 72	0.003	0.010	0.010	0.033	0.052	0.001	0.001	**0.001**	0.001	0.001
Operation 123	0.053	0.363	0.333	0.005	2E+01	0.019	0.013	0.016	0.015	**0.005**
Operation 82	0.094	0.346	0.346	0.440	1.147	0.054	0.040	0.049	0.044	**0.026**
Operation 47	0.614	1.942	1.560	**0.234**	8E+05	0.454	0.375	0.383	0.320	0.383
Operation 108	0.213	0.450	0.450	0.327	0.437	0.088	0.058	0.099	0.071	**0.034**
Operation 31	0.036	0.040	0.040	0.055	8E+03	0.065	**0.032**	0.034	0.034	0.034
Operation 40	0.483	0.970	0.970	**0.121**	2E+09	0.235	0.179	0.225	0.196	0.144
Operation 129	0.907	2.132	1.828	**0.031**	0.036	0.708	0.432	0.443	0.369	0.237
Operation 247	0.707	1.338	1.079	5.175	2E+06	0.581	0.427	**0.420**	0.514	0.482
Operation 119	5.721	7.793	7.377	**0.502**	0.698	4.654	3.095	3.501	3.417	2.640
Operation 141	4.912	6.375	5.690	4.574	4.980	5.408	3.314	2.858	2.910	**2.164**
Operation 73	6.976	8.352	8.250	5.079	7.058	6.242	5.046	4.994	5.853	**4.255**
Operation 274	1E+02	2E+02	2E+02	5E+02	4E+02	2E+02	1E+02	**1E+02**	2E+02	2E+02
Operation 190	4E+02	4E+02	4E+02	2E+02	**2E+02**	6E+02	4E+02	4E+02	4E+02	4E+02
	4E+01	4E+01	4E+01	5E+01	1E+08	6E+01	**4E+01**	**4E+01**	**4E+01**	**4E+01**

Table 13: Mean squared error for ten models across a subset of operations

Mean Average Error

The biggest problem with MSE is that the errors are squared before averaged, resulting in large errors having a high factor in the metric. This means that MSE is more useful when large errors are specifically undesirable. Mean average error (MAE) on the other hand measures the average magnitude of errors and is not subject to large errors holding more weight. As such, MAE is another useful metric we consider when choosing our models. Like MSE, MAE also ranges from 0 to ∞ with 0 being the best and is indifferent to the direction of errors. [65] Out-of-same MAE is defined as:

$$MAE = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} |y_i - \hat{y}_i|$$

Equation 38

Again, we see the same four models performing the best. The MAE values of the same 15 operation subset are shown in Table 14.

MAE	Ridge	Lasso	EN	SVM	SGD	CART	RF	GBT	KNN	ANN
Operation 72	0.019	0.033	0.033	0.090	0.079	0.004	**0.004**	0.004	0.004	0.008
Operation 123	0.077	0.425	0.406	0.061	1.145	0.016	0.016	0.027	**0.014**	0.034
Operation 82	0.181	0.392	0.392	0.203	0.623	0.075	0.075	0.092	**0.071**	0.084
Operation 47	0.413	0.990	0.870	0.169	3E+02	0.091	**0.086**	0.126	0.143	0.195
Operation 108	0.274	0.437	0.437	0.148	0.290	0.092	0.092	0.141	0.094	**0.090**
Operation 31	0.103	0.109	0.109	0.103	6E+01	0.123	0.096	0.095	**0.094**	0.098
Operation 40	0.369	0.586	0.586	0.143	6E+03	**0.140**	0.142	0.182	0.141	0.163
Operation 129	0.517	1.070	0.975	**0.098**	0.103	0.203	0.185	0.209	0.146	0.156
Operation 247	0.498	0.784	0.707	0.668	8E+02	0.324	**0.316**	0.329	0.346	0.349
Operation 119	0.985	1.449	1.399	**0.325**	0.490	0.413	0.401	0.666	0.363	0.381
Operation 141	0.908	1.126	1.025	0.577	0.926	0.500	0.482	0.562	**0.409**	0.477
Operation 73	1.633	2.038	2.013	0.952	1.685	**0.918**	0.969	1.058	1.303	1.051
Operation 274	8.346	8.490	8.671	9.044	1E+01	8.809	7.141	**7.013**	7.619	7.444
Operation 190	1E+01	1E+01	1E+01	**7.575**	8.333	1E+01	8.932	9.377	8.595	9.393
	1.851	2.124	2.098	1.440	5E+02	1.556	**1.353**	**1.420**	**1.382**	**1.423**

Table 14: Mean average error for ten models across a subset of operations

After looking at both metrics, it was clear that four models were best at predicting the response variable—Random Forest, Gradient Boosted Trees, K-Nearest Neighbor, and the Artificial Neural Network. All of which are not very interpretable. CART and SVM were the two best interpretable models; however, since our primary goal is predicting CT accurately, we do not continue to tune and analyze these models.

6.7.2 Model Tuning and Final Selection

Now that four models appear to provide us the best prediction, we need to tune their hyperparameters and use performance metrics to choose the best model. The parameters we are interested in tuning are found in Table 15.

We will go about tuning these variables through cross-validation. The goal of cross-validation is to test the model's predictive ability to predict new data that were not used in estimating it in order to prevent overfitting. The specific cross-validation method we use is k-fold cross-validation. Like before, we first split the set into a training set and a test set. We then divide the training set into k folds. For fold j, we use the other $k-1$ folds to construct a model and test that model on fold j. We repeat this for each of the k folds and for each value of a hyperparameter we are considering. We then fit the predictive model to all of the training data with the optimally tuned parameters and evaluate performance on the test set. Most commonly 5 or 10-fold cross-validation is used. A visual depiction of this can be found in Figure 26.

Hyperparameters Tuned	**K-Nearest Neighbor**
	Number of neighbors
Artificial Neural Network	Uniform or inverse distance
Number of hidden layers	**Random Forest**
Activation function used in hidden layers	Number of trees
Number of nodes	Max predictor variables per split
Momentum	Max tree depth
Decay	Min samples per split
Batch Size	Min samples per leaf
Number of iterations and threshold	Use bootstrapping or no
Weight optimization	Regularization penalty
Regularization penalty	**Gradient Boosted Trees**
Learning rate for weight updates	Learning rate of each new tree
Initial learning rate used	Similar to Random Forest

Table 15: Hyperparameters we look into tuning for each of the top four models

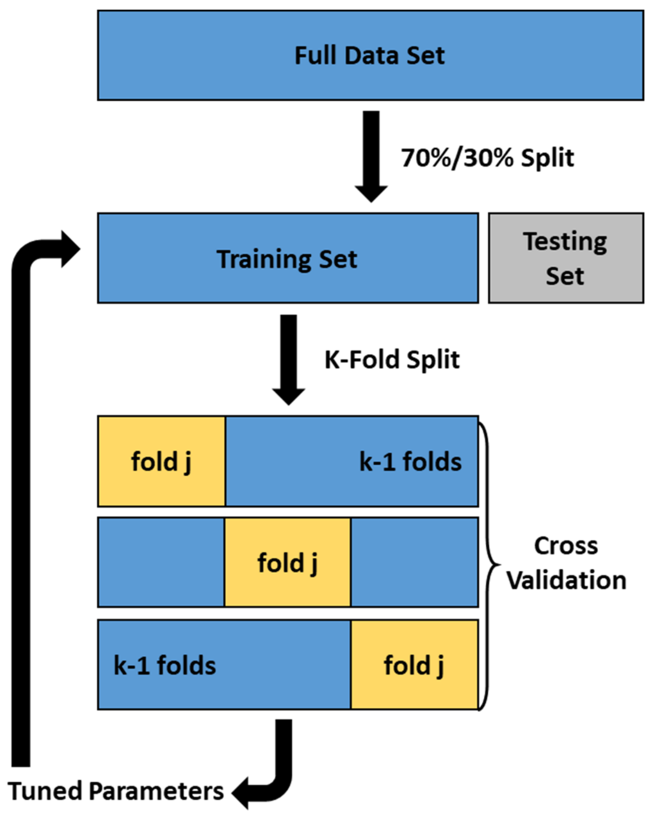

Figure 26: K-Fold Cross-Validation

K-Nearest Neighbor

We consider tuning only two parameters of our KNN model—the number of neighbors and the weighting scale of nearby trees. Because it only involves two parameters, we can see the effect of changing each through graphs as shown in Figure 27 and Figure 28.

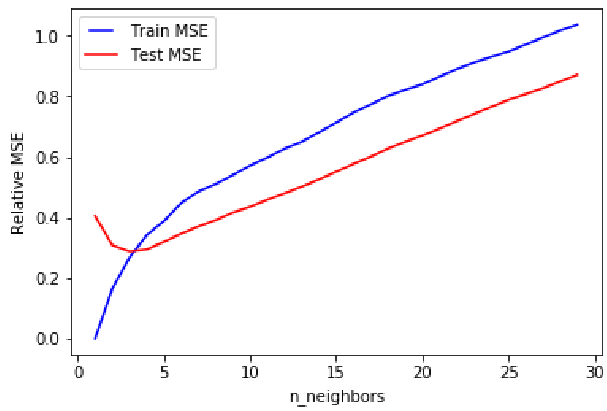

Figure 27: MSE as the number of nearest neighbors increases

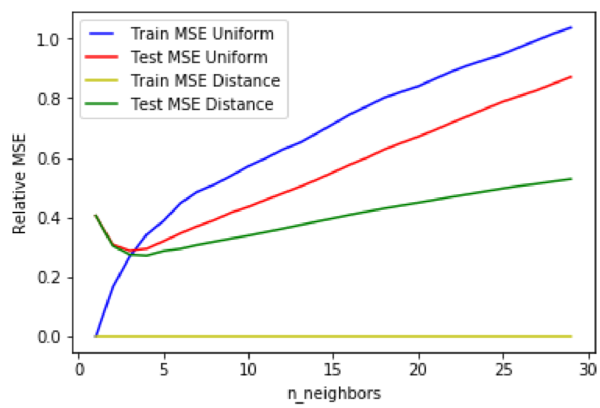

Figure 28: MSE according to weighting scale as the number of neighbors increases

From these graphs, it is shown that four neighbors appear to produce the lowest MSE and using weighting factors of the inverse distance from each sample further reduces MSE. This is verified through cross-validation. As a result of tuning, we were able to improve our MSE by 15.63% and MAE by 23.13%, where we define improvement as $\frac{MSE_{base}-MSE_{tuned}}{MSE_{base}} * 100\%$ and $\frac{MAE_{base}-MAE_{tuned}}{MAE_{base}} * 100\%$.

Random Forest, Gradient Boosted Tree, Artificial Neural Network

With Random Forests needing seven parameters tuned, a graphical interpretation is not as useful. Instead, we use a random search cross-validation technique followed by a full grid search cross-validation. To perform the random search cross-validation technique, we first define a grid of hyperparameter ranges, randomly sample combinations from the grid, and perform k-fold cross-validation with each combination. The randomization enables us to save hours of computing time by not testing every single combination and the cross-validation reduces the chance of overfitting. [72] [73] Once the set of best parameters is defined from the random search cross-validation, we can then concentrate our final grid search cross-validation on a smaller set of hyperparameters. As a result of tuning, we were able to improve our MSE by 17.6% at the cost of reducing our MAE by 15.7%. MSE improved because it was the performance metric that was optimized.

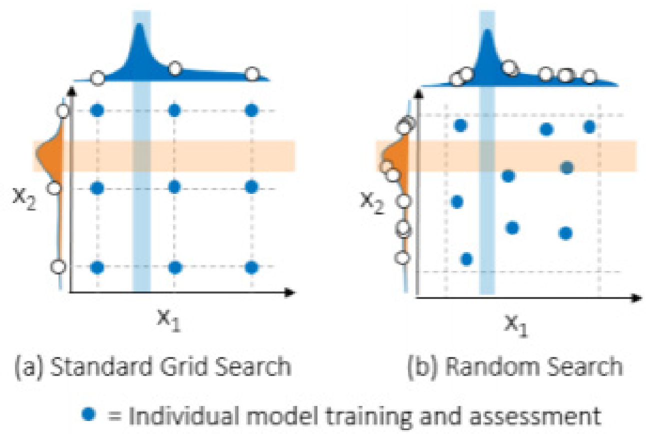

Figure 29: Standard grid search vs random search [73]

A similar approach to the tuning of the Random Forest hyperparameters was used for the GBT and ANN. As a result of tuning, we were able to improve our MSE by 13.8% and MAE by 8.6% for the GBT and 11.1% and 20.6% for the ANN.

6.7.3 Final Selection

After tuning the hyperparameters, the Random Forest model was shown to have the best overall performance on average across operations when measured by MSE. Final overall performance metrics across a random subset of the 326 operations with known demand in the next five years are shown in Table 16. This test set included 273 of the 326 operations and all 41 key operations.

Method	MSE	MAE
RF	62.45	1.55
GBT	65.09	1.44
KNN	70.73	1.11
ANN	63.18	1.41

Table 16: Final overall performance metrics across operations with known demand for the top four models

Using this RF, we can now test our hypothesis that CT can be more accurately predicted via supervised machine learning methods than the currently implemented method.

CH. 7 RESULTS AND DISCUSSION

This chapter discusses the results which demonstrate the achieved predictive and optimization performance of the capacity planning model. We begin this chapter by evaluating the three major components of the strategic capacity planning optimization model—static capacity utilization and OEE calculations, the machine learning model to predict CT, and the MINLP optimization model. This is followed by the business impact of the final model before discussing its limitations.

7.1 EVALUATION OF THE STRATEGIC CAPACITY PLANNING MODEL

This section evaluates each component of the strategic capacity planning optimization model individually. First, results and features from the static capacity utilization and OEE model are demonstrated. Next, results from the machine learning model are discussed and compared against the currently used predictions. Lastly, the output results and sensitivity from the MINLP optimization model are evaluated.

7.1.1 Static Capacity Utilization and OEE Model

As shown in 4.4 STATIC CAPACITY UTILIZATION AND OEE RESULTS, we have successfully been able to graphically represent capacity utilization through automated data mining algorithms. By taking confirmed demand, projected demand, and the current production plan to meet both, we can add manufacturing metrics from transactional and IIoT data to calculate current capacity utilization. Two screenshots from the model are shown in Figure 30 and Figure 31. Each shows an operation's capacity utilization and OEE as it stood when the model was opened.

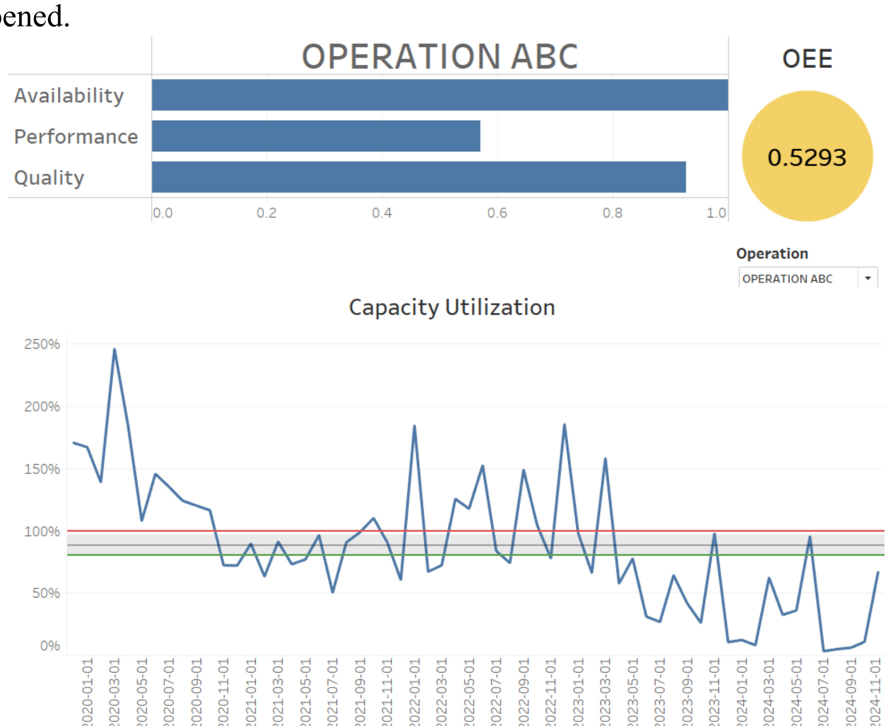

Figure 30: Screenshot of capacity utilization and OEE model for Operation ABC

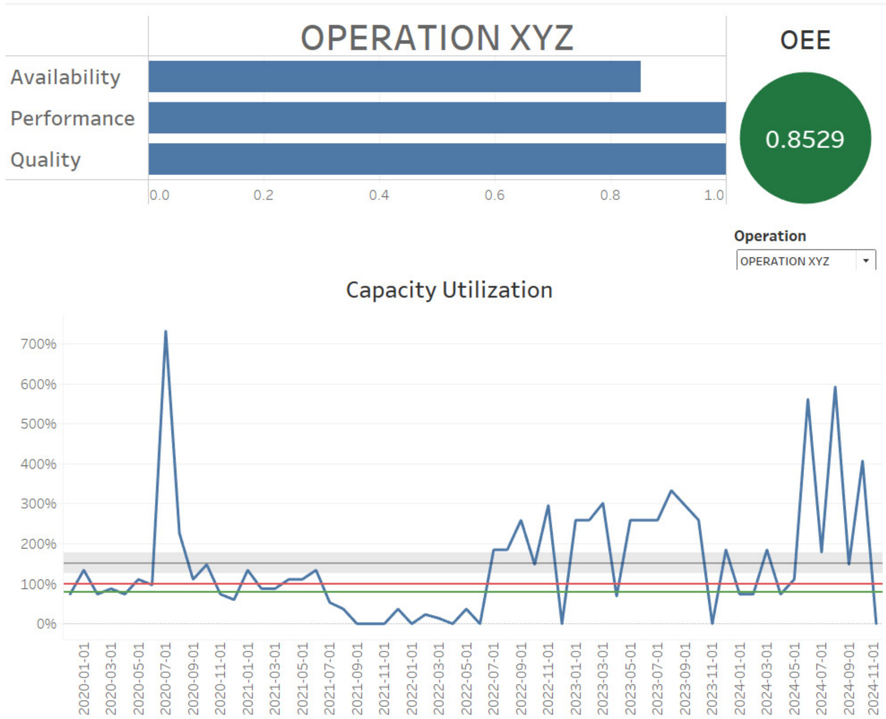

Figure 31: Screenshot of capacity utilization and OEE model for Operation XYZ

Other features of the model can also be enabled to get more information or a clearer picture. For example, Figure 32 breaks down confirmed demand and predicted demand on the same operation as Figure 31. This can be used to assess what is primarily driving capacity utilization in a given month.

