Forecasting Parts Demand Using Service Data and Machine Learning

Volume 1

Sergio Posadas

Carl M. Kruger

Catherine M. Beazley

Russell S. Salley

John A. Stephenson

Esther C. Thron

Justin D. Ward

January 2020

NOTICE:

THE VIEWS, OPINIONS, AND FINDINGS CONTAINED IN THIS REPORT ARE THOSE OF LMI AND SHOULD NOT BE CONSTRUED AS AN OFFICIAL AGENCY POSITION, POLICY, OR DECISION, UNLESS SO DESIGNATED BY OTHER OFFICIAL DOCUMENTATION.

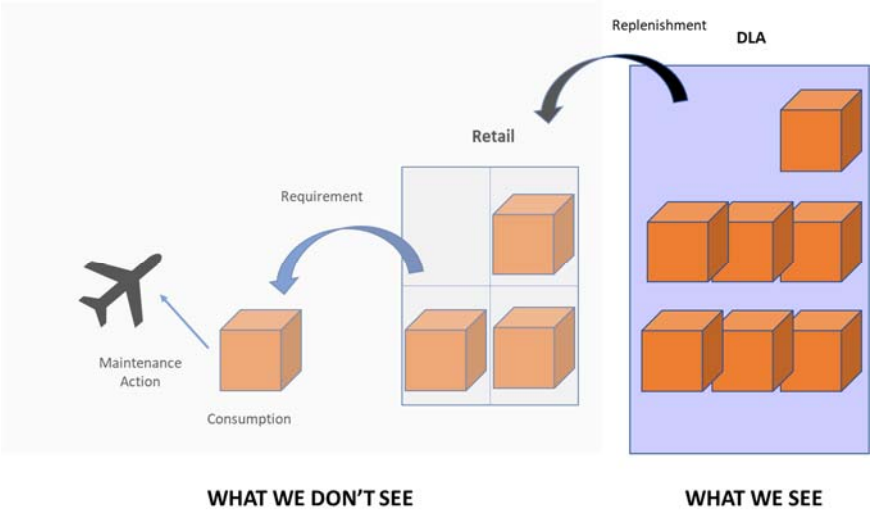
Executive Summary

LMI completed a research and development (R&D) project to evaluate machine learning (ML) as a potential approach for improving forecasts of part demands. **Data sparsity limited the success of ML across a wide range of techniques.** As a result of this research focused recommendations for follow-on R&D are identified to explore the successful elements of this investigation.

Due to the sparsity of the available demand data, using Maintenance and Availability Data Warehouse™ (MADW™) data alone is insufficient to improve the Defense Logistics Agency (DLA) demand forecasts beyond those using established DLA models. The F/A-18E/F was selected as the trial platform for the project. Data sparsity was encountered across all platform component part demands. However, **ML models of maintenance events, rather than part demands, delivered improved forecasting over time series modeling while addressing data sparsity concerns.** Further research should be performed to determine whether this approach can enhance customer service, leading to improved readiness of the supported weapon systems.

To improve further analysis, use of additional data, beyond MADW™, should be explored to develop a more complete view of supply chain demands. The multi-echelon supply system should be modeled for added precision. Consumption data along with both retail and wholesale demands should be included in the follow-on models. Figure ES-1 shows the partial view provided by the data available for this R&D project compared to the full range of supply chain demands.

Figure ES-1. Using Consumption Data for Acquisition



Forecasting demand for repair parts for weapon systems is a challenging objective that requires advanced analytics. The demand varies greatly, maintenance patterns change, and multiple inventory levels shroud true consumption patterns. Through R&D, DLA wants to create and test a minimum viable product using ML techniques. LMI applied these techniques on historical data from Service maintenance records and a Condition Based Maintenance Plus (CBM+) program to forecast parts demand. Improved forecasts would enable DLA to better manage the supply chain, enhancing support to retail customers.

DLA's objective is to explore the value of applying ML to Service historical maintenance records, which contain detailed information. This R&D project focuses on the F/A-18E/F Generator Converter Unit (GCU), a top degrader for the aircraft fleet.

As the GCU is identified by the Navy as a top degrader there was in fact less sparsity than for other components. Key insights from the GCU analysis include the following:

- **Performance of ML models using demand data is limited by the sparsity of data.** Infrequent part orders cause the data sparsity over time. The best-performing ML models consistently under-predict part demands.
- **ML models using maintenance action forecasting mitigate the data sparsity issue** and perform better than time series models for maintenance actions. Using ML to predict maintenance action should be further investigated as a means of improving part demand forecasts across other platforms and parts.
- Change point detection may be useful in **signaling the need for a different ML model**.

We recommend five follow-on options, directly related to ML modeling, to improve parts forecasting. We also recommend an alternative to ML rooted in simulation modeling:

1. *Predict maintenance events and associated usage bills of materiel (BOMs)*. This recommendation expands on promising results from maintenance action forecasting with ML to overcome sparsity. Using data available in the MADW™

and applying ML techniques, we can produce a BOM for use with predicted maintenance events:

- Building on the promising results, use ML to forecast the organizational (O-level) and intermediate (I-level) maintenance requirements for the asset.
- Use ML to identify a BOM for each identified maintenance event.
- Use this BOM to forecast parts required for the maintenance availability.

2. *Predict changes in maintenance requirements*. Current forecasting models use historical demand data without regard for changes in requirements due to changes in airframe construct, updates to components, or aging. Using the data available in the MADW™ and applying ML, we can predict changing maintenance requirements:
   - Identify changes discovered in recent maintenance events.
   - Identify changes in parts requirements associated with these changes.
   - Identify future requirements on the basis of these changes.
3. *Use ML to improve readiness and acquisition using consumption data*. From experience, by using consumption data available at the retail level and applying ML, we can help improve overall readiness and availability of the airframes. DLA will see actual consumption levels, rather than waiting and relying on wholesale requisitioning. This approach will more accurately reflect usage and changes in usage at the National Item Identification Number level in a timelier manner. If historical replenishment has been for 100 units, but consumption has declined by 50 percent, DLA can anticipate a decrease in frequency or quantities of replenishment requisitions. Knowing this can help prevent over-stocking inventory (and tying up acquisition funding) as well as under-stocking (resulting in out-of-stocks and decreased customer service):
   - Identify actual consumption at the O/I levels.
   - Identify retail issues to support consumption.
   - Associate the retail issues to on-hand inventory and identify changes at the retail level.
   - Recommend acquisition of wholesale inventories from observed trends in the consumption data.
4. *Dynamically train ML models*. Dynamically choosing models over time can help improve accuracy. Utilizing change points can be helpful if not many reside within the data and lag times are insufficient to cause problems. Evaluating models on the basis of past performance is also a viable solution, but it can be time consuming.

   The MADW™ provides operational-level demand data for individual parts over time. These data furnish consumption information not found in the current DLA wholesale requisitioning data. Using data from both of these systems in concert with other available data (platform operational and DLA acquisition data) renders a more holistic picture of parts support and a subsequent improvement in forecasting for each layer of inventory.
5. *As an alternative to ML, use predictive simulation modeling to determine future part demands*. To leverage service maintenance data to improve part forecasts,

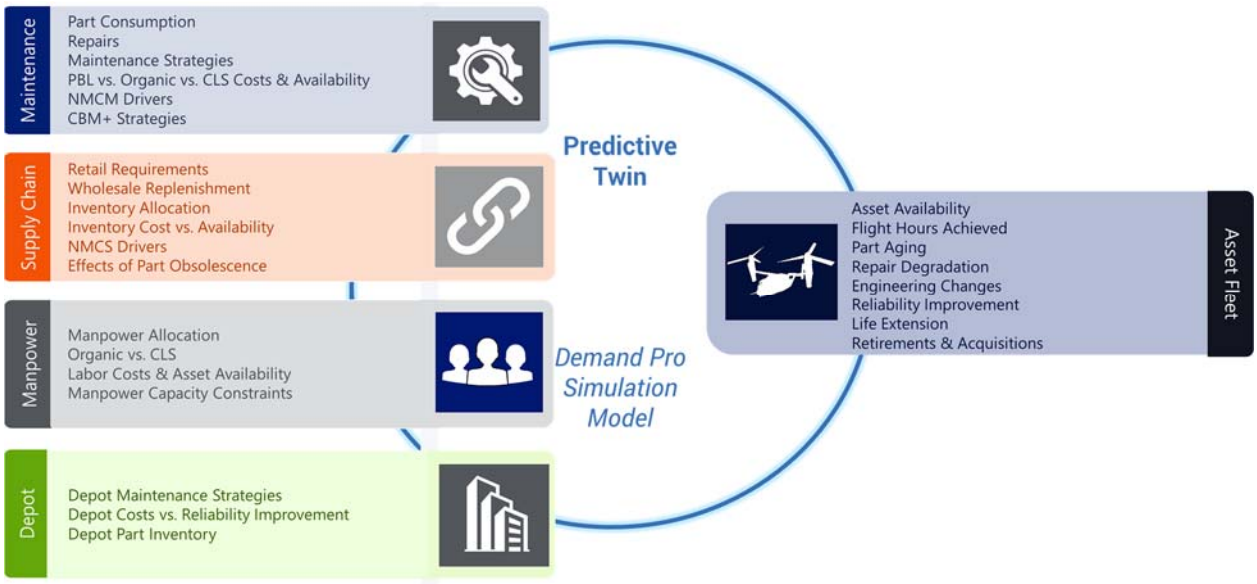
DLA can use predictive modeling (using asset-focused, high-resolution simulation) to balance inventory levels beyond the limitations of statistical models and traditional forecasting.

LMI successfully applied an asset-focused, high resolution simulation, Demand Pro, to many Department of Defense programs over the last 20 years. We have applied this proven capability to Air Force parts support for data cleansing, predictive maintenance actions, and ready for testing.

This platform has predicted part requirements at the operational level as well as the intermediate and depot levels. Demand Pro delivers a capability that far surpasses forecasting tools. Through past performance, we have demonstrated that this predictive simulation platform and prescriptive analytics are more accurate and deeper than widely used traditional forecasting methods.

Leveraging predictive simulation modeling can provide DLA with the holistic approach needed to capture the relationships between maintenance actions at the asset level, part consumption, retail requirements, and wholesale replenishment as summarized in Figure ES-2. Model elements from the simulation are listed and can be applied to predictive analysis according to the desired study objectives.

Figure ES-2. Leveraging Predictive Simulation Modeling



Note: CLS = contract logistics support; NMCM = not mission capable maintenance; NMCS = not mission capable supply; PBL = performance-based logistics.

Contents

Preface ..... **xi**
Chapter 1 Introduction **1-1**
Background 1-1
Objective 1-1
Chapter 2 Technical Concept and Approach **2-1**
Component Selection 2-1
Analysis Method 2-1
Task 1: Establish Working Group 2-2
Task 2: Analyze F/A-18 GCU 2-2
Task 3: Develop Predictive Models 2-2
Task 4: Compare Forecast with DLA Data 2-2
Task 5: Develop Business Cases for Predictive Models 2-2
Task 6: Deliver Final Report with Recommendations 2-3
Assumptions and Constraints 2-3
Data Collection and Analysis 2-3
MADW™ 2-3
Historical Demand Data 2-4
Historical Maintenance Data 2-5
Model Training and Evaluation 2-5
Machine Learning 2-5
Chapter 3 Analysis and Results **3-1**
Parts Demand Forecasting 3-1
Data 3-1
Exploratory Data Analysis 3-2
Evaluation Measures 3-6
Time Series Modeling 3-6
ML Modeling 3-9
Results 3-12
Additional Analysis: Part Order Classification 3-19
Maintenance Action Forecasting 3-20
Data 3-20

Evaluation Methods 3-21
Time Series Modeling 3-21
ML Modeling 3-21
Results 3-24
Additional Analysis 3-28
Chapter 4 Conclusions 4-1
Chapter 5 Recommendations for Further R&D 5-1
Predict Maintenance Events and Associated Usage BOMs 5-1
Predict Changes in Maintenance Requirements 5-2
Use ML to Improve Readiness and Acquisition Using Consumption Data 5-2
Dynamically Train ML Models 5-3
Use Predictive Simulation Modeling to Determine Future Part Demands 5-3
Appendix A Model Descriptions
Appendix B Detailed Modeling Method
Appendix C Recommendations for Future R&D Projects
Appendix D Dynamic ML Training Models
Appendix E Examples of Simulation-Driven Inventory Demand Predictions
Appendix F Abbreviations

Figures

Figure 2-1. MADW™ 2-4
Figure 3-1. Modeling Process 3-2
Figure 3-2. Second Quarter 2011 Parts Demand 3-3
Figure 3-3. Demand by Quarter 3-4
Figure 3-4. *k*-Means Clustering Using Three Clusters 3-5
Figure 3-5. Squared Distance for Different Cluster Amounts 3-5
Figure 3-6. Clustered NIINs over Time 3-6
Figure 3-7. Example Sliding Window 3-8
Figure 3-8. Autocorrelation 3-10
Figure 3-9. Partial Autocorrelation 3-10
Figure 3-10. Comparison of Time Series Models for NIIN 010050515 3-13
Figure 3-11. Comparison of Time Series Models for NIIN 014554498 3-13
Figure 3-12. Count of NIINs by Percent of Days with No Orders 3-14
Figure 3-13. Comparison of ML Models for NIIN 010050515 3-17
Figure 3-14. Comparison of ML Models for NIIN 014938822 3-17

Figure 3-15. P-R Curve for Logistics Ridge Regression 3-20
Figure 3-16. Autocorrelation 3-22
Figure 3-17. Partial Autocorrelation 3-22
Figure 3-18. Decomposition Plots 3-23
Figure 3-19. Comparison of Time Series Models 3-25
Figure 3-20. Comparison of ML Models 3-26
Figure 3-21. Random Forest Comparison 3-26
Figure 3-22. Random Forest and Moving Average Comparison 3-27
Figure 3-23. Weekly Number of Maintenance Events with Change Points 3-29
Figure 3-24. Cumulative Weekly Number of Maintenance Events with Change Points 3-29
Figure 3-25. Daily Number of Maintenance Events 3-30
Figure 3-26. Weekly Number of Maintenance Events 3-30
Figure 5-1. From Forecast Maintenance to Forecast Parts 5-1
Figure 5-2. Maintenance Patterns Help Planners 5-2
Figure 5-3. Using Consumption Data for Acquisition 5-3

Tables

Table 2-1. RCB Top Degrading F/A-18E/F Parts with Root Cause 2-1
Table 3-1. Time Series Format 3-9
Table 3-2. Supervised ML Format 3-9
Table 3-3. Model Evaluation Metrics for NIIN 010050515 3-12
Table 3-4. Model Evaluation Metrics for NIIN 014554498 3-12
Table 3-5. Best Models for Most Frequently Ordered NIINs 3-13
Table 3-6. Ridge Regression Model Results 3-15
Table 3-7. Overall ML Model Error 3-15
Table 3-8. Best ML Model Performance by NIIN 3-16
Table 3-9. Evaluation Metrics for Each Model for NIIN 010050515 3-17
Table 3-10. Scores for Each Model 3-18
Table 3-11. Overall Best Model Performance by NIIN 3-18
Table 3-12. Evaluation Metrics 3-19
Table 3-13. MSE and MAE for Each Trained Model 3-24
Table 3-14. MSE and MAE for Each Model 3-25
Table 3-15. MSE and MAE Averages and Random Forest Comparison 3-27
Table 3-16. Random Forest and Moving Average Comparison 3-28
Table 5-1. Demand Pro Modeling versus Traditional Forecasting 5-4

Preface

This research and development project seeks to create and test a minimum viable product using machine learning (ML) techniques on historical data from Service maintenance records to improve parts demand forecasting.

The report has two volumes:

1. Volume 1 addresses the efforts in predictive modeling using maintenance data.
2. Volume 2 discusses use of Condition Based Maintenance Plus data and methods.

In this volume, we tried to apply various ML models to evaluate past maintenance events and part demands to predict future part orders.

Chapter 1
Introduction

The Defense Logistics Agency (DLA) wants to apply machine learning (ML) techniques on data generated by the weapon systems' integrated Condition Based Maintenance Plus (CBM+) program to improve its retail customer support through better supply chain management.

This research and development (R&D) project seeks to create and test a minimum viable product (MVP) using ML techniques on historical data from Service maintenance records and a CBM+ program to improve parts demand forecasting. The customers for this effort are DLA J3 (J31 Mission Support, National Account Managers, and J34 Process Owners), J6, DLA Land and Maritime, Troop Support, and Distribution. The Service program manager (PM), weapon system, and associated maintenance depot are all key stakeholders.

Background

DLA manages over 2 million unique spare parts. These items are not all stocked, many have no demand, some are shipped directly from suppliers to DLA customers, and some are buy-on-demand. DLA uses two types of forecasting. The first employs statistical models using DLA historical demand data. The second initiates with the customer organization and is finalized through collaboration. Both are based on item supply data and are filtered across multiple levels of inventory. Forecasts that produce too little stock result in backorders and may decrease readiness. Forecasts that produce too much stock consume DLA acquisition funds and depot space, incurring the associated costs of maintaining inventory.

DLA does not exploit information-rich Service maintenance data to develop parts forecasts. The private sector has known the value of this point of consumption data for years. Traditional DLA supply forecasts, based on depot demand information, have errors in types, quantities, timing, and location of required parts.

Objective

DLA's objective is to explore the value of applying ML to Service historical maintenance records, which contain detailed information.

This R&D project seeks to answer the following question: Does the analysis of Service historical maintenance records improve parts forecasts and resulting supply support?

Chapter 2
Technical Concept and Approach

The initial concept was to apply ML to available data to improve forecasting of DLA-managed items. From this position, we would evaluate and execute predictive models to compare with current DLA statistical forecasting and develop business case analyses for these predictive models. Infrequent part orders create data sparsity over time. As the project progressed, the sparsity of data led to two determinations: the predictive models did not improve on the existing statistical forecasting, and without this improvement there was no need to develop business cases.

In concert with a DLA-established technical working group (TWG), we did develop and execute predictive models and formulated follow-on options to properly leverage these new technologies.

Component Selection

To focus the R&D project, we selected a critical component to analyze: the F/A-18E/F Generator Converter Unit (GCU). Navy Reliability Control Board (RCB) root cause analysis revealed the GCU as a top degrader (Table 2-1).

Focusing the analysis on the GCU was approved by the TWG in September. The GCU also presented a high return on investment opportunity with respect to readiness. While the analysis and results are limited to the GCU, the methodology is repeatable for any component.

Table 2-1. RCB Top Degrading F/A-18E/F Parts with Root Cause

Rank	Degrader name	Root cause
1	Outboard Leading Edge Flap Lower Fairing	Maintenance-induced damage, over-torque during removal and replacement. Compounded by demand forecast transfer from Naval Supply Systems Command to DLA.
2	GCU	G2/G3 not designed to withstand alternating current non-linear electrical loads. Erroneous removal while troubleshooting wiring discrepancies.
3	Arresting Tailhook Assembly	Nicks, gouges, and corrosion driving excessive repair turnaround time at scheduled overhaul. Compounded by inaccurate forecast model, quality evaluation at fleet readiness center, and limited number of repairable components.

Analysis Method

The functional approach included the following tasks:

1. Establish a TWG and define a DLA retail support operational use case.
2. Analyze the F/A-18 GCU.
3. Develop predictive models.

4. Compare the forecast with actual DLA forecasts.
5. Develop a business cases for predictive models.
6. Deliver a final report with recommendations.

Task 1: Establish Working Group

DLA established a TWG to guide and support the project. LMI provided input to and received guidance from the TWG to adjust project actions and execution while maintaining scope, schedule, and budget for the overall effort.

Task 2: Analyze F/A-18 GCU

2.1. Characterize the capability of CBM+ in a specific weapon system component of the subsystem as a baseline. **With and the TWG's approval, LMI focused on the F/A-18 GCU to determine the availability of historical and CBM+ maintenance data.** As the GCU is identified by the Navy as a top degrader there was in fact less sparsity than for other components. We gathered the CBM+ data from the Services and found the DLA data in the Enterprise Business System. We used these data as the baseline for all subsequent modeling efforts.

2.2. Develop a predictive model approach for CBM+. Through our partner Global Strategic Solutions, LLC (GSS), the team investigated a method to link data from selected CBM+ enabled subsystems to DLA supply data and determine the ML tools required for analysis. (See Volume 2 for all CBM+ method applications.)

2.3. Supply the forecast using ML and historical data. Use ML and LMI's Maintenance and Availability Data Warehouse™ (MADW™) to develop models for improving demand forecasting.

Task 3: Develop Predictive Models

The predictive models developed apply the techniques described in Appendix A. (Because the developed models were unable to improve the parts forecast using available data, we did not identify an MVP.)

Task 4: Compare Forecast with DLA Data

Compare current forecasting models with those developed under the project to determine whether the predictive model improves demand forecasting. (As noted in Task 3, we did not develop an MVP and thus did not complete this task.) Additionally, DLA does not have operational level forecasts, only depot level forecasts. Therefore, comparing the LMI models (which used operational level data) to DLA forecast models using depot data equates to comparing organizational level to depot level maintenance actions. These two distinctly different types of maintenance are not expected to produce similar demands.

Task 5: Develop Business Cases for Predictive Models

From the results of the previous tasks, develop a business case analysis that includes operational and implementation costs, data availability, and benefits to DLA. (Because we did not develop the models beyond the GCU, as noted in Task 3, we did not do this.)

Task 6: Deliver Final Report with Recommendations

The final report includes details on all models, procedures, and use for supply chain functional experts and data scientists.

Assumptions and Constraints

Maintenance events are assumed to require a specific, consistent set of parts. Part frequency per maintenance action is assumed to be constant with respect to time. These assumptions bound the modeling effort and address gaps in data. As additional details and data become available, these assumptions may be removed.

DLA part orders may be constrained by changes in operations tempo, aging equipment, changes in maintenance concept, or budget considerations. Orders are not classified into these categories. These constraints define the processes and procedures modeled by the predictive analysis.

Data Collection and Analysis

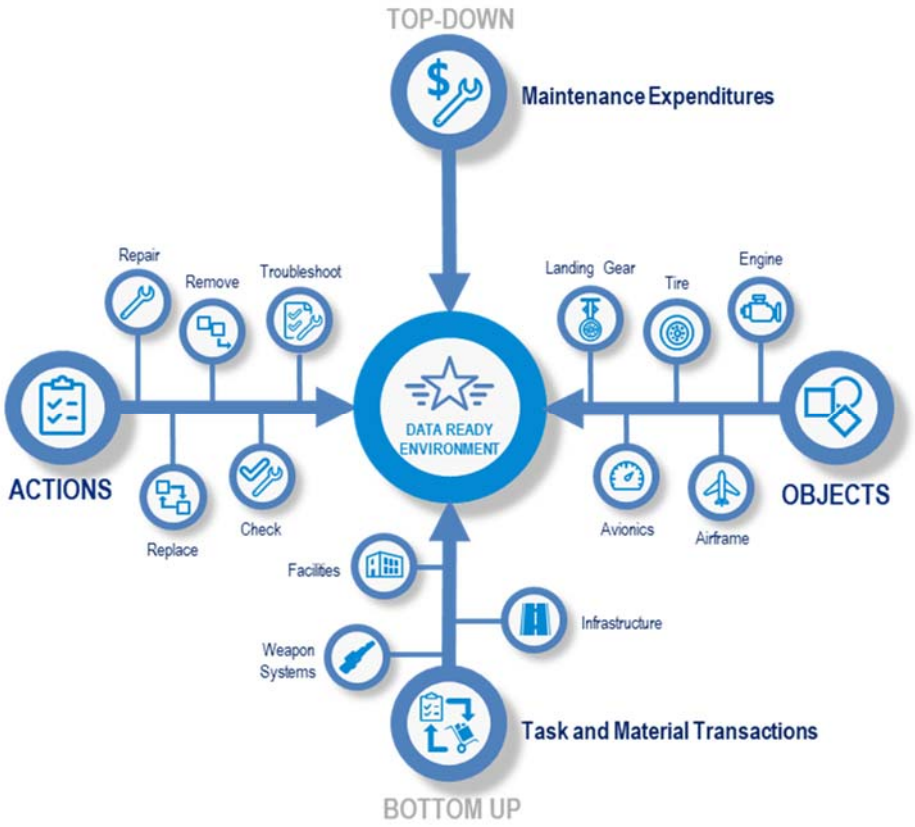
Input data was smoothed, parsed and conditioned with several techniques for use across varied ML models. Data parsing and smoothing include:

- Changed ML modeling of part demands to maintenance events to support smoothing
- Parsed data by time by establishing lags in various models
- Examined seasonality (found to not be an issue)
- Conducted change point detection isolating groups of data by demand frequency (led to the recommendation to dynamically train ML models).

MADW™

The MADW™ (Figure 2-1) is the primary source for data and analysis. This decision support tool integrates and stores maintenance and availability data for equipment, weapon systems, infrastructure, and facilities across the Services for each level of maintenance (field, intermediate, and depot), provider (organic or commercial), and nature of cost (labor and materials).

Figure 2-1. MADW™



Since 2005, LMI has been collecting Service maintenance data in the MADW™. These data include all available maintenance data curated through Natural Language Processing and other techniques. The database helps the Department of Defense (DoD) understand the cost and availability impacts on weapon system sustainment. The curation supports comprehensive analysis through ML. The MADW™ contains over 1 billion records from 42 data sources. Records in the MADW™ are cleaned, standardized, resolved, and reconciled.

In the MADW™, a set of flexible tools provides information on historical maintenance questions related to cost and availability. The MADW™ uses predictive and diagnostic modeling to answer questions related to materiel availability, with a focus on

- data materiel availability,
- efficiency,
- impact, and
- insights and analysis.

Historical Demand Data

The MADW™ contains detailed information on all supply transactions required for maintenance actions. This information is predominantly categorical, describing the National Item Identification Numbers (NIINs) ordered, the quantities, when the orders are

placed, and the weapon system for which they are ordered. With this information, the MADW™ can identify the supply transactions uniquely linked to labor transactions, offering an opportunity to understand how specific NIINs contribute to availability loss. For modeling, we used data from 2009–18, containing every order for any NIIN made for work involving a GCU. We grouped transactions by day and NIIN to get a table containing each day, an NIIN, and the number of orders for that NIIN placed on that day (backfilling days of no order with zero). The full information taken from the MADW™ demand data includes

- NIIN,
- order date, and
- total number of orders of each NIIN on each date.

Historical Maintenance Data

The MADW™ contains detailed information on all labor transactions in numerous data fields, including cost, availability impacts, and other descriptive information on maintenance actions such as the system and subsystem maintained, the maintenance performed (repair, replace, treat, etc.), when the transaction began, and the end item maintained. As noted, the data covered transactions in 2009–18. From these data, we aggregated the total number of maintenance actions performed on a given day. To add granularity to the scope of the problem, we filtered by both maintenance action and end item. All possible maintenance actions were used and filtered to one specific end item, the GCU. The full information taken from the MADW™ maintenance action data includes

- action start date,
- end item, and
- number of maintenance actions on each date.

Model Training and Evaluation

(Appendix A contains detailed model descriptions.)

Machine Learning

We applied various ML models, including ridge regression, random forest, Poisson regression, gradient boosting, elastic net, least absolute shrinkage and selection operator (LASSO) regression, and *k*-nearest neighbors (*k*-NN). These ML models evaluate past maintenance events or part demands to predict future part orders. (Appendix A details each of these models.)

We explored two approaches to parts forecasting. First, we used historical demand data to predict NIIN-level demand. The results suggested that this approach would not be significantly better than DLA's current forecasting methods. Second, we forecast the number of maintenance actions and information on the average number of parts used in a given maintenance action to predict NIIN-level demand. Both forecasting approaches started with a time series input (weekly parts consumed per NIIN for parts forecasting and weekly number of maintenance actions for maintenance forecasting) and implemented many of the same ML and deep learning methods. To evaluate performance, we compared those methods with baseline time series methods.

In both approaches, we converted time series data to a schema suitable for supervised learning, and then applied ML and deep learning methods to the data. To convert the data into a supervised ML format, we lagged the data back 40 weeks (each column represents the value of 40 previous time steps for a data point) and added categorical time features to the data. This schema enabled us to encapsulate all the time series aspects of the data into a format suitable for supervised ML. By lagging the data by 40 weeks, autocorrelation and seasonality were considered in the time series models and increased the number of features for ML. The value is predicted 8 months into the future from the current values; thus, the response is the value 32 weeks into the future.

With the data in a suitable format, we trained both parametric models and non-parametric models to evaluate which type performs best for this data. Parametric models fit an equation to the data and are unlikely to fit to noise. Non-parametric methods do not fit an equation to the data so they can describe more complex patterns in data. For all models in both approaches, hyperparameters were tuned to optimize model performance.

ML models for maintenance action forecasting included the following:

- Parametric Methods:
  - Ridge regression
  - LASSO regression
  - Elastic net
- Non-parametric Methods:
  - Random forest regression
  - Gradient boosted regression
  - *k*-NN
  - Kernel ridge regression
  - One dimensional convolutional neural network
  - Recurrent neural network.

ML models for parts forecasting included the following:

- Parametric Methods:
  - Ridge regression
  - Poisson regression
  - LASSO regression
  - Elastic net regression
- Non-parametric Methods:
  - Random forest regression
  - Gradient boosted regression
  - *k*-NN.

Chapter 3
Analysis and Results

The analysis consists of two forecasting approaches: parts demand forecasting and maintenance forecasting. These approaches involve a similar data setup and ML approach but diverge in their analysis and results. Each section in this chapter details an approach and consists of the approach's data setup, preliminary data analysis, modeling method, modeling result analysis, and conclusion.

Data sparsity limited the success of ML across a wide range of techniques. As a result of this research focused recommendations for follow-on R&D are identified to explore the successful elements of this investigation.

Parts Demand Forecasting

The first approach to parts forecasting is to use historical demand quantity to predict future demand quantity. This approach assumes a relationship between past and future quantity demanded.

We used time series methods to establish baseline results. We then applied ML models to predicting parts demand. (Appendix B details the modeling method.)

Exploratory data analysis on the parts demand data included NIIN clusters analysis using unsupervised ML methods, statistical analysis on demand increases at the end of a fiscal quarter and end of a fiscal year, and classification to predict whether there will be an order on a given day. Figure 3-1 shows the modeling process.

Data

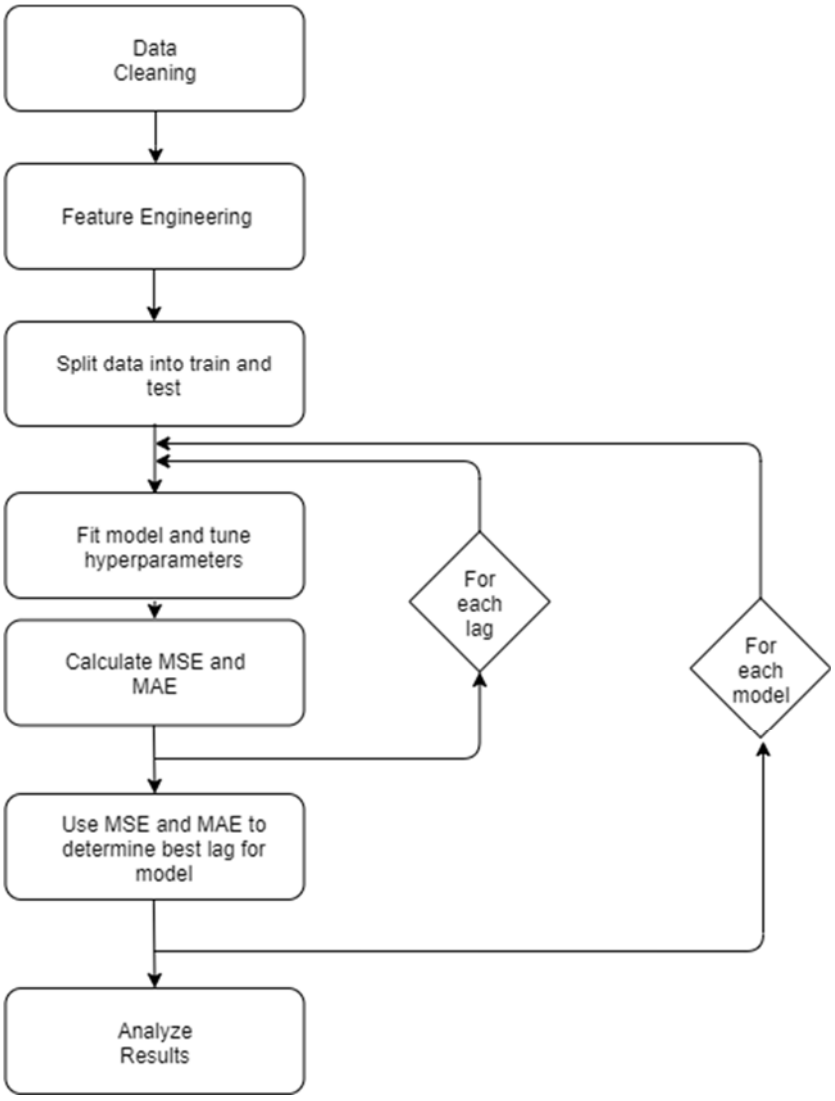
Source

We used MADW™ data sourced from the Aviation Financial Analysis Tool (AFAST) to model demand. Each row in the data corresponds to an order of a given NIIN at a specific time and location. The AFAST-sourced MADW™ data provide information on operational-level demand for NIINs.

Cleaning

We applied several filters to the MADW™ data. First, we filtered the data by *TMS* (Type/Model/Series) and *ServiceWBS* (Service work breakdown structure, an alternative to work unit code [WUC]) so the data contained only demands for GCU parts on F/A-18s. After this filtering, the data consisted of demand records for 1,419 NIINs. The majority of NIINs are ordered on less than 2 days in the data set. Since neither time series methods nor ML methods perform well on such sparse data, we further filtered the data to include only NIINs ordered on more than 2 days. The resulting data set consisted of 487 NIINs. Finally, we reduced the data to contain only NIINs DLA currently forecasts, resulting in the final data set with demand records for 330 NIINs.

Figure 3-1. Modeling Process



Note: MAE = mean absolute error; MSE = mean squared error.

We then aggregated the data into daily quantities by NIIN. After filtering and aggregating, the data set consisted of NIIN, nonzero demand quantity, and date of corresponding demand. The data are subsetted to consist of dates from January 2009 through December 2018 to account for data collection inconsistencies before 2009 and in 2019. The data consisted of every instance of a nonzero demand quantity, so we added missing dates and a corresponding zero for demand quantity for each NIIN.

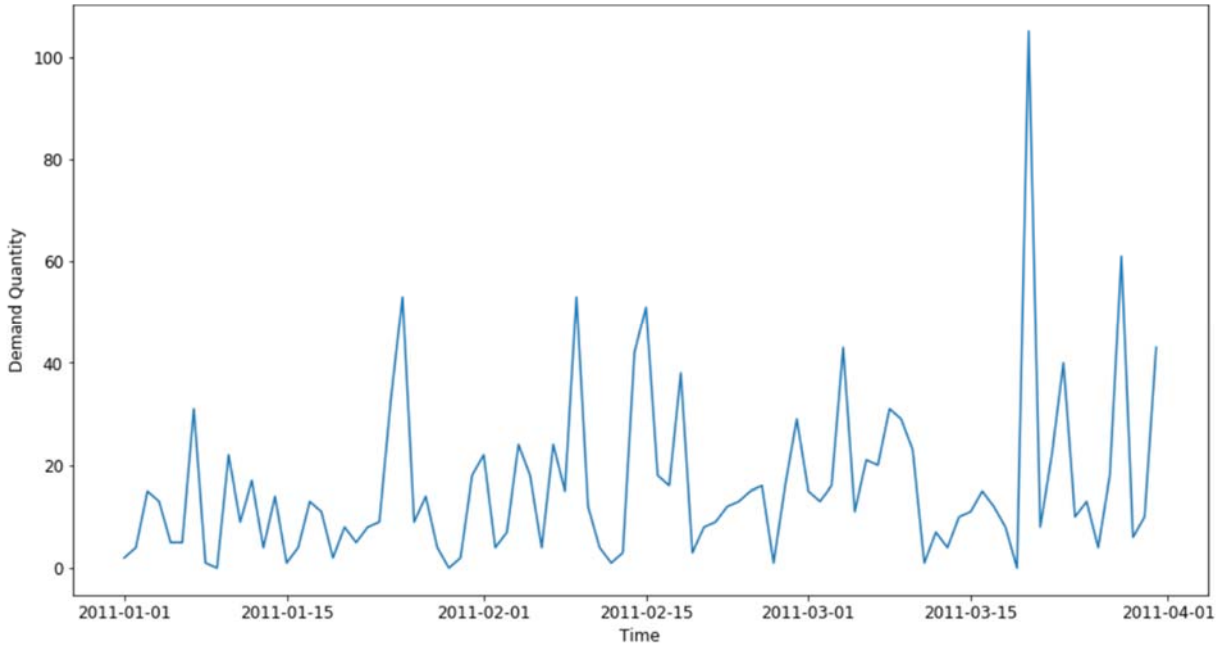
Exploratory Data Analysis

End-of-Quarter Spikes

At the end of a fiscal quarter or fiscal year, planners may order more parts to use underspent funds. This phenomenon may appear as spikes in demand quantity at the end of fiscal quarters and at the end of the fiscal year. To evaluate end-of-fiscal-quarter spikes, we split the data by fiscal quarter and used *t*-tests comparing the difference in

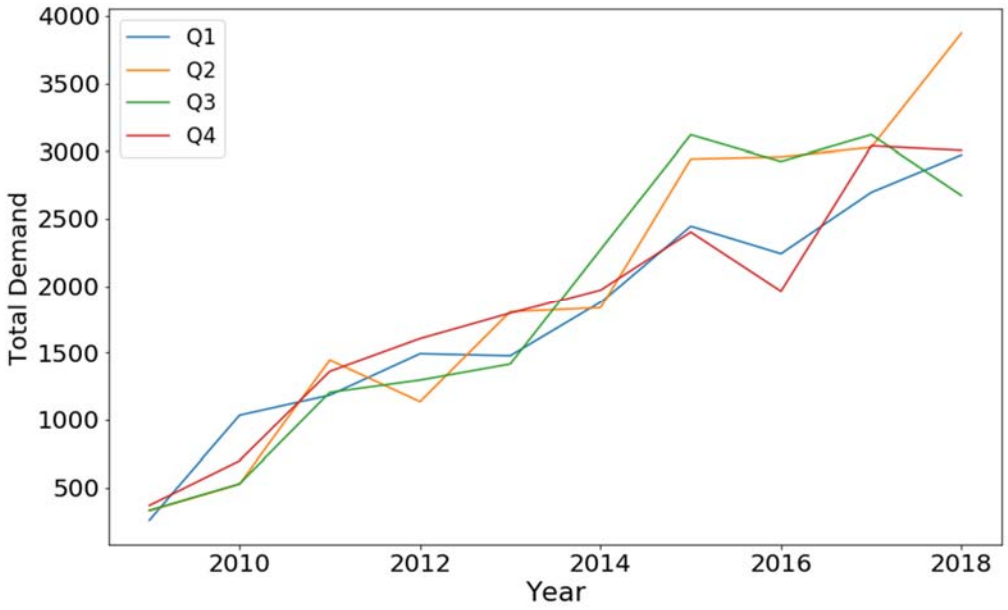
means over the last month of the quarter to the rest of the quarter. The *t*-tests showed that a few quarters had statistically significant increases in order quantity at the end of the quarter at the 0.05 significance level. Of all quarters in 2009–18, only the fourth quarter of 2010, the second quarter of 2011, and the fourth quarter of 2015 showed a significant increase in parts demanded at the end of the quarter. Figure 3-2 shows the plot of the second quarter of 2011. The plot shows a clear increase in demand at the end the quarter. Since only 3 of 40 quarters showed significant increases at the end of the quarter, we concluded planners are not demanding more parts at the end of the quarter.

Figure 3-2. Second Quarter 2011 Parts Demand



Likewise, planners may order more parts at the end of a fiscal year to exhaust funds. Behavior like this would result in higher demand quantity in the fourth quarter than in other quarters. To evaluate whether planners ordered significantly more in the fourth quarter, we applied a one-way analysis of variance (ANOVA) test to see whether all four quarters have the same mean. The test achieved a high *p*-value at the 0.05 significance level. All four quarters have statistically similar means throughout the data; thus, planners are not ordering significantly more at the end of the fiscal year. Figure 3-3 shows the total amount of demand quantity by quarter over a 10-year period. The plot confirms the ANOVA test results, as no quarter is consistently higher than others. The plot also shows that demand consistently increases over time for all four quarters.

Figure 3-3. Demand by Quarter



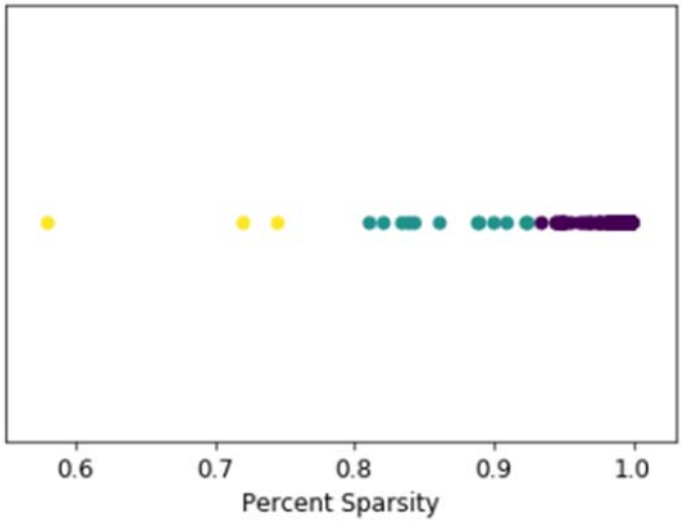
The results of the *t*-test and the ANOVA test suggest that, overall, there is little evidence of an increase in demand at the end of fiscal quarters and fiscal years. Transformation of data or predictions is not required to account for spikes in the data from planners exhausting underspent funds.

NIIN Clustering

We explored clustering NIINs according to sparsity to break down the data set and train on groups of NIINs with similar sparsity separately. Ideally, if the data set is clustered well, training and testing on NIINs by cluster would improve model accuracy because sparse NIINs will not cause the model to under-predict for frequently ordered NIINs.

Our clustering methods first trained on one-dimensional data that consisted of the NIIN and the number of days that NIIN had no orders. We clustered the data with *k*-means clustering and density-based spatial clustering of applications with noise (DBSCAN). The *k*-means clustering identified separations between data points that made some sense visually, but DBSCAN was unable to identify anything meaningful. Results from clustering the one-dimensional data alone did not provide confident clusters. Figure 3-4 shows the clusters assigned by the *k*-means clustering algorithm using three clusters. Each point represents the percentage of days a specific NIIN is not ordered over the entire period. Each color represents a different cluster.

Figure 3-4. k-Means Clustering Using Three Clusters

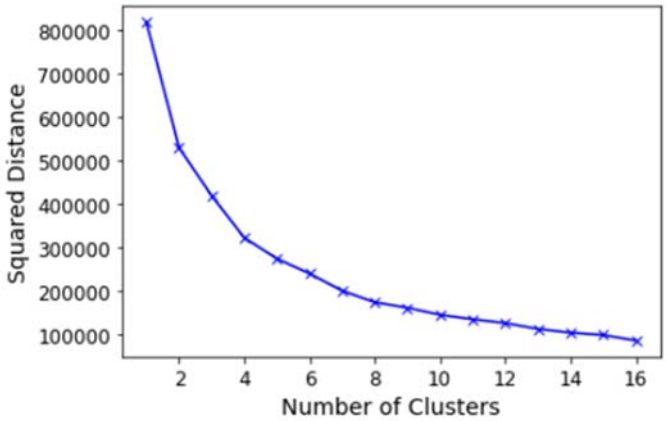


We next applied clustering by the entire weekly time series to each NIIN. This data format consists of each individual NIIN's weekly time series data as rows in a multidimensional data frame. Next, we compared *k*-means, DBSCAN, Gaussian mixture models (GMMs), and hierarchical clustering.

We selected the number of clusters for these algorithms using plots of the relationship between the sum of squared Euclidean distance between each point and the center of the nearest cluster and the number of clusters. Figure 3-5 shows the plot of the sum of squared distance as the *k*-means clustering algorithm generates increasing numbers of clusters. The plot shows an elbow that falls around four clusters. This elbow means that when the algorithm generates more than four clusters, they are unlikely to add significant value to the model. Other clustering algorithms also indicated four clusters may be best.

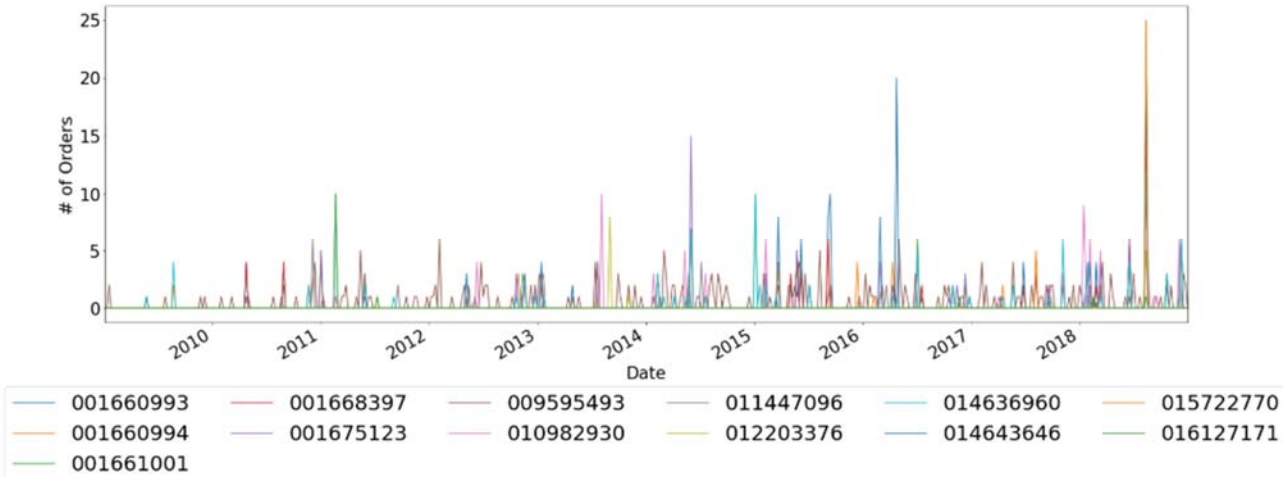
All clustering algorithms clustered the data similarly except DBSCAN, which labeled nearly all data as noise. We identified no similarities among the time series data of the clustered NIINs.

Figure 3-5. Squared Distance for Different Cluster Amounts



All the clustering methods except DBSCAN use distance-based metrics. We next applied cosine similarity to improve clustering results. Figure 3-6 shows results from one cluster generated from DBSCAN using cosine similarity. Each line represents the weekly demand for a single NIIN over the entire period 2009–18. Clusters where different NIINs have similar order dates (regardless of number of orders) are considered successful clusters. Using DBSCAN with cosine similarity, we grouped the time series into six clusters with similar time series data. Despite these groupings, most of the NIINs are classified as noise. Because of the inconsistencies between different clustering techniques, clustering is not applied in further modeling.

Figure 3-6. Clustered NIINs over Time



Evaluation Measures

We used the following measures to evaluate and compare models:

- MSE, which measures the squared difference between the actual and predicted values.
- MAE, which measures the absolute difference between actual and predicted values.
- Period in stock (PIS), which measures how accurately and how consistently a model predicts.
- Mean, the average of the test set prediction.
- Standard deviation (SD), a descriptive statistic that measures how far spread out predictions are from their mean. Calculated for test set predictions.
- Standard error (SE), a summary statistic that estimates how close the sample mean error is likely to be from the true population mean error. Calculated for test set predictions.