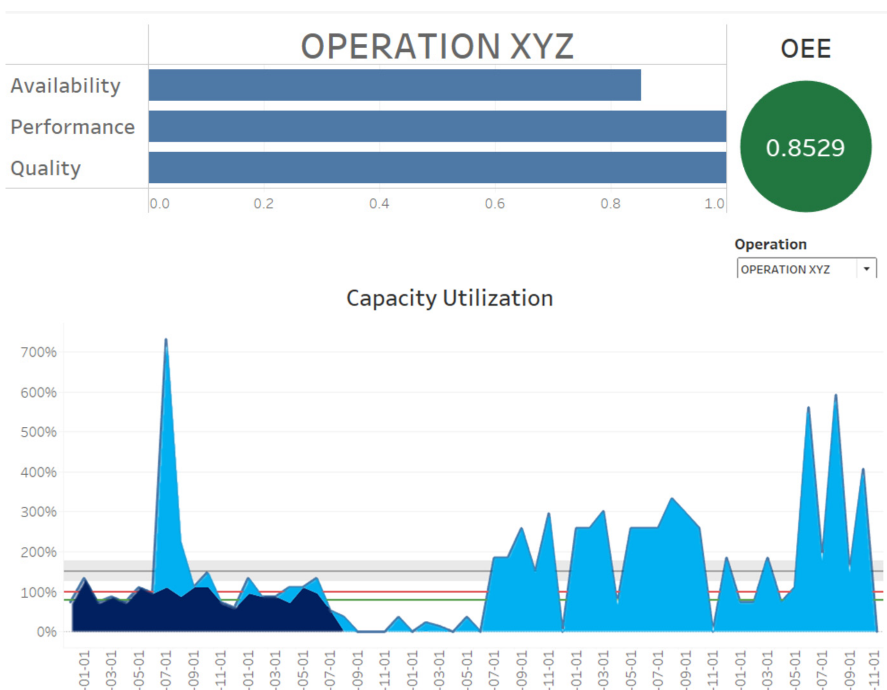

Figure 32: Screenshot of capacity utilization with breakdown of confirmed demand and predicted demand

On the other hand, Figure 33 shows an overview of OEE across selected factory operations that can be used by factory managers to compare operations.

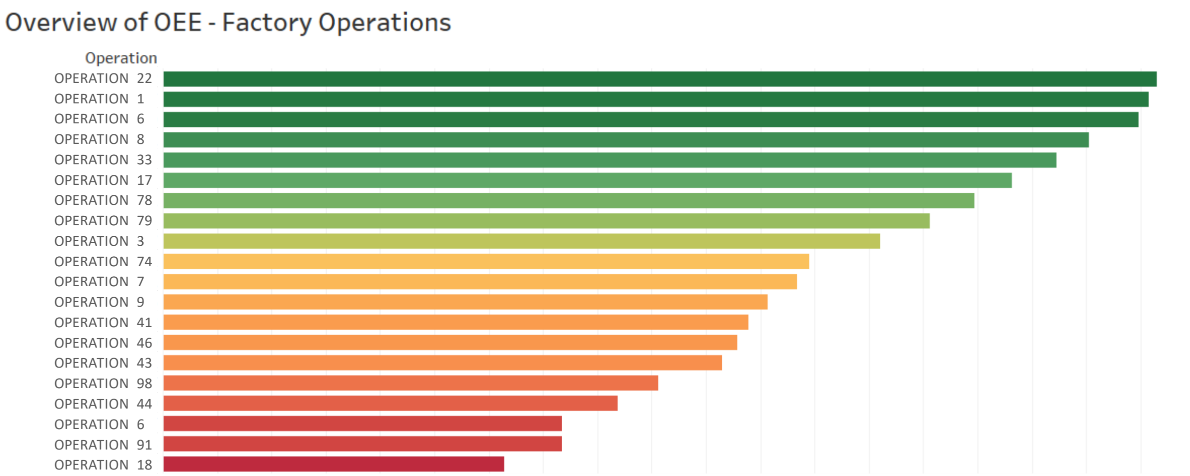

Figure 33: Screenshot of an overview of OEE across selected factory operations

While we were unable to verify our OEE metrics against any current company metrics, we were able to verify our capacity utilization model. More specifically, we were able to compare our capacity utilization model against the most recent manual capacity modeling effort. When similar assumptions are made for the number of machines, first pass yield, availability, and the number of shifts, both models yield similar results and we are statistically unable to differentiate them.

7.1.2 Machine Learning Model to Predict CT

After completing our machine learning framework as described in 6.5 MACHINE LEARNING FRAMEWORK OVERVIEW, we found that a random forest model provided us with the best predictions, on average, when measured by MSE, which checks with previous research as discussed in 3.2 MACHINE LEARNING FOR CYCLE TIME PREDICTION. In order to test our hypothesis that CT can be more accurately predicted via supervised machine learning methods than the currently implemented method, we must compare the predictive power of both.

Using the same test set data that looks across a random subset of 326 operations with demand in the next five years, we can find how accurate our model is compared to the company's currently employed static predictions. As seen in Table 17, the overall average MAE is 76.3% lower for the RF models and the overall average MSE is 80% lower for the RF models, showing the predictive power of using the RF predictions over the current prediction. In Figure 34, we can more easily see that the RF models have less error than the current static predictions across our test set. The error associated with all 273 operations of the test set can be found in APPENDIX: RANDOM FOREST MODEL PERFORMANCE AND SENSITIVITY.

	Random Forest		Currently Used Predictions	
	MSE	MAE	MSE	MAE
Operation 1	0.671434457	0.271687557	46.33640098	3.979033876
Operation 2	0.424853893	0.270926202	12.21947414	2.497983742
Operation 3	13.47601789	1.68818025	89.31287805	6.011059015
Operation 4	0.574243785	0.23300326	79.21263658	7.313318589
Operation 5	0.030928201	0.017076563	6.128977374	2.35767552
…	…	…	…	…
Operation 269	16.07593397	0.787887034	173.1794712	6.512791979
Operation 270	4.554396031	0.811519624	12.58125257	1.715680081
Operation 271	2.635523458	1.081666921	7.303275674	1.652077124
Operation 272	25.57065526	1.820699159	486.9145805	16.27615539
Operation 273	0.87211642	0.699639208	0.801467026	0.732636753
Overall Average	**62.45**	**1.55**	**314.17**	**6.54**

Table 17: Random Forest vs Current Prediction Error Metrics

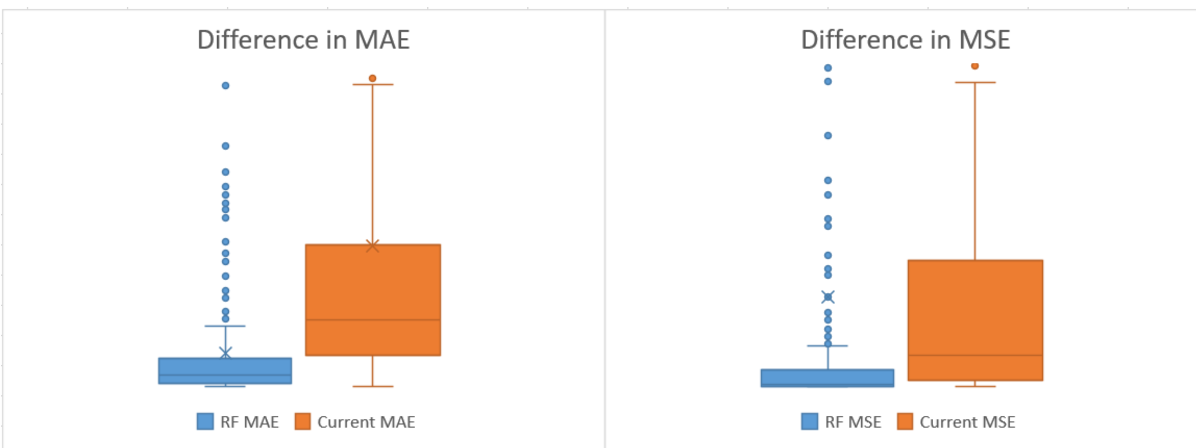

Figure 34: A graphical depiction of the difference between the new RF and Current Prediction Errors

Since we were deliberate about eliminating some features and selecting relevant features to include in the model in 6.3 FEATURE SELECTION, it would follow that the features that ended up in the model were useful for prediction. Ultimately, this process should have led to better performance on the test set by avoiding overfitting. In this section, we remove a few features we hypothesized to be important and compare the average error of the new model without a specific feature to the full model.

The three features we chose to remove, one at a time, were Date Info such as day of the week, Queue Data such as length of queue, and Historical CT Info which provided the most recent processing time, queue time, and cycle time. We chose these features due to having the highest feature importance as described in 6.3 FEATURE SELECTION. As expected, the MAE and the MSE increased when these features were removed as shown in Figure 35 and Figure 36. We can also see that the Queue Data and Historical CT Info, which had the highest feature importance, caused the largest increase in model error when removed.

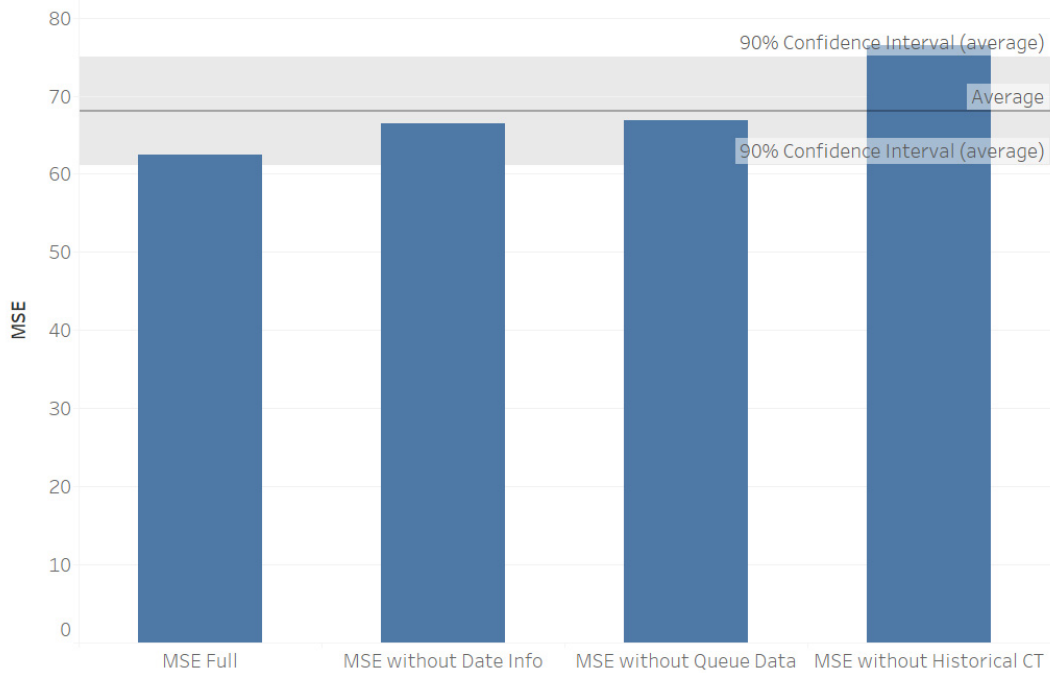

Figure 35: Comparison of model MSE after removing specific features

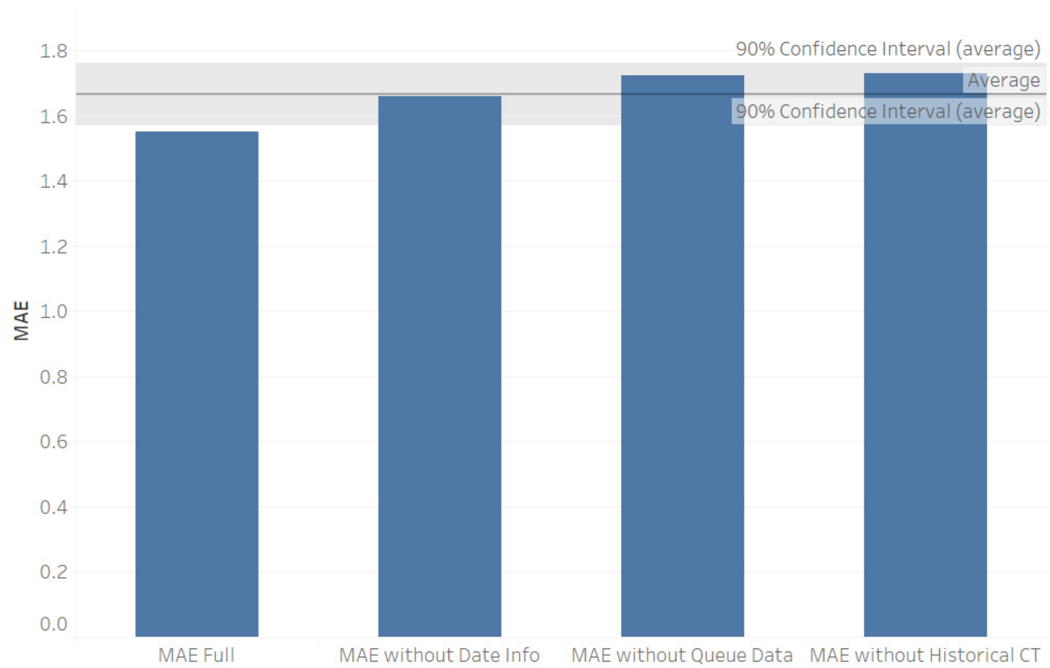

Figure 36: Comparison of model MAE after removing specific features

7.1.3 MINLP Optimization Model

Following a similar approach to solving as described in 2.5 STRATEGIC CAPACITY PLANNING but using the model defined in 5.5 MIXED-INTEGER NONLINEAR OPTIMIZATION, we can solve for key management decisions with respect to strategic capacity planning. While the MINLP takes many hours to solve, it can be solved to optimality within a tolerance of 1.00e-04. The total cost of production, machines, and workforce as defined in the objective function is minimized. From this minimization and the constraints placed on the model, eight operations required additional machines/workstations, production had to be shifted, and operational inventory had to be used.

Evaluating Model Output

In the final model solved, we narrowed the operations to the 41 key operations as discussed in 5.5 MIXED-INTEGER NONLINEAR OPTIMIZATION. To further reduce the size of the model, we also decided to set OEE metrics to their original values, making the assumption that they will remain constant over the next five years. Lastly, we assumed that the length of shifts (ls) would remain constant. Under these assumptions, our model outputs ten pieces of valuable information: $sh_{ot}, wd_{ot}, m_{ot}, b_{ot}, wf_{ot}, h_{ot}, f_{ot}, pr_{iot}, in_{iot}$, and the $Objective\ Value$. In other words, the model outputs how many shifts and workdays an operation should have, how many machines/workstations we need to add and when, how many people we should hire or fire and when, how much of each item we should make and when, how much of each item we should make ahead of time and hold in inventory and when, and an estimate of how much all of these decisions will cost us.

In Figure 37, one can see a visible depiction of the model's output over the 60-month time period analyzed. At the top, the original CU as it stands today is shown. Note that the CU goes over 200% on multiple occasions. Below the original CU is the optimized CU. Due to the constraints of the model, CU never surpasses 100%. Finally, below the CU graphs, decisions required to be made to make the optimized CU a reality are shown. From Figure 37, one can see that there are five primary management decisions required to obtain the optimal strategic capacity plan for this single operation.

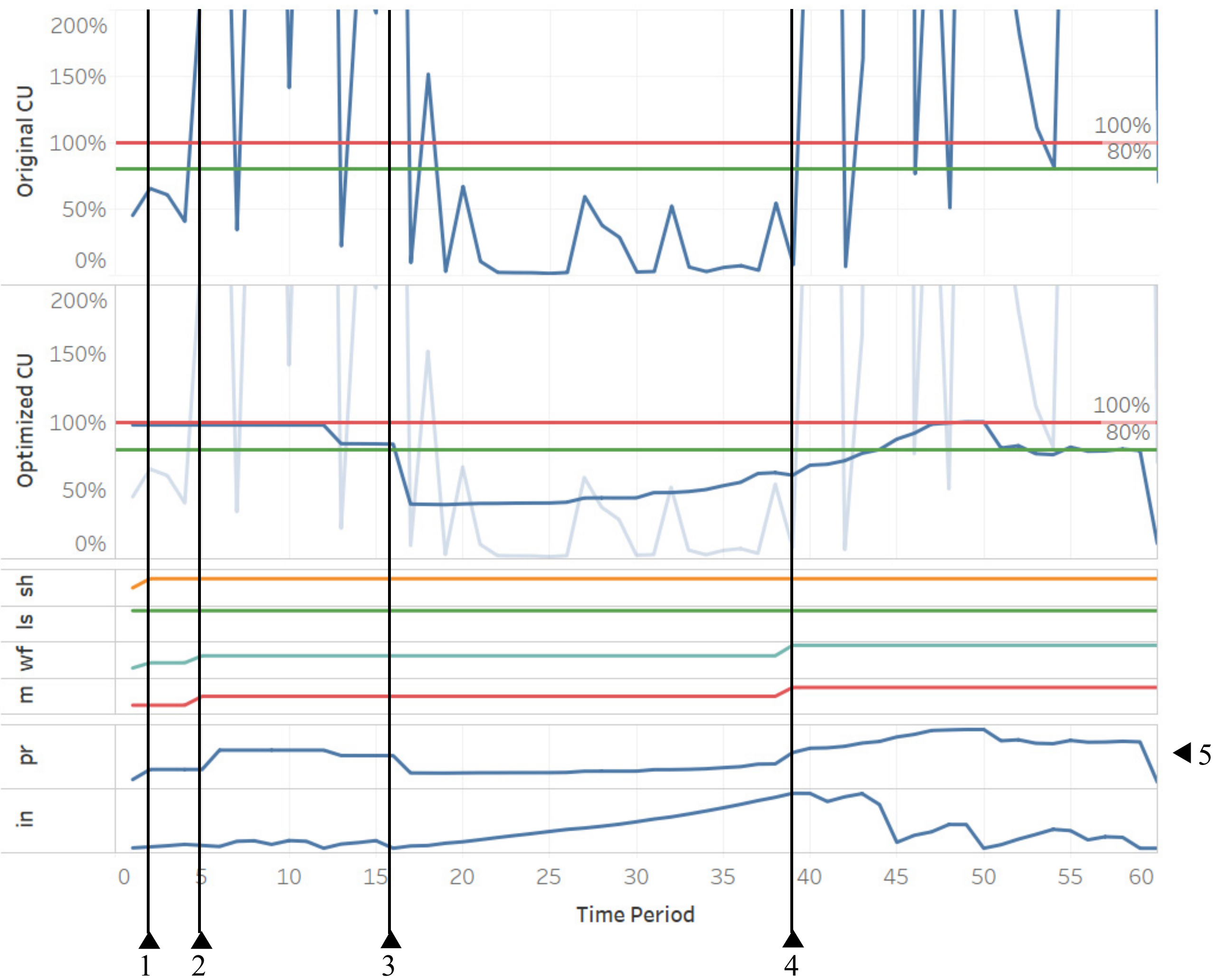

Figure 37: Visual depiction of optimization model results for a single operation

The five management decisions required are:

1. Increase number of shifts and required workforce to fill those shifts as soon as possible.
2. Add a second machine and increase workforce to operate the new machine by month 5.
3. Begin holding inventory for anticipated demand jump after month 39.
4. Add a third machine and increase workforce to operate the new machine by month 39.
5. Relatively flatten production over time by following newly optimized production plan.