Time Series Modeling

Before evaluating ML models for parts forecasting, we used time series models to explore simple, traditional forecasting methods and establish baseline results. Including

time series modeling also helped to exhaustively evaluate forecasting methods on the data.

Time series methods model data as a function of time. They are effective only when previous values in time can predict future values in time. Time series models take in two variables that have a one-to-one relationship: time and corresponding value. To forecast demand, the time series models take in the time and the corresponding total quantity ordered for a NIIN.

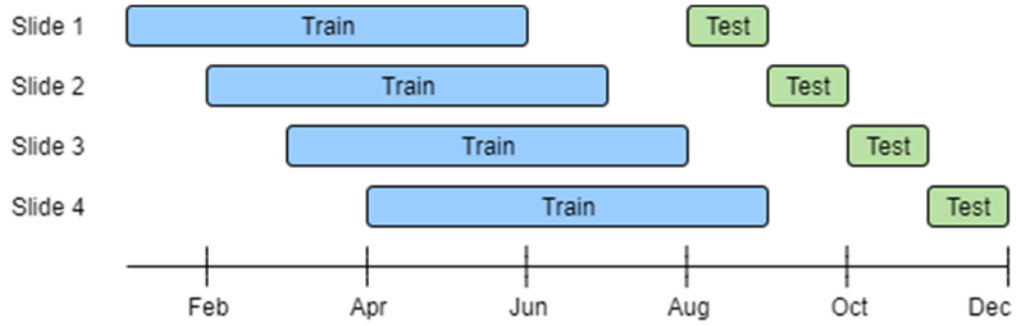
Data Setup

In preparation for time series modeling, we evaluated the data for stationarity to ensure they are appropriate for the models. Stationary data have consistent statistical properties over time. The data have consistent variance and covariance throughout time. The data may have changing mean, but it must change in the same manner with time. Data need to be stationary for some time series modeling to be effective. The augmented Dickey-Fuller (ADF) test determines whether the data are stationary. ADF is a statistical hypothesis test with the null hypothesis that the data are non-stationary. It defines critical values on the basis of change in a stream of data over a set number of time intervals. A *p*-value represents the significance of the test findings. A *p*-value greater than 0.05 indicates that the null hypothesis was rejected. Sufficient evidence exists to establish that the data are stationary. Every NIIN tested using the ADF test returns a *p*-value greater than 0.05 when looking at daily-level data. When looking at weekly data, nearly every NIIN tested using the ADF test returns a *p*-value greater than 0.05 (except a few, usually highly sparse NIINs, that do indicate non-stationarity). Because the majority of NIINs proved stationary under both aggregation levels, we assume the data are stationary. Thus, transformation and differencing of data are not required before time series modeling.

An 8-month lead is incorporated into the time series models by setting a gap between the end of training and the start of testing. We used a sliding window approach to make predictions, establishing start and end dates for the training and testing sets. We then slid the model predictions forward by the length of the testing set and repeated the process through the entire testing data set. This process enables creation of disjoint sets of test predictions for each slide that, when combined, completely cover the entire testing set. The testing set consisted of 28 days to aggregate the predictions into roughly a 1-month prediction for each slide. We used this interval to make 28-day time series predictions and weekly ML predictions, aggregated by 4 weeks into comparable, monthly predictions.

Figure 3-7 illustrates an example of a sliding window. In this example, the training window is 5 months long, the gap between train and test for lead time is 2 months long, the testing window is 1 month long, and the entire test set is 4 months long. The example shows the four slides required to make enough disjoint testing window predictions to cover the testing set and shows that, while the testing windows are disjoint, the training windows may overlap.

Figure 3-7. Example Sliding Window



We repeated the sliding window approach on many training window sizes to test their effect on accuracy. Training start dates vary while the initial training end date, initial testing start date, testing window length, and gap length stay constant. This permits the training window length to vary while maintaining the gap between training and testing and the number of predictions each slide makes. All time series models evaluated training windows including monthly increments from 30 days to 1 year and yearly increments from 1 year to 7 years, except for exponential smoothing, which did not evaluate training windows of 1 month through 3 months because of modeling restrictions.

Method

For each NIIN, window, and time series model, we did the following:

1. Filtered data by a given NIIN because the time series methods can only model one variable at a time.
2. Split data into initial training and testing. Set the start of testing to April 14, 2017, the end of training to September 2, 2016, and the start of training according to the number of days in the given window length. We slid these dates according to the sliding window methods explained above.
3. For the given NIIN, model, and window length, fit the model to the training window and tune hyperparameters over the testing set by grid searching every possible parameter combination and choosing the model that provides the prediction with the lowest possible MAE for the monthly aggregated test set.
4. Calculated MSE and MAE for given NIIN, window, and model.

Model Development

For an exhaustive model evaluation, we used moving average, simple smoothing, exponential smoothing, Fourier, and Prophet to model and make predictions on the data. (Appendix A details these models.) These time series methods can only predict one variable, so each model trained and tested NIINs individually. As a result, these time series methods cannot relate demand patterns between NIINs. Some time series methods can relate multiple variables in one model, but we did not explore them due to weak support for these algorithms in Python.

Models are trained on data describing NIINs with less than 90 percent sparsity. NIINs with nonzero orders on at least 10 percent of days. The testing window for results is a 22-month period, ranging from April 13, 2017, to November 22, 2018.

ML Modeling

Unlike time series methods, ML methods can model complex relationships and incorporate additional information. More complex modeling methods like many ML methods tend to produce results with lower bias but higher variance, which may be useful given the sparsity inherent in the data set. Many ML methods can also find relationships between variables and use those relationships in prediction. Likewise, they may be able to find relationships between NIINs and use those relationships to increase forecast accuracy. The role of these relationships in ML modeling may be enough to overcome the sparsity challenges.

Feature Engineering

We converted the data into a supervised ML format by lagging the data. Lagging a variable involves recursively adding a column for the variable at the previous time step to the original data frame for the number of lags specified. Tables 3-1 and 3-2 show an example transformation of time series format to a supervised ML format lagged back 2 days. The first table shows the data in a time series format with two columns: date and corresponding quantity. The second shows the original data frame with two additional columns representing each lag from lagging quantity back 2 days. The first two rows of the supervised ML data setup will be deleted before modeling because they are incomplete. When lagging back *N* time steps, the first *N* rows in the data will always be deleted for being incomplete.

Table 3-1. Time Series Format

Date	Quantity
01-01-2012	5
01-02-2012	3
01-03-2012	8
01-04-2012	7
01-05-2013	9

Table 3-2. Supervised ML Format

Date	Quantity (*t*–2)	Quantity (*t*–1)	Quantity (*t*)
01-01-2012	N/A	N/A	5
01-02-2012	N/A	5	3
01-03-2012	5	3	8
01-04-2012	3	8	7
01-05-2012	8	7	9

Lagging is a traditional method for converting time series data to supervised ML data. When the data are lagged with the correct number of lags, a row of the data will include

all the row's time series aspects, namely seasonality and autocorrelation. Seasonality is the cyclical pattern in time series data that repeats throughout the data set. Autocorrelation is the amount with which a time point is affected by all previous points. Similarly, partial autocorrelation evaluates the relationship between observations at two different time periods without including the effect of the observations at intermediate time periods.

Figures 3-8 and 3-9 are the plots for autocorrelation and partial autocorrelation for NIIN 016432402. This NIIN is representative of the majority of the NIINs in the data set. The plots consist of vertical lines for each lagged value up to 60 weeks prior. The height of each line in the autocorrelation (Figure 3-8) and partial autocorrelation (Figure 3-9) signifies the correlation coefficient between the given lagged value and present value.

Figure 3-8. Autocorrelation

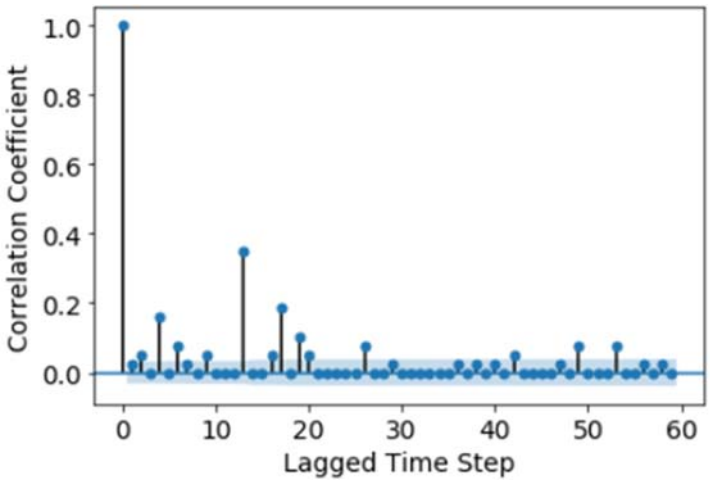
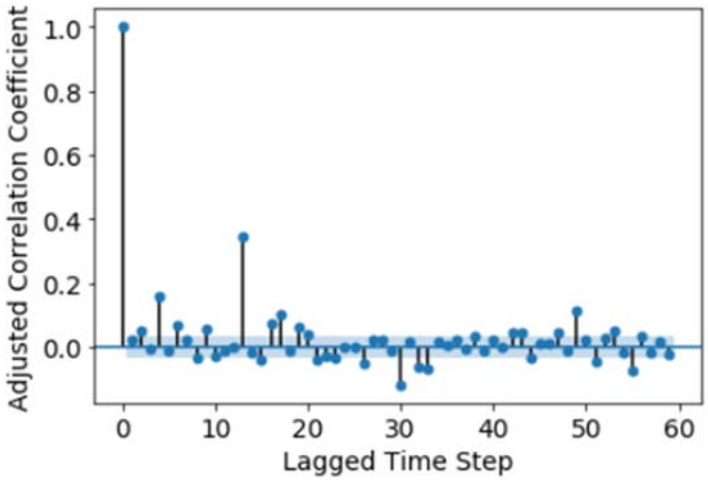


Figure 3-9. Partial Autocorrelation



The partial autocorrelation plot removes the correlation caused by all intermediary time intervals and the autocorrelation plot does not. The greater the correlation coefficient, the more related the specific lagged value and present value are. The shaded region represents the boundary for statistically significant correlation coefficients. Lagged

values with lines that extend outside of this shaded region are significantly related to the present value. Because the furthest lag that extends out of the shaded region in both plots is around 55 weeks, the plots show significant lags up to 55 weeks. Likewise, the majority of NIINs show similar plots with significant lags up to 55 weeks prior.

The autocorrelation plot also shows no seasonality for the majority of the NIINs as there is no cyclical behavior in the autocorrelations. Thus, lagging far enough back to include all the autocorrelations is sufficient.

In addition to the 55 lags, we applied one hot encoding to the month to predict ahead 32 weeks. Predicting ahead 32 weeks is equivalent to predicting ahead 8 months, a long enough time to address DLA's lead times.

Method

ML models often perform best with as few variables as possible that still represent the data, so we explored all possible lag values to find the best lag for each model. With each iteration of the method below, we removed the latest lag. For each model, 55 lags are evaluated.

For each lag and model, we did the following:

1. Split data into initial training and testing by setting the testing start date to April 14, 2017.
2. For the given model and number of lags, fit the model and tuned hyperparameters with randomized grid search. (Appendix B shows the sampled hyperparameter sets, and the final, best hyperparameter sets for each model.)
3. Calculated MSE and MAE for given model and number of lags.

Model Development

We tested the models on monthly aggregations from April 2017 through November 2018 to directly compare ML model predictions and evaluation metrics with those from the time series models.

We predicted demand quantity 8 months into the future and compared results for the following ML models:

- Ridge regression
- Random forest regression
- Poisson regression
- Gradient boosted regression
- LASSO regression
- *k*-NN
- Elastic net regression.

(Appendix A details each of these models.)

Results

Time Series Results

Tables 3-3 and 3-4 show the values for the three measures for each of the time series models for NIIN 010050515, the most frequently ordered NIIN in the data, and NIIN 014554498, an infrequently ordered NIIN that is representative of the data. Both list the best models for their respective NIIN in order, with better performing models listed before worse performing ones. The models are ranked according to each model's MSE and MAE.

Table 3-3. Model Evaluation Metrics for NIIN 010050515

Model	MSE	MAE	PIS	Mean	SD	SE
Exponential smoothing	210.57	11.36	−2,222.04	45.28	8.06	1.76
Simple smoothing	208.21	11.64	−2,558.60	48.73	8.46	1.85
Prophet	242.50	11.48	−1,773.83	49.40	17.61	3.84
Fourier	231.24	12.54	−2,413.11	43.67	5.80	1.27
Moving average	239.60	13.04	−2,772.29	42.36	7.65	1.67

Table 3-4. Model Evaluation Metrics for NIIN 014554498

Model	MSE	MAE	PIS	Mean	SD	SE
Moving average	1.01	0.60	−155.05	0.10	0.05	0.01
Exponential smoothing	1.09	0.55	−176.00	0.00	0.00	0.00
Simple smoothing	1.02	0.74	−132.17	0.29	0.25	0.06
Fourier	1.08	0.69	−163.50	0.16	0.24	0.05
Prophet	1.35	0.95	−102.86	0.55	0.57	0.12

For both NIINs, all PIS values are negative and large in magnitude. Such values indicate that all models for both NIINs consistently under-predict the true values. Because all models under-predict for frequently ordered NIINs and for infrequently ordered NIINs, the models overall have high statistical bias toward under-prediction and are thus not performing well. Exponential smoothing performed among the top two models for both NIINs, but otherwise different models perform best for different NIINs.

The evaluation metrics for each NIIN are not directly comparable, as they are not standardized and can be misleading due to data sparsity. The metrics are low for NIIN 014554498 because it is ordered infrequently, so predicting zero values for all instances can yield good model evaluation metrics but is a useless model. This result is the norm for the data; the majority of the NIINs in the data behave similarly to NIIN 014554498 and achieve similar, under-predicted results.

Figures 3-10 and 3-11 show the plots for monthly predicted quantity and monthly actual quantity for NIINs 010050515 and 014554498 over the test set. The plots show the different order frequencies for the two NIINs and that, generally, the models under-predict the actual values.

Figure 3-10. Comparison of Time Series Models for NIIN 010050515

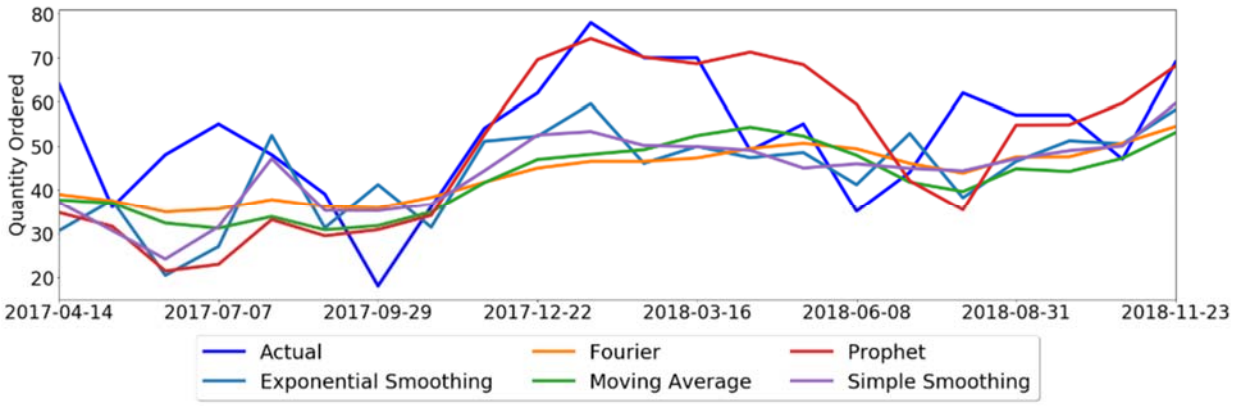


Figure 3-11. Comparison of Time Series Models for NIIN 014554498

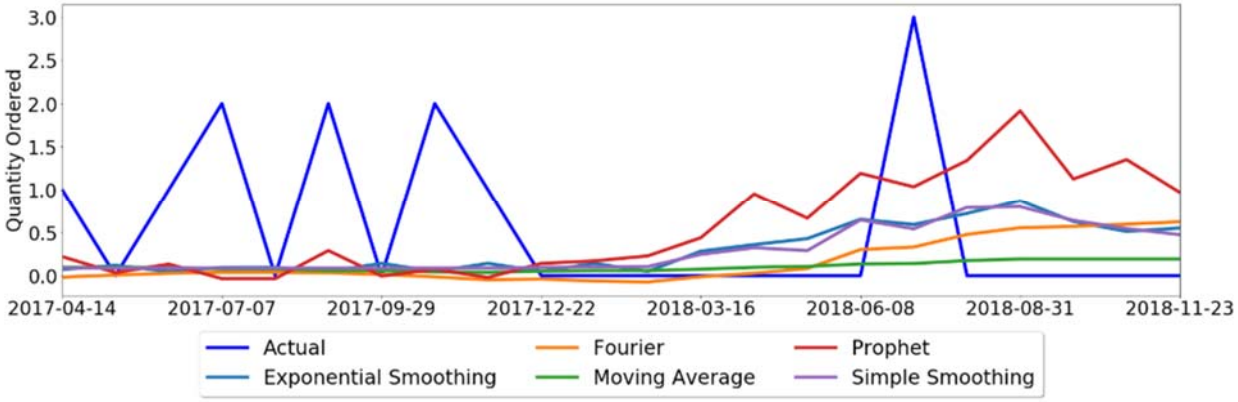


Table 3-5 shows the best model, according to MSE and MAE scores, for each of the 11 most frequently ordered NIINs. These results indicate that moving average and exponential smoothing are best for the majority of frequently ordered NIINs.

Table 3-5. Best Models for Most Frequently Ordered NIINs

Model	NIIN	MSE	MAE	PIS	Mean	SD	SE
Simple smoothing	001651942	57.43	5.89	−96.65	13.63	1.03	0.22
Moving average	014554498	1.01	0.60	−155.05	0.10	0.05	0.01
Moving average	011192008	283.79	10.79	−539.81	12.18	1.36	0.30
Exponential smoothing	014938822	4.68	1.65	8.00	4.14	1.14	0.25
Moving average	014793739	8.55	2.11	−36.56	5.15	0.24	0.05
Fourier	000546940	155.26	9.39	−1,649.11	26.64	6.92	1.51
Exponential smoothing	009507783	8.76	2.31	−509.88	8.94	1.29	0.28
Moving average	014793776	488.76	17.51	−2,338.49	42.22	3.27	0.71
Fourier	012223502	673.52	22.30	−2,210.28	97.74	19.96	4.36
Fourier	014793835	58.85	5.67	−1,031.31	17.82	2.31	0.50
Exponential smoothing	010050515	210.57	11.36	−2,222.04	45.28	8.06	1.76

Table 3-5. Best Models for Most Frequently Ordered NIINs

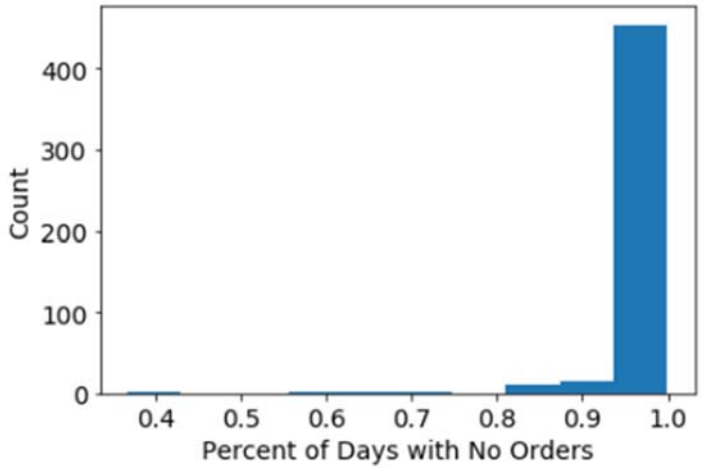
Model	NIIN	MSE	MAE	PIS	Mean	SD	SE
Simple smoothing	001651942	57.43	5.89	−96.65	13.63	1.03	0.22
Moving average	014554498	1.01	0.60	−155.05	0.10	0.05	0.01
Moving average	014793815	5.93	1.92	−206.65	4.37	0.12	0.03

Time series methods are generally ineffective at predicting far into the future and tend to perform better with more data. The accuracy of the models used here suffered from incorporating an 8-month gap between training and testing for lead times and from the limited amount of data.

ML Results

While briefly exploring results using all DLA-forecasted NIINs present in the MADW™, examining the most frequently ordered NIINs is more productive due to data sparsity. The majority of NIINs in the final, filtered data set are still ordered infrequently. The histogram in Figure 3-12 shows the frequency for different levels of sparsity. The vast majority of NIINs have no orders on at least 95 percent of the days in the data, demonstrating the data sparsity.

Figure 3-12. Count of NIINs by Percent of Days with No Orders



With such sparsity, modeling methods often suffer. Initially, all NIINs are modeled regardless of sparsity, but as expected, these modeling efforts produce poor results by consistent under-predicting, as the extremely sparse NIINs affected the predictions for the less sparse NIINs. Table 3-6 shows results from a ridge regression model with MSE, MAE, average PIS scores per NIIN, and general statistics for groups of NIINs with different levels of sparsity.

Table 3-6. Ridge Regression Model Results

% days with no orders	No. of values	MSE	MAE	PIS	Mean	SD	SE	Avg no. of orders
>99	271	3.80	0.31	−267.48	0.04	0.49	0.01	26.188192
> 95 and ≤ 99	36	82.24	3.19	−1,666.81	1.90	7.30	0.26	214.138889
> 90 and ≤ 99	12	38.22	4.13	−1,294.17	4.00	5.45	0.34	564.083333
< 90	11	304.39	11.08	−7,800.82	21.92	23.83	1.54	2,514.818182

Table 3-5 shows that as sparsity increases, MSE and MAE improve. However, these metrics are misleading. For the three sparsest groups, the models tend to predict all zero values since there are few, low-valued orders seen in the mean and average number of orders columns in Table 3-5. While models that predict all zero values can achieve better MSE and MAE metrics for these sparse groups, they are useless for parts forecasting. Conversely, the models produced variable predictions for NIINs in the least sparse group. While these models achieved worse MSE and MAE values, they produced useful predictions. Thus, we only included the 11 NIINs in the group with less than 90 percent sparsity for modeling.

The sparsity constraint is addressed by filtering out all NIINs ordered on less than 10 percent of the data and aggregating the data by week for the ML approach. While the initial results indicated that these groups are also under-predicted, isolating only the most frequently ordered parts may alleviate this issue. The resulting data set contained 11 NIINs. We tested all models in the 22-month period from April 2017 through November 2018. We evaluated overall performance with MSE and MAE. PIS measures sequential consistency and can only be evaluated at the NIIN level.

For each model, we compared the predictions resulting from the model training iteration with the lowest MSE and MAE for each NIIN. To directly compare results among all models, we aggregated predictions into approximately monthly predictions. For the time series results, we aggregate the daily predictions into disjoint 28-day predictions, and for the ML results, we aggregate the weekly predictions into disjoint 4-week predictions. With these aggregations over the same testing set, we could directly compare all model results for each NIIN with plots and tables describing monthly MSE, MAE, and PIS.

Ridge regression performed best, achieving the lowest overall MSE and MAE scores on the filtered data set consisting of the 11 NIINs. However, raw evaluation metrics defined our ranking system and statistical tests never evaluated these rankings. Future work would include statistical testing to evaluate whether models statistically performed better than each other. Table 3-7 shows the overall evaluation metric for each model.

Table 3-7. Overall ML Model Error

Model	MSE	MAE	Mean	SD	SE
Ridge	322.16	10.67	24.06	24.14	1.53
Random forest	393.89	11.24	23.46	19.80	1.25
Elastic net	351.16	11.29	21.70	23.85	1.51
LASSO	477.40	13.28	17.63	19.90	1.26

Table 3-7. Overall ML Model Error

Model	MSE	MAE	Mean	SD	SE
Ridge	322.16	10.67	24.06	24.14	1.53
Gradient boosting	583.74	12.96	20.10	17.44	1.11
k-NN	534.20	13.42	17.64	18.49	1.17
Poisson	836.82	16.95	18.57	17.07	1.08

While ridge regression performed best overall, different models performed best for different individual NIINs. Table 3-8 shows the performance for the 11 most frequently ordered NIINs with the best-performing model according to MSE and MAE, corresponding MSE, MAE, and PIS scores. The testing set sparsity is included for each NIIN to evaluate whether there is a relationship between testing sparsity and prediction accuracy. No discernible pattern is present, but there is a consistent under-prediction for all but the sparsest NIINs. The table shows the relationships between error metrics and sparsity and that random forest and ridge regression perform best for the most individual NIINs.

Table 3-8. Best ML Model Performance by NIIN

NIIN	Model	MSE	MAE	PIS	Mean	SD	SE	% zeros
014938822	Random forest	4.95	1.77	22	4.14	0.34	0.07	42.86
001651942	Poisson regression	69.82	6.27	−319	12.73	3.28	0.72	27.47
011192008	Ridge regression	309.77	12.23	53	13.55	4.51	0.98	48.35
009507783	Random forest	14.59	3.05	−598	8.59	1.47	0.32	17.58
014793739	Ridge regression	11.95	2.68	−53	4.95	2.12	0.46	23.08
014793835	Random forest	65.45	6.36	−905	19.59	4.01	0.87	8.79
012223502	Ridge regression	1,603.64	35.00	−6,945	82.68	16.57	3.62	2.20
014793776	Gradient boosting	482.82	16.45	−2,189	41.32	11.71	2.56	18.68
010050515	Random forest	281.73	13.73	−3,012	41.14	6.46	1.41	1.10
014793815	Gradient boosting	4.18	1.55	−114	5.27	1.29	0.28	31.87
000546940	Random forest	182.82	9.91	−1,562	26.73	4.00	0.87	1.10

Because nearly all PIS scores in Table 3-8 are negative and large in magnitude, even the best-performing models tend to under-predict. The final filtered data set consisted of 11 NIINs. With so few NIINs, it is computationally feasible to train, tune, and test each individually. However, it would not be feasible to train, tune, and test individual models for each NIIN in the larger set of DLA-forecasted items.

Although the error metrics for some of the NIINs suggest the models were adequately forecasted, only a few NIINs achieved meaningful results. Figure 3-13 shows the results of the model trained on the 11 most frequently ordered NIINs in the full data set for one of those NIINs, 010050515. The plot shows that all models under-predict and that even the best-performing models do not quite mimic the true behavior of the data.

Figure 3-13. Comparison of ML Models for NIIN 010050515

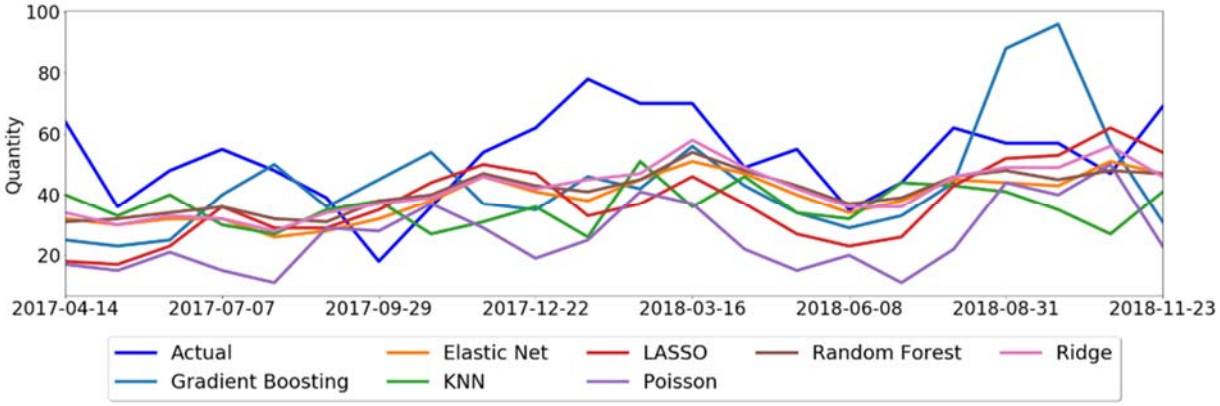


Table 3-9 shows the evaluation metrics for each model for NIIN 010050515 corresponding to the above results. It shows that all models consistently under-predict for this NIIN as all PIS values are negative and large in magnitude.

Table 3-9. Evaluation Metrics for Each Model for NIIN 010050515

Model	MSE	MAE	PIS	Mean	SD	SE
Random forest	281.73	13.73	−3,012	41.14	6.46	1.41
Ridge	269.95	13.77	−3,046	41.55	8.07	1.76
Elastic net	325.50	15.14	−3,620	39.09	7.32	1.60
k-NN	471.91	18.18	−3,829	36.05	6.60	1.44
LASSO	475.27	18.73	−4,422	37.32	12.34	2.69
Gradient boosting	523.45	19.91	−2,807	44.05	17.95	3.92
Poisson	981.27	27.73	−6,839	25.95	10.94	2.39

The same results can be seen below for the NIIN with the overall lowest MSE and MAE scores, 014938822. Figure 3-14 displays the results of all ML models on the prediction set, and Table 3-10 shows the scores for each model.

Figure 3-14. Comparison of ML Models for NIIN 014938822

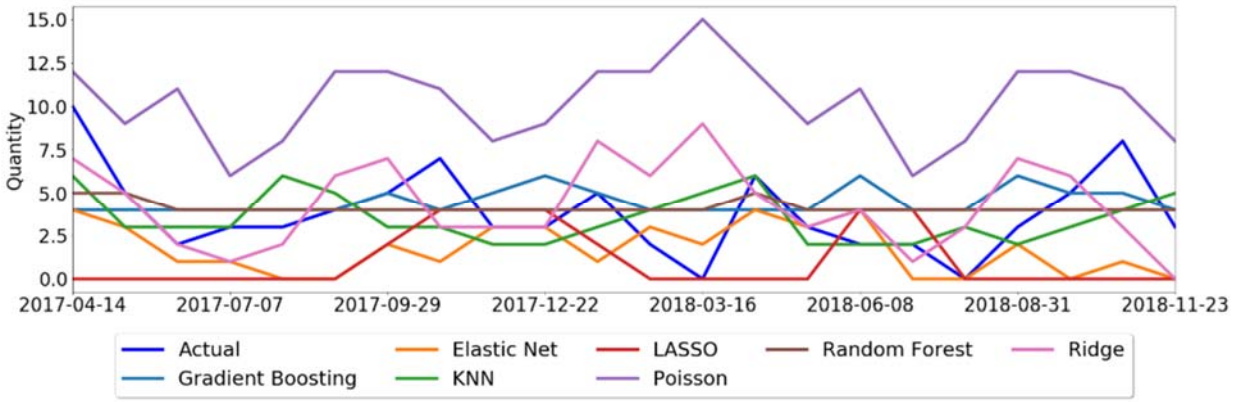


Table 3-10. Scores for Each Model

Model	MSE	MAE	PIS	Mean	SD	SE
Random forest	4.95	1.77	22	4.14	0.34	0.07
k-NN	5.50	1.86	−119	3.50	1.37	0.30
Boost	6.59	2.05	65	4.50	0.72	0.16
Ridge	9.55	2.27	74	4.27	2.42	0.53
Elastic net	10.55	2.55	−561	1.73	1.42	0.31
LASSO	16.18	3.27	−766	1.09	1.68	0.37
Poisson	49.82	6.45	1,552	10.27	2.24	0.49

While the MSE and MAE scores for these predictions are very close to zero, which can indicate that the model fit closely to the actual values, the graph indicates that the model is not fitting properly to the data. The best-performing model according to the evaluation metrics is random forest, but random forest predicted a flat line and did not come close to fitting the actual data. This result runs counter to the results for NIIN 010050515, which show comparatively high MSE and MAE scores, but has a plot indicating a somewhat close fit. The difference in evaluation metrics and prediction plots emphasizes that MSE and MAE can be misleading. PIS scores also seem to vary greatly depending on the model. Ranging from severe over-predictions for Poisson to severe under-predictions for some others.

Accurate prediction using ML requires little sparsity and relevant features in the data. The available data did not meet either requirement. The data are naturally sparse due to infrequent demand. Modeling all NIINs together introduces more sparsity because one hot encoding of NIINs is required to combine them into one data set. The data contained no useful features.

Comparing ML and Time Series Results

For NIIN 010050515, nearly all the time series models performed better than the ML models according to all metrics. Exponential smoothing and simple smoothing performed best overall with the lowest MSE, MAE, and PIS magnitude. However, none of the models had predictions that fit close to the actual values on the plots.

Likewise, for nearly all NIINs, a time series method performed best, with only 2 of 11 NIINs showing best results with an ML model. Overall, moving average, exponential smoothing, and Fourier performed best for the most NIINs. The time series methods overwhelmingly outperform ML methods, so the latter must not have been able to overcome the sparsity challenges by relating NIINs in the model. Table 3-11 shows the best overall model for each NIIN.

Table 3-11. Overall Best Model Performance by NIIN

NIIN	Model	MSE	MAE	PIS	Mean	SD	SE	% zeros
014938822	Random forest	4.95	1.77	22.00	4.14	0.34	0.07	42.86
001651942	Poisson	69.82	6.27	−319.00	12.73	3.28	0.72	27.47
011192008	Moving average	227.28	9.85	464.62	14.44	4.32	0.80	48.35

Table 3-11. Overall Best Model Performance by NIIN

NIIN	Model	MSE	MAE	PIS	Mean	SD	SE	% zeros
014938822	Random forest	4.95	1.77	22.00	4.14	0.34	0.07	42.86
009507783	Exponential smoothing	9.16	2.43	−633.45	8.44	1.64	0.31	17.58
014793739	Moving average	8.55	2.24	185.26	4.94	0.46	0.09	23.08
014793835	Fourier	55.46	5.60	−1319.87	17.06	2.47	0.46	8.79
012223502	Fourier	750.30	23.55	76.86	95.73	19.61	3.64	2.20
014793776	Moving average	461.22	17.06	−4,954.58	40.00	5.03	0.93	18.68
010050515	Exponential smoothing	206.79	11.21	−2,163.50	43.78	7.78	1.45	1.10
014793815	Gradient boosting	4.18	1.55	−114.00	5.27	1.29	0.28	31.87
000546940	Exponential smoothing	135.82	8.99	−1,929.15	27.05	8.36	1.55	1.10

Overall, the evaluation metrics and the plots do not show that any of the time series or ML methods are truly capturing the underlying patterns in the data and predicting future demand well. The predictions do not consistently follow the trend or magnitude of the data. Parts forecasting with demand data thus does not predict well enough to render value to DLA.

Additional Analysis: Part Order Classification

We tested ML methods for classification as predictors of whether orders will exist 8 months into the future. The daily data is lagged by 1 month, included categorical time variables, and predicted 8 months into the future. This data setup is the same as the data setup for ML for parts forecasting except that the data are binary and orders are coded with a value of one; otherwise, a value of zero indicates no orders.

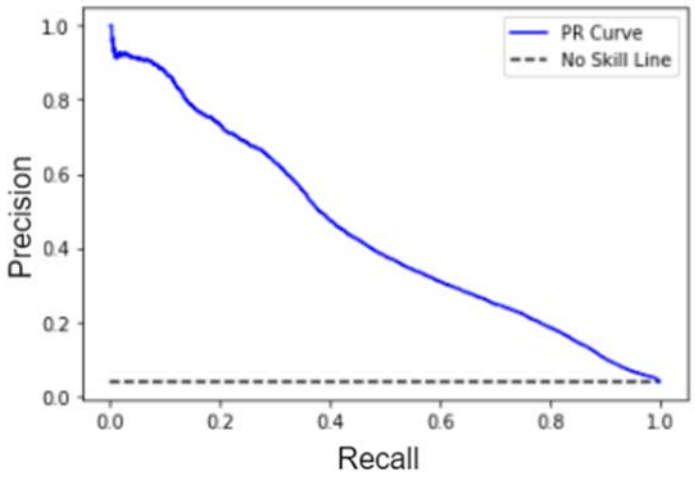
We evaluated random forest for classification, logistic ridge regression, linear discriminant analysis, *k*-NN, and naïve Bayes models using precision and recall. Precision is the percentage of the time an order exists, given a prediction by the model. Recall is the percentage of time the model predicts an order when an order exists. Overall, logistic ridge regression performed the best on the basis of the evaluation metrics. Table 3-12 shows the evaluation metrics for logistic ridge regression and random forest for classification.

Table 3-12. Evaluation Metrics

Model	Precision (%)	Recall (%)	Total error (%)
Logistic ridge regression for classification	44.2	16.8	78.0
Random forest for classification	32.0	16.8	76.5

We employed precision-recall (P-R) curves of each model to evaluate performance. P-R curves are used instead of receiver operating characteristic curves as a visual evaluation tool due to class imbalance. All models showed P-R curves indicating poor performance. Ideally a P-R curve will approach the upper right corner where P=1 and R=1. Instead, the models produced P-R curves similar to the P-R curve for logistic ridge regression in Figure 3-15.

Figure 3-15. P-R Curve for Logistics Ridge Regression



The P-R curves indicate the classification models performed poorly. We applied resampling methods in an attempt to improve the class imbalance, but resampling did not improve the P-R curves and the evaluation metrics.

Maintenance Action Forecasting

Parts forecasting with demand quantity did not predict well enough to render value to DLA, so we explored maintenance action forecasting as a second approach. The goal was to leverage the relationship between the number of maintenance actions and the quantity demanded for an NIIN to forecast the amount demanded for an NIIN. This approach assumes a constant relationship between the number of maintenance actions and the percentages of each NIIN involved in maintenance actions.

Time series methods that predict the number of maintenance actions on the GCU establish a baseline metric for prediction accuracy. We compared results from ML and deep learning methods with the baseline results from time series methods. The number of maintenance actions is modeled as a proxy for demand forecasting, as accurate predictions for maintenance can lead to accurate predictions for parts demand.

Data

Source

MADW™ data sourced from the Decision Knowledge Programming for Logistics Analysis and Technical Evaluation (DECKPLATE) is an integrated data environment for modeling maintenance. Each row represents a maintenance action performed on a specific end item. The DECKPLATE-sourced MADW™ data provides details on operational-level maintenance actions.

Cleaning

We filtered the data according to *TMS* and *ServiceWBS* so they contained only demands for GCU parts on F/A-18s. We aggregated the number of maintenance actions by day, trimmed the dates to range from January 2009 through December 2018, and filled missing dates with zero values. We filtered dates using the start of the maintenance,

OPNStartDate, because many maintenance actions take more than 1 day to complete. The final data set consisted of the date and the corresponding total number of maintenance actions. With this aggregation, data sparsity is eliminated and the data are better suited for ML models. The ML data are aggregated by week, and the time series models use daily level information.

Evaluation Methods

We evaluated models using MSE and MAE and used SD and SE to describe the spread of the predictions.

Time Series Modeling

We developed time series models for various techniques: moving average, simple smoothing, Fourier, exponential smoothing, and Prophet. Each model tests different lengths of training windows on testing accuracy. We used MSE and MAE to find the best window and best corresponding hyperparameters for each model.

We took a sliding window approach to make predictions, looping it through different windows with each model to find the best training size. Sliding windows are also used in time series forecasting for demand and are explained there. However, only one variable is predicted in maintenance action forecasting with time series; thus, filtering data for each model is not required. The windows range in monthly increments from 30 days to 1 year, and then yearly increments from 1 year to 7 years. Exponential smoothing tested these same training windows, aside from the first 6 monthly windows, which were used intermittently, because they were sometimes too small for some modeling parameters to work.

For each window and model, we did the following:

1. Split data into initial training and testing. Set the testing start date to April 13, 2017, and the training end date to September 1, 2016.
2. For the given model, and window length, fit the model and tuned hyperparameters by searching every possible parameter combination for the one that resulted in the lowest possible MAE over the monthly aggregated test set.
3. Calculated MSE and MAE for given window and model.

ML Modeling

Feature Engineering

We transformed data using the same method applied to demand data.

Figure 3-16, the autocorrelation plot, shows that up to 40 weeks of prior data are correlated with the current data within a 95 percent confidence bound. Forty lags are included in the models to capture the autocorrelation in the data.

Figure 3-16. Autocorrelation

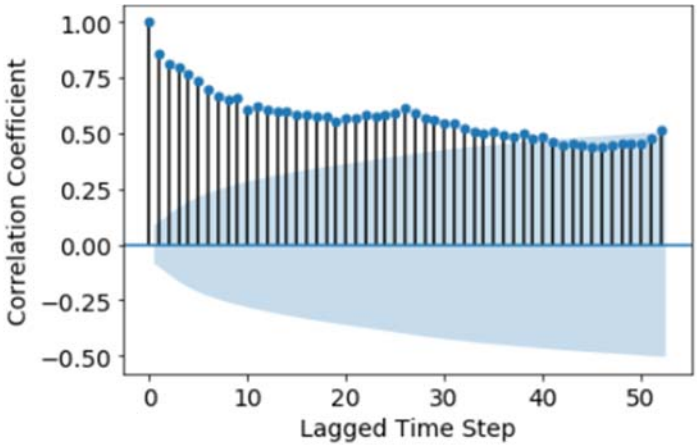


Figure 3-17, the partial autocorrelation plot, shows that a lag of nearly 40 is independently correlated with the data. These two plots emphasize that much historical data (7–10 months) is needed to capture the correlation in the data.

Figure 3-17. Partial Autocorrelation

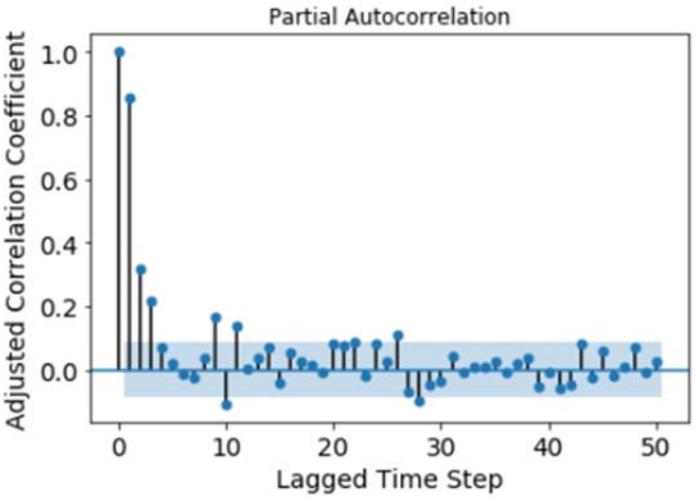
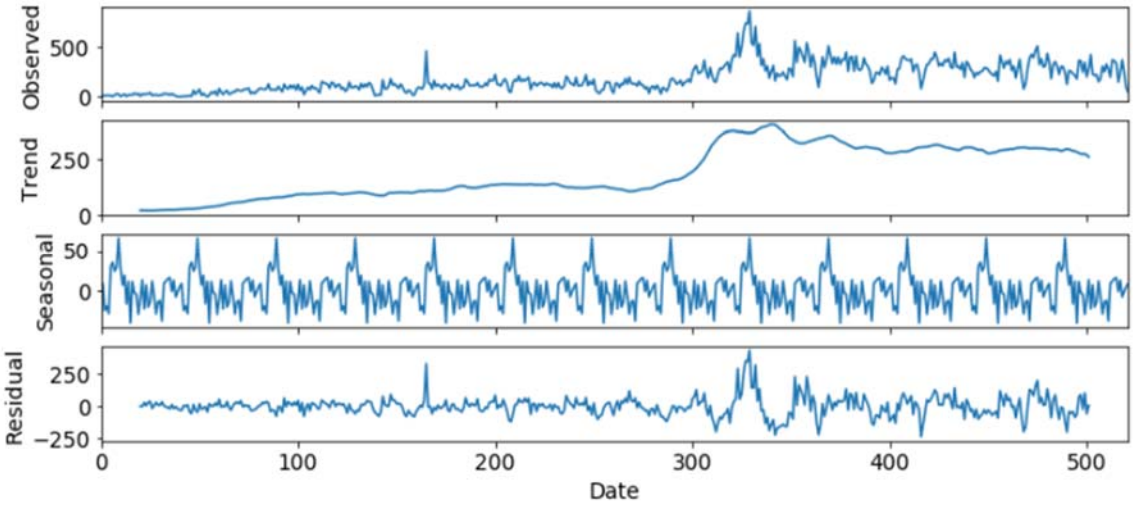


Figure 3-18 illustrates the decomposition plots, which show the seasonality and trend in the data. A decomposition plot breaks down time series data into its four component parts and plots each component sequentially. The third plot from the top in the set below shows the seasonality in the number of maintenance events on the GCU. The plot shows that there is seasonality that cycles roughly every 8 months. To capture seasonality in the data, we include 8 months of historical data in the lagged data frame.

Figure 3-18. Decomposition Plots



Given the autocorrelation and seasonality, we lagged the data by 40 weeks. We evaluated feature importance to determine specific lag for each model. Including more than 40 lags unnecessarily increases the number of regressors in the data and decreases performance by essentially adding noise to the model.

Month is one hot encoded to add time variables to the data. We used the data value of the point 8 months (32 weeks) into the future as the response. We set the gap between training and prediction to 8 months to capture DLA's large lead times.

Modeling

Again, because ML models often perform best with as few variables as possible that still represent the data, each model loops through each number of possible lags to find the best number of lags for the model. With each iteration, the model sequentially includes one less lag. With this process, each model evaluates each possible set of lags from 1 lag through 40 lags.

For each model and lag, we did the following:

1. Split data into initial training and testing by setting the testing start date to April 13, 2017.
2. For the given model and number of lags, we fit the model and tune hyperparameters with randomized grid search. (Appendix B shows a table with the hyperparameter sets from which we sampled, and the final, best hyperparameter sets for each model.)
3. Calculated MSE and MAE for a given model and number of lags.

Development

We applied many of the same models used in modeling demand quantity. We included two neural networks to test whether more complex models performed better than

traditional ML approaches. We selected the best-performing hyperparameters and lag for each model. We explored several ML techniques (described in Appendix A):

- Ridge regression
- Random forest regression
- Gradient boosted regression
- LASSO regression
- Elastic net
- *k*-NN
- Kernel ridge regression
- Convolutional neural network
- Recurrent neural network.

Results

Time Series Results

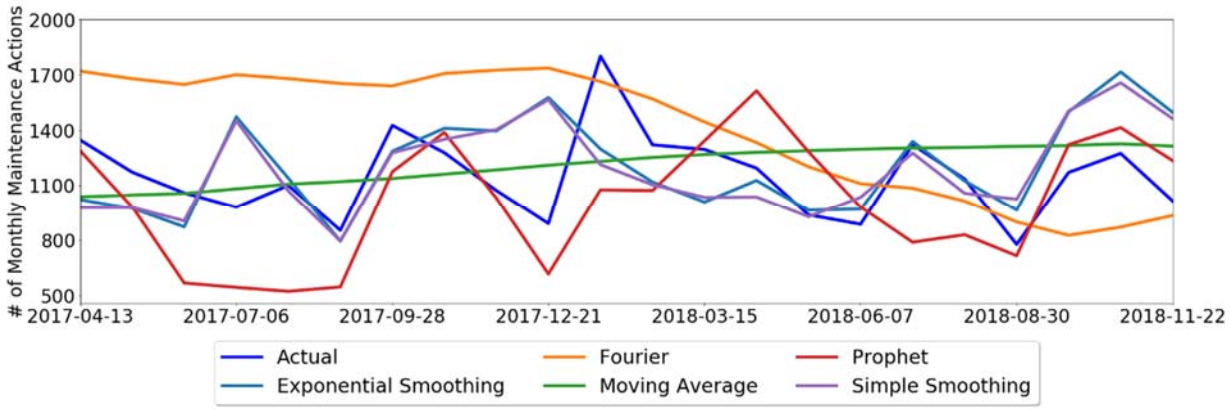
Models are ranked on the basis of MSE and MAE (Table 3-13). The table lists the models in order from best to worst performance according to the evaluation metrics. Moving average performed best by far, with MSE and MAE values much smaller than those for the other models.