Some other management decisions that do not come from the model, but should be thought about:

1. Be extra cautious during months when capacity utilization is at 100%. Consider increasing workforce or performance metrics at operation, which is currently 90.88%. If uncomfortable with 100% utilization, consider the tradeoff to further constrain CU.
2. If another machine is needed by month 39, how long does it take to acquire, install, and test the machine before production can begin on it? If it takes three months, then the process should begin by month 36 at the latest.

3. With CU dipping for a few months before increasing again, can we better utilize our workers or shift their operation without laying them off? Our current model has a very high cost associated with firing so the model does not suggest it, but it remains an option.

While the model provides an optimal baseline for strategic capacity planning, there are many more decisions that are needed to make the optimal baseline a reality. The model does not have the functionality to make all of them and has multiple limitations. These limitations are discussed further in 7.3 MODEL LIMITATIONS.

Evaluating Model Sensitivity

If integer constraints are relaxed, we see a ~9% decrease in the objective function and a ~4% decrease in the number of machines/workstations needed. We also see that the time to solve it reduced by almost 98%. If the company instead chooses to maintain its current production schedule and increase capacity by only adding machines/workstations, then the objective function increases by 376% as shown in Table 18. Again, this model is much faster to solve than the base model, but it requires much more capital to meet demand.

	MINLP (base)	**Relaxed - NLP**	**MINLP (in=0)**
Total Cost	0%	-8.89%	376.38%
# Machines/Workstations	0%	-4.41%	78.19%
Total Inventory Held	0%	-0.70%	-100.00%
Time to Solve Model	0%	-97.51%	-90.39%

Table 18: Percent change in cost, machines, inventory and time across various models

Table 18 displays some interesting results. The first is that shifting production has a huge cost savings benefit. The second is that removing the integer constraints does not change the optimal solution by much; however, it does solve in a much faster time, relatively.

Our original model constrained capacity utilization to be less than or equal to 100%; however, it may be in the company's best interest to restrict this to a different value to account for variability in demand. Table 19 shows the percent changes in cost, number of machines/workstations required, inventory level, and time to solve as CU is restricted to different levels.

	Max CU Allowed				
	100%	**95%**	**90%**	**85%**	**80%**
Total Cost	0.00%	13.72%	24.03%	41.18%	48.08%
# Machines/Workstations	0.00%	6.50%	11.49%	19.93%	23.52%
Total Inventory Held	0.00%	-4.92%	4.20%	4.29%	16.84%
Time to Solve Model	0.00%	-11.52%	95.88%	-32.38%	-36.57%

Table 19: Percent change of select decision variables as capacity utilization constraint is varied

As expected, total cost and the number of machines/workstations required went up as CU was further restricted. Total inventory also went up when CU was restricted to 90%, 85%, and 80%, but went down when restricted to 95%. We believe this initial reduction is due to the original added flexibility from the additional machines/workstations. Time to solve the model also generally went down as more resources and inventory were added; however, it went up when CU was restricted to 90%.

While we made the decision to assume OEE metrics would maintain their original values and not improve or deteriorate, we can look at how our model outputs would change if OEE did change. As expected, in Table 20 we see that cost and number of machines/workstations are reduced when OEE metrics improve and increased when OEE metrics deteriorate. For example, if we can achieve a 5% improvement in OEE across all key operations, we would be able to lower the cost by 13.73% and the number of machines/workstations required by 7.13%. The management questions are then: 1) Is it worth investing capital to improve OEE by 5%? and 2) Is a 5% improvement possible? This changes depending on where OEE originally sits. It's much easier to achieve a 5% improvement when OEE is at 50% than when OEE is at 85%.

	Change in OEE Metrics				
	-10%	**-5%**	**Base**	**+5%**	**+10%**
Total Cost	44.56%	20.57%	0.00%	-13.73%	-34.29%
# Machines/Workstations	21.98%	9.95%	0.00%	-7.13%	-17.14%
Total Inventory Held	6.04%	2.39%	0.00%	-16.20%	-10.90%
Time to Solve Model	5.64%	-71.78%	0.00%	11.56%	-51.66%

Table 20: Percent change of select decision variables as OEE metrics are changed

7.2 BUSINESS IMPACT

The benefits of automating capacity and OEE modeling, more accurately predicting CT, and optimizing strategic capacity utilization mostly come from the enhanced visibility into the company's operations; however, we can financially estimate some of these benefits to demonstrate the potential business impact of the models. The performance results of the models with several assumptions will be used to estimate business impact.

While the actual savings of the models require proprietary knowledge of the company's cost structure and financial positions, we can get an estimate through public information and assumptions. If we assume that only 1% of the total 2018 capital expenditures (CAPEX) for IDS is used to increase capacity of the CCA factory by increasing the number of machines, we can use the percentage change of cost to assess the savings. From Table 18, we know that it requires 376% more in capital to increase capacity utilization by increasing machines only. With 1% of 2018 CAPEX equaling $2.42M, we can expect this model to annually save the company an estimated $1.78M. [74] We consider this the maximum expected savings since the company does not only increase capacity by increasing the number of machines.

Since the capacity utilization model is automated, it enables us to reassign a portion of the team of engineers working on capacity utilization to work on something different. It cannot reduce the entire team due to monitoring and maintenance; however, we can conservatively assume that this model can reduce the number of engineers by two. If these engineers both had $100k salaries, the model saves the company $1.94M in perpetuity, assuming a 10.29% discount rate. [75]

Altogether, the implementations of all three individual model components provide the opportunity to automatically monitor capacity utilization and OEE across the factory, predict CT of lots more accurately, and strategically optimize capacity planning. Under numerous assumptions, the potential business impact for Raytheon is $1.98M or .0384% of the IDS operating cost per year. [74] Nevertheless, there are additional long term and secondary benefits of the model that are even harder to estimate. Savings from the model's ability to provide more accurate CTs for planning purposes, pinpoint operational inefficiencies through OEE metrics, and other secondary uses were not included in the estimation.

7.3 MODEL LIMITATIONS

As with all models, our model comes with multiple limitations that must be known to be able to make informed decisions relating to strategic capacity planning. In this section, we discuss some of these limitations. We begin with limitations on the input side of the model and end with limitations on the output side of the model.

7.3.1 Input Limitations

A commonly thrown around phrase when discussing models is "garbage in, garbage out." Bad data can severely limit the ability of a model. While all data is susceptible to errors, the transactional data in our model is especially susceptible. An operator can log into a lot and forget to sign out, artificially inflating production metrics associated with that lot, or an engineer can put a lot on hold without properly documenting the reason, hiding the fact that the lot is expected to behave differently. Even though our model does some outlier filtering, many of these errors will inevitably hinder the accuracy of our model.

Our model also relies on automated data mining of currently available databases. Therefore, it is limited by the continued availability of these data. If these data are not continuously updated and maintained, then the model will lose its functionality. Furthermore, if the database structure or location is changed then the model will also lose its functionality.

We also make many assumptions throughout our model. One of the biggest assumptions we make is that predicted demand retrieved from Kinaxis is accurate. While we know this is not true, it is the only data available for long term capacity planning. Understanding that demand is variable and will most likely not equal this prediction must be acknowledged.

7.3.2 Output Limitations

In regards to the random forest model used to predict CT, we chose a model based on prediction ability, deliberately putting interpretability as a secondary priority. Although this gave us a lower predictive error, losing interpretability does limit our ability to pinpoint why a CT is higher or lower for a given lot/item.

In regards to the optimization model, we were required to significantly reduce the operations our model optimized for due to the limited computing power we have, the complexity of the factory, and the nonlinear aspect of the model. Furthermore, while the optimization model outputs planning decisions such as when a machine needs to be added, it does not incorporate the time required to acquire, install, and test the machine before production. Similarly, it also does not incorporate the training time required for new employees or the ability of one employee to work at multiple operations.

CH. 8 CONCLUSION AND RECOMMENDATIONS

"All models are approximations. Assumptions, whether implied or clearly stated, are never exactly true. All models are wrong, but some models are useful. So the question you need to ask is not 'Is the model true?' (it never is) but 'Is the model good enough for this particular application?'"

–George Box et al. [76]

The optimization model developed in this thesis combines the capability of automated data mining algorithms, the predictive power of machine learning, and the optimization ability of mathematical programming for strategic capacity planning. This chapter provides an overview of the principal findings, recommendations for implementation, as well as suggested areas for future work and applications.

8.1 SUMMARY OF FINDINGS AND CONTRIBUTIONS

We developed a capacity visualization tool that mines millions of pieces of demand and historical manufacturing data from current sources to calculate capacity utilization under the current production plan and OEE for operations in the CCA factory. In order to calculate capacity utilization, we first calculated capacity available and capacity required. In order to calculate OEE, we first calculated its three parts—Availability, Performance, and Quality. Using SQL queries embedded in scheduled, executable Python scripts, we automated the retrieval of required data and these calculations. By enabling all of this automation to happen in the background without the need for user action and using a commonly known interface such as Tableau, we were able to lower the learning curve and assist in the adoption of the tool. We demonstrate the visual output of this model in 4.4 STATIC CAPACITY UTILIZATION AND OEE RESULTS and 7.1 EVALUATION OF THE STRATEGIC CAPACITY PLANNING MODEL.

We also developed 326 machine learning models to predict the cycle time of items moving through each operation in the CCA factory that has expected demand in the next five years. Transactional and automated manufacturing data along with material and component data were ingested to generate features to be used for prediction such as queue length and historical CT, which we demonstrate to enable a 6.57% and 18.29% decrease in MSE, respectively. Given the large number of features available to be incorporated into the models, we chose to use feature selection before training our models and evaluating them against test data. The best model type, on average, was found to be a random forest. Across the randomized 273 operations used in the test set, it was found to have an MSE of 62.45 and MAE of 1.55, on average. Both metrics outperform the current predicted CT when evaluated on the same test set. The overall average MAE is 76.3% lower and the overall average MSE is 80% lower. Therefore, we conclude that CT can be more accurately predicted via supervised machine learning methods than the currently implemented method. More on these results can be found in 6.7 MODEL BUILDING, TRAINING, AND TESTING and 7.1 EVALUATION OF THE STRATEGIC CAPACITY PLANNING MODEL.

Lastly, we developed an optimization model that integrates the capacity utilization and machine learning models in its constraints to provide us with recommended decisions for resource and demand planning. At first, we started with a simple LP model and added constraints to match key decisions Raytheon needs for capacity planning. Our final model became a MINLP with appropriate constraints and an objective function that minimized costs associated with production, inventory, workforce, and machine/workstation planning. After solving the model to optimality, we obtained recommended decisions for resource and demand planning to minimize the overall cost associated with strategic capacity planning. Across the 41 key operations defined by subject matter experts, we found that eight of them required an increase in resources when we also enable production planning and inventory. We demonstrated that not allowing production and inventory optimization increased cost by 376.38% and the number of resources by 78.19%. We also demonstrated that improving OEE metrics by 5% reduced cost by 13.73% and the number of resources by 7.13%. More on these results can be found in 7.1 EVALUATION OF THE STRATEGIC CAPACITY PLANNING MODEL.

The results obtained confirm that models developed enable identification of capacity utilization across operations in the factory and provide recommendations on the sequence and the timing of machine purchases and workforce adjustments for strategic capacity planning. We can then design systems to monitor and improve operations over time, more accurately assess demand due dates and schedule resources, and make better capacity planning decisions involving trade-offs between finance, output, and risk. Under several assumptions, the potential business impact is $1.98M as estimated in 7.2 BUSINESS IMPACT.

8.2 RECOMMENDATIONS FOR IMPLEMENTATION

The models built enable Raytheon to make better decisions when planning factory capacity in the long term and get a clearer picture of operational health in the short term; however, they are required to be properly implemented and scaled to have a sustainable effect on the company. Even if a model or tool is capable of greatly benefiting the company, poor implementation can result in these benefits not being realized. This section analyzes the organization through a strategic lens, political lens, and cultural lens to better understand how to enable a more successful implementation effort. It then summarizes the findings.

8.2.1 Three Lens Analysis for Implementation

Strategic Lens: The structure of Raytheon is complex as the company has different aspects of divisional and functional structuring. The overall company is divisionally structured with four primary business units as discussed in 2.2 RAYTHEON COMPANY OVERVIEW. These business units operate relatively independently with different presidents. Within these business units, the structure becomes very functional and hierarchical. The main functional areas are sales, operations, engineering, supply chain, and finance. While the modernization and innovation group works across functional areas within IDS, the business unit is siloed overall with communication primarily occurring through formal meetings in conference rooms.

One thing that is not siloed and aligns the business unit, however, is the financial incentive. Each employee receives a percentage of her/his base pay as an annual bonus. This percentage is directly tied to how well the business unit did in each measurement of bookings, net sales, operating income, and free cash flow goals. Due to the siloing of functional areas and the incentive structure, rolling out a new model will need to be done to each area individually with specifics of how this will move the needle on the most relevant measurement to that area. How the models are described to employees in each area greatly influences its adoption rate. Describing what is in it for them and how it will help solve their pain points will increase the adoption rate.

Political Lens: Raytheon is very traditional with power coming primarily from formal positions. The hierarchical organization provides a good guide to the vertical power system. Those at the top have the power of processes such as resource allocation, information flows, employee performance evaluation, task assignment, etc. Those in management positions also have additional financial incentives aligned to their specific areas, tying the political and strategic lens together. These key influencers that can greatly benefit from a capacity model and also have the ability to influence the people that work with them are vital for successful model implementation. By persuading them to adopt the new model, they can catalyze the promulgation and sustainment of it. Furthermore, while employees naturally resist change, some are less resistant than others. Identifying and giving these more innovative employees access to the models first will enable them to learn and teach other employees.

Cultural Lens: Depending on where a person goes in Raytheon, he/she may find himself/herself overdressed or underdressed. In the finance and sales area, slacks are generally worn. On the other side of the site, where manufacturing occurs, jeans are often the staple outfit. These vastly different work environments lead to various subcultures within a single site. One thing that spans all areas, however, is the belief that each employee is helping our nation's military and making the world a safer place. Due to this cultural environment, the implementation of a new model needs to be communicated slightly differently depending on the subcultural norms of the area. For all areas, however, sharing how the model will make the world a safer place will go a long way.

8.2.2 Implementation Summary

Based on the above three-lens-analysis, three key actions need to be completed to effectively implement and sustain a new model. From the strategic lens analysis, one needs to roll out the model in each area of Raytheon individually and explain how their incentives align with the success of model implementation. From the political lens analysis, one needs to get support from key influencers to promulgate it. Lastly, from the cultural analysis, one needs to communicate effectively based on each subculture.

Sustainable implementation also requires action to overcome some limitations of the model in its current state. For example, the current data sources need to continue to be updated and maintained in their current structure. Operators also need to be trained on how important it is that they log accurate information. Additionally, adding more automated data collection systems to key operations would enable more accurate analyses. Finally, adopting any new model and tool can have a significant impact on how some employees work, so their opinions matter. While the tool was developed in Tableau, a currently used user interface, having users provide feedback and continuously adapting the tool to their feedback and needs will make employees more motivated and engaged with the tool.

8.3 FUTURE RESEARCH AND APPLICATIONS

We were successful in combing data mining algorithms, machine learning methods, and optimization models; however, there are multiple actions that can be done to better apply our models and several assumptions and simplifications that can be challenged in order to develop better models.

In order to better apply the developed models, the following actions are proposed. First, the models and scripts should be moved from a local machine to a server with better computing power. Second, generated feature data should be stored on a central database so that calculations are not required each time the model is opened. Third, automated scheduled runs of the model to not interfere with current work should be created. These runs should include retraining the machine learning models with new data to maintain accuracy over time. The recommended interval for updating the capacity visualization data is daily so that it can be used to monitor factory health, while it is recommended that the machine learning and optimization models be updated monthly to be used in S&OP meetings. Fourth, the optimization model currently uses Julia/Jump, which is not commonly used at Raytheon. Adapting the code to Python could help with maintenance and sustainability of the model. Finally, a better front end with an easy-to-use graphical user interface (GUI) should be added to enable better use of the models.

In order to advance the developed models, the following work is proposed. First, we recommend either linearizing the optimization model or using an advanced computer to solve for all 326 operations with known demand in the next five years. While we only optimized for 41 key operations, there is a possibility that the factory becomes capacity limited by an operation that was not classified as a key operation. Second, we assumed that predicted demand was correct and did not perform simulations to test for certain demand variability. We recommend running the models through variability testing. Third, we decided to choose only one final machine learning method for CT prediction across all operations for simplicity purposes. Nevertheless, some operations perform better with other machine learning methods. It is conceivable to choose a different machine learning method for each operation so that error is minimized even further. Fourth, we assumed that the nonlinearity was necessary to gain sufficient capacity planning knowledge; however, if fewer decision variables are desired, the model can be converted to linear and thus would solve much faster. We also implemented an integer constraint to make the model more realistic; however, we demonstrated that the relaxed model solves much faster and does not greatly affect the results. Finally, due to the model's ability to update itself when a new operation or item is created, we believe that this same model could be deployed to areas outside of the CCA factory.

REFERENCES

[1] Deloitte, "2020 Global Aerospace and Defense Industry Outlook," 2019.

[2] D. Canning, "Defense Industry Technology Innovation," Jabil, 2020. [Online]. Available: https://www.jabil.com/blog/defense-industry-technology-innovation.html.

[3] M. Casavant, "Predicting Competitor Restructuring using Machine Learning Methods," *MIT LGO Thesis,* 2019.

[4] United States Government Accountability Office (GAO), "Weapon Systems Annual Assessment," 2019.

[5] M. Thompson, "The Incredibly Shrinking Defense Industry," Project on Government Oversight, 2019. [Online]. Available: https://www.pogo.org/analysis/2019/08/the-incredibly-shrinking-defense-industry/.

[6] A. Macias, "American firms rule the $398 billion global arms industry: Here's a roundup of the world's top 10 defense contractors, by sales," CNBC, 10 January 2019. [Online]. Available: https://www.cnbc.com/2019/01/10/top-10-defense-contractors-in-the-world.html.

[7] MarketLine, "Raytheon Company SWOT Analysis," 2019.

[8] "Raytheon Business Units," Raytheon, 2019. [Online]. Available: https://www.raytheon.com/ourcompany/businesses.