Table 3-13. MSE and MAE for Each Trained Model

Model	MSE	MAE	Mean	SD	SE
Moving average	66,731.3	199.56	1,210.91	98.22	21.43
Exponential smoothing	91,047.2	237.50	1,207.64	248.42	54.21
Simple smoothing	93,426.5	246.60	1,186.71	235.80	51.46
Prophet	110,248.0	274.43	1,014.76	324.29	70.77
Fourier	192,495.0	372.33	1,401.99	330.64	72.15

Figure 3-19 shows the plot of the predicted number of monthly maintenance actions for each time series model as well as the actual values. The plot shows that the simple flat line estimate from moving average fits a seemingly linear model to the actual values. The plot also shows how the other time series models do not model the same trend of the actual values.

Figure 3-19. Comparison of Time Series Models



None of the time series methods model maintenance actions despite training on different window sizes and tuning the models to account for seasonality and autocorrelation.

ML Results

We evaluated and ranked the models using a combination of MSE and MAE. Random forest performed best, closely followed by gradient boosted trees, and then the ridge. Table 3-14 shows the MSE and MAE for each model. The table is ordered from best- to worst-performing model.

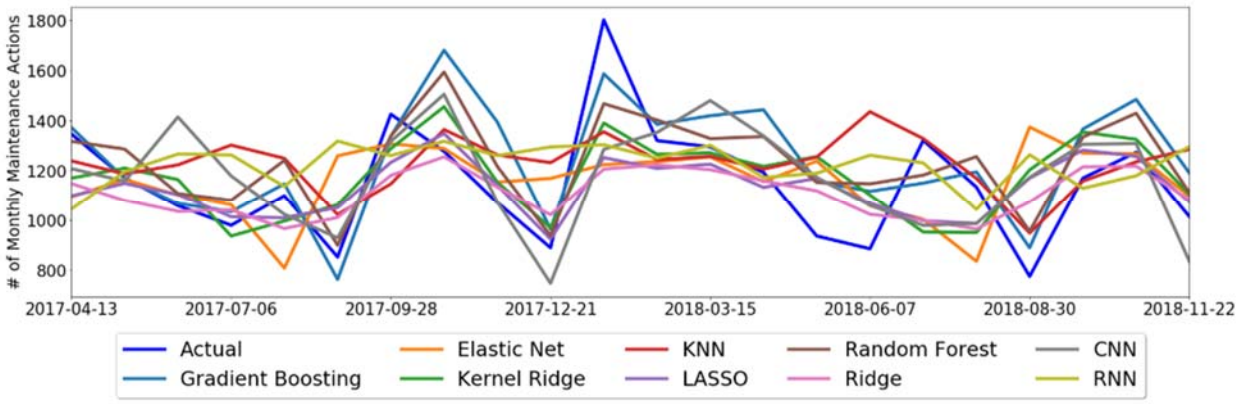
Table 3-14. MSE and MAE for Each Model

Model	MSE	MAE	Mean	SD	SE
Random forest	25,608.20	136.91	1,233.64	171.77	37.48
Gradient boosting	31,145.00	143.41	1,242.23	221.74	48.39
Ridge	37,875.60	146.59	1,108.23	89.70	19.58
LASSO	39,512.10	148.41	1,132.50	107.76	23.52
Kernel ridge	40,220.20	159.77	1,173.50	146.91	32.06
k-NN	52,281.20	173.09	1,232.91	103.89	22.67
Convolutional neural network (CNN)	47,252.40	175.80	1,174.93	198.27	43.27
Elastic net	65,985.20	190.50	1,156.59	139.46	30.43
Recurrent neural network (RNN)	66,693.40	201.44	1,227.20	79.10	17.26

We tested the models over the 22-month period from April 2017 through November 2018. Nearly all ML methods are visually able to model the data and show lower MAE and MSE than those from time series models.

Figure 3-20 shows the plot of actual and predicted values for all fitted ML models. The plot shows that all ML methods predict spikes in maintenance actions as well as periods with less variation. Figure 3-20 shows the evaluation metrics for the ML models. These models deliver promising results that support continued evaluation with other assets and components.

Figure 3-20. Comparison of ML Models



No models exhibited noticeable statistical bias, as none consistently under-predicted or over-predicted values. Figure 3-20 shows the plot of monthly model predictions and monthly actual values for number of maintenance events. The plot shows the lack of model bias and that, overall, most of the models follow the actual value trend.

We compared the results from the best-performing model, random forest, with predictions created with different ensembles of individual model predictions. Figure 3-21 shows monthly predictions for random forest, actual monthly predictions, averaged predictions from all models, and averaged predictions from the best two models, random forest and gradient boosted trees. The plot shows that the mean of the best two models seems to be somewhat closer to the actual values than random forest alone and that the overall mean is further away from the actual values than both random forest and the mean of random forest and gradient boosted trees.

Figure 3-21. Random Forest Comparison

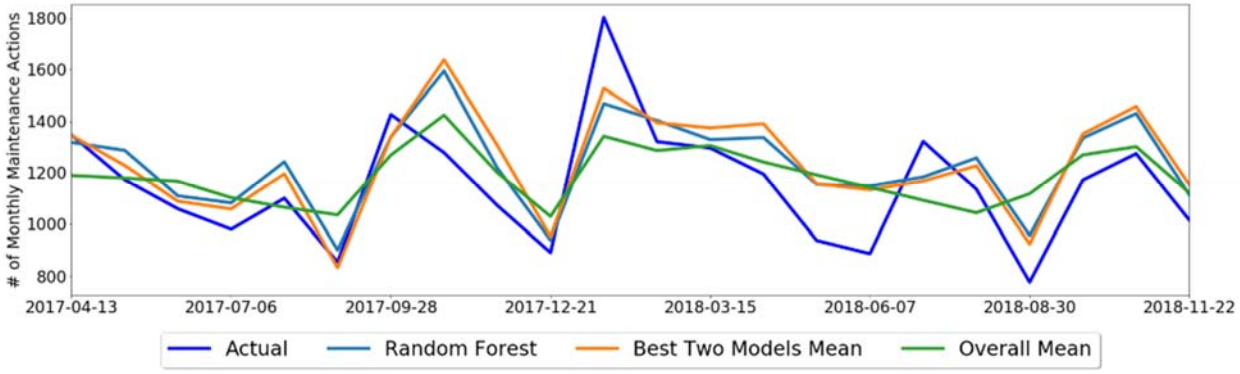


Table 3-15 shows the MSE and MAE for these average values and compares them with the MSE and MAE for random forest. The plot and table indicate that random forest alone and the mean of random forest and gradient boosting performed similarly according to MSE and MAE, but random forest achieves lower SD and SE. Random forest thus performs marginally better than the mean of random forest and gradient boosting.

Table 3-15. MSE and MAE Averages and Random Forest Comparison

Model	MSE	MAE	Mean	SD	SE
Random forest	25,608.20	136.91	1,233.64	171.77	37.48
Best two model means	26,760.30	136.70	1,237.93	194.27	42.39
Overall mean	32,389.90	142.63	1,186.86	105.43	23.01

Although random forest performed best, the mean of random forest and gradient boosting still performed well. This result suggests that averaging predictions may lead to reasonable estimates of the future number of maintenance actions. Further exploration would be needed to see whether a more sophisticated combination of predictions leads to even better performance and improvement over random forest. However, ensemble models come with the computational cost of training multiple models. Combining predictions is only worthwhile if its improvement in accuracy outweighs the computational cost of training multiple models.

Unlike the demand data, the maintenance data are not sparse and thus better suited for ML than demand data. Thus, traditional ML methods performed well, and even simple time series methods had some success.

ML methods significantly under-predicted when trained with less lags than the autocorrelation and seasonality plots suggested are necessary. Increasing the complexity of the problem with more columns in the data and more sophisticated, nonparametric models like the CNN removed the issue with under-prediction.

Comparing ML and Time Series Results

Figure 3-22 shows the plot of the best time series model, moving average, and the best ML model, random forest. Moving average predicts a flatline estimate over the entire testing window, whereas random forest fits closely to the plot of actual maintenance action counts. No time series models, including ones that can make more complex forecasts, were able to model how the data change over time. The MSE and MAE scores seen in Table 3-16 also indicate that random forest performs better, with lower MSE and MAE scores.

Figure 3-22. Random Forest and Moving Average Comparison

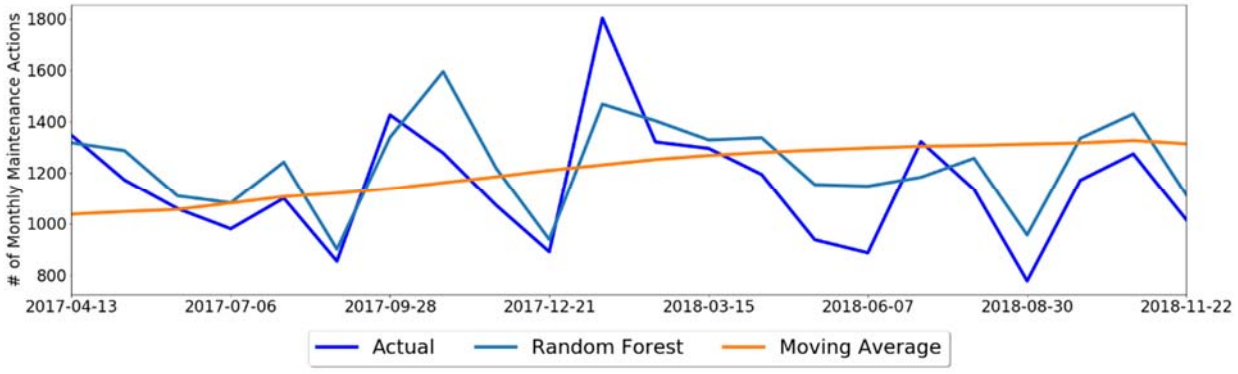


Table 3-16. Random Forest and Moving Average Comparison

Model	MSE	MAE	Mean	SD	SE
Random forest	25,608.20	136.91	1,233.64	171.77	37.48
Moving average	66,731.30	199.56	1,210.91	98.22	21.43

ML accurately predicts maintenance actions, but future research is needed to apply these predictions to forecasting parts. Additional comparisons will show how consistently the ML models outperform time series models. Although the five models developed here do not perform better, other time series models not explored may perform better. Other components and other assets should be included in further exploration.

Finally, in Figure 3-22, random forest performs unusually well. This performance is due to intentional training data selection to include the middle change point in Figure 3-23. When the training data did not include this change point, all models performed much worse, with plots of predicted values much different from actual values. Additionally, results from exponential smoothing and simple smoothing also closely match the actual values. These two models were trained on a data window different from the training set for the ML models, yet still achieve results close to the actual values.

Additional Analysis

Two additional modeling techniques can further enhance maintenance action predictions. Change point detection can find changes in historical trends, and anomaly detection can detect unexpected points in the data. Both excursions can enhance training data cleansing; change points can help define training data boundaries, and removing anomalous points from training data may help ensure they are representative.

Change Point Detection

We explored changes in data trends with change point analysis. Bayesian change point detection is applied on the basis of Facebook's package Prophet. We confirmed change points in the data using the cumulative sum control chart algorithm and validated them with cumulative plots of the data. Cumulative plots show the cumulative total number of maintenance actions. When the average trend changes, the rate of change in the cumulative plot will change, appearing like a piecewise linear model. These plots are aggregated from daily data to be as smooth and detailed as possible.

We found three changes in maintenance actions trends when aggregating by week. The change points occur around May 9, 2013; April 23, 2015; and November 10, 2016. These change points agree with changes in the cumulative plot of weekly maintenance actions. In this plot, changes in the rate of maintenance action count are indicated by points where plot concavity shifts. Figure 3-23 shows the weekly number of maintenance actions (with change points marked in red).

Figure 3-23. Weekly Number of Maintenance Events with Change Points

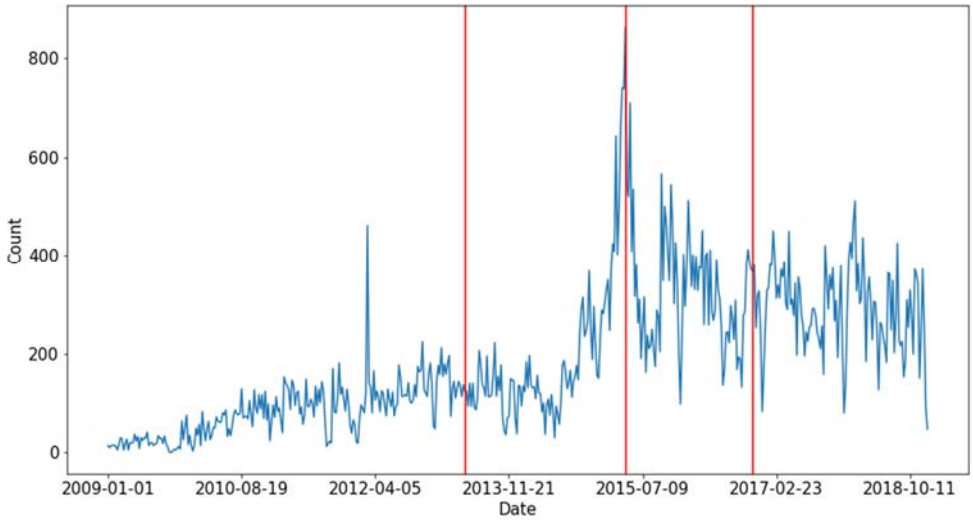
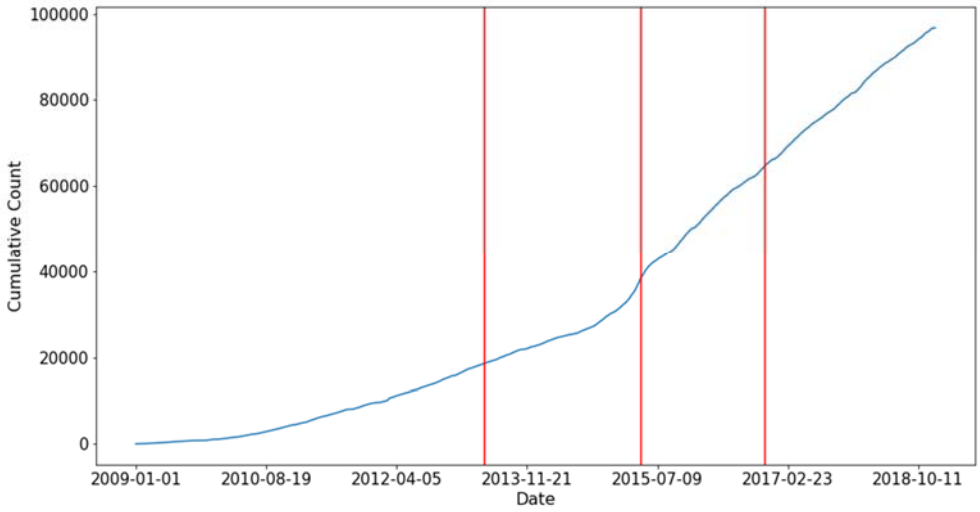


Figure 3-24 shows the cumulative weekly number of maintenance actions (again with change points marked in red).

Figure 3-24. Cumulative Weekly Number of Maintenance Events with Change Points



Change points can be useful in developing an appropriate training set that ultimately improves testing accuracy. For instance, only including the data since the most recent change point may lead to the training data being more relevant to the testing data because new and relevant data may predict future data better. In the future, finding the best way to incorporate change points into training may improve modeling accuracy by signaling the need for retraining or replacing the ML model.

Anomaly Detection

We evaluated 40-week time streams for anomalies using isolation forest, local outlier factor, and elliptic envelope. (Appendix A details these ML methods for anomaly detection.) We used the majority vote from the three methods to label anomalous points.

On the weekly data, we found four periods of anomalous data. These areas correspond to March 2012, March–June 2015, mid-October 2015–mid-May 2016, and February–mid-March 2018. On the monthly data, we see one period of anomalous data: mid-July 2014–mid-May 2015. This period overlaps with two of the anomalous periods in the weekly data. Figure 3-25 shows the daily and Figure 3-26 shows the weekly number of maintenance events labeled with anomalous points. In these plots, a red "x" indicates that the 40-week stream of data leading up to that point is found as anomalous by at least two of the anomaly detection methods.

Figure 3-25. Daily Number of Maintenance Events

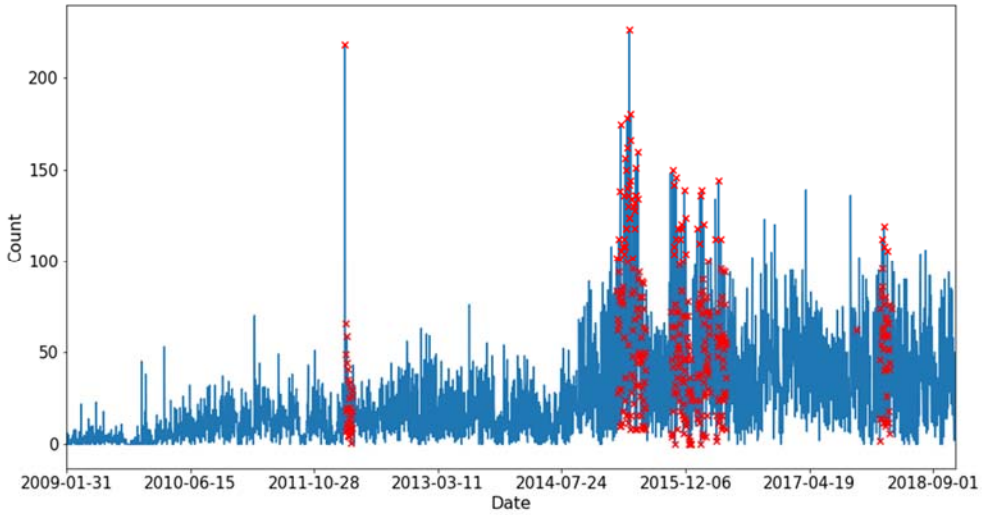
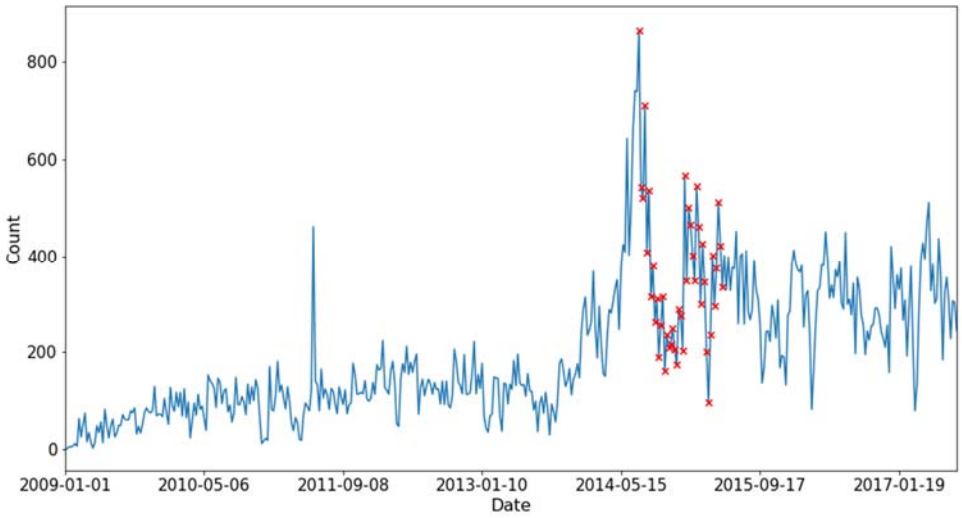


Figure 3-26. Weekly Number of Maintenance Events



The anomalous points could be removed from the data set to build a more consistent training data set. However, to effectively remove anomalous activity for better training, anomalous points need to be validated as truly abnormal for the data. These anomalous periods are likely to be related to policy and historical events. Removing them from the data would thus remove relevant historical data and lead to a less robust model for the future.

Chapter 4
Conclusions

The analysis completed supports three general conclusions:

- Performance of ML models using demand data is limited by the sparsity of data. The best-performing ML models consistently under-predict part demands.
- ML models using maintenance action forecasting addressed the problems with data sparsity and performed better than time series models for maintenance actions. Using ML to predict maintenance action should be further investigated as a means to improve part demand forecasts.
- Change point detection may be useful to signal the need for a different ML model.

This R&D effort provides answers to the following analysis question: Does the analysis of Service historical maintenance records improve parts forecasts and resulting supply support?

When applied to the F/A-18E/F GCU, ML models using Service historical records are limited due to sparse part demands. However, when modeling maintenance events, the sparsity in the data is reduced and the accuracy of ML modeling improves over time series models. Further investigation could produce results across other platforms and parts.

To improve further analysis, use of additional data, beyond MADW™, should be explored to develop a more complete view of supply chain demands. The multi-echelon supply system should be modeled for added precision. Consumption data along with both retail and wholesale demands should be included in follow-on models.

Chapter 5
Recommendations for Further R&D

Through this R&D project, we found that using only the currently available MADW™ data is insufficient to improve DLA demand forecasts beyond those using established DLA models due to the sparsity of the available data. However, the introduction of maintenance actions does produce improved predictions over time series modeling for maintenance actions. Further exploring this approach across other assets and parts may produce results that enhance customer service and lead to improved readiness of the supported weapon systems.

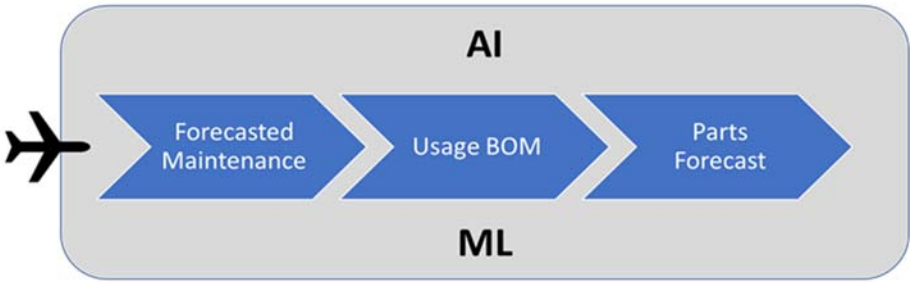
DLA should consider the following options and strategies to improve the overall accuracy and fidelity of parts forecasting. Our recommendations include efforts that require no R&D as well as those accomplished through DLA R&D. We recommend four options directly related to ML modeling to improve parts forecasting and an alternative to ML rooted in simulation modeling. Appendix C details the program management requirements for each recommendation.

Predict Maintenance Events and Associated Usage BOMs

Using the data available in the MADW™ for the F/A-18E/F variants and applying ML, LMI can produce a usage bill of materiel (BOM) to be used for the predicted maintenance events. The comprehensive data in the MADW™ contain many quantitative and qualitative fields that describe the objects maintained on the weapon system, cost and availability metrics, and availability loss for the maintenance event. Predicting maintenance events by Service WUCs can potentially provide improved parts forecasting for DLA:

- Use ML to forecast the O-level and I-level maintenance requirements for the airframe.
- Use ML to identify a BOM for each identified maintenance event by WUC.
- Use this BOM to forecast parts required for the maintenance availability (Figure 5-1).

Figure 5-1. From Forecast Maintenance to Forecast Parts

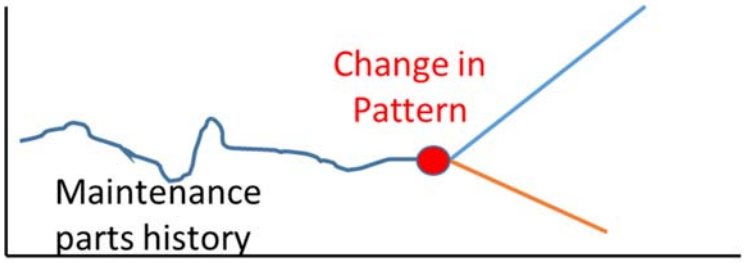


Predict Changes in Maintenance Requirements

Using the data available in the MADW™ for the F/A-18E/F variants and applying ML, LMI can predict changing maintenance requirements. Current forecasting models use historical demand data without regard to changes in requirements due to changes in airframe construct, updates to components, or age:

- Identify changes in maintenance patterns discovered in recent maintenance events.
- Automatically alert DLA planners to intervene and discontinue using the now incorrect statistical forecasts (Figure 5-2).

Figure 5-2. Maintenance Patterns Help Planners



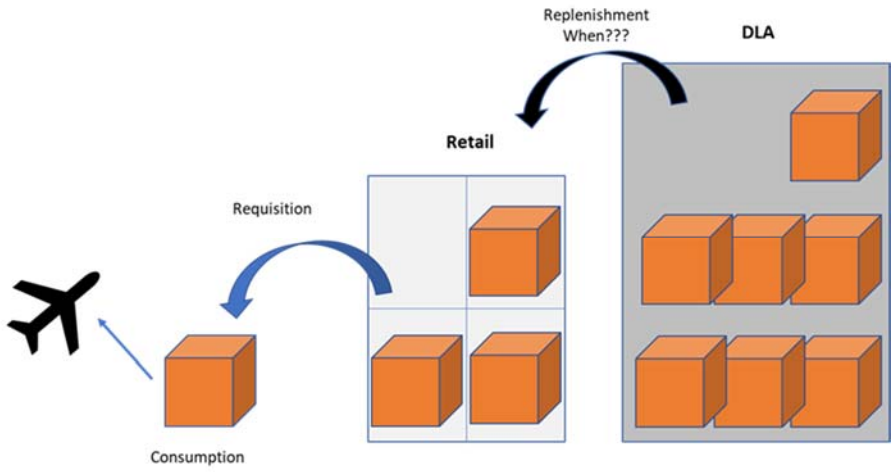
Use ML to Improve Readiness and Acquisition Using Consumption Data

Feeding curated MADW™ consumption data available for the F/A-18E/F variants into DLA's existing models may improve those forecasts. Use of consumption data is a best practice in the private sector. Until now, consumption data have been unavailable to DLA.

To assess using these data, we will employ LMI's Financial and Inventory Simulation Model™ (FINISM™). LMI and DLA R&D jointly developed FINISM™ for use on prior projects. It simulates DLA's existing forecasting models and is a good tool to evaluate how forecasts do (or do not) improve using consumption rather than wholesale supply data:

- Identify ML to curate actual parts consumption data for O/I-level maintenance.
- Feed the consumption data into the models DLA uses for parts forecasts.
- Repeat using the demands that DLA normally uses to drive its forecasts.
- Use ML to identify which populations perform better using consumption data (Figure 5-3).

Figure 5-3. Using Consumption Data for Acquisition



Dynamically Train ML Models

Dynamically choosing models over time can help improve accuracy. Utilizing change points can be helpful if not many reside within the data and lag times are insufficient to cause problems. Evaluating models on the basis of past performance is also a viable solution, but it can be time consuming. (Appendix D details this approach.)

The MADW™ provides operational-level demand data for individual parts over time. These data furnish consumption information not found in the current DLA wholesale requisitioning data. Using data from both of these systems in concert with other available data (platform operational and DLA acquisition data) renders a more holistic picture of parts support and a subsequent improvement in forecasting for each layer of inventory.

Use Predictive Simulation Modeling to Determine Future Part Demands

To leverage service maintenance data in the development of improved DLA part forecasts, predictive modeling using asset-focused, high-resolution simulation can serve to balance inventory levels beyond the limitations of statistical models and traditional forecasting. LMI has developed an asset-focused, high-resolution simulation—Demand Pro—and applied it across many DoD programs over the last 20 years:

- *Asset-focused*. The Demand Pro simulation platform models each individual asset in a fleet or each weapon system in a service inventory.
- *High-resolution*. Maintenance, supply, logistics, and operations are detailed for each asset. Five levels of indenture are modeled in the asset structure.

This platform has been applied to predict future part requirements at the operational level as well as intermediate and depot levels. Demand Pro delivers a capability that far surpasses forecasting tools. Through past performance, LMI has demonstrated the accuracy and depth of its predictive simulation platform and prescriptive analytics vs. widely used traditional forecasting methods (Table 5-1). (Appendix E details this approach.)

Table 5-1. Demand Pro Modeling versus Traditional Forecasting

Demand Pro modeling and analysis	Traditional forecasting
Focuses on assets, their components, operations, and the activities that sustain these assets through both planned and unplanned maintenance. Models a holistic, detailed asset life cycle, including components, operation, reliability, maintenance, sustainment, and supply.	Views demands in isolation. Fails to connect assets and their operation to performance metrics and costs in an accurate, detailed manner.
Uses historical data to establish an initial condition of assets and components, and then generates volumes of predictive output data for mining to represent a set of possible future outcomes. Maps the possibilities of future outcomes to enable well-informed decisions and prescribe effective actions.	Relies heavily on historical data, thus requiring insights to be tightly coupled to past events. This fundamental flaw creates a gap between forecasts and actual observations in dynamic systems where the future does not equal the past.
Leverages many replications of possible future outcomes to enable a deep understanding of risk, uncertainty, and confidence in reported metrics.	Because only one set of historical data exists (it is not possible to repeat the past to generate new outcomes), attempts to separate natural variability from meaningful relationships fail repeatedly.
Eliminates the need for simplifying assumptions. Explicitly quantifies and represents reliability, maintenance task duration, shipment delays, repair effectiveness, and other sources of uncertainty. Models aging on a component level. Includes complex representation of probabilistic age, including independent component aging and restoration and updating age distributions on the basis of failure or maintenance events.	Requires many over-simplifying assumptions about the nature of inputs that include uncertainty, for example, exponentially distributed failure rates and exponential maintenance task duration. Assumes that asset, parts, and components do not age over years of operation. Assumes maintenance is unchanging over time.

.

Appendix A
Model Descriptions

Time Series Models

Time series models employ raw time series data (a quantity or measurement and a time when that information is collected) and attempt to capture how the data change over time and use them to forecast what they will look like in the future. Several time series models are used to predict demand quantity and number of maintenance events.

Moving Average

Moving average defines a lookback window length and averages all the data points in this time window to obtain the estimate of the value for a future point in time. The lookback window maintains a constant length and moves forward with each time step. This is a simple model with no real assumptions about the nature of the incoming data. It tends to predict very well on constant data and worse in any cases where the variance is high. It is unable to capture more complex patterns in time series data.

Exponential Weighted Moving Average/Simple Smoothing

Like moving average, exponential weighted moving average (EWMA) averages data points from a time window to use for forecasting. However, these data points are not all treated equally. Values further back in time are weighted exponentially less than more recent points. The main assumption here is that more recent history is more indicative of future points than older points. These models work well in situations where moving averages perform well but are less sensitive to changes in the data. These models still tend to perform worse in situations where the variance in high. They are unable to capture more complex patterns in time series data.

Exponential Smoothing

Exponential smoothing takes the idea from EWMA, weighting all previous data points in how they contribute to the estimate of the future, and expands it to include information about trends and seasonality present in the data. Trends are simply a representation of the direction the data are moving over time. Seasonality refers to patterns within the time series data that occur at consistent intervals in time. The smoothing algorithms can incorporate the information from the weighted average, trends, and seasonality to develop a forecast. These models work best with clear trends or seasonality in the data that can be fit and where the data can mostly be represented by a combination of these elements and the weighted average. One of the biggest difficulties with using these algorithms is determining the correct trend and seasonality needed to properly model the data. Also, when data have inconsistent trends or seasonality the predictions can be off.

Fourier

Forecasting using Fourier transformations use the rule that any stream of non-zero values can be broken down into some number of sine waves. During Fourier forecasting,

the time series data are broken down into the set of sine waves that can be combined to re-create it. Instead of re-creating the time series, a select number of the sine waves that most represent the data are combined to give a smoothed representation of the time series. Use of these models assumes the time series is composed of a set of patterns that happen intermittently (seasonality) that the Fourier waves can capture and project into the future. They are particularly strong at modeling when the data have very strong evidence of seasonality. Conversely, the models are weak when the data do not contain many repeating patterns or change frequently and inconsistently.

Prophet

This model is used for forecasting and change point detection.

Facebook's Prophet is a widely used forecasting tool that takes an "analyst-in-the-loop approach" by combining statistical forecasting methods with an analyst's domain knowledge to achieve optimal forecasting accuracy. Rather than learning temporal relationships in the data like with traditional time series methods for forecasting, Prophet fits a curve to the historical data with generalized additive models, Fourier transformations, and categorical features to address trend, seasonality, and holidays. Its equation is $y(t) = g(t) + s(t) + h(t) + epsilon$, where $g(t)$ is the trend of the data, $s(t)$ is the seasonality in the data, $h(t)$ is the categorical features indicating holidays in the data, and epsilon is the normally distributed error.

Prophet fits the trend, $g(t)$, to the data with piecewise logistic growth model if the data describe growth that will saturate at carrying capacity or piecewise linear model otherwise. It automatically finds the change points in the data and uses them to define the cut points in the piecewise models. To forecast, Prophet assumes that the future data will follow the same trend as the historical data and have the same change point frequency and associated rate of change magnitudes. To address seasonality, $s(t)$, Prophet adds Fourier series to fit the periodicity. Finally, to fit a function for holidays, $h(t)$, Prophet can take in lists of holidays and their dates to add an indicator function to represent whether any given date is a holiday.

To optimize this equation, Prophet uses Bayesian inference methods to update the distributions of all the model parameters and determine the model's forecasting uncertainty and offers the analyst opportunities to alter the model's saturation capacities, change points, holidays and seasonality, and smoothing parameters based on their expertise in the domain.

ML for Regression

These models are used to predict the demand quantity. The following supervised ML methods all take in a continuous response variable and a set of features that are used to predict the response variable called regressors. Each description includes how the model works and the model's assumptions, strengths, and weaknesses.

Random Forest

Random forest creates uncorrelated decision trees with class assignments in the leaves by randomly sampling the observation set with replacement and fitting a decision tree to each random set. It makes predictions by applying all the decisions trees to the observation and taking the average predicted value. Random forest is interpretable,

unlikely to fit to noise, and can indicate which regressors are important to the model. However, it is ineffective for non-stationary data and does not perform well with highly non-linear relationships.

Ridge Regression

Ridge regression fits a linear line to the data points such that the total difference between each point and the line is minimal. It weights coefficients for regressors to allow for some regressors to be more important to the model than others. Ridge regression is thus a regularized version of linear regression. The model assumes linear relationships between the regressors and response, lack of outliers, and that the error between the actual points and the line are independent and identically distributed (IID). The errors are independent from one another and follow a normal distribution centered at a mean of zero. Ridge regression is unlikely to overfit, computationally fast, and interpretable but it is only effective with linear relationships.

LASSO Regression

LASSO regression fits a linear line to data points much in the same way that ridge regression does. It also weights coefficients to permit certain regressors to be more important to the model than others. The major distinction between ridge and LASSO is that LASSO's weighting allows some coefficients to be driven all the way to zero. This allows LASSO to determine feature importance within possible regressors. The model has the same assumptions as ridge: a linear relationship between regressors and response, lack of outliers, and IID errors. LASSO is unlikely to overfit, computationally efficient (though slightly less than ridge), and interpretable. It is only effective in modeling linear relationships.

Elastic Net Regression

Elastic net regression works as a combination of both ridge and LASSO regression occurring at the same time. It weighs regressor coefficients according to both models. The model again assumes a linear relationship between regressors and response, lack of outliers, and IID errors. It is computationally efficient (though less than either LASSO or ridge) and can capture more complex relationships than LASSO or ridge. It is somewhat less interpretable than either ridge or LASSO and more prone to overfitting than either of those models. It is again only effective at modeling linear relationships.

Kernel Ridge Regression

Kernel ridge is a version of ridge regression that utilizes a "kernel trick" to transform the data into another dimensional space during fitting. In the original data space, the data may not be linear and regression models would not be effective, but by transforming the data it may have a linear relationship with the response. It is a non-parametric version of ridge regression. The major assumption of this algorithm is that transforming the data to another dimensional will uncover a linear relationship between the data and response where normal ridge regression assumptions hold.

Poisson Regression

Like ordinary least squares regression, Poisson regression fits a line to the data such that the total difference between the points and the line is minimal. However, it assumes the response data follow a Poisson distribution rather than a Gaussian distribution.

Gradient Boosting

The gradient boosting model creates several decision trees and combines them to minimize loss for the best predictive ability. A decision tree is essentially a series of conditions that filter the data into different groups, with each node of the tree representing a split in the data according to the condition. For instance, a node can be whether a variable is less than a specified value, and all observations meeting that condition are filtered by that node and have similar response variables. Gradient boosting can minimize a few loss functions, and each loss function is used in different scenarios. To forecast quantity demanded, Poisson loss is minimized because the data consist of count totals and the Poisson distribution specifically models discrete counts. Gradient boosting is flexible, computationally fast, and can indicate what features are important to the model. However, it lacks interpretability, can fit to noise, requires tuning many hyperparameters, and is ineffective for non-stationary data.

Feed Forward Neural Network

A feed forward neural network (FFNN) is a dense layered system of equations that optimizes internal weights with MSE to be able to take input values and deliver an approximate output prediction after training on a large amount of data. FFNNs are good at handling non-linear relationships and work well with large data sets. However, they can fit to noise easily, are ineffective with highly sparse data sets, difficult to interpret, and train slowly.

CNN

CNN is similar to an FFNN in that it passes input through a series of nested equations to approximate an output. The CNN differs by applying special functions to the input space to try and detect patterns or relationships between different columns. After processing the input space, the output is then put through an FFNN and used to estimate an output. CNN’s have been successful in finding patterns in sequential data and are less computationally complex than other neural network approaches to capture this information. Like FFNNs, they fit to noise easily, are ineffective with highly sparse data sets, difficult to interpret, and train slowly. Also, if no relevant patterns exist in the input space, they will not be effective.

RNN

RNN works very similarly to the previously mentioned CNN. It attempts to capture information in sequential data and pass that information into an FFNN to approximate an output prediction. It performs well in the same situations as the CNN but is specifically designed for modeling time series data, so it can, in some cases, perform better. Like FFNNs, it fits to noise easily, is ineffective with highly sparse data sets, is difficult to interpret, and trains slowly. Also, if the data are not purely a function of the sequence, modeling over it will not be effective.

Classification Models

Classification models train similarly to regression models but learn to predict a binary outcome. Specifically, they can learn to predict whether something will or will not happen. In this research, classification models trained to predict an order for a given part would occur on a given day.

Logistic Regression

Logistic regression models the probability that an observation belongs to a specific class by fitting a logistic sigmoid function to the predictor variables. This function transforms the output from the regressors to values between zero and one to return probabilities that can be mapped to class assignments. The model selects a threshold for these probabilities. All observations with a class probability above this threshold are assigned to one class while observations with a class probability below the threshold are assigned to the other. The regressors for this model are whatever set of predictors the data scientist finds useful. The response variable for logistic regression is the list of predicted binary class assignments. Behind the scenes, logistic regression minimizes cross-entropy to fit the best logistic sigmoid function for the best predictive ability. It optimizes parameters in the cross-entropy function to do so. Use of logistic regression requires a linear relationship between regressors and the response, lack of multicollinearity, lack of outlier, and IID errors. Logistic regression is a good model choice for interpretability, but is unstable when the two classes are very distinct.

Naïve Bayes

Naïve Bayes finds the probability an observation belongs to a given class by using Bayes rule and assuming each regressor is an independent feature. Bayes rule is the equation relating the conditional probability of a class with the marginal conditional probabilities of the features where X is the class and y and z are regressors. Its equation is:

$$P(X|y,z) = P(y|X) * P(z|X) * P(y) * P(X)$$

The model calculates the probability that each feature indicates each class, and each conditional probability among features. To predict the class to which an observation belongs, the model calculates the probability of the observation belonging to each class by formatting the problem into Bayes rule and plugging in the feature probabilities and conditional probabilities. The model returns the class with the greatest probability. Naïve Bayes is interpretable, fast, and insensitive to irrelevant regressors and cannot represent complex behavior. Thus, this technique is unlikely to over-fit to training data and can quickly adapt to changing data. However, it assumes that the features are independent regardless of whether they truly are.

Random Forest for Classification

Random forest for classification works the same as random forest for regression. The only difference is that it contains class assignments in the leaves rather than continuous numbers. As before, random forest creates uncorrelated decision trees with class assignments in the leaves by randomly sampling the observation set with replacement and fitting a decision tree to each random set. It makes predictions by applying all the decisions trees to the observation and taking the majority vote. Random forest is

interpretable and unlikely to fit to noise, and can indicate which regressors are important to the model. However, it is ineffective for non-stationary data and does not perform well with highly non-linear relationships.

k-NN

k-NN uses feature similarity to predict the class assignment for an observation. It plots an observation in the training data's feature space, finds the *K* (the chosen number of neighbors to use) closest training observations to the point according to the specified distance metric, and uses the class assignment with the highest frequency as the observation's prediction. *k*-NN's hyperparameters are *K*, which is the number of neighbors to evaluate, and the distance metric. The data scientist can optimize prediction accuracy by evaluating different *K* values and different distance metrics to minimize classification error on the training set then use the best *K* and the best distance metric found to predict on testing data. *k*-NN makes no data assumptions other than that observations in the same class are nearest to each other in feature space. *k*-NN is a good option because it is non-parametric and does not assume any underlying data distribution. *k*-NN is a lazy algorithm and does not generalize based on the training data and trains a model quickly. *k*-NN is interpretable. However, *k*-NN is computationally expensive because it stores all the training information and takes a long time to make predictions since it needs to evaluate every training point before it can choose those closest points for each testing point.

Support Vector Classifier

A support vector classifier creates a separating hyperplane that divides the two classes in feature space. The hyperplane is a maximal separating hyperplane. The hyperplane separates the classes such that the hyperplane is the furthest from observations in all classes and separates classes to minimize classification error. It assumes that the data are IID.

Linear Discriminant Analysis

Linear discriminant analysis (LDA) calculates the probability of an observation's belonging to a class given the predictor variables by fitting a probability density function to the predictor variables for each class then using Bayes rule to find the probability of a class given each probability distribution. The model uses these probabilities to define decision boundaries in feature space that surround observations in each class. In the case of a binary classification, LDA defines a line that separates observations in one class from the other; observations falling in the region for a class are classified as belonging to that class. It assumes that the predictor variables are drawn from a Gaussian distribution or a multivariate Gaussian distribution and that they have a common covariance matrix. LDA is a good model choice because it does not require hyperparameter tuning, it is stable even if classes are extremely distinct or contain few observations, and it works well with multi-class classification problems.

Gradient Boosting for Classification

Gradient boosting for classification works in the same manner as gradient boosting for regression but minimizes classification loss. As explained in regression, gradient boosting creates several decision trees and combines them to minimize loss for the best predictive ability. A decision tree is essentially a series of conditions that filter the data

into different groups, with each node of the tree representing a split in the data according to the condition. For instance, a node can be whether a variable is less than a specified value, and all observations meeting that condition are filtered by that node and have similar response variables. Gradient boosting is a flexible model that can handle non-linear relationships in data, is computationally fast, and can indicate which features are important to the model. However, it lacks interpretability, can fit to noise, requires extensive hyperparameter tuning, and is ineffective for non-stationary data.

Anomaly Detection

Anomaly detection algorithms look for data points in a data set that are somehow different from most of the data set. Identifying streams of maintenance action counts that differ from the majority of normal maintenance action count patterns could be valuable to increasing DLA awareness of upcoming demand changes. Anomaly detection algorithms model maintenance action count data to identify these anomalous points.

Isolation Forest

Isolation forest labels points that are remote from most of the data as anomalies. It uses decision trees with random splits to partition the data. As outliers are infrequent, different from most of the data, and remote in feature space, random partitioning should cause the outliers to lie close to the root of the tree. Outliers need fewer random splits of the tree to be separated from most of the data. Isolation forest calculates an anomaly score for each of the points using aspects of the tree and classifies points according to the score, with scores close to one indicating anomalies and scores less than 0.5 indicating non-anomalies. The points with the greatest anomaly scores are ultimately classified as anomalies.

Elliptic Envelope

Elliptic envelope assumes the data are Gaussian and fits an ellipse to surround most of the data. It classifies points lying outside of this ellipse as anomalous. The model fits this ellipse such that the percent of observations outside of it is equal to the contamination, the specified percentage of points expected to be outliers. The data scientist must use domain knowledge to specify the contamination.

Local Outlier Factor

Local outlier factor classifies points with low local density as anomalous. Points with low local density are those that are remote in feature space. The algorithm calculates each point’s local density and compares it to the average of its neighbors’ local densities. If the point is less dense that its neighbors, then it is more remote than its neighbors. It classifies the most remote points as anomalies.

Clustering Methods

Clustering methods group data points according to different measurements of similarity. They are unsupervised learning algorithms that require manual evaluation once they execute. In this research, clustering methods grouped data about specific NIINs over time to identify similar NIINs.

k-Means Clustering

k-means clustering assumes a set of data has a specific number of clusters into which the data can be broken down. It then tries to optimally assign these clusters so that each data point is close to the nearest (by distance) cluster's mean while keeping clusters as far apart as possible. The algorithm assumes that the data being clustered can be adequately represented by distance metrics and that the mean of the data is a meaningful representation. One of the difficulties in using this algorithm is that it is necessary to decide the correct number of clusters, a subjective decision.

Hierarchical Clustering

Hierarchical clustering works similarly to *k*-means in that it is given a set number of clusters and tries to optimally place each cluster. Instead of optimally matching clusters and points on optimal distance from a mean, points are assigned too many clusters initially and the algorithm works backwards to group the closest clusters together. Again, the number of clusters to use is a subjective manual decision.

DBSCAN

DBSCAN works differently from the previous algorithms described. DBSCAN can use any measure of similar or close data points and tries to find groups of data points very close together based on the selected measure. Any area where lacking a significantly dense representation of data points (low density areas) will be marked as noise. The major strength of this method is that there is no need to manually determine cluster count. The drawback is that it can be difficult to determine the correct thresholds for how dense an area should be to be considered a cluster.