[9] "Raytheon and United Technologies Aerospace Businesses to Combine in Merger of Equals," UTC, 2019. [Online]. Available: https://www.utc.com/en/news/2019/06/09/raytheon-and-united-technologies-aerospace-businesses-to-combine-in-merger-of-equals.

[10] C. A. Harper, Handbook of materials and processes for electronics, New York: McGraw-Hill, 1970.

[11] B. White, "Through Hole vs Surface Mount," Freedom CAD Services, 26 July 2018. [Online]. Available: https://www.freedomcad.com/2018/07/26/through-hole-vs-surface-mount-which-should-you-use/.

[12] "Through Hole vs. Surface Mount Technology," PCBoard, 2015. [Online]. Available: http://www.pcboardrework.com/through-hole-vs-surface-mount/.

[13] G. Gallego, "Production Management," [Online]. Available: http://www.columbia.edu/~gmg2/4000/pdfold/throughput.pdf.

[14] J. D. C. Little, "A Proof for the Queuing Formula: $L = \lambda W$," *Operations Research,* p. 383–387, 1961.

[15] "Capacity Utilization," Corporate Finance Institute, 2020. [Online]. Available: https://corporatefinanceinstitute.com/resources/knowledge/economics/capacity-utilization/.

[16] S. Willems, "Chapter 10 - Process Flow Analysis," in *Operations Management*, 2018.

[17] M. A. Reveley, "A capital equipment capacity planning methodology for aerospace parts manufacturing in a high-mix, low volume environment," *MIT Thesis,* 2012.

[18] N. Geng and Z. Jiang, "A review on strategic capacity planning for the semiconductor," *International Journal of Production Research,* p. 3639–3655, 2009.

[19] S. Hood, S. Bermon and F. Barahona, " Capacity planning under demand uncertainty for semiconductor manufacturing," *IEEE Transactions on Semiconductor Manufacturing,* p. 273–280, 2003.

[20] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2020.

[21] D. H. Stamatis, The OEE Primer: Understanding Overall Equipment Effectiveness, Reliability, and Maintainability, 2010.

[22] S. Nakajima, Introduction to TPM: Total Productive Maintenance, Productivity Press, 1988.

[23] S. A. Irani, "Production system design for high-mix, low-volume manufacturing," *Industrial and Systems Engineering Magazine,* vol. 52, no. 1, 2020.

[24] S. Eickemeyer, F. Herde, P. Irudayaraj and P. Nyhuis, "Decision models for capacity planning in a regeneration environment," *International Journal of Production Research,* vol. 52, no. 23, pp. 7007-7026, 2014.

[25] C. Florensa, P. Garcia-Herreros, P. Misra, E. Arslan, S. Mehta and I. E. Grossmann, "Capacity planning with competitive decision-makers: Trilevel MILP formulation, degeneracy, and solution approaches," *European Journal of Operational Research,* vol. 262, no. 2, pp. 449-463, 2017.

[26] R. Zhou and L. Li, "Tactical capacity planning for semiconductor manufacturing: MILP models and scalable distributed parallel algorithms," *AIChE Journal,* vol. 62, no. 11, 2016.

[27] "Julia Opt," 2019. [Online]. Available: https://www.juliaopt.org/JuMP.jl/stable/installation/.

[28] R. Zhou and L. Li, "Joint capacity planning and distribution network optimization of coal supply chains under uncertainty," *AIChE Journal,* vol. 64, no. 4, 2017.

[29] Y. Kristianto and A. Gunasekaran, "A global optimization for sustainable multi-domain global manufacturing," *Computers and Operations Research,* vol. 89, p. 307–323, 2018.

[30] F. T. S. Chan, N. Li, S. H. Chung and M. Saadat, "Management of sustainable manufacturing systems-a review on mathematical problems," *International Journal of Production Research,* vol. 55, no. 4, p. 1210–1225, 2017.

[31] O. Kröger, C. Coffrin, H. Hijazi and H. Nagarajan, "Juniper: An Open-Source Nonlinear Branch-and-Bound Solver in Julia," in *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, Springer International Publishing, 2018, pp. 377-386.

[32] H. Nagarajan, M. Lu, S. Wang, R. Bent and K. Sundar, "An adaptive, multivariate partitioning algorithm for global optimization of nonconvex programs," *Journal of Global Optimization,* 2019.

[33] S. H. Chung and H. W. Huang, "Cycle time estimation for wafer fabwith engineering lots," *IIE Trans,* vol. 34, p. 105–118, 2002.

[34] J. C. Wood, "Cost and cycle time performance of fabs based on inte-grated single-wafer processing," *IEEE Trans. Semiconduct. Manufact.,* vol. 10, no. 1, p. 98–111, 1997.

[35] Y. D. Kim, J. Kim, S. Lim and H. Jun, "Due-date basedscheduling and control policies in a multiproduct semiconductor waferfabrication facility," *IEEE Tran. Semiconduct. Manufact.,* vol. 11, no. 1, p. 155–164, 1998.

[36] L. F. Atherton and R. W. Atherton, *Wafer Fabrication: Factory Performance and Analysis,* 1995.

[37] P. Backus, M. Janakiram, S. Mowzoon, C. Runger and A. Bhargava, "Factory Cycle-Time Prediction With a Data-Mining Approach," *Semiconductor Manufacturing, IEEE Transactions,* vol. 19, pp. 252 - 258, 2006.

[38] A. Raddon and B. Grigsby, "Throughput time forecasting model," *Proc. IEEE/SEMI Advanced Semiconductor Manufacturing Conf,* p. 430–433, 1997.

[39] S. T. Enns, "A dynamic forecasting model for Job shop flowing prediction and tardiness control," *Int. J. Production Res.,* vol. 33, no. 5, p. 2817–2834, 1995.

[40] A. C. Kaplan and A. T. Unal, "A probabilistic cost-based due date as-signment model for job shops," *Int. J. Production Res.,* vol. 31, no. 12, p. 2817–2834, 1993.

[41] D. Liao and C. Wang, "Neural-network-based delivery time estimates for prioritized 300-mm automatic material handling operations," *IEEE Trans. Semiconduct. Manufact.,* vol. 17, no. 3, p. 324–332, 2004.

[42] J. Wang and J. Zhang, "Big data analytics for forecasting cycle time in semiconductor wafer fabrication system," *International Journal of Production Research,* 2016.

[43] J. Wang, "Big Data Driven Cycle Time Parallel Prediction for Production Planning in Wafer Manufacturing," *Enterprise Information Systems,* vol. 12, no. 6, p. 714–732, 2018.

[44] Y. Meidan, B. Lerner, M. Hassoun and G. Rabinowitz, "Data Mining for Cycle Time Key Factor Identification and Prediction in Semiconductor Manufacturing," *IFAC Proceedings Volumes,* vol. 42, no. 4, p. 217–222, 2009.

[45] I. Tirkel, "Forecasting flow time in semiconductor manufacturing using knowledge discovery in databases," *International Journal of Production Research,* vol. 51, no. 18, pp. 5536-5548, 2013.

[46] R. Rosalina, *A Comparison of Machine Learning Algorithms in Manufacturing Production Process,* 2019.

[47] L. Lingitz, V. Gallina, F. Ansari, D. Gyulai, A. Pfeiffer, W. Sihn and L. Monostori, *Lead time prediction using machine learning algorithms: A case study by a semiconductor manufacturer,* 2018.

[48] SAP, "Calculating Production Dates," [Online]. Available: https://help.sap.com/.

[49] Kinaxis, "Demand Planning," 2017. [Online]. Available: https://www.kinaxis.com/sites/default/files/resources/rapidresponse-demand-planning-brochure-kinaxis.pdf.

[50] S. C. Graves, "Manufacturing Planning and Control," MIT, 1999.

[51] P. Durlinger, Inventory and holding costs, 2015.

[52] G. Louppe, "Understanding Random Forests," *University of Liège Dissertation,* 2014.

[53] J. Kronqvist, D. Bernal, A. Lundell and I. Grossmann, "A review and comparison of solvers for convex MINLP," *Optimization and Engineering,* vol. 20, p. 397–455, 2019.

[54] D. Jurafsky, "Minimum Edit Distance," [Online]. Available: https://web.stanford.edu/class/cs124/lec/med.pdf.

[55] A. Cohen, "FuzzyWuzzy," [Online]. Available: https://pypi.org/project/fuzzywuzzy/.

[56] A. Y. Ng, "On feature selection: Learning with Exponentially many Irrelevant Features as Training Examples," *MIT Thesis,* 1998.

[57] I. Jolliffe, Principal Component Analysis, New York: Springer-Verlag New York, Inc., 2002.

[58] M. Brems, "A One-Stop Shop for Principal Component Analysis," Towards Data Science, 17 April 2017. [Online]. Available: https://towardsdatascience.com/a-one-stop-shop-for-principal-component-analysis-5582fb7e0a9c.

[59] W. Koehrsen, "A Feature Selection Tool for Machine Learning in Python," Towards Data Science, 22 June 2018. [Online]. Available: https://towardsdatascience.com/a-feature-selection-tool-for-machine-learning-in-python-b64dd23710f0.

[60] R. Tamura, K. Kobayashi, Y. Takano, R. Miyashiro, K. Nakata and T. Matsui, "Mixed integer quadratic optimzation formulationsfor eliminating multicollinearity based on variance inflation factor," *Journal of Global Optimization,* vol. 2019, p. 431–446, 73.

[61] G. Louppe, L. Wehenkel, A. Sutera and P. Geurts, "Understanding variable importances in Forests of randomized trees," *Advances in Neural Information Processing Systems,* vol. 26, 2013.

[62] L. Breiman, *Random Forests,* 2001.

[63] L. Juárez-Orozco, O. Martinez Manzanera, S. Nesterov, S. Kajander and J. Knuuti, "The machine learning horizon in cardiac hybrid imaging," *European Journal of Hybrid Imaging,* vol. 2, 2018.

[64] scikit-learn Developers, "Supervised learning," scikit-learn - Machine Learning in Python, [Online]. Available: https://scikit-learn.org/stable/supervised_learning.html.

[65] C. Fogarty, *Engineering Statistics and Data Science,* 2018.

[66] A. J. Smola and B. Scholkopf, "A Tutorial on Support Vector Regression," 2003.

[67] T. Zhang, "Solving large scale linear prediction problems using stochastic gradient descent algorithms," *Proceedings of ICML,* 2004.

[68] C. Molnar, Interpretable Machine Learning, 2020.

[69] S. Gollapudi and V. Laxmikanth, Practical Machine Learning, Packt Publishing, 2016.

[70] T. Liu, Z. Li, Y. Tang, D. Yang, S. Jin and J. Guan, "The Application of the Machine Learning Method in Electromyographic Data," *IEEE Access,* vol. 8, pp. 9196-9208, 2020.

[71] On Splitting Training and Validation Set: A Comparative Study of Cross-Validation, Bootstrap and Systematic Sampling for Estimating the Generalization Performance of Supervised Learning.

[72] J. Bergstra and Y. Bengio, "Random Search for Hyper-Parameter Optimization," *Journal of Machine Learning Research,* vol. 13, pp. 281-305, 2012.

[73] P. Koch, B. Wujek, O. Golovidov and S. Gardner, "Automated Hyperparameter Tuning for Effective Machine Learning," *SAS Global Forum Proceedings,* 2017.

[74] Raytheon, "Form 10-K," 2018.

[75] "Raytheon Co. Dividend Discount Model," Stock Analysis on Net, 2020. [Online]. Available: https://www.stock-analysis-on.net/NYSE/Company/Raytheon-Co/DCF/DDM.

[76] G. E. P. Box, A. Luceño and M. d. C. Paniagua-Quinones, Statistical Control by Monitoring and Adjustment, Hoboken, NJ: John Wiley & Sonce, Inc., 2009.

APPENDIX: RANDOM FOREST MODEL PERFORMANCE AND SENSITIVITY

Operation	Features	Measure	Error
1	All	MSE	0.6714
1	No Queue	MSE	0.9277
1	No Date	MSE	0.6890
1	No Historical CT	MSE	1.4761
1	All	MAE	0.2717
1	No Queue	MAE	0.3225
1	No Date	MAE	0.3118
1	No Historical CT	MAE	0.5448
2	All	MSE	0.4249
2	No Queue	MSE	0.4698
2	No Date	MSE	0.5019
2	No Historical CT	MSE	0.1575
2	All	MAE	0.2709
2	No Queue	MAE	0.3014
2	No Date	MAE	0.3199
2	No Historical CT	MAE	0.1785
3	All	MSE	13.4760
3	No Queue	MSE	14.5674
3	No Date	MSE	15.6623
3	No Historical CT	MSE	9.7668
3	All	MAE	1.6882
3	No Queue	MAE	1.8302
3	No Date	MAE	1.9512
3	No Historical CT	MAE	0.9925
4	All	MSE	0.5742
4	No Queue	MSE	0.5991
4	No Date	MSE	0.6998
4	No Historical CT	MSE	0.4146
4	All	MAE	0.2330
4	No Queue	MAE	0.2443
4	No Date	MAE	0.2891
4	No Historical CT	MAE	0.2641
5	All	MSE	0.0309
5	No Queue	MSE	0.0361
5	No Date	MSE	0.0314
5	No Historical CT	MSE	0.0000
5	All	MAE	0.0171
5	No Queue	MAE	0.0263
5	No Date	MAE	0.0182
5	No Historical CT	MAE	0.0009
6	All	MSE	26.7741
6	No Queue	MSE	31.8666
6	No Date	MSE	27.4122
6	No Historical CT	MSE	42.7731
6	All	MAE	1.0126
6	No Queue	MAE	1.1499
6	No Date	MAE	1.0920
6	No Historical CT	MAE	1.4644
7	All	MSE	1.1659
7	No Queue	MSE	1.2103
7	No Date	MSE	1.4262
7	No Historical CT	MSE	0.2614
7	All	MAE	0.3948
7	No Queue	MAE	0.4311
7	No Date	MAE	0.4642

Operation	Features	Measure	Error
137	All	MAE	0.0685
137	No Queue	MAE	0.0867
137	No Date	MAE	0.0668
137	No Historical CT	MAE	0.0597
138	All	MSE	0.1809
138	No Queue	MSE	0.2093
138	No Date	MSE	0.1869
138	No Historical CT	MSE	0.1356
138	All	MAE	0.1743
138	No Queue	MAE	0.1890
138	No Date	MAE	0.1725
138	No Historical CT	MAE	0.0923
139	All	MSE	2.0701
139	No Queue	MSE	1.9285
139	No Date	MSE	2.1940
139	No Historical CT	MSE	1.1231
139	All	MAE	0.9748
139	No Queue	MAE	0.9708
139	No Date	MAE	1.0205
139	No Historical CT	MAE	0.5783
140	All	MSE	4.3990
140	No Queue	MSE	4.5802
140	No Date	MSE	4.7012
140	No Historical CT	MSE	1.8235
140	All	MAE	1.0633
140	No Queue	MAE	1.0893
140	No Date	MAE	1.2633
140	No Historical CT	MAE	0.4810
141	All	MSE	3.3143
141	No Queue	MSE	3.9953
141	No Date	MSE	3.1904
141	No Historical CT	MSE	4.1976
141	All	MAE	0.4821
141	No Queue	MAE	0.5395
141	No Date	MAE	0.5255
141	No Historical CT	MAE	0.4794
142	All	MSE	2.8764
142	No Queue	MSE	2.9848
142	No Date	MSE	3.4647
142	No Historical CT	MSE	4.9499
142	All	MAE	0.3348
142	No Queue	MAE	0.3869
142	No Date	MAE	0.3900
142	No Historical CT	MAE	0.6424
143	All	MSE	0.0110
143	No Queue	MSE	0.0115
143	No Date	MSE	0.0244
143	No Historical CT	MSE	0.0816
143	All	MAE	0.0403
143	No Queue	MAE	0.0430
143	No Date	MAE	0.0639
143	No Historical CT	MAE	0.1704
144	All	MSE	3.0404
144	No Queue	MSE	4.1853
144	No Date	MSE	3.0780

7	No Historical CT	MAE	0.1358
8	All	MSE	8.2116
8	No Queue	MSE	9.1153
8	No Date	MSE	10.4395
8	No Historical CT	MSE	3.8442
8	All	MAE	1.3402
8	No Queue	MAE	1.4504
8	No Date	MAE	1.5667
8	No Historical CT	MAE	0.8843
9	All	MSE	9.8336
9	No Queue	MSE	10.3202
9	No Date	MSE	11.6741
9	No Historical CT	MSE	6.4625
9	All	MAE	1.1317
9	No Queue	MAE	1.1977
9	No Date	MAE	1.2973
9	No Historical CT	MAE	0.7014
10	All	MSE	6.8617
10	No Queue	MSE	7.1351
10	No Date	MSE	6.9810
10	No Historical CT	MSE	3.3937
10	All	MAE	0.7801
10	No Queue	MAE	0.8439
10	No Date	MAE	0.8274
10	No Historical CT	MAE	0.4481
11	All	MSE	918.2197
11	No Queue	MSE	900.1747
11	No Date	MSE	916.3198
11	No Historical CT	MSE	925.6870
11	All	MAE	11.2260
11	No Queue	MAE	11.2056
11	No Date	MAE	11.2448
11	No Historical CT	MAE	11.1966
12	All	MSE	0.5874
12	No Queue	MSE	0.6254
12	No Date	MSE	0.6791
12	No Historical CT	MSE	0.1879
12	All	MAE	0.4287
12	No Queue	MAE	0.4508
12	No Date	MAE	0.4728
12	No Historical CT	MAE	0.1975
13	All	MSE	1.0937
13	No Queue	MSE	1.2029
13	No Date	MSE	1.4653
13	No Historical CT	MSE	0.5328
13	All	MAE	0.2293
13	No Queue	MAE	0.2517
13	No Date	MAE	0.2917
13	No Historical CT	MAE	0.2039
14	All	MSE	0.9942
14	No Queue	MSE	1.1671
14	No Date	MSE	1.1687
14	No Historical CT	MSE	1.1480
14	All	MAE	0.2598
14	No Queue	MAE	0.2878
14	No Date	MAE	0.3127
14	No Historical CT	MAE	0.2749
15	All	MSE	15.6080
144	No Historical CT	MSE	3.8520
144	All	MAE	1.2367
144	No Queue	MAE	1.3208
144	No Date	MAE	1.1221
144	No Historical CT	MAE	1.1299
145	All	MSE	0.5127
145	No Queue	MSE	0.5617
145	No Date	MSE	0.5375
145	No Historical CT	MSE	0.3260
145	All	MAE	0.2869
145	No Queue	MAE	0.3074
145	No Date	MAE	0.3027
145	No Historical CT	MAE	0.2319
146	All	MSE	0.0035
146	No Queue	MSE	0.0047
146	No Date	MSE	0.0083
146	No Historical CT	MSE	0.0030
146	All	MAE	0.0303
146	No Queue	MAE	0.0365
146	No Date	MAE	0.0425
146	No Historical CT	MAE	0.0255
147	All	MSE	222.9491
147	No Queue	MSE	238.5030
147	No Date	MSE	207.5432
147	No Historical CT	MSE	120.3129
147	All	MAE	8.5960
147	No Queue	MAE	9.0893
147	No Date	MAE	8.3879
147	No Historical CT	MAE	6.7459
148	All	MSE	262.5254
148	No Queue	MSE	401.7564
148	No Date	MSE	176.6970
148	No Historical CT	MSE	318.7019
148	All	MAE	8.5594
148	No Queue	MAE	9.8476
148	No Date	MAE	7.4793
148	No Historical CT	MAE	10.0666
149	All	MSE	81.5103
149	No Queue	MSE	83.2159
149	No Date	MSE	71.1825
149	No Historical CT	MSE	117.2469
149	All	MAE	4.3778
149	No Queue	MAE	4.6551
149	No Date	MAE	4.2466
149	No Historical CT	MAE	6.1519
150	All	MSE	30.0709
150	No Queue	MSE	33.2637
150	No Date	MSE	27.1213
150	No Historical CT	MSE	23.4896
150	All	MAE	1.9774
150	No Queue	MAE	2.1448
150	No Date	MAE	1.7598
150	No Historical CT	MAE	1.9636
151	All	MSE	1.8537
151	No Queue	MSE	2.1653
151	No Date	MSE	3.9571
151	No Historical CT	MSE	0.4105
151	All	MAE	0.5514