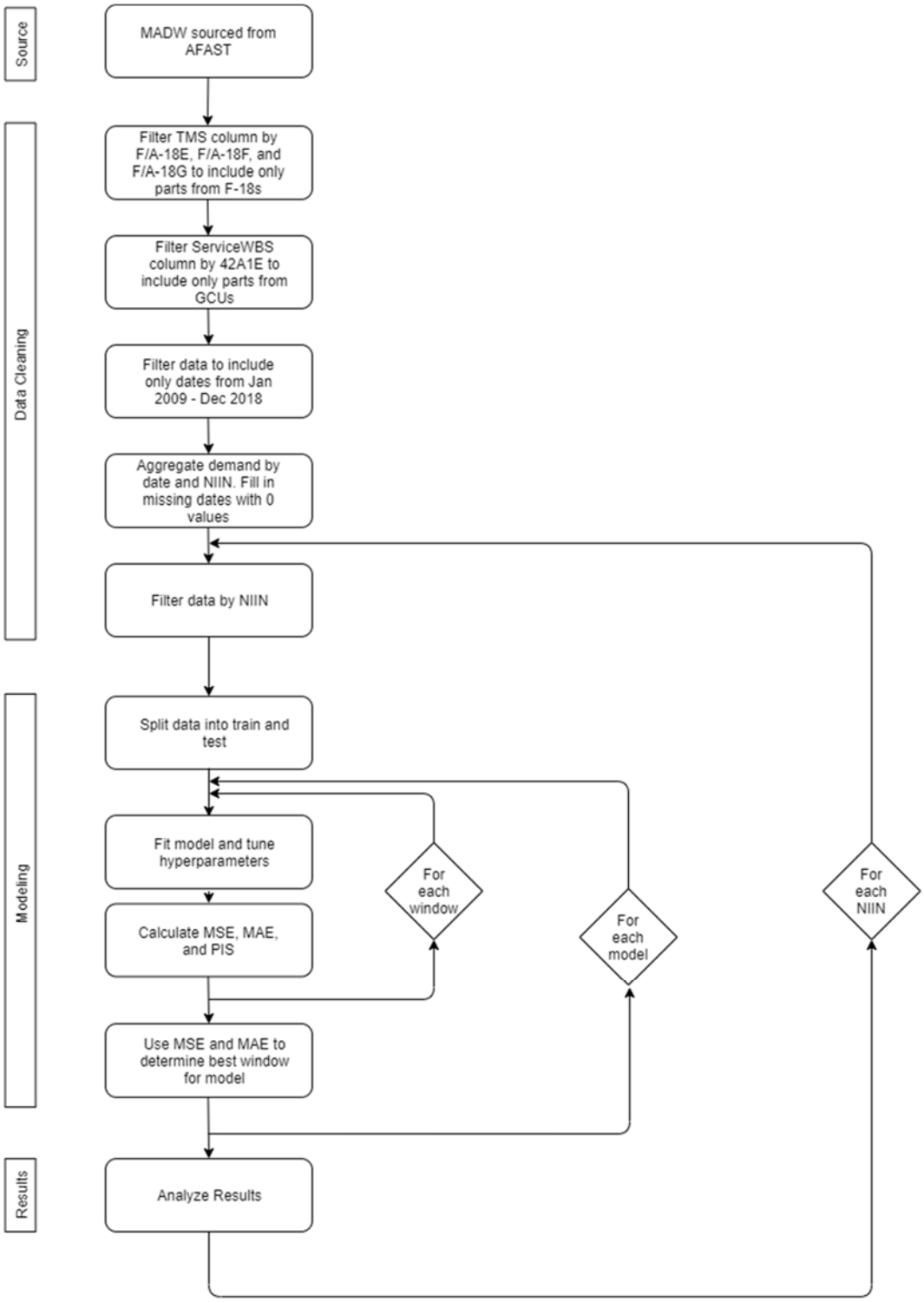
GMMs

GMMs work similarly to *k*-means clustering in that they try to optimize clusters on the basis of distance between clusters and nearness of points within clusters. The main difference is that instead of giving a hard assignment to the cluster to which a point belongs, the model is probabilistic and there is a probability of any point belonging to any cluster. These models are useful in cases with low confidence on the appropriate number of clusters to use.

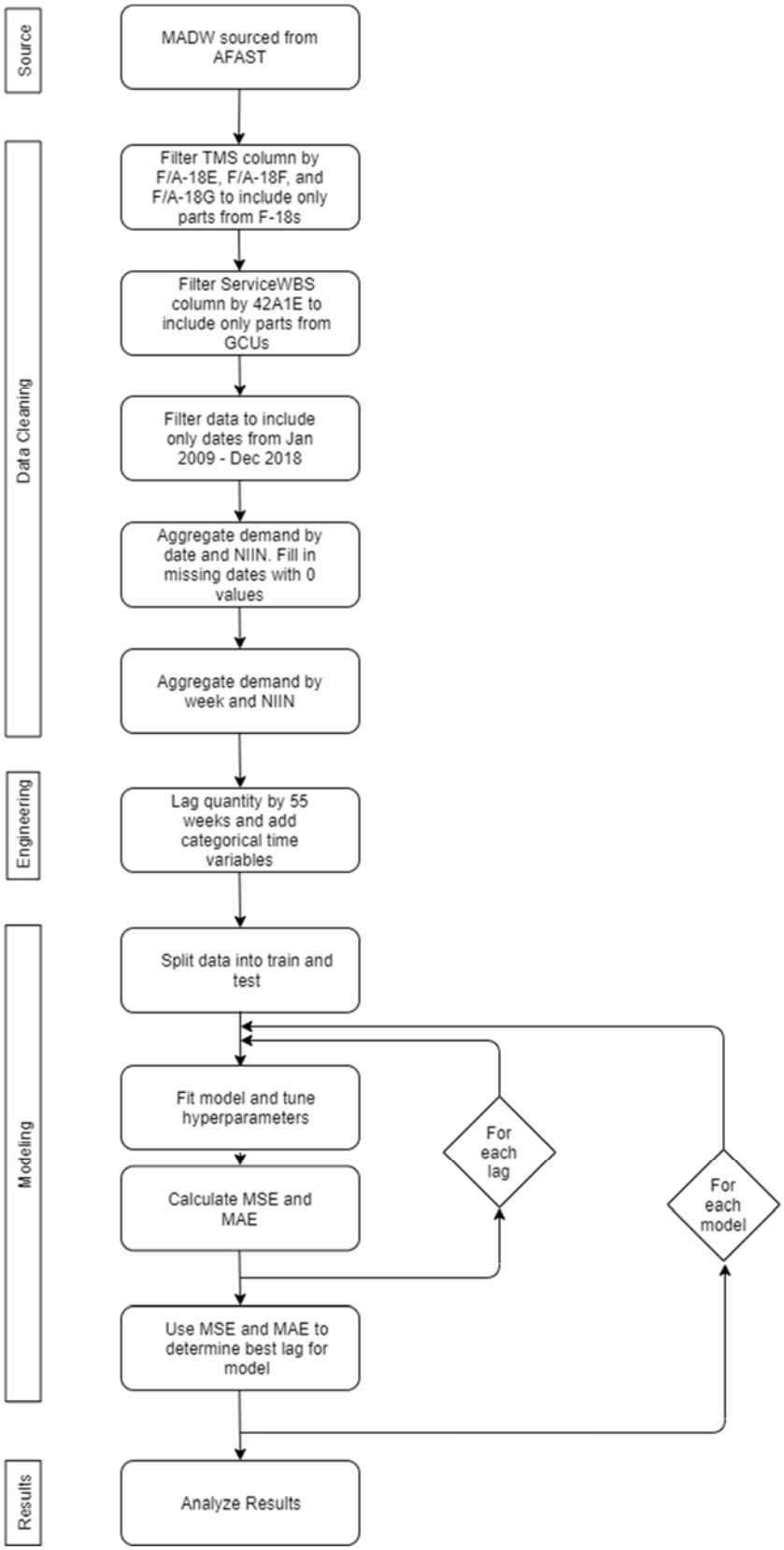
Appendix B
Detailed Modeling Method

The following flowcharts illustrate the steps used during this project for the multiple ML efforts.

Time Series Using Part Demands

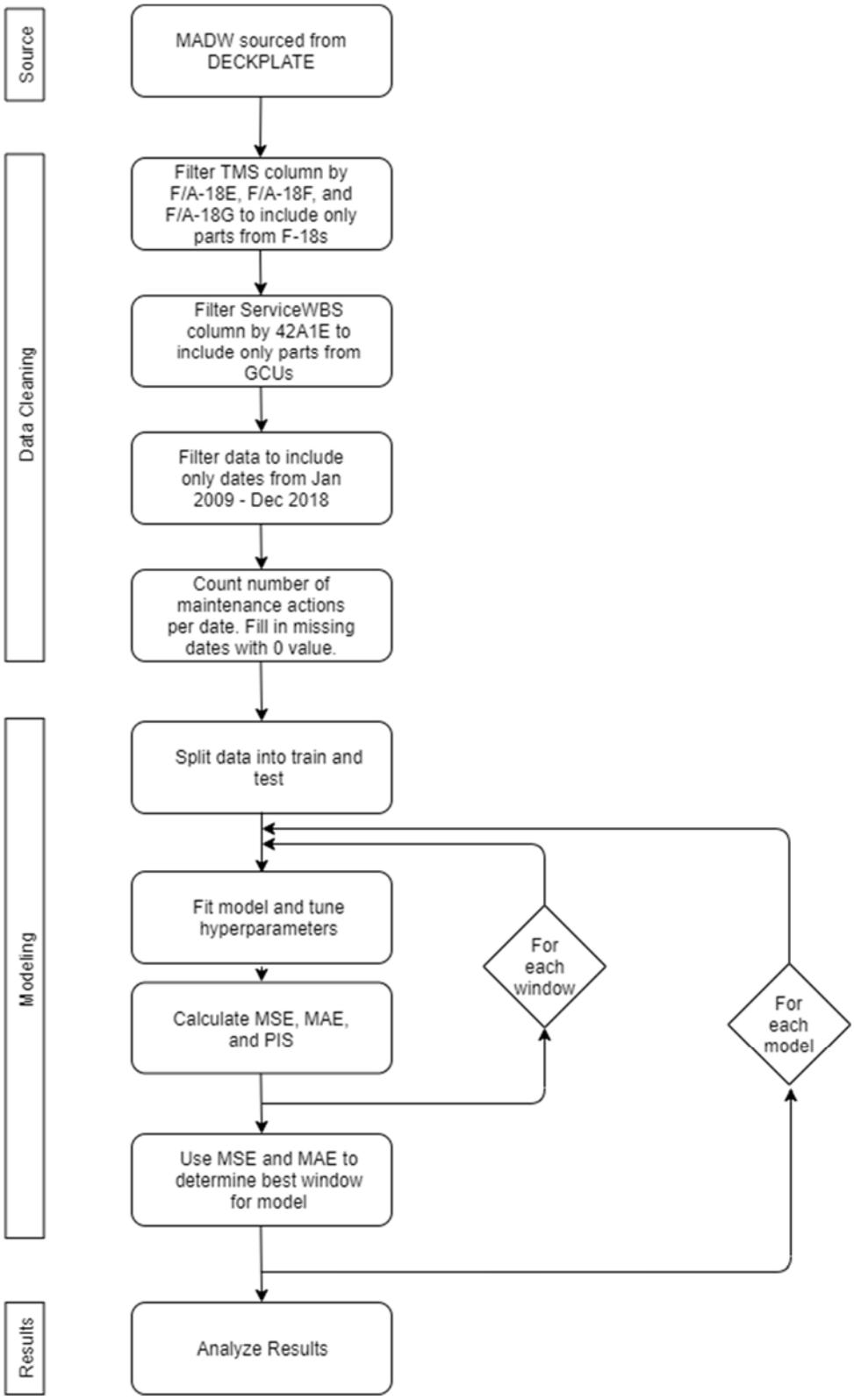


Machine Learning Using Part Demands



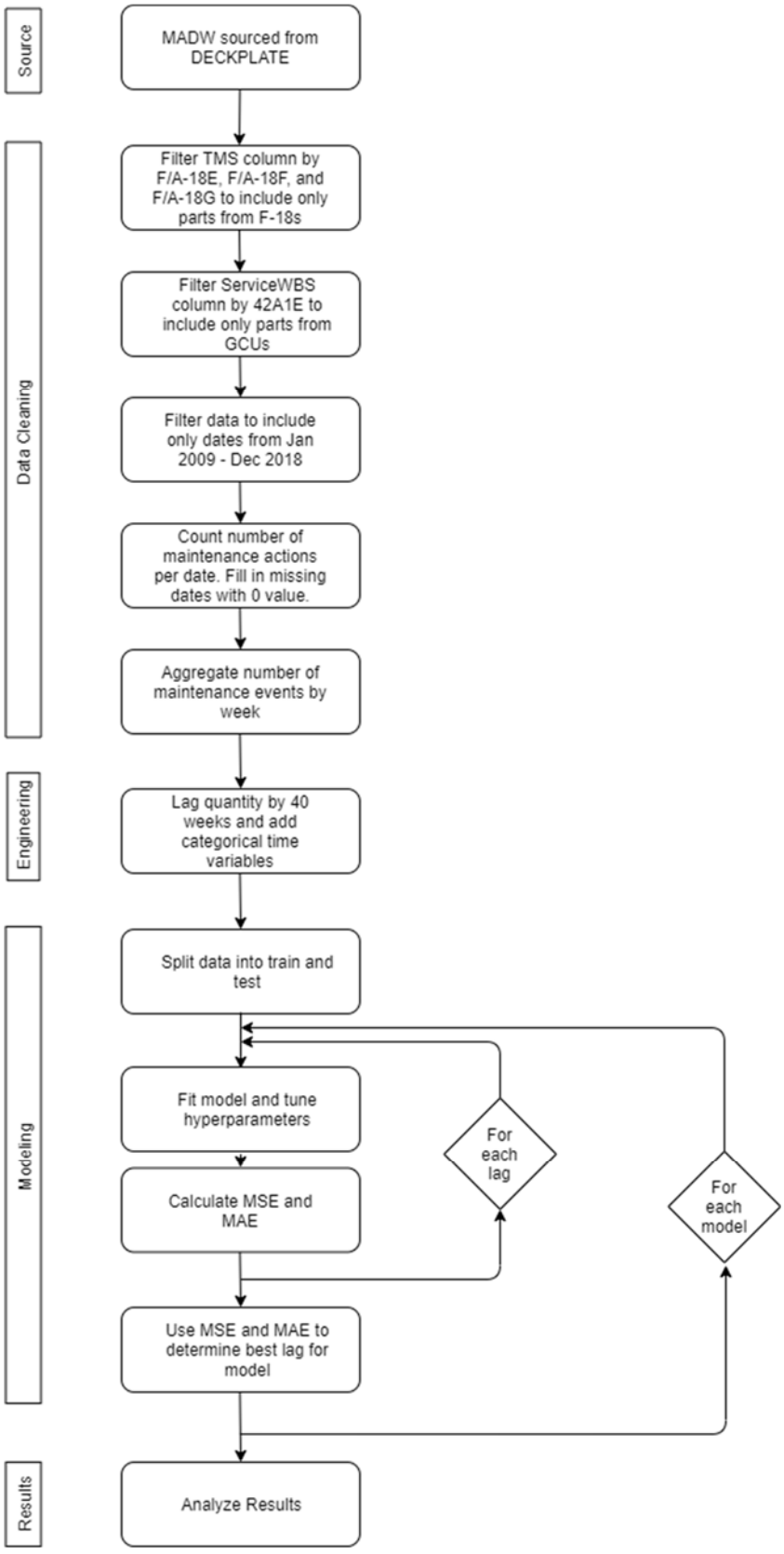
Model	Hyperparameter set	Best hyperparameters
Gradient boosting	min_child_weight: [1, 5, 10], gamma: [0.5, 1, 1.5, 2, 5], subsample: [0.6, 0.8, 1.0], colsample_bytree: [0.6, 0.8, 1.0], max_depth: [3, 4, 5], reg_lambda: [1e-2, 1e-1, 0.5, 0.99], objective: [‘reg:squarederror’, count:poisson’]	Min_child_weight: 10 Gamma: 5 Subsample: 1.0 Colsample_bytree: 1.0 Max_depth: 3 Objective: ‘count:poisson’
Ridge regression	alpha: [1, 0.1, 0.01, 0.001, 0.0001, 0] fit_intercept: [True, False] solver: [‘auto’, ‘svd’, ‘cholesky’, ‘lsqr’, ‘sparse_cg’] normalize: [True, False]	Alpha: 0.001 Fit_intercept: False Normalize: True Solver: ‘lsqr’
Random forest	bootstrap: [True, False], max_depth: [5, 10, 20, 40, 60, 80, 100, None], max_features: [‘auto’, ‘sqrt’], min_samples_leaf: [1, 2, 4], min_samples_split: [2, 5, 10], n_estimators: [200, 400, 800, 1200, 1600, 2000]	Bootstrap: True Max_depth: None Max_features: ‘sqrt’ Min_samples_leaf: 4 Min_samples_split: 10 N_estimators: 800
k-NN	n_neighbors: [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15] weights: [‘distance’, ‘uniform’] algorithm: [‘auto’, ‘ball_tree’, ‘kd_tree’, ‘brute’] leaf_size: [20-30] p: [1, 2, 3, 4, 5]	N_neighbors: 6 Weights: ‘uniform’ Algorithm: ‘brute Leaf_size: 25 P: 1
LASSO regression	alpha: [0.01, 1.01, 2.01, 3.01, 4.01, 5.01, 6.01, 7.01, 8.01, 9.01, 10.01] fit_intercept: [True, False] normalize: [True, False] precompute: [True, False]	Alpha: 3.01 Fit_intercept: False Normalize: False Precompute: True
Elastic net regression	alpha: [0.01, 1.01, 2.01, 3.01, 4.01, 5.01, 6.01, 7.01, 8.01, 9.01, 10.01] l1_ratio: np.arange(0.01, 0.999, 0.25) fit_intercept: [True, False] normalize: [True, False] precompute: [True, False]	Alpha: 5.1 L1_ratio: 0.51 Fit_intercept: False Normalize: False Precompute: True
Poisson regression	Use l1 regularization. The python library “statsmodels” finds best parameters while training.	

Time Series Using Maintenance Events



Model	Hyperparameter set	Best hyperparameters
Fourier	number_of_harmonics: [all integers from 0 to 20]	number_of_harmonics: 1
Exponential Smoothing	trend: [‘add’, ‘mul’, None seasonal: [‘add’, ‘mul’, None] seasonal_periods: [30, 60, 90, 1] damped: [True, False]	trend: None seasonal: ‘add’ seasonal_periods: 90 damped: False

Machine Learning Using Maintenance Events



Model	Hyperparameter set	Best hyperparameters
Gradient boosting	min_child_weight: [1, 5, 10], gamma: [0.5, 1, 1.5, 2, 5], subsample: [0.6, 0.8, 1.0], colsample_bytree: [0.6, 0.8, 1.0], max_depth: [3, 4, 5], reg_lambda: [1e-2, 1e-1, 0.5, 0.99], objective: ['reg:squarederror', count:poisson']	Best Lag: 39 Min_child_weight: 1 Gamma: 5 Subsample: 0.6 Max_depth: 5 Reg_lambda: 0.5 Objective: "reg:squarederror"
Ridge regression	alpha: [1, 0.1, 0.01, 0.001, 0.0001, 0] fit_intercept: [True, False] solver: ['auto', 'svd', 'cholesky', 'lsqr', 'sparse_cg'] normalize: [True, False]	Best lag: 23 Alpha: 1 Fit_intercept: True Solver: "sparse_cg" Fit_intercept: True
Random forest	bootstrap: [True, False], max_depth: [5, 10, 20, 40, 60, 80, 100, None], max_features: ['auto', 'sqrt'], min_samples_leaf: [1, 2, 4], min_samples_split: [2, 5, 10], n_estimators: [200, 400, 800, 1200, 1600, 2000]	Best Lag: 28 Bootstrap: False Max_depth: 80 Max_features: 'sqrt' Min_samples_leaf: 1 Min_samples_split: 2
k-NN	n_neighbors: [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15] weights: ['distance', 'uniform'] algorithm: ['auto', 'ball_tree', 'kd_tree', 'brute'] leaf_size: [20-30] p: [1, 2, 3, 4, 5]	Best Lag: 8 N_neighbors: 5 Weights: "distance" Algorithm: "ball_tree" Leaf_size: 32 P: 3
LASSO regression	alpha: [0.01, 1.01, 2.01, 3.01, 4.01, 5.01, 6.01, 7.01, 8.01, 9.01, 10.01] fit_intercept: [True, False] normalize: [True, False] precompute: [True, False]	Best Lag: 25 Alpha: 2.01 Fit_intercept: True Normalize: False Precompute: True
Elastic net regression	alpha: [0.01, 1.01, 2.01, 3.01, 4.01, 5.01, 6.01, 7.01, 8.01, 9.01, 10.01] l1_ratio: np.arange(0.01, 0.999, 0.25) fit_intercept: [True, False] normalize: [True, False] precompute: [True, False]	Best Lag: 16 Alpha: 0.1 L1_ratio: 0.51 Fit_intercept: False Normalize: True Precompute: False
Kernel-ridge regression	alpha: [0.1-15] in 0.5 increments kernel: ['linear', 'laplacian', 'rbf', 'polynomial', 'sigmoid'] degree: [2, 3, 4]	Best Lag: 31 Kernel: 'linear'

Model	Hyperparameter set	Best hyperparameters
CNN	# of Convolutional Layers: [1, 2, 3] # of feed-forward layers: [1, 2, 3] # of nodes, Convolutional Layer 1: [512, 256, 128, 64] Convolutional Layer 1 kernel size: 4 # of nodes, Convolutional Layer 2: [32, 64, 128] Convolutional Layer 2 kernel size: 3 # of nodes, Convolutional Layer 3: [16, 32] Convolutional Layer 3 kernel size: 3 # of nodes, feed-forward Layer 1: [1000, 500, 400, 300, 100] # of nodes, feed-forward Layer 2: [400, 200, 100] # of nodes, feed-forward Layer 3: [150, 50] Dropout rate: [None, 0.05, 0.01, 0.25] Optimizer: ['SGD', 'RMSprop', 'Adagrad', 'Adadelta', 'Adam', 'Adamax', 'Nadam']	Best Lag: 27[a] # of Convolutional Layers: 2 # of feed-forward layers: 2 # of nodes, Convolutional Layer 1: 128 # of nodes, Convolutional Layer 2: 64 # of nodes, feed-forward Layer 1: 400 # of nodes, feed-forward Layer 2: 200 Dropout rate: None Optimizer: 'Adam'
RNN	# of Recurrent Layers: [2, 1] # of Feed-Forward Layers: [1, 2, 3] Convolutional Layer 1 kernel size: 4 # of nodes, Recurrent Layer 1: [32, 64, 128] # of nodes, Recurrent Layer 2: [16, 32] # of nodes, feed-forward Layer 1: [1000, 500, 400, 300, 100] # of nodes, feed-forward Layer 2: [400, 200, 100] # of nodes, feed-forward Layer 3: [150, 50] Dropout rate: [None, 0.05, 0.01, 0.25] Optimizer: ['SGD', 'RMSprop', 'Adagrad', 'Adadelta', 'Adam', 'Adamax', 'Nadam']	Best Lag: 19[a] # of Recurrent Layers: 1 # of Feed-Forward Layers: 2 # of nodes, Recurrent Layer 1: 128 # of nodes, feed-forward Layer 1: 400 # of nodes, feed-forward Layer 1: 200 Dropout rate: None Optimizer: 'Adam'

[a] For neural networks, we only tried every lag for one pre-defined model and then grid searched both the max lag (40) and the one found to be best during this initial loop. This was done due to the computational time involved in running neural networks.

Appendix C
Recommendations for Future R&D Projects

We recommend DLA consider the following projects for their potential in the improvement of forecasting customer needs.

Predict Maintenance Events and Associated Usage BOMs

Objectives

Use ML to identify and predict discrete maintenance events and generate the BOM required for each event.

Problem Description

Rather than awaiting wholesale replenishment requisitions, DLA can generate a parts forecast using ML to predict specific weapon systems maintenance requirements.

Benefits

Visibility into predicted maintenance requirements provides the opportunity to ensure all required parts and materiel are available at the time needed. This will result in more efficient maintenance (no awaiting parts), increased collaboration as the parts requirements can be viewed across all levels of inventory, and increased platform availability.

Customer

DLA planning community.

Technical Concept and Approach

Using the data available in MADW™ and applying ML, LMI can produce a usage BOM to be used for the predicted maintenance events. The comprehensive data in the MADW™ contain many quantitative and qualitative fields that describe the objects maintained on the weapon system, cost and availability metrics, and availability loss for the maintenance event. Predicting maintenance events by Service WUCs can potentially provide improved parts forecasting for DLA:

- Use ML to forecast the O-level and I-level maintenance requirements for the airframe.
- Use ML to identify a BOM for each identified maintenance event by WUC.
- Use this BOM to forecast parts required for the maintenance availability.

Team Member Roles

DLA R&D will be the PM for this effort. The DLA Center of Planning Excellence will be the functional experts. LMI will provide ML and supply chain subject matter experts and execute the contract.

Predict Changes in Maintenance Requirements

Objectives

Use ML to identify and predict changes in maintenance requirements.