15	No Queue	MSE	17.2733
15	No Date	MSE	16.1871
15	No Historical CT	MSE	15.3777
15	All	MAE	0.9926
15	No Queue	MAE	1.1245
15	No Date	MAE	1.0568
15	No Historical CT	MAE	1.0793
16	All	MSE	1.1979
16	No Queue	MSE	1.2114
16	No Date	MSE	1.1817
16	No Historical CT	MSE	1.0569
16	All	MAE	0.1298
16	No Queue	MAE	0.1321
16	No Date	MAE	0.1295
16	No Historical CT	MAE	0.1409
17	All	MSE	14.9939
17	No Queue	MSE	16.9171
17	No Date	MSE	15.0085
17	No Historical CT	MSE	9.2439
17	All	MAE	1.0803
17	No Queue	MAE	1.1895
17	No Date	MAE	1.1848
17	No Historical CT	MAE	1.0093
18	All	MSE	5.5072
18	No Queue	MSE	6.0310
18	No Date	MSE	5.8019
18	No Historical CT	MSE	6.5295
18	All	MAE	0.6635
18	No Queue	MAE	0.7302
18	No Date	MAE	0.6962
18	No Historical CT	MAE	0.7312
19	All	MSE	0.0000
19	No Queue	MSE	0.0000
19	No Date	MSE	0.0000
19	No Historical CT	MSE	0.0000
19	All	MAE	0.0013
19	No Queue	MAE	0.0013
19	No Date	MAE	0.0014
19	No Historical CT	MAE	0.0006
20	All	MSE	0.8240
20	No Queue	MSE	0.9296
20	No Date	MSE	0.8144
20	No Historical CT	MSE	0.8469
20	All	MAE	0.2607
20	No Queue	MAE	0.2805
20	No Date	MAE	0.2751
20	No Historical CT	MAE	0.2455
21	All	MSE	0.0009
21	No Queue	MSE	0.0009
21	No Date	MSE	0.0010
21	No Historical CT	MSE	0.0006
21	All	MAE	0.0052
21	No Queue	MAE	0.0055
21	No Date	MAE	0.0053
21	No Historical CT	MAE	0.0079
22	All	MSE	0.3846
22	No Queue	MSE	0.3868
22	No Date	MSE	0.4474

151	No Queue	MAE	0.5920
151	No Date	MAE	0.9033
151	No Historical CT	MAE	0.2244
152	All	MSE	2.5980
152	No Queue	MSE	1.4918
152	No Date	MSE	6.0036
152	No Historical CT	MSE	0.8520
152	All	MAE	0.9966
152	No Queue	MAE	0.7925
152	No Date	MAE	1.6829
152	No Historical CT	MAE	0.5114
153	All	MSE	11.9988
153	No Queue	MSE	10.9650
153	No Date	MSE	15.0204
153	No Historical CT	MSE	11.9468
153	All	MAE	2.5650
153	No Queue	MAE	2.3715
153	No Date	MAE	2.8026
153	No Historical CT	MAE	2.4645
154	All	MSE	0.0240
154	No Queue	MSE	6.2928
154	No Date	MSE	0.0001
154	No Historical CT	MSE	0.0001
154	All	MAE	0.1314
154	No Queue	MAE	2.2040
154	No Date	MAE	0.0045
154	No Historical CT	MAE	0.0044
155	All	MSE	51.9418
155	No Queue	MSE	101.5463
155	No Date	MSE	115.4980
155	No Historical CT	MSE	13.1945
155	All	MAE	6.9315
155	No Queue	MAE	9.2197
155	No Date	MAE	9.2563
155	No Historical CT	MAE	3.5538
156	All	MSE	1.8679
156	No Queue	MSE	2.1031
156	No Date	MSE	2.3660
156	No Historical CT	MSE	1.0696
156	All	MAE	0.2365
156	No Queue	MAE	0.3007
156	No Date	MAE	0.3097
156	No Historical CT	MAE	0.1349
157	All	MSE	91.9342
157	No Queue	MSE	100.3348
157	No Date	MSE	97.2003
157	No Historical CT	MSE	117.4143
157	All	MAE	3.4345
157	No Queue	MAE	3.8605
157	No Date	MAE	3.7432
157	No Historical CT	MAE	4.0879
158	All	MSE	49.5430
158	No Queue	MSE	61.0658
158	No Date	MSE	53.3498
158	No Historical CT	MSE	79.4780
158	All	MAE	1.6996
158	No Queue	MAE	1.9763
158	No Date	MAE	1.8690

22	No Historical CT	MSE	0.0829
22	All	MAE	0.1639
22	No Queue	MAE	0.1786
22	No Date	MAE	0.1877
22	No Historical CT	MAE	0.1253
23	All	MSE	0.1730
23	No Queue	MSE	0.1838
23	No Date	MSE	0.2131
23	No Historical CT	MSE	0.0204
23	All	MAE	0.2082
23	No Queue	MAE	0.2175
23	No Date	MAE	0.2890
23	No Historical CT	MAE	0.0875
24	All	MSE	0.7490
24	No Queue	MSE	0.8291
24	No Date	MSE	0.8888
24	No Historical CT	MSE	0.8463
24	All	MAE	0.2136
24	No Queue	MAE	0.2410
24	No Date	MAE	0.2462
24	No Historical CT	MAE	0.2617
25	All	MSE	0.2817
25	No Queue	MSE	0.3266
25	No Date	MSE	0.2956
25	No Historical CT	MSE	0.2583
25	All	MAE	0.0987
25	No Queue	MAE	0.1131
25	No Date	MAE	0.1117
25	No Historical CT	MAE	0.1448
26	All	MSE	0.7107
26	No Queue	MSE	0.8405
26	No Date	MSE	0.7252
26	No Historical CT	MSE	0.7980
26	All	MAE	0.1771
26	No Queue	MAE	0.2008
26	No Date	MAE	0.1815
26	No Historical CT	MAE	0.1992
27	All	MSE	1.1734
27	No Queue	MSE	1.2375
27	No Date	MSE	1.5526
27	No Historical CT	MSE	0.6009
27	All	MAE	0.5960
27	No Queue	MAE	0.6063
27	No Date	MAE	0.7021
27	No Historical CT	MAE	0.3634
28	All	MSE	0.1222
28	No Queue	MSE	0.1305
28	No Date	MSE	0.1379
28	No Historical CT	MSE	0.5184
28	All	MAE	0.0865
28	No Queue	MAE	0.0876
28	No Date	MAE	0.0962
28	No Historical CT	MAE	0.2617
29	All	MSE	2.0457
29	No Queue	MSE	2.0964
29	No Date	MSE	2.4323
29	No Historical CT	MSE	0.6352
29	All	MAE	0.8253

158	No Historical CT	MAE	2.2727
159	All	MSE	663.2981
159	No Queue	MSE	665.0149
159	No Date	MSE	660.4867
159	No Historical CT	MSE	2667.1240
159	All	MAE	5.8300
159	No Queue	MAE	6.0257
159	No Date	MAE	5.4867
159	No Historical CT	MAE	37.5798
160	All	MSE	0.3017
160	No Queue	MSE	0.3234
160	No Date	MSE	0.3401
160	No Historical CT	MSE	0.2250
160	All	MAE	0.2032
160	No Queue	MAE	0.2265
160	No Date	MAE	0.2302
160	No Historical CT	MAE	0.1024
161	All	MSE	134.2060
161	No Queue	MSE	146.5893
161	No Date	MSE	137.6488
161	No Historical CT	MSE	223.7528
161	All	MAE	3.7006
161	No Queue	MAE	4.1926
161	No Date	MAE	3.9119
161	No Historical CT	MAE	5.5885
162	All	MSE	47.0846
162	No Queue	MSE	54.5123
162	No Date	MSE	54.1597
162	No Historical CT	MSE	50.1103
162	All	MAE	2.1711
162	No Queue	MAE	2.5538
162	No Date	MAE	2.4362
162	No Historical CT	MAE	2.2489
163	All	MSE	112.2401
163	No Queue	MSE	125.9914
163	No Date	MSE	120.3702
163	No Historical CT	MSE	90.1404
163	All	MAE	6.2428
163	No Queue	MAE	6.5229
163	No Date	MAE	6.6323
163	No Historical CT	MAE	4.5501
164	All	MSE	0.0035
164	No Queue	MSE	0.0044
164	No Date	MSE	0.0059
164	No Historical CT	MSE	0.0006
164	All	MAE	0.0137
164	No Queue	MAE	0.0160
164	No Date	MAE	0.0209
164	No Historical CT	MAE	0.0089
165	All	MSE	0.0137
165	No Queue	MSE	0.0152
165	No Date	MSE	0.0124
165	No Historical CT	MSE	0.0065
165	All	MAE	0.0631
165	No Queue	MAE	0.0696
165	No Date	MAE	0.0639
165	No Historical CT	MAE	0.0497
166	All	MSE	3.8065

29	No Queue	MAE	0.8834
29	No Date	MAE	0.9155
29	No Historical CT	MAE	0.3750
30	All	MSE	0.0886
30	No Queue	MSE	0.1417
30	No Date	MSE	0.1882
30	No Historical CT	MSE	0.0352
30	All	MAE	0.1020
30	No Queue	MAE	0.1116
30	No Date	MAE	0.1483
30	No Historical CT	MAE	0.0783
31	All	MSE	0.0321
31	No Queue	MSE	0.0340
31	No Date	MSE	0.0330
31	No Historical CT	MSE	0.0304
31	All	MAE	0.0960
31	No Queue	MAE	0.0993
31	No Date	MAE	0.0979
31	No Historical CT	MAE	0.0940
32	All	MSE	1.2780
32	No Queue	MSE	1.3466
32	No Date	MSE	2.9429
32	No Historical CT	MSE	0.2934
32	All	MAE	0.2531
32	No Queue	MAE	0.3066
32	No Date	MAE	0.9066
32	No Historical CT	MAE	0.1671
33	All	MSE	0.2910
33	No Queue	MSE	0.3385
33	No Date	MSE	0.6948
33	No Historical CT	MSE	0.0707
33	All	MAE	0.1152
33	No Queue	MAE	0.1270
33	No Date	MAE	0.2730
33	No Historical CT	MAE	0.0477
34	All	MSE	11.9565
34	No Queue	MSE	112.3294
34	No Date	MSE	47.7642
34	No Historical CT	MSE	13.1112
34	All	MAE	2.1427
34	No Queue	MAE	6.0835
34	No Date	MAE	4.0136
34	No Historical CT	MAE	1.1847
35	All	MSE	1.9190
35	No Queue	MSE	3.0525
35	No Date	MSE	2.7814
35	No Historical CT	MSE	14.9029
35	All	MAE	0.1425
35	No Queue	MAE	0.2240
35	No Date	MAE	0.2028
35	No Historical CT	MAE	1.1583
36	All	MSE	4.9759
36	No Queue	MSE	6.0488
36	No Date	MSE	4.6140
36	No Historical CT	MSE	28.0314
36	All	MAE	0.4165
36	No Queue	MAE	0.4512
36	No Date	MAE	0.4003

166	No Queue	MSE	4.6394
166	No Date	MSE	3.9901
166	No Historical CT	MSE	1.6185
166	All	MAE	0.8099
166	No Queue	MAE	0.9286
166	No Date	MAE	0.8815
166	No Historical CT	MAE	0.3724
167	All	MSE	8.9533
167	No Queue	MSE	8.9965
167	No Date	MSE	9.3891
167	No Historical CT	MSE	6.0654
167	All	MAE	0.5904
167	No Queue	MAE	0.6474
167	No Date	MAE	0.6330
167	No Historical CT	MAE	0.5352
168	All	MSE	7.6592
168	No Queue	MSE	7.9687
168	No Date	MSE	8.3928
168	No Historical CT	MSE	12.5797
168	All	MAE	0.6141
168	No Queue	MAE	0.7062
168	No Date	MAE	0.6315
168	No Historical CT	MAE	1.1162
169	All	MSE	3.7525
169	No Queue	MSE	2.8034
169	No Date	MSE	4.2352
169	No Historical CT	MSE	5.5280
169	All	MAE	0.7848
169	No Queue	MAE	0.7771
169	No Date	MAE	0.8191
169	No Historical CT	MAE	0.9020
170	All	MSE	0.1801
170	No Queue	MSE	0.2203
170	No Date	MSE	0.1575
170	No Historical CT	MSE	0.0062
170	All	MAE	0.2318
170	No Queue	MAE	0.2443
170	No Date	MAE	0.2647
170	No Historical CT	MAE	0.0566
171	All	MSE	4.9959
171	No Queue	MSE	5.7115
171	No Date	MSE	5.0193
171	No Historical CT	MSE	1.9574
171	All	MAE	1.2849
171	No Queue	MAE	1.4761
171	No Date	MAE	1.2825
171	No Historical CT	MAE	1.0711
172	All	MSE	2.8501
172	No Queue	MSE	3.2211
172	No Date	MSE	2.9128
172	No Historical CT	MSE	2.2464
172	All	MAE	0.4560
172	No Queue	MAE	0.5419
172	No Date	MAE	0.4745
172	No Historical CT	MAE	0.2885
173	All	MSE	6.7000
173	No Queue	MSE	5.9931
173	No Date	MSE	6.9618

36	No Historical CT	MAE	1.3717
37	All	MSE	5.9228
37	No Queue	MSE	6.6486
37	No Date	MSE	6.3620
37	No Historical CT	MSE	9.1387
37	All	MAE	0.4995
37	No Queue	MAE	0.5521
37	No Date	MAE	0.5532
37	No Historical CT	MAE	0.7083
38	All	MSE	0.2921
38	No Queue	MSE	0.8527
38	No Date	MSE	0.2161
38	No Historical CT	MSE	0.1699
38	All	MAE	0.1606
38	No Queue	MAE	0.2445
38	No Date	MAE	0.1305
38	No Historical CT	MAE	0.0816
39	All	MSE	0.2903
39	No Queue	MSE	1.6556
39	No Date	MSE	2.6297
39	No Historical CT	MSE	0.0452
39	All	MAE	0.1719
39	No Queue	MAE	0.2807
39	No Date	MAE	0.4358
39	No Historical CT	MAE	0.0663
40	All	MSE	0.1789
40	No Queue	MSE	0.2351
40	No Date	MSE	0.2315
40	No Historical CT	MSE	0.2908
40	All	MAE	0.1417
40	No Queue	MAE	0.1654
40	No Date	MAE	0.1706
40	No Historical CT	MAE	0.2370
41	All	MSE	2.9358
41	No Queue	MSE	3.1537
41	No Date	MSE	3.5906
41	No Historical CT	MSE	3.0938
41	All	MAE	0.5277
41	No Queue	MAE	0.5960
41	No Date	MAE	0.5714
41	No Historical CT	MAE	0.5275
42	All	MSE	52.6903
42	No Queue	MSE	85.1460
42	No Date	MSE	51.8846
42	No Historical CT	MSE	86.3845
42	All	MAE	4.2633
42	No Queue	MAE	5.1694
42	No Date	MAE	4.3160
42	No Historical CT	MAE	4.1490
43	All	MSE	2.1031
43	No Queue	MSE	3.3127
43	No Date	MSE	5.5332
43	No Historical CT	MSE	8.8194
43	All	MAE	0.3084
43	No Queue	MAE	0.3628
43	No Date	MAE	0.5898
43	No Historical CT	MAE	0.5365
44	All	MSE	40.4395

173	No Historical CT	MSE	5.4870
173	All	MAE	0.7896
173	No Queue	MAE	0.8067
173	No Date	MAE	0.8259
173	No Historical CT	MAE	0.9991
174	All	MSE	5265.4696
174	No Queue	MSE	4903.4158
174	No Date	MSE	6140.5701
174	No Historical CT	MSE	4358.9539
174	All	MAE	42.1918
174	No Queue	MAE	40.9215
174	No Date	MAE	45.0551
174	No Historical CT	MAE	40.5622
175	All	MSE	3.4799
175	No Queue	MSE	5.5626
175	No Date	MSE	3.3816
175	No Historical CT	MSE	2.4077
175	All	MAE	0.4451
175	No Queue	MAE	0.7900
175	No Date	MAE	0.4486
175	No Historical CT	MAE	0.3588
176	All	MSE	0.7719
176	No Queue	MSE	0.7879
176	No Date	MSE	0.8424
176	No Historical CT	MSE	0.5825
176	All	MAE	0.4051
176	No Queue	MAE	0.4407
176	No Date	MAE	0.4500
176	No Historical CT	MAE	0.2957
177	All	MSE	2.6366
177	No Queue	MSE	2.8598
177	No Date	MSE	2.8634
177	No Historical CT	MSE	1.7500
177	All	MAE	0.8726
177	No Queue	MAE	0.9529
177	No Date	MAE	0.9357
177	No Historical CT	MAE	0.6287
178	All	MSE	41.8901
178	No Queue	MSE	44.6267
178	No Date	MSE	43.4900
178	No Historical CT	MSE	60.7250
178	All	MAE	2.7593
178	No Queue	MAE	3.1246
178	No Date	MAE	2.8080
178	No Historical CT	MAE	3.2418
179	All	MSE	82.3842
179	No Queue	MSE	85.0651
179	No Date	MSE	81.4083
179	No Historical CT	MSE	106.8293
179	All	MAE	2.4806
179	No Queue	MAE	2.7039
179	No Date	MAE	2.5709
179	No Historical CT	MAE	3.2482
180	All	MSE	0.3180
180	No Queue	MSE	0.3588
180	No Date	MSE	0.5358
180	No Historical CT	MSE	0.0330
180	All	MAE	0.2650