Problem Description

Current DLA forecasting models rely on historical demand data without regard to changes in requirements due to changes in airframe construct, updates to components, or simply age.

Benefits

Visibility into predicted maintenance requirements offers the opportunity to ensure all required parts and materiel are available at the time needed. This will result in more efficient maintenance (no awaiting parts), increased collaboration as the parts requirements can be viewed across all levels of inventory, and increased platform availability.

Customer

DLA planning community.

Technical Concept and Approach

Using the data available in the MADW™ and applying ML, LMI can identify changes in maintenance events. The comprehensive data in the MADW™ contain many quantitative and qualitative fields that describe the objects maintained on the weapon system, cost and availability metrics, and availability loss for the maintenance event. Predicting maintenance events by Service WUCs can potentially provide improved parts forecasting for DLA:

- Identify changes in maintenance patterns discovered in recent maintenance events.
- Automatically alert DLA planners to intervene and discontinue using the now incorrect statistical forecasts.

Team Member Roles

DLA R&D will be the PM for this effort. The DLA Center of Planning Excellence will be the functional experts. LMI will provide ML and supply chain subject matter experts and execute the contract.

Use ML to Improve Readiness and Acquisition Using Consumption Data

Objectives

Use consumption data available at the retail level and apply ML to improve airframe overall readiness/availability and assist in the acquisition strategy development.

Problem Description

Currently, DLA only sees wholesale requisitions and creates its acquisition strategy on the basis of these data.

Benefits

Visibility into actual consumption data supports a more accurate understanding of parts usage and potential changes in usage. This will assist DLA in the development of the acquisition strategy by providing a more current and granular picture of item use, resulting in a greater ability to anticipate and react to end user needs.

Second tier benefits will include more efficient stocking (reduced over-stocking and tying up of acquisition funds) and more effective stocking (reduced out-of-stocks).

Customer

DLA planning community.

Technical Concept and Approach

Feeding curated MADW™ consumption data into DLA's existing models may improve those forecasts. Use of consumption data is a best practice in the private sector. Until now, consumption data have been unavailable to DLA.

To assess using these data, we will employ LMI's FINISM™. LMI and DLA R&D jointly developed FINISM™ for use on prior projects. It simulates DLA's existing forecasting models and is a good tool to evaluate how forecasts do (or do not) improve using consumption rather than wholesale supply data:

- Identify ML to curate actual parts consumption data for O/I-level maintenance.
- Feed the consumption data into the models DLA uses for parts forecasts.
- Repeat using the demands that DLA normally uses to drive its forecasts.
- Use ML to identify which populations perform better using consumption data.

Team Member Roles

DLA R&D will be the PM for this effort. DLA Center of Planning Excellence will be the functional experts. LMI will provide ML and supply chain subject matter experts and execute the contract.

Dynamically Train ML Models

Objectives

Leverage service maintenance data to improve DLA parts forecasts, using exploratory data analysis (EDA) to identify change points potentially impacting usage.

Problem Description

Currently, DLA relies on historical demand when developing forecasts. This does not account for changes in maintenance due to operational tempo, changes in maintenance requirements, changes in components, or age of end use platform. Even if an accurate model is identified for use in forecasting, change points may significantly alter the accuracy of that model.

Benefits

Identifying change points and dynamically choosing models over time can improve accuracy by reducing response time to actual events.

Customer

DLA planning community.

Technical Concept and Approach

Dynamically choosing models over time can help improve accuracy. Utilizing change points can be helpful if not many reside within the data and lag times are insufficient to cause problems. Evaluating models on the basis of past performance is also a viable solution, but it can be time consuming.

MADW™ provides operational-level demand data for individual parts over time. These data furnish consumption information not found in the current DLA wholesale requisitioning data. Using data from both of these systems in concert with other available data (platform operational and DLA acquisition data) renders a more holistic picture of parts support and a subsequent improvement in forecasting for each layer of inventory.

Team Member Roles

DLA R&D will be the PM for this effort. The DLA Center of Planning Excellence will be the functional experts. LMI will provide ML and supply chain subject matter experts and execute the contract.

Use Predictive Simulation Modeling to Determine Future Part Demands

Objectives

Leverage service maintenance data in the development of improved DLA part forecasts, using predictive modeling employing an asset-focused, high-resolution simulation,

Demand Pro, to balance inventory levels beyond the limitations of statistical models and traditional forecasting.

Problem Description

Currently, DLA relies on historical demand when developing forecasts. This does not account for changes in maintenance due to operational tempo, changes in maintenance requirements, changes in components, or age of end use platform. Even if an accurate model is identified for use in forecasting, change points may significantly alter the accuracy of that model.

Benefits

Demand Pro has the following benefits:

- *Asset-focused*. The Demand Pro simulation platform models each individual asset in a fleet or each weapon system in a service inventory.
- *High-resolution*. Maintenance, supply, logistics, and operations are detailed for each asset. Five levels of indenture are modeled in the asset structure.

This platform has been applied to predict future part requirements at the operational level as well as intermediate and depot levels. Demand Pro delivers a capability that far surpasses forecasting tools. Through past performance, LMI has demonstrated the accuracy and depth of its predictive simulation platform and prescriptive analytics vs. widely used traditional forecasting methods.

Customer

DLA planning community.

Technical Concept and Approach

Use an asset-focused, high-resolution simulation capability to model each asset in a population as follows:

- Conduct EDA on maintenance data to identify change points.
- Evaluate model performance after the change point is identified.
- Take one of two options:
  - Select the most accurate model on the basis of this evaluation.
    - Pros
      - Only need one active model at a time.
      - Historical predictions do not have to be stored.
      - Clear delineation between old and new.
    - Cons
      - Lag time impacts currency of the model.
      - If change points happen frequently, the model may not be able to maintain accuracy.

- Evaluate models on the basis of past performance.
    - Pros
        - Minimizes lag time.
        - Everything is adjustable.
        - Weights data as required.
        - Potentially detects patterns over time and switches to the best model for the pattern.
    - Cons
        - More involved to determine parameters that constitute a model change.
            - How good does the model have to be compared with the others?
            - How long must it sustain its lead?
        - More storage and process time is required to maintain.

Team Member Roles

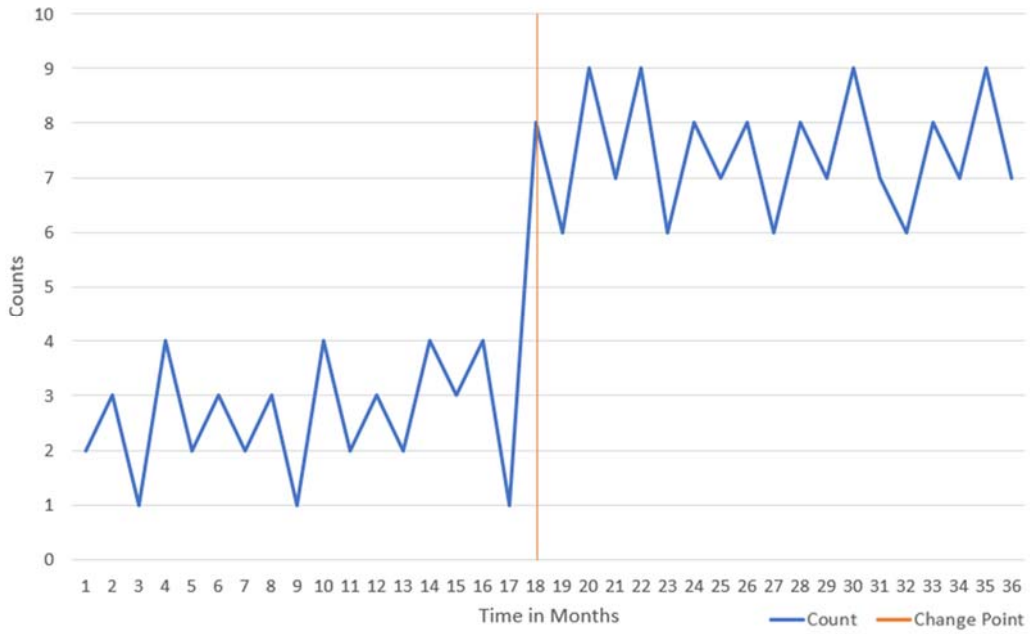
DLA R&D will be the PM for this effort. The DLA Center of Planning Excellence will be the functional experts. LMI will provide ML and supply chain subject matter experts and execute the contract.

Appendix D
Dynamic ML Training Models

Problem Statement

EDA on the maintenance action data found three change points. These change points can be the result of real-life events, including changes in fleet activity, weather patterns, and aging fleets. With statistically different sections of data separated by these change points and potential reasoning for each change point, using different models on different time periods of the data will lead to better forecasts. We propose using change points to improve current modeling with an automated approach to choosing training data and models (Figure D-1).

Figure D-1. Maintenance Events Over Time



Proposed Solutions

Reselect Model after Change Point

This method would train a new model after change point detection. There would need to be a lag time in order to collect new data on which to train. For different aggregations, lag times will cover larger periods of information. For example, if you want five samples on which to train, depending on your aggregations, it could be 5 days, 5 weeks, or 5 months.

Pros

This approach has the following pros:

- Only needs one active model at a time.
- Historical predictions do not have to be stored.
- Has clear delineation between old data and new. The data scientist can decide how to weight/discard data for training.

Cons

It has the following cons:

- Lag time on training makes it tough to keep models current.
- If change points happen often enough, the models would not be able to keep up.

Evaluate Models on the Basis of Past Performance

This method would keep all models active. All model predictions would be stored along with the actual results (Table D-1). Accuracy would be measured over time, and a new model will be selected if it outperforms the others for a long enough time.

Table D-1. Example Schema

Date	Model A pred.	Model B pred.	Model C pred.	Actual	A %Err	B %Err	C %Err
Jan-19	8	6	7	7	14.28571	−14.2857	0
Feb-19	6	4	10	5	20	−20	100
Mar-19	10	8	9	9	11.11111	−11.1111	0
Apr-19	11	11	5	10	10	10	−50
May-19	9	8	4	8	12.5	0	−50
Jun-19	13	11	12	12	8.333333	−8.33333	0
Jul-19	14	15	6	14	0	7.142857	−57.1429
Aug-19	15	16	12	16	−6.25	0	−25
Sep-19	13	12	20	13	0	−7.69231	53.84615
Oct-19	16	13	5	15	6.666667	−13.3333	−66.6667

A is almost always over predicting by one, B is usually predicting under the correct amount. C has the best rate of getting 100 percent, but also has the most variance away from the mean. Depends on customer needs.

Pros

This approach has the following pros:

- Minimizes lag time on training. Able to keep up with current data.
- Everything is adjustable.

- Can include change point data to weight the current state more heavily.
- Could potentially detect patterns over time and switch to the best model for the pattern.

Cons

It has the following cons:

- More involved to determine parameters that constitute a model change.
  - How good does the model have to be compared with others?
  - How long must it sustain its lead?
- More storage and processing time required to maintain.

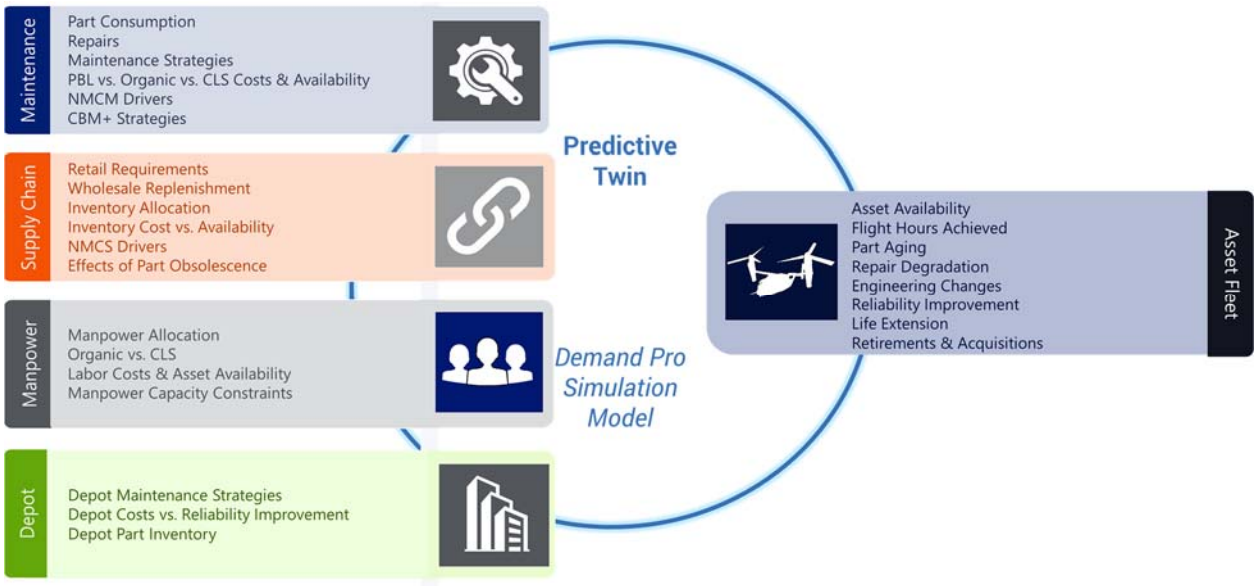
Conclusion

Dynamically choosing models over time can help improve accuracy. Utilizing change points can be helpful if few exist within the data and lag times are not big enough to cause problems. Evaluating models on the basis of past performance is also a viable solution, though it can be time consuming. Overall, it depends on the customer and the project timeline.

Appendix E
Examples of Simulation-Driven Inventory Demand Predictions

Predictive simulation modeling with Demand Pro provides the holistic approach needed to capture the relationships between maintenance actions at the asset level, part consumption, retail requirements, and wholesale replenishment as summarized in Figure E-1. Model elements from the simulation modeling are listed and can be applied to predictive analysis according to the desired study objectives.

Figure E-1. Leveraging Predictive Simulation Modeling



CLS = contract logistics support; NMCM = not mission capable maintenance; NMCS = not mission capable supply; PBL = performance-based logistics.

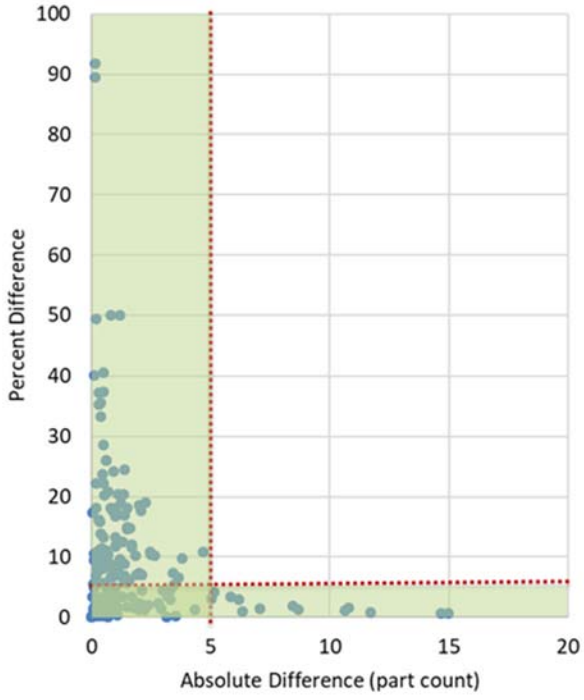
Ground Vehicle Fleet 1: Within 5% or Five Total Parts

A Demand Pro model of this asset fleet was developed and validated against observed part orders. Accurate simulated part counts will result in either a small percentage difference or small absolute difference between historical and predicted orders for each component. Parts with many orders are expected to have a small percentage difference between historical and predicted parts orders; however, the parts might still have a large absolute difference. For example, the tow bar had 2,158 orders accumulated in 1999–2017 and 2,173 orders predicted in the Demand Pro model operating the same number of fleet miles. The difference is 15 parts ordered but less than 1 percent. On the other hand, parts with a low number of orders historically are expected to predict a low number of orders but result in a high percentage difference between historical and predicted parts ordered. For example, the manifold intake has one order in 1999–2017 and a

predicted average of 1.4 orders in the Demand Pro model, demonstrating a small absolute difference of less than one but a large percent difference of 40 percent.

In Figure E-2, both the percentage difference and absolute difference between historical and predicted part orders are plotted. The customer set a maximum threshold of 5 percent difference or five parts for absolute difference define the regions shaded in green. All parts usage predictions fall within this threshold.

Figure E-2. Parts Ordered Validity Plot from the Demand Pro Model for Asset Fleet 1



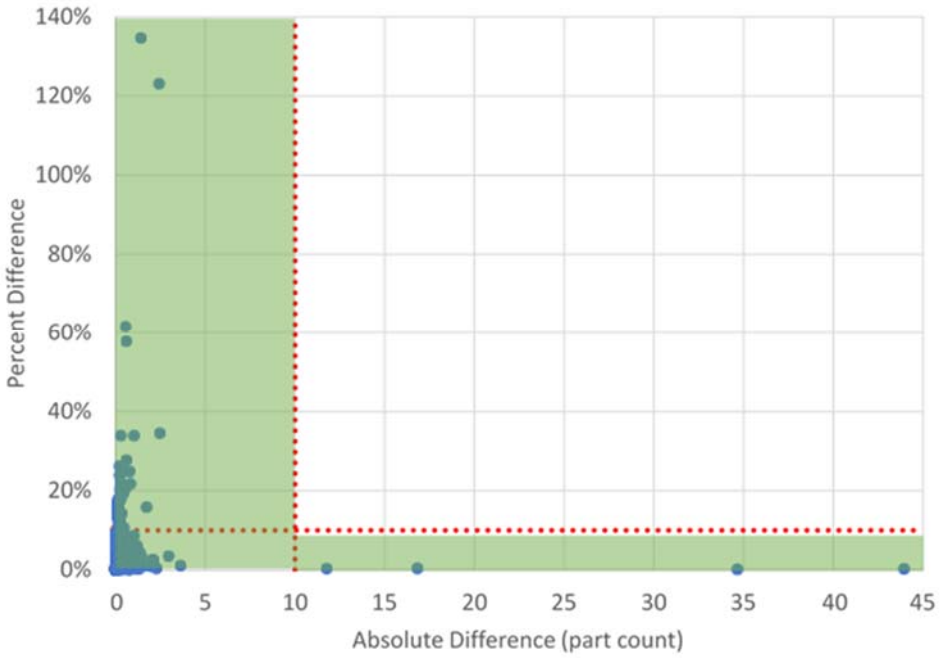
Ground Vehicle Fleet 2: Within 3% or Three Parts

Part orders by component are counted in the model output and compared with historical parts ordered in maintenance records. The number of predicted part orders is related to failure rate and operational profiles in the model. Parts with many orders are expected to have a small percentage difference between historical and predicted parts orders; however, the parts might still have a large absolute difference. For example, the track shoe assembly has 23,556 orders from February 2012 to July 2017, and 23,591 orders predicted in the Demand Pro model during the same period. The difference is 35 parts ordered but less than 1 percent. On the other hand, parts with a low number of orders historically are expected to predict a low number of orders, but a high percentage difference between historical and predicted parts ordered. For example, the skate mount assembly has one order from February 2012 to July 2017 and has a predicted average of 1.6 orders in the Demand Pro model, demonstrating a small absolute difference of less than one but a large percent difference of 60 percent.

Both the percentage difference and absolute difference between historical and predicted part orders are plotted in Figure E-3. The customer set a maximum threshold of

10 percent difference or 10 parts for absolute difference define the regions shaded in green. All parts usage predictions fall within three absolute parts or 3 percent difference.

Figure E-3. Parts Ordered Validity Plot from the Demand Pro Model for Asset Fleet 2

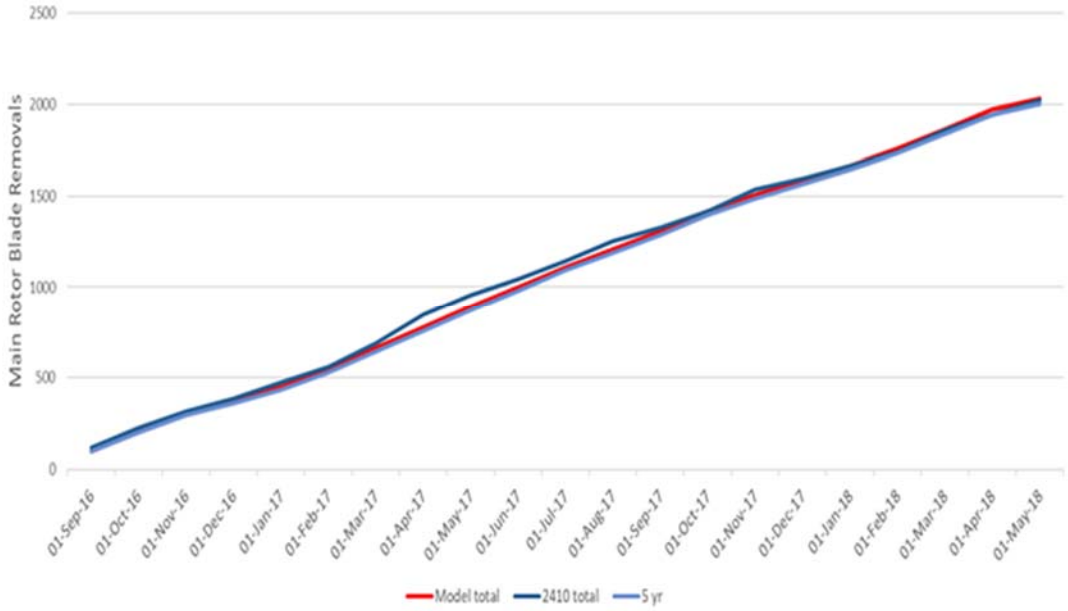


Aviation Fleet 3: Within 1% Accuracy

The results for fleets 1 and 2 inventory demand predictions came from analysis of ground vehicle fleets. These examples included sparse data sets without information on all the maintenance transactions on tracked parts. Fleet 3 is an aviation fleet with a robust transactional data base. The millions of maintenance records provided for this fleet are conditioned and fed into a reliability-centered maintenance analysis, which produces failure distributions customized to the degree of resolution supported by the data. Custom failure distributions may be developed for part numbers on specific aircraft variants, operating at a specific location, with a specified number of previous repairs. These factors are used to improve the accuracy of the failure distributions used in the predictive simulation.

From this improved model, supported by robust data, even more accurate results are produced. Figure E-4 plots actual main rotor blade removals in dark blue. Simulation predictions using customized failure distributions generated from 5 years of historical data are plotted in light blue. Similar simulation predictions wing 10 years of historical data are plotted in red. Both sets of predictions fall within 1 percent of the observed part removals.

Figure E-4. Actual vs. Model Prediction for Rotor Blade Removals



Predictive Simulation Modeling Tools and Methods

LMI's Demand Pro platform employs discrete event simulation along with proprietary tools and techniques in the modeling and analysis effort. Together with the expertise of LMI's data scientists, the predictive analysis software provides detailed, accurate, time-dependent insights.

LMI data scientists employ this asset-focused, high-resolution simulation platform to model vehicle fleet operations and sustainment. The simulation represents each asset in the fleet to predict future inventory requirements, readiness, and cost. Every maintenance event is captured: part failures, removals, inspection, along with tearing down components into subparts and rebuilding them. Consumption of supply along with part shipments and their costs are captured as simulation events. Each individual asset operation is modeled along with component failures that occur during the operation. The simulation scenarios used analysis periods that can range from a few weeks or months to years or decades. These scenarios model each discrete event encountered by an asset, including part failures, repairs, maintenance actions, shipments, and inventory status.

To develop Demand Pro simulation models, LMI evaluates available data and data sources for relevancy in support of model development. LMI leverages 20 years of experience with Demand Pro, applying varied DoD supply and maintenance data to support rapid modeling and analysis. To enhance development of Demand Pro models, LMI leverages the MADW™: a decision support tool that integrates and stores maintenance and availability data for equipment, weapon systems, infrastructure, and facilities across DoD Services for each level of maintenance (field, intermediate, depot), provider (organic, commercial) and nature