44	No Queue	MSE	43.7170
44	No Date	MSE	45.5735
44	No Historical CT	MSE	46.3447
44	All	MAE	3.3674
44	No Queue	MAE	3.5539
44	No Date	MAE	3.9216
44	No Historical CT	MAE	2.8789
45	All	MSE	9.2510
45	No Queue	MSE	10.4194
45	No Date	MSE	8.9372
45	No Historical CT	MSE	3.3215
45	All	MAE	1.2112
45	No Queue	MAE	1.3787
45	No Date	MAE	1.2126
45	No Historical CT	MAE	0.8271
46	All	MSE	35.0376
46	No Queue	MSE	47.1766
46	No Date	MSE	44.1538
46	No Historical CT	MSE	27.7796
46	All	MAE	3.7369
46	No Queue	MAE	4.6382
46	No Date	MAE	4.0418
46	No Historical CT	MAE	2.6937
47	All	MSE	0.3753
47	No Queue	MSE	0.3750
47	No Date	MSE	0.3747
47	No Historical CT	MSE	0.6550
47	All	MAE	0.0864
47	No Queue	MAE	0.0926
47	No Date	MAE	0.0878
47	No Historical CT	MAE	0.2823
48	All	MSE	0.1821
48	No Queue	MSE	0.1331
48	No Date	MSE	0.3226
48	No Historical CT	MSE	0.1255
48	All	MAE	0.1265
48	No Queue	MAE	0.1065
48	No Date	MAE	0.1637
48	No Historical CT	MAE	0.1135
49	All	MSE	35.5184
49	No Queue	MSE	34.0331
49	No Date	MSE	39.3909
49	No Historical CT	MSE	35.7464
49	All	MAE	2.6519
49	No Queue	MAE	2.7125
49	No Date	MAE	3.1838
49	No Historical CT	MAE	2.4372
50	All	MSE	0.0253
50	No Queue	MSE	0.0250
50	No Date	MSE	0.0270
50	No Historical CT	MSE	0.0272
50	All	MAE	0.0559
50	No Queue	MAE	0.0607
50	No Date	MAE	0.0588
50	No Historical CT	MAE	0.0480
51	All	MSE	0.1098
51	No Queue	MSE	0.1194
51	No Date	MSE	0.1119

180	No Queue	MAE	0.2815
180	No Date	MAE	0.3600
180	No Historical CT	MAE	0.0849
181	All	MSE	2.4361
181	No Queue	MSE	2.4608
181	No Date	MSE	2.3025
181	No Historical CT	MSE	0.1560
181	All	MAE	0.6291
181	No Queue	MAE	0.6925
181	No Date	MAE	0.7008
181	No Historical CT	MAE	0.1414
182	All	MSE	2.7148
182	No Queue	MSE	2.9806
182	No Date	MSE	3.3752
182	No Historical CT	MSE	2.3742
182	All	MAE	0.7295
182	No Queue	MAE	0.8278
182	No Date	MAE	1.0609
182	No Historical CT	MAE	0.5026
183	All	MSE	24.6166
183	No Queue	MSE	55.0763
183	No Date	MSE	36.1214
183	No Historical CT	MSE	17.2989
183	All	MAE	1.9863
183	No Queue	MAE	2.8743
183	No Date	MAE	2.4785
183	No Historical CT	MAE	1.4593
184	All	MSE	0.0123
184	No Queue	MSE	0.0073
184	No Date	MSE	0.0089
184	No Historical CT	MSE	0.0284
184	All	MAE	0.0699
184	No Queue	MAE	0.0642
184	No Date	MAE	0.0687
184	No Historical CT	MAE	0.1258
185	All	MSE	0.5222
185	No Queue	MSE	2.6773
185	No Date	MSE	0.4899
185	No Historical CT	MSE	0.0001
185	All	MAE	0.1571
185	No Queue	MAE	0.5795
185	No Date	MAE	0.2575
185	No Historical CT	MAE	0.0049
186	All	MSE	0.0000
186	No Queue	MSE	0.0002
186	No Date	MSE	0.0000
186	No Historical CT	MSE	0.0000
186	All	MAE	0.0001
186	No Queue	MAE	0.0045
186	No Date	MAE	0.0002
186	No Historical CT	MAE	0.0002
187	All	MSE	247.8226
187	No Queue	MSE	272.2499
187	No Date	MSE	241.3805
187	No Historical CT	MSE	119.5563
187	All	MAE	8.2367
187	No Queue	MAE	8.7657
187	No Date	MAE	7.2349

51	No Historical CT	MSE	0.1116
51	All	MAE	0.0592
51	No Queue	MAE	0.0952
51	No Date	MAE	0.0527
51	No Historical CT	MAE	0.0508
52	All	MSE	5.1986
52	No Queue	MSE	5.3071
52	No Date	MSE	5.9275
52	No Historical CT	MSE	7.4308
52	All	MAE	1.0274
52	No Queue	MAE	0.9939
52	No Date	MAE	1.0966
52	No Historical CT	MAE	1.1779
53	All	MSE	1.5088
53	No Queue	MSE	1.4911
53	No Date	MSE	2.1480
53	No Historical CT	MSE	1.4951
53	All	MAE	0.3525
53	No Queue	MAE	0.3633
53	No Date	MAE	0.6406
53	No Historical CT	MAE	0.3526
54	All	MSE	6.8077
54	No Queue	MSE	8.6095
54	No Date	MSE	8.2933
54	No Historical CT	MSE	6.4331
54	All	MAE	0.8093
54	No Queue	MAE	0.9498
54	No Date	MAE	1.0274
54	No Historical CT	MAE	1.5242
55	All	MSE	3.8539
55	No Queue	MSE	4.9044
55	No Date	MSE	7.0592
55	No Historical CT	MSE	0.0269
55	All	MAE	0.4596
55	No Queue	MAE	0.5647
55	No Date	MAE	1.3414
55	No Historical CT	MAE	0.1077
56	All	MSE	0.0007
56	No Queue	MSE	0.0638
56	No Date	MSE	0.0149
56	No Historical CT	MSE	0.0028
56	All	MAE	0.0087
56	No Queue	MAE	0.0289
56	No Date	MAE	0.0331
56	No Historical CT	MAE	0.0277
57	All	MSE	1.1906
57	No Queue	MSE	1.0033
57	No Date	MSE	1.3532
57	No Historical CT	MSE	0.0558
57	All	MAE	0.2498
57	No Queue	MAE	0.2522
57	No Date	MAE	0.2591
57	No Historical CT	MAE	0.1043
58	All	MSE	0.1388
58	No Queue	MSE	0.2202
58	No Date	MSE	0.1318
58	No Historical CT	MSE	0.0904
58	All	MAE	0.0890

187	No Historical CT	MAE	3.6854
188	All	MSE	1.3114
188	No Queue	MSE	1.3997
188	No Date	MSE	1.4617
188	No Historical CT	MSE	1.8260
188	All	MAE	0.3652
188	No Queue	MAE	0.3892
188	No Date	MAE	0.3982
188	No Historical CT	MAE	0.4103
189	All	MSE	10.6911
189	No Queue	MSE	11.8009
189	No Date	MSE	13.5132
189	No Historical CT	MSE	13.1957
189	All	MAE	0.8686
189	No Queue	MAE	0.9831
189	No Date	MAE	1.0275
189	No Historical CT	MAE	0.9385
190	All	MSE	371.0312
190	No Queue	MSE	449.2621
190	No Date	MSE	374.3181
190	No Historical CT	MSE	353.8077
190	All	MAE	8.9325
190	No Queue	MAE	9.7680
190	No Date	MAE	7.9274
190	No Historical CT	MAE	7.3109
191	All	MSE	1.9675
191	No Queue	MSE	2.0763
191	No Date	MSE	1.6331
191	No Historical CT	MSE	1.0328
191	All	MAE	0.7796
191	No Queue	MAE	0.8008
191	No Date	MAE	0.6961
191	No Historical CT	MAE	0.4873
192	All	MSE	5.3080
192	No Queue	MSE	5.2939
192	No Date	MSE	5.5256
192	No Historical CT	MSE	6.0133
192	All	MAE	0.7137
192	No Queue	MAE	0.7208
192	No Date	MAE	0.7487
192	No Historical CT	MAE	0.9062
193	All	MSE	8.8397
193	No Queue	MSE	10.0052
193	No Date	MSE	7.6283
193	No Historical CT	MSE	6.0632
193	All	MAE	2.3620
193	No Queue	MAE	2.3647
193	No Date	MAE	2.3189
193	No Historical CT	MAE	2.0291
194	All	MSE	1217.4814
194	No Queue	MSE	1605.2643
194	No Date	MSE	1215.3589
194	No Historical CT	MSE	1004.2776
194	All	MAE	20.1646
194	No Queue	MAE	21.4322
194	No Date	MAE	22.3162
194	No Historical CT	MAE	19.1506
195	All	MSE	2.5083

58	No Queue	MAE	0.1094
58	No Date	MAE	0.0865
58	No Historical CT	MAE	0.0535
59	All	MSE	1.5798
59	No Queue	MSE	1.9748
59	No Date	MSE	1.3427
59	No Historical CT	MSE	0.0134
59	All	MAE	0.3777
59	No Queue	MAE	0.5745
59	No Date	MAE	0.3431
59	No Historical CT	MAE	0.0843
60	All	MSE	175.6584
60	No Queue	MSE	224.8423
60	No Date	MSE	181.5098
60	No Historical CT	MSE	230.3942
60	All	MAE	4.6107
60	No Queue	MAE	5.1034
60	No Date	MAE	4.6524
60	No Historical CT	MAE	6.7784
61	All	MSE	0.3425
61	No Queue	MSE	0.4014
61	No Date	MSE	0.3929
61	No Historical CT	MSE	0.0854
61	All	MAE	0.2215
61	No Queue	MAE	0.2500
61	No Date	MAE	0.2603
61	No Historical CT	MAE	0.0971
62	All	MSE	2.6896
62	No Queue	MSE	2.5772
62	No Date	MSE	2.6634
62	No Historical CT	MSE	7.0044
62	All	MAE	0.5402
62	No Queue	MAE	0.5166
62	No Date	MAE	0.5446
62	No Historical CT	MAE	1.7443
63	All	MSE	0.4982
63	No Queue	MSE	0.5355
63	No Date	MSE	0.5762
63	No Historical CT	MSE	0.6137
63	All	MAE	0.3752
63	No Queue	MAE	0.3862
63	No Date	MAE	0.4055
63	No Historical CT	MAE	0.3695
64	All	MSE	0.2126
64	No Queue	MSE	0.2088
64	No Date	MSE	0.2190
64	No Historical CT	MSE	0.2188
64	All	MAE	0.0909
64	No Queue	MAE	0.0931
64	No Date	MAE	0.0939
64	No Historical CT	MAE	0.1084
65	All	MSE	23.9499
65	No Queue	MSE	23.4525
65	No Date	MSE	7.0818
65	No Historical CT	MSE	155.3645
65	All	MAE	3.2737
65	No Queue	MAE	3.3605
65	No Date	MAE	1.6643

195	No Queue	MSE	2.2642
195	No Date	MSE	2.6176
195	No Historical CT	MSE	2.5474
195	All	MAE	1.1449
195	No Queue	MAE	1.2082
195	No Date	MAE	1.1085
195	No Historical CT	MAE	1.0441
196	All	MSE	0.4627
196	No Queue	MSE	0.2978
196	No Date	MSE	0.5383
196	No Historical CT	MSE	1.1128
196	All	MAE	0.1523
196	No Queue	MAE	0.1386
196	No Date	MAE	0.1814
196	No Historical CT	MAE	0.2943
197	All	MSE	212.9816
197	No Queue	MSE	308.5900
197	No Date	MSE	244.9870
197	No Historical CT	MSE	45.9857
197	All	MAE	6.7754
197	No Queue	MAE	8.3562
197	No Date	MAE	7.9757
197	No Historical CT	MAE	2.7129
198	All	MSE	0.0118
198	No Queue	MSE	0.0084
198	No Date	MSE	0.0102
198	No Historical CT	MSE	0.0136
198	All	MAE	0.0669
198	No Queue	MAE	0.0531
198	No Date	MAE	0.0553
198	No Historical CT	MAE	0.0678
199	All	MSE	1.2621
199	No Queue	MSE	2.0790
199	No Date	MSE	1.3736
199	No Historical CT	MSE	1.6569
199	All	MAE	0.6192
199	No Queue	MAE	0.7832
199	No Date	MAE	0.6184
199	No Historical CT	MAE	0.7360
200	All	MSE	0.3374
200	No Queue	MSE	0.6130
200	No Date	MSE	0.6995
200	No Historical CT	MSE	0.0135
200	All	MAE	0.2174
200	No Queue	MAE	0.2540
200	No Date	MAE	0.3939
200	No Historical CT	MAE	0.0760
201	All	MSE	2.3398
201	No Queue	MSE	2.4377
201	No Date	MSE	2.0182
201	No Historical CT	MSE	0.2681
201	All	MAE	1.4957
201	No Queue	MAE	1.5206
201	No Date	MAE	1.3256
201	No Historical CT	MAE	0.4266
202	All	MSE	24.8732
202	No Queue	MSE	30.8111
202	No Date	MSE	51.1632

65	No Historical CT	MAE	6.4226
66	All	MSE	0.0000
66	No Queue	MSE	0.0000
66	No Date	MSE	0.0010
66	No Historical CT	MSE	0.0000
66	All	MAE	0.0001
66	No Queue	MAE	0.0002
66	No Date	MAE	0.0065
66	No Historical CT	MAE	0.0000
67	All	MSE	146.6731
67	No Queue	MSE	139.0913
67	No Date	MSE	132.0232
67	No Historical CT	MSE	140.1944
67	All	MAE	7.8860
67	No Queue	MAE	8.0052
67	No Date	MAE	7.2261
67	No Historical CT	MAE	7.1342
68	All	MSE	0.3507
68	No Queue	MSE	0.5864
68	No Date	MSE	0.3286
68	No Historical CT	MSE	0.1825
68	All	MAE	0.2596
68	No Queue	MAE	0.3128
68	No Date	MAE	0.2602
68	No Historical CT	MAE	0.2887
69	All	MSE	0.5538
69	No Queue	MSE	1.0727
69	No Date	MSE	0.5248
69	No Historical CT	MSE	1.1745
69	All	MAE	0.1231
69	No Queue	MAE	0.1642
69	No Date	MAE	0.1129
69	No Historical CT	MAE	0.5604
70	All	MSE	1.7649
70	No Queue	MSE	2.3860
70	No Date	MSE	1.9079
70	No Historical CT	MSE	1.5330
70	All	MAE	0.9027
70	No Queue	MAE	1.0207
70	No Date	MAE	0.8938
70	No Historical CT	MAE	0.4591
71	All	MSE	15.6789
71	No Queue	MSE	20.8133
71	No Date	MSE	19.2687
71	No Historical CT	MSE	6.7899
71	All	MAE	0.7745
71	No Queue	MAE	0.9079
71	No Date	MAE	0.9316
71	No Historical CT	MAE	0.3402
72	All	MSE	0.0010
72	No Queue	MSE	0.0009
72	No Date	MSE	0.0012
72	No Historical CT	MSE	0.0013
72	All	MAE	0.0036
72	No Queue	MAE	0.0035
72	No Date	MAE	0.0036
72	No Historical CT	MAE	0.0069
73	All	MSE	5.0458

202	No Historical CT	MSE	23.5244
202	All	MAE	1.5685
202	No Queue	MAE	2.2505
202	No Date	MAE	1.9935
202	No Historical CT	MAE	1.3377
203	All	MSE	0.1285
203	No Queue	MSE	0.1277
203	No Date	MSE	0.1647
203	No Historical CT	MSE	0.1506
203	All	MAE	0.3137
203	No Queue	MAE	0.2881
203	No Date	MAE	0.3638
203	No Historical CT	MAE	0.3526
204	All	MSE	1.4412
204	No Queue	MSE	1.5485
204	No Date	MSE	1.9043
204	No Historical CT	MSE	0.7375
204	All	MAE	0.4050
204	No Queue	MAE	0.4388
204	No Date	MAE	0.5476
204	No Historical CT	MAE	0.2221
205	All	MSE	1.9395
205	No Queue	MSE	1.7503
205	No Date	MSE	2.6250
205	No Historical CT	MSE	0.0781
205	All	MAE	0.9029
205	No Queue	MAE	0.9075
205	No Date	MAE	1.1020
205	No Historical CT	MAE	0.1476
206	All	MSE	0.0000
206	No Queue	MSE	0.0000
206	No Date	MSE	0.0000
206	No Historical CT	MSE	0.0000
206	All	MAE	0.0010
206	No Queue	MAE	0.0013
206	No Date	MAE	0.0010
206	No Historical CT	MAE	0.0003
207	All	MSE	2.1410
207	No Queue	MSE	2.5966
207	No Date	MSE	7.3447
207	No Historical CT	MSE	1.5430
207	All	MAE	0.8458
207	No Queue	MAE	0.9780
207	No Date	MAE	1.7766
207	No Historical CT	MAE	0.5958
208	All	MSE	11.0642
208	No Queue	MSE	17.4171
208	No Date	MSE	10.1752
208	No Historical CT	MSE	10.1155
208	All	MAE	2.6158
208	No Queue	MAE	3.5218
208	No Date	MAE	2.5557
208	No Historical CT	MAE	2.4886
209	All	MSE	0.8029
209	No Queue	MSE	1.4240
209	No Date	MSE	1.3029
209	No Historical CT	MSE	3.7920
209	All	MAE	0.2473

73	No Queue	MSE	5.4471
73	No Date	MSE	5.7835
73	No Historical CT	MSE	3.0417
73	All	MAE	0.9688
73	No Queue	MAE	1.0912
73	No Date	MAE	1.0979
73	No Historical CT	MAE	0.8292
74	All	MSE	0.0012
74	No Queue	MSE	0.0017
74	No Date	MSE	0.0015
74	No Historical CT	MSE	0.0011
74	All	MAE	0.0237
74	No Queue	MAE	0.0289
74	No Date	MAE	0.0278
74	No Historical CT	MAE	0.0206
75	All	MSE	0.0016
75	No Queue	MSE	0.0018
75	No Date	MSE	0.0015
75	No Historical CT	MSE	0.0016
75	All	MAE	0.0250
75	No Queue	MAE	0.0285
75	No Date	MAE	0.0242
75	No Historical CT	MAE	0.0210
76	All	MSE	0.1972
76	No Queue	MSE	0.2215
76	No Date	MSE	0.2598
76	No Historical CT	MSE	0.0994
76	All	MAE	0.1649
76	No Queue	MAE	0.1837
76	No Date	MAE	0.2019
76	No Historical CT	MAE	0.1466
77	All	MSE	0.0751
77	No Queue	MSE	0.0954
77	No Date	MSE	0.1929
77	No Historical CT	MSE	0.0753
77	All	MAE	0.1282
77	No Queue	MAE	0.1820
77	No Date	MAE	0.2475
77	No Historical CT	MAE	0.0978
78	All	MSE	11.8156
78	No Queue	MSE	11.2166
78	No Date	MSE	12.1532
78	No Historical CT	MSE	10.3564
78	All	MAE	2.0858
78	No Queue	MAE	2.1964
78	No Date	MAE	2.0835
78	No Historical CT	MAE	1.6110
79	All	MSE	0.2649
79	No Queue	MSE	0.2723
79	No Date	MSE	0.3863
79	No Historical CT	MSE	0.0721
79	All	MAE	0.1489
79	No Queue	MAE	0.1555
79	No Date	MAE	0.1874
79	No Historical CT	MAE	0.1011
80	All	MSE	0.5587
80	No Queue	MSE	0.5950
80	No Date	MSE	0.6490

209	No Queue	MAE	0.3675
209	No Date	MAE	0.4321
209	No Historical CT	MAE	0.8542
210	All	MSE	1.9807
210	No Queue	MSE	9.5604
210	No Date	MSE	1.5390
210	No Historical CT	MSE	1.2714
210	All	MAE	0.6976
210	No Queue	MAE	1.0765
210	No Date	MAE	0.6105
210	No Historical CT	MAE	0.6308
211	All	MSE	0.0489
211	No Queue	MSE	0.1077
211	No Date	MSE	0.1190
211	No Historical CT	MSE	0.0152
211	All	MAE	0.0947
211	No Queue	MAE	0.1239
211	No Date	MAE	0.1166
211	No Historical CT	MAE	0.0502
212	All	MSE	144.1951
212	No Queue	MSE	146.9571
212	No Date	MSE	125.7453
212	No Historical CT	MSE	274.5878
212	All	MAE	5.2825
212	No Queue	MAE	5.8740
212	No Date	MAE	4.8198
212	No Historical CT	MAE	7.5823
213	All	MSE	0.0003
213	No Queue	MSE	0.0015
213	No Date	MSE	0.0014
213	No Historical CT	MSE	0.0001
213	All	MAE	0.0126
213	No Queue	MAE	0.0324
213	No Date	MAE	0.0327
213	No Historical CT	MAE	0.0102
214	All	MSE	2.3856
214	No Queue	MSE	2.1269
214	No Date	MSE	5.7175
214	No Historical CT	MSE	15.0275
214	All	MAE	1.2804
214	No Queue	MAE	1.1388
214	No Date	MAE	2.1025
214	No Historical CT	MAE	2.3953
215	All	MSE	2.9757
215	No Queue	MSE	16.4116
215	No Date	MSE	3.6935
215	No Historical CT	MSE	3.6586
215	All	MAE	0.8226
215	No Queue	MAE	1.7606
215	No Date	MAE	1.1261
215	No Historical CT	MAE	0.7550
216	All	MSE	13.2011
216	No Queue	MSE	13.0116
216	No Date	MSE	13.9465
216	No Historical CT	MSE	2.8757
216	All	MAE	1.7018
216	No Queue	MAE	1.6534
216	No Date	MAE	1.8464

80	No Historical CT	MSE	0.1929
80	All	MAE	0.2902
80	No Queue	MAE	0.3223
80	No Date	MAE	0.3181
80	No Historical CT	MAE	0.1543
81	All	MSE	232.9886
81	No Queue	MSE	239.9817
81	No Date	MSE	229.4022
81	No Historical CT	MSE	636.8554
81	All	MAE	3.5128
81	No Queue	MAE	3.8704
81	No Date	MAE	3.3610
81	No Historical CT	MAE	11.6005
82	All	MSE	0.0399
82	No Queue	MSE	0.0428
82	No Date	MSE	0.0645
82	No Historical CT	MSE	0.0158
82	All	MAE	0.0746
82	No Queue	MAE	0.0778
82	No Date	MAE	0.0976
82	No Historical CT	MAE	0.0619
83	All	MSE	50.3198
83	No Queue	MSE	51.5547
83	No Date	MSE	50.2625
83	No Historical CT	MSE	55.2611
83	All	MAE	2.3859
83	No Queue	MAE	2.8300
83	No Date	MAE	2.4068
83	No Historical CT	MAE	2.6086
84	All	MSE	0.0041
84	No Queue	MSE	0.0038
84	No Date	MSE	0.0010
84	No Historical CT	MSE	0.0006
84	All	MAE	0.0332
84	No Queue	MAE	0.0330
84	No Date	MAE	0.0187
84	No Historical CT	MAE	0.0148
85	All	MSE	0.0701
85	No Queue	MSE	0.0774
85	No Date	MSE	0.0713
85	No Historical CT	MSE	0.0612
85	All	MAE	0.1726
85	No Queue	MAE	0.1816
85	No Date	MAE	0.1739
85	No Historical CT	MAE	0.1391
86	All	MSE	0.2573
86	No Queue	MSE	0.2680
86	No Date	MSE	0.3880
86	No Historical CT	MSE	0.1647
86	All	MAE	0.1530
86	No Queue	MAE	0.1652
86	No Date	MAE	0.2028
86	No Historical CT	MAE	0.1470
87	All	MSE	95.0271
87	No Queue	MSE	98.8684
87	No Date	MSE	95.9077
87	No Historical CT	MSE	123.3475
87	All	MAE	3.6711

216	No Historical CT	MAE	0.8696
217	All	MSE	14.5546
217	No Queue	MSE	28.4479
217	No Date	MSE	12.4961
217	No Historical CT	MSE	20.9589
217	All	MAE	2.0145
217	No Queue	MAE	2.9772
217	No Date	MAE	1.8570
217	No Historical CT	MAE	1.4295
218	All	MSE	3.4086
218	No Queue	MSE	3.5566
218	No Date	MSE	8.2367
218	No Historical CT	MSE	0.0345
218	All	MAE	0.9139
218	No Queue	MAE	1.1324
218	No Date	MAE	1.6070
218	No Historical CT	MAE	0.1184
219	All	MSE	6.0104
219	No Queue	MSE	12.2509
219	No Date	MSE	7.6888
219	No Historical CT	MSE	2.4974
219	All	MAE	1.3589
219	No Queue	MAE	1.9440
219	No Date	MAE	1.6844
219	No Historical CT	MAE	0.8267
220	All	MSE	19.0311
220	No Queue	MSE	18.7957
220	No Date	MSE	18.4915
220	No Historical CT	MSE	21.7163
220	All	MAE	1.6756
220	No Queue	MAE	1.7072
220	No Date	MAE	1.7438
220	No Historical CT	MAE	2.1541
221	All	MSE	0.0554
221	No Queue	MSE	0.0614
221	No Date	MSE	0.0670
221	No Historical CT	MSE	0.0367
221	All	MAE	0.1066
221	No Queue	MAE	0.1204
221	No Date	MAE	0.1151
221	No Historical CT	MAE	0.0889
222	All	MSE	0.2578
222	No Queue	MSE	4.7820
222	No Date	MSE	0.1623
222	No Historical CT	MSE	0.6643
222	All	MAE	0.3759
222	No Queue	MAE	1.7161
222	No Date	MAE	0.3842
222	No Historical CT	MAE	0.6036
223	All	MSE	8.2086
223	No Queue	MSE	7.9117
223	No Date	MSE	6.2273
223	No Historical CT	MSE	9.0146
223	All	MAE	1.5858
223	No Queue	MAE	1.6309
223	No Date	MAE	1.5313
223	No Historical CT	MAE	1.9636
224	All	MSE	7.0147

87	No Queue	MAE	3.8229
87	No Date	MAE	3.7542
87	No Historical CT	MAE	7.1821
88	All	MSE	0.0966
88	No Queue	MSE	0.0950
88	No Date	MSE	0.0928
88	No Historical CT	MSE	0.0426
88	All	MAE	0.0869
88	No Queue	MAE	0.0884
88	No Date	MAE	0.0936
88	No Historical CT	MAE	0.0618
89	All	MSE	77.7464
89	No Queue	MSE	82.0798
89	No Date	MSE	75.3812
89	No Historical CT	MSE	59.8758
89	All	MAE	3.7191
89	No Queue	MAE	4.2175
89	No Date	MAE	3.7117
89	No Historical CT	MAE	4.4532
90	All	MSE	1.1034
90	No Queue	MSE	1.1647
90	No Date	MSE	1.4307
90	No Historical CT	MSE	0.5772
90	All	MAE	0.3165
90	No Queue	MAE	0.3208
90	No Date	MAE	0.5177
90	No Historical CT	MAE	0.1492
91	All	MSE	10.4627
91	No Queue	MSE	12.2336
91	No Date	MSE	11.2947
91	No Historical CT	MSE	9.1272
91	All	MAE	1.2577
91	No Queue	MAE	1.4157
91	No Date	MAE	1.3601
91	No Historical CT	MAE	1.5373
92	All	MSE	0.5669
92	No Queue	MSE	0.6323
92	No Date	MSE	0.6809
92	No Historical CT	MSE	0.7637
92	All	MAE	0.1799
92	No Queue	MAE	0.2044
92	No Date	MAE	0.2112
92	No Historical CT	MAE	0.3039
93	All	MSE	1.6638
93	No Queue	MSE	2.0622
93	No Date	MSE	2.1608
93	No Historical CT	MSE	1.8766
93	All	MAE	0.3083
93	No Queue	MAE	0.3528
93	No Date	MAE	0.4724
93	No Historical CT	MAE	0.3694
94	All	MSE	6.4389
94	No Queue	MSE	7.8694
94	No Date	MSE	7.1885
94	No Historical CT	MSE	2.4074
94	All	MAE	0.5902
94	No Queue	MAE	0.6406
94	No Date	MAE	0.6361

224	No Queue	MSE	6.1412
224	No Date	MSE	9.4225
224	No Historical CT	MSE	7.5076
224	All	MAE	1.9695
224	No Queue	MAE	1.7432
224	No Date	MAE	2.6584
224	No Historical CT	MAE	1.5161
225	All	MSE	3.2838
225	No Queue	MSE	3.7428
225	No Date	MSE	3.9627
225	No Historical CT	MSE	0.5563
225	All	MAE	0.9029
225	No Queue	MAE	0.9835
225	No Date	MAE	1.0341
225	No Historical CT	MAE	0.3822
226	All	MSE	0.0211
226	No Queue	MSE	0.0239
226	No Date	MSE	0.0283
226	No Historical CT	MSE	0.0526
226	All	MAE	0.0813
226	No Queue	MAE	0.0844
226	No Date	MAE	0.0886
226	No Historical CT	MAE	0.0872
227	All	MSE	0.0259
227	No Queue	MSE	0.4269
227	No Date	MSE	0.0373
227	No Historical CT	MSE	0.0191
227	All	MAE	0.0523
227	No Queue	MAE	0.1964
227	No Date	MAE	0.0638
227	No Historical CT	MAE	0.0255
228	All	MSE	2.7121
228	No Queue	MSE	3.2538
228	No Date	MSE	3.4131
228	No Historical CT	MSE	1.0635
228	All	MAE	0.6303
228	No Queue	MAE	0.7384
228	No Date	MAE	0.8176
228	No Historical CT	MAE	0.3533
229	All	MSE	1.6299
229	No Queue	MSE	4.3323
229	No Date	MSE	2.1285
229	No Historical CT	MSE	0.9182
229	All	MAE	0.8552
229	No Queue	MAE	1.0460
229	No Date	MAE	0.9589
229	No Historical CT	MAE	0.5246
230	All	MSE	1383.4350
230	No Queue	MSE	1461.0930
230	No Date	MSE	1393.0517
230	No Historical CT	MSE	3660.8625
230	All	MAE	14.0064
230	No Queue	MAE	13.7986
230	No Date	MAE	13.6465
230	No Historical CT	MAE	31.6665
231	All	MSE	7.6654
231	No Queue	MSE	8.0985
231	No Date	MSE	8.6062

94	No Historical CT	MAE	0.4586
95	All	MSE	0.3115
95	No Queue	MSE	0.6500
95	No Date	MSE	0.3524
95	No Historical CT	MSE	0.2154
95	All	MAE	0.2524
95	No Queue	MAE	0.4338
95	No Date	MAE	0.2683
95	No Historical CT	MAE	0.1880
96	All	MSE	0.1965
96	No Queue	MSE	0.1725
96	No Date	MSE	0.2139
96	No Historical CT	MSE	0.0664
96	All	MAE	0.0828
96	No Queue	MAE	0.0835
96	No Date	MAE	0.0844
96	No Historical CT	MAE	0.1035
97	All	MSE	51.3912
97	No Queue	MSE	175.7317
97	No Date	MSE	43.7695
97	No Historical CT	MSE	88.6479
97	All	MAE	4.1135
97	No Queue	MAE	8.2148
97	No Date	MAE	3.1240
97	No Historical CT	MAE	6.7134
98	All	MSE	11.6870
98	No Queue	MSE	16.1043
98	No Date	MSE	9.2667
98	No Historical CT	MSE	15.4952
98	All	MAE	2.6235
98	No Queue	MAE	2.7749
98	No Date	MAE	2.2335
98	No Historical CT	MAE	3.0101
99	All	MSE	12.8278
99	No Queue	MSE	12.6621
99	No Date	MSE	15.6801
99	No Historical CT	MSE	19.1629
99	All	MAE	1.1033
99	No Queue	MAE	1.1112
99	No Date	MAE	1.4389
99	No Historical CT	MAE	1.9206
100	All	MSE	3.9905
100	No Queue	MSE	4.0551
100	No Date	MSE	4.1477
100	No Historical CT	MSE	2.4683
100	All	MAE	0.7266
100	No Queue	MAE	0.7843
100	No Date	MAE	0.7457
100	No Historical CT	MAE	0.4260
101	All	MSE	33.5174
101	No Queue	MSE	45.4335
101	No Date	MSE	30.3593
101	No Historical CT	MSE	158.1376
101	All	MAE	3.1877
101	No Queue	MAE	3.7730
101	No Date	MAE	3.0017
101	No Historical CT	MAE	8.0846
102	All	MSE	8.3660

231	No Historical CT	MSE	10.1001
231	All	MAE	0.7976
231	No Queue	MAE	0.8836
231	No Date	MAE	0.9410
231	No Historical CT	MAE	0.9142
232	All	MSE	18.4332
232	No Queue	MSE	19.9756
232	No Date	MSE	19.6872
232	No Historical CT	MSE	26.3961
232	All	MAE	1.1222
232	No Queue	MAE	1.2417
232	No Date	MAE	1.1864
232	No Historical CT	MAE	1.7564
233	All	MSE	0.1385
233	No Queue	MSE	0.1522
233	No Date	MSE	0.1833
233	No Historical CT	MSE	0.0532
233	All	MAE	0.1474
233	No Queue	MAE	0.1540
233	No Date	MAE	0.1762
233	No Historical CT	MAE	0.1153
234	All	MSE	1.1916
234	No Queue	MSE	1.2531
234	No Date	MSE	1.3419
234	No Historical CT	MSE	0.7533
234	All	MAE	0.3785
234	No Queue	MAE	0.3981
234	No Date	MAE	0.4215
234	No Historical CT	MAE	0.2492
235	All	MSE	479.3272
235	No Queue	MSE	538.3945
235	No Date	MSE	425.6994
235	No Historical CT	MSE	436.1916
235	All	MAE	8.0107
235	No Queue	MAE	8.1800
235	No Date	MAE	8.0429
235	No Historical CT	MAE	7.7471
236	All	MSE	4.0348
236	No Queue	MSE	3.9889
236	No Date	MSE	4.3450
236	No Historical CT	MSE	9.9434
236	All	MAE	0.9939
236	No Queue	MAE	1.0047
236	No Date	MAE	0.9772
236	No Historical CT	MAE	1.5014
237	All	MSE	0.2658
237	No Queue	MSE	0.2832
237	No Date	MSE	0.2884
237	No Historical CT	MSE	0.1206
237	All	MAE	0.2808
237	No Queue	MAE	0.3041
237	No Date	MAE	0.2952
237	No Historical CT	MAE	0.1548
238	All	MSE	16.9071
238	No Queue	MSE	16.2340
238	No Date	MSE	18.0697
238	No Historical CT	MSE	21.7422
238	All	MAE	0.8916

102	No Queue	MSE	8.1053
102	No Date	MSE	7.6165
102	No Historical CT	MSE	0.9888
102	All	MAE	1.4859
102	No Queue	MAE	1.5431
102	No Date	MAE	1.6028
102	No Historical CT	MAE	0.6998
103	All	MSE	0.0005
103	No Queue	MSE	0.0006
103	No Date	MSE	0.0007
103	No Historical CT	MSE	0.0006
103	All	MAE	0.0183
103	No Queue	MAE	0.0183
103	No Date	MAE	0.0182
103	No Historical CT	MAE	0.0197
104	All	MSE	8.7816
104	No Queue	MSE	10.4742
104	No Date	MSE	10.1351
104	No Historical CT	MSE	4.5870
104	All	MAE	0.9098
104	No Queue	MAE	1.0494
104	No Date	MAE	1.0428
104	No Historical CT	MAE	0.8389
105	All	MSE	0.0589
105	No Queue	MSE	0.0617
105	No Date	MSE	0.0592
105	No Historical CT	MSE	0.0086
105	All	MAE	0.1091
105	No Queue	MAE	0.1160
105	No Date	MAE	0.1048
105	No Historical CT	MAE	0.0493
106	All	MSE	5.8356
106	No Queue	MSE	6.2485
106	No Date	MSE	6.2509
106	No Historical CT	MSE	4.0961
106	All	MAE	0.8150
106	No Queue	MAE	0.8801
106	No Date	MAE	0.9049
106	No Historical CT	MAE	0.6455
107	All	MSE	0.5762
107	No Queue	MSE	0.8248
107	No Date	MSE	0.6854
107	No Historical CT	MSE	0.7954
107	All	MAE	0.3458
107	No Queue	MAE	0.4309
107	No Date	MAE	0.3924
107	No Historical CT	MAE	0.3267
108	All	MSE	0.0583
108	No Queue	MSE	0.0683
108	No Date	MSE	0.0881
108	No Historical CT	MSE	0.0259
108	All	MAE	0.0919
108	No Queue	MAE	0.1000
108	No Date	MAE	0.1223
108	No Historical CT	MAE	0.0651
109	All	MSE	0.1114
109	No Queue	MSE	0.4898
109	No Date	MSE	0.1461

238	No Queue	MAE	0.9063
238	No Date	MAE	0.9593
238	No Historical CT	MAE	1.1901
239	All	MSE	0.0405
239	No Queue	MSE	0.0542
239	No Date	MSE	0.0458
239	No Historical CT	MSE	0.0036
239	All	MAE	0.0751
239	No Queue	MAE	0.0886
239	No Date	MAE	0.0808
239	No Historical CT	MAE	0.0297
240	All	MSE	1.4932
240	No Queue	MSE	1.8124
240	No Date	MSE	1.6336
240	No Historical CT	MSE	2.4087
240	All	MAE	0.2956
240	No Queue	MAE	0.3816
240	No Date	MAE	0.3451
240	No Historical CT	MAE	0.6300
241	All	MSE	0.0885
241	No Queue	MSE	0.1079
241	No Date	MSE	0.0844
241	No Historical CT	MSE	0.0106
241	All	MAE	0.1325
241	No Queue	MAE	0.1463
241	No Date	MAE	0.1271
241	No Historical CT	MAE	0.0899
242	All	MSE	0.3791
242	No Queue	MSE	0.4479
242	No Date	MSE	0.4113
242	No Historical CT	MSE	0.2390
242	All	MAE	0.3296
242	No Queue	MAE	0.3451
242	No Date	MAE	0.3488
242	No Historical CT	MAE	0.2157
243	All	MSE	0.0093
243	No Queue	MSE	0.0177
243	No Date	MSE	0.0086
243	No Historical CT	MSE	0.0003
243	All	MAE	0.0340
243	No Queue	MAE	0.0439
243	No Date	MAE	0.0342
243	No Historical CT	MAE	0.0110
244	All	MSE	19.3405
244	No Queue	MSE	22.5655
244	No Date	MSE	20.1393
244	No Historical CT	MSE	76.6502
244	All	MAE	1.6430
244	No Queue	MAE	1.7484
244	No Date	MAE	1.7881
244	No Historical CT	MAE	7.4748
245	All	MSE	14.8820
245	No Queue	MSE	23.3659
245	No Date	MSE	12.6019
245	No Historical CT	MSE	12.3506
245	All	MAE	1.6239
245	No Queue	MAE	1.8855
245	No Date	MAE	1.5389

109	No Historical CT	MSE	0.0727
109	All	MAE	0.0731
109	No Queue	MAE	0.1144
109	No Date	MAE	0.0859
109	No Historical CT	MAE	0.1362
110	All	MSE	0.2935
110	No Queue	MSE	0.3467
110	No Date	MSE	0.4518
110	No Historical CT	MSE	0.2281
110	All	MAE	0.1779
110	No Queue	MAE	0.2085
110	No Date	MAE	0.2335
110	No Historical CT	MAE	0.1324
111	All	MSE	0.0927
111	No Queue	MSE	0.1116
111	No Date	MSE	0.1222
111	No Historical CT	MSE	0.0161
111	All	MAE	0.1120
111	No Queue	MAE	0.1220
111	No Date	MAE	0.1343
111	No Historical CT	MAE	0.0473
112	All	MSE	0.1147
112	No Queue	MSE	0.1297
112	No Date	MSE	0.2442
112	No Historical CT	MSE	0.0156
112	All	MAE	0.0484
112	No Queue	MAE	0.0507
112	No Date	MAE	0.0938
112	No Historical CT	MAE	0.0176
113	All	MSE	0.0023
113	No Queue	MSE	0.0026
113	No Date	MSE	0.0023
113	No Historical CT	MSE	0.0004
113	All	MAE	0.0199
113	No Queue	MAE	0.0226
113	No Date	MAE	0.0201
113	No Historical CT	MAE	0.0143
114	All	MSE	0.3291
114	No Queue	MSE	0.3519
114	No Date	MSE	0.4028
114	No Historical CT	MSE	0.1675
114	All	MAE	0.1865
114	No Queue	MAE	0.1994
114	No Date	MAE	0.2249
114	No Historical CT	MAE	0.1455
115	All	MSE	0.0028
115	No Queue	MSE	0.0025
115	No Date	MSE	0.0039
115	No Historical CT	MSE	0.0070
115	All	MAE	0.0078
115	No Queue	MAE	0.0081
115	No Date	MAE	0.0099
115	No Historical CT	MAE	0.0136
116	All	MSE	0.0074
116	No Queue	MSE	0.0126
116	No Date	MSE	0.0069
116	No Historical CT	MSE	0.0039
116	All	MAE	0.0139

245	No Historical CT	MAE	1.4939
246	All	MSE	264.0304
246	No Queue	MSE	280.1531
246	No Date	MSE	274.4967
246	No Historical CT	MSE	218.2599
246	All	MAE	9.3309
246	No Queue	MAE	9.7569
246	No Date	MAE	9.2249
246	No Historical CT	MAE	7.0148
247	All	MSE	0.4270
247	No Queue	MSE	0.4610
247	No Date	MSE	0.4778
247	No Historical CT	MSE	0.2620
247	All	MAE	0.3159
247	No Queue	MAE	0.3338
247	No Date	MAE	0.3433
247	No Historical CT	MAE	0.2467
248	All	MSE	0.0453
248	No Queue	MSE	0.0451
248	No Date	MSE	0.0579
248	No Historical CT	MSE	0.0249
248	All	MAE	0.1057
248	No Queue	MAE	0.1075
248	No Date	MAE	0.1151
248	No Historical CT	MAE	0.0757
249	All	MSE	18.8291
249	No Queue	MSE	13.3684
249	No Date	MSE	32.7791
249	No Historical CT	MSE	39.8064
249	All	MAE	2.8093
249	No Queue	MAE	2.3917
249	No Date	MAE	3.6474
249	No Historical CT	MAE	3.7348
250	All	MSE	18.4910
250	No Queue	MSE	26.4232
250	No Date	MSE	23.5443
250	No Historical CT	MSE	0.0002
250	All	MAE	1.0371
250	No Queue	MAE	1.3581
250	No Date	MAE	1.1591
250	No Historical CT	MAE	0.0067
251	All	MSE	0.0001
251	No Queue	MSE	0.0000
251	No Date	MSE	0.0000
251	No Historical CT	MSE	0.0000
251	All	MAE	0.0052
251	No Queue	MAE	0.0044
251	No Date	MAE	0.0052
251	No Historical CT	MAE	0.0043
252	All	MSE	0.0061
252	No Queue	MSE	0.0579
252	No Date	MSE	0.0098
252	No Historical CT	MSE	0.0000
252	All	MAE	0.0318
252	No Queue	MAE	0.0983
252	No Date	MAE	0.0596
252	No Historical CT	MAE	0.0007
253	All	MSE	1.0216

116	No Queue	MAE	0.0178
116	No Date	MAE	0.0121
116	No Historical CT	MAE	0.0068
117	All	MSE	0.9174
117	No Queue	MSE	1.0530
117	No Date	MSE	1.2066
117	No Historical CT	MSE	0.4846
117	All	MAE	0.5550
117	No Queue	MAE	0.6131
117	No Date	MAE	0.6680
117	No Historical CT	MAE	0.3161
118	All	MSE	13.6055
118	No Queue	MSE	13.9741
118	No Date	MSE	21.1542
118	No Historical CT	MSE	4.0176
118	All	MAE	2.4669
118	No Queue	MAE	2.4038
118	No Date	MAE	3.3089
118	No Historical CT	MAE	1.5945
119	All	MSE	3.0954
119	No Queue	MSE	3.6010
119	No Date	MSE	4.9335
119	No Historical CT	MSE	1.9559
119	All	MAE	0.4013
119	No Queue	MAE	0.4808
119	No Date	MAE	0.7696
119	No Historical CT	MAE	0.1519
120	All	MSE	1.0421
120	No Queue	MSE	1.0837
120	No Date	MSE	1.1213
120	No Historical CT	MSE	0.7462
120	All	MAE	0.2584
120	No Queue	MAE	0.2757
120	No Date	MAE	0.2866
120	No Historical CT	MAE	0.2133
121	All	MSE	0.8601
121	No Queue	MSE	1.1448
121	No Date	MSE	1.8906
121	No Historical CT	MSE	0.1033
121	All	MAE	0.5209
121	No Queue	MAE	0.5944
121	No Date	MAE	0.8434
121	No Historical CT	MAE	0.1814
122	All	MSE	0.2522
122	No Queue	MSE	0.3063
122	No Date	MSE	0.4800
122	No Historical CT	MSE	0.1146
122	All	MAE	0.2571
122	No Queue	MAE	0.2845
122	No Date	MAE	0.4148
122	No Historical CT	MAE	0.1478
123	All	MSE	0.0129
123	No Queue	MSE	0.0179
123	No Date	MSE	0.0340
123	No Historical CT	MSE	0.0053
123	All	MAE	0.0160
123	No Queue	MAE	0.0211
123	No Date	MAE	0.0293

253	No Queue	MSE	0.9879
253	No Date	MSE	1.0127
253	No Historical CT	MSE	1.1254
253	All	MAE	0.6861
253	No Queue	MAE	0.6797
253	No Date	MAE	0.6268
253	No Historical CT	MAE	0.6660
254	All	MSE	86.5914
254	No Queue	MSE	121.5961
254	No Date	MSE	84.9149
254	No Historical CT	MSE	40.3383
254	All	MAE	5.1473
254	No Queue	MAE	6.1693
254	No Date	MAE	5.6097
254	No Historical CT	MAE	3.3669
255	All	MSE	0.9653
255	No Queue	MSE	1.0826
255	No Date	MSE	1.0911
255	No Historical CT	MSE	0.4833
255	All	MAE	0.4117
255	No Queue	MAE	0.4416
255	No Date	MAE	0.3946
255	No Historical CT	MAE	0.3040
256	All	MSE	32.3778
256	No Queue	MSE	30.6092
256	No Date	MSE	31.7650
256	No Historical CT	MSE	48.8153
256	All	MAE	1.4731
256	No Queue	MAE	1.5054
256	No Date	MAE	1.5380
256	No Historical CT	MAE	2.4869
257	All	MSE	62.4045
257	No Queue	MSE	78.5111
257	No Date	MSE	59.2497
257	No Historical CT	MSE	17.4186
257	All	MAE	5.9182
257	No Queue	MAE	6.9197
257	No Date	MAE	5.8494
257	No Historical CT	MAE	3.3135
258	All	MSE	28.6660
258	No Queue	MSE	24.5511
258	No Date	MSE	28.7517
258	No Historical CT	MSE	34.9892
258	All	MAE	2.8205
258	No Queue	MAE	2.5586
258	No Date	MAE	2.9204
258	No Historical CT	MAE	2.9681
259	All	MSE	3.3611
259	No Queue	MSE	6.4399
259	No Date	MSE	3.4294
259	No Historical CT	MSE	1.4681
259	All	MAE	1.1259
259	No Queue	MAE	1.8049
259	No Date	MAE	1.1808
259	No Historical CT	MAE	0.5437
260	All	MSE	117.3521
260	No Queue	MSE	121.7190
260	No Date	MSE	118.6950

123	No Historical CT	MAE	0.0060
124	All	MSE	0.2824
124	No Queue	MSE	0.2923
124	No Date	MSE	0.3648
124	No Historical CT	MSE	0.0892
124	All	MAE	0.1743
124	No Queue	MAE	0.1857
124	No Date	MAE	0.2096
124	No Historical CT	MAE	0.1168
125	All	MSE	18.0047
125	No Queue	MSE	17.9321
125	No Date	MSE	20.2656
125	No Historical CT	MSE	23.7585
125	All	MAE	1.0746
125	No Queue	MAE	1.1251
125	No Date	MAE	1.2083
125	No Historical CT	MAE	1.3462
126	All	MSE	62.5000
126	No Queue	MSE	70.7959
126	No Date	MSE	69.5158
126	No Historical CT	MSE	125.2783
126	All	MAE	2.5265
126	No Queue	MAE	2.8318
126	No Date	MAE	2.7663
126	No Historical CT	MAE	4.1465
127	All	MSE	1.0953
127	No Queue	MSE	1.1555
127	No Date	MSE	1.4065
127	No Historical CT	MSE	0.6024
127	All	MAE	0.3168
127	No Queue	MAE	0.3442
127	No Date	MAE	0.4539
127	No Historical CT	MAE	0.1841
128	All	MSE	0.6447
128	No Queue	MSE	0.7193
128	No Date	MSE	0.8763
128	No Historical CT	MSE	0.1309
128	All	MAE	0.2746
128	No Queue	MAE	0.2988
128	No Date	MAE	0.3611
128	No Historical CT	MAE	0.1000
129	All	MSE	0.4317
129	No Queue	MSE	0.4543
129	No Date	MSE	0.5282
129	No Historical CT	MSE	0.3216
129	All	MAE	0.1854
129	No Queue	MAE	0.1957
129	No Date	MAE	0.2183
129	No Historical CT	MAE	0.2618
130	All	MSE	384.5342
130	No Queue	MSE	379.7439
130	No Date	MSE	423.9955
130	No Historical CT	MSE	275.2573
130	All	MAE	10.0054
130	No Queue	MAE	10.2933
130	No Date	MAE	11.0561
130	No Historical CT	MAE	6.9185
131	All	MSE	0.7435
260	No Historical CT	MSE	130.7416
260	All	MAE	4.4900
260	No Queue	MAE	4.8252
260	No Date	MAE	4.5571
260	No Historical CT	MAE	4.7827
261	All	MSE	1.6171
261	No Queue	MSE	1.7060
261	No Date	MSE	1.7111
261	No Historical CT	MSE	0.8314
261	All	MAE	0.5695
261	No Queue	MAE	0.6056
261	No Date	MAE	0.5977
261	No Historical CT	MAE	0.3803
262	All	MSE	26.5197
262	No Queue	MSE	28.0117
262	No Date	MSE	32.5681
262	No Historical CT	MSE	23.6398
262	All	MAE	1.6983
262	No Queue	MAE	1.9136
262	No Date	MAE	2.0939
262	No Historical CT	MAE	1.2290
263	All	MSE	12.0650
263	No Queue	MSE	15.4598
263	No Date	MSE	15.2854
263	No Historical CT	MSE	5.1777
263	All	MAE	0.8592
263	No Queue	MAE	1.0333
263	No Date	MAE	1.2413
263	No Historical CT	MAE	0.3386
264	All	MSE	4.7182
264	No Queue	MSE	5.5147
264	No Date	MSE	10.3639
264	No Historical CT	MSE	2.1599
264	All	MAE	0.4849
264	No Queue	MAE	0.5782
264	No Date	MAE	1.0189
264	No Historical CT	MAE	0.2226
265	All	MSE	3.1148
265	No Queue	MSE	3.2856
265	No Date	MSE	3.1678
265	No Historical CT	MSE	3.4351
265	All	MAE	0.9814
265	No Queue	MAE	1.0308
265	No Date	MAE	1.0335
265	No Historical CT	MAE	0.8293
266	All	MSE	0.3176
266	No Queue	MSE	0.3164
266	No Date	MSE	0.3444
266	No Historical CT	MSE	0.0504
266	All	MAE	0.1752
266	No Queue	MAE	0.1800
266	No Date	MAE	0.1997
266	No Historical CT	MAE	0.1194
267	All	MSE	6.2079
267	No Queue	MSE	6.7959
267	No Date	MSE	6.8166
267	No Historical CT	MSE	5.5299
267	All	MAE	0.7529

131	No Queue	MSE	0.8701
131	No Date	MSE	0.8564
131	No Historical CT	MSE	0.2216
131	All	MAE	0.4056
131	No Queue	MAE	0.4566
131	No Date	MAE	0.3582
131	No Historical CT	MAE	0.2654
132	All	MSE	1959.5211
132	No Queue	MSE	1960.5576
132	No Date	MSE	1984.3373
132	No Historical CT	MSE	1973.4292
132	All	MAE	15.6054
132	No Queue	MAE	16.4584
132	No Date	MAE	16.0977
132	No Historical CT	MAE	14.7347
133	All	MSE	7.6882
133	No Queue	MSE	6.2042
133	No Date	MSE	8.2480
133	No Historical CT	MSE	5.3903
133	All	MAE	1.2866
133	No Queue	MAE	1.0174
133	No Date	MAE	1.3271
133	No Historical CT	MAE	0.8391
134	All	MSE	2.1070
134	No Queue	MSE	4.2612
134	No Date	MSE	3.0737
134	No Historical CT	MSE	0.7660
134	All	MAE	0.4800
134	No Queue	MAE	0.7613
134	No Date	MAE	0.7652
134	No Historical CT	MAE	0.4039
135	All	MSE	1.1041
135	No Queue	MSE	1.4509
135	No Date	MSE	1.1568
135	No Historical CT	MSE	0.4768
135	All	MAE	0.6809
135	No Queue	MAE	0.8204
135	No Date	MAE	0.6953
135	No Historical CT	MAE	0.4029
136	All	MSE	0.0750
136	No Queue	MSE	0.4881
136	No Date	MSE	0.5357
136	No Historical CT	MSE	0.0304
136	All	MAE	0.1589
136	No Queue	MAE	0.3170
136	No Date	MAE	0.3718
136	No Historical CT	MAE	0.1006
137	All	MSE	0.0243
137	No Queue	MSE	0.0498
137	No Date	MSE	0.0240
137	No Historical CT	MSE	0.0279

267	No Queue	MAE	0.8207
267	No Date	MAE	0.8773
267	No Historical CT	MAE	1.0425
268	All	MSE	36.8799
268	No Queue	MSE	20.3439
268	No Date	MSE	40.0959
268	No Historical CT	MSE	92.9418
268	All	MAE	0.8541
268	No Queue	MAE	0.7969
268	No Date	MAE	1.0493
268	No Historical CT	MAE	4.0842
269	All	MSE	16.0759
269	No Queue	MSE	15.0228
269	No Date	MSE	19.5621
269	No Historical CT	MSE	26.5556
269	All	MAE	0.7879
269	No Queue	MAE	0.7521
269	No Date	MAE	0.9558
269	No Historical CT	MAE	1.2543
270	All	MSE	4.5544
270	No Queue	MSE	7.1612
270	No Date	MSE	4.6391
270	No Historical CT	MSE	0.4027
270	All	MAE	0.8115
270	No Queue	MAE	0.9635
270	No Date	MAE	0.7521
270	No Historical CT	MAE	0.3084
271	All	MSE	2.6355
271	No Queue	MSE	7.8212
271	No Date	MSE	1.5516
271	No Historical CT	MSE	6.1738
271	All	MAE	1.0817
271	No Queue	MAE	1.6469
271	No Date	MAE	0.8302
271	No Historical CT	MAE	1.0841
272	All	MSE	25.5707
272	No Queue	MSE	30.7128
272	No Date	MSE	62.6289
272	No Historical CT	MSE	23.9419
272	All	MAE	1.8207
272	No Queue	MAE	2.0558
272	No Date	MAE	3.0756
272	No Historical CT	MAE	1.2359
273	All	MSE	0.8721
273	No Queue	MSE	0.9684
273	No Date	MSE	0.7934
273	No Historical CT	MSE	0.4153
273	All	MAE	0.6996
273	No Queue	MAE	0.7488
273	No Date	MAE	0.6877
273	No Historical CT	MAE	0.4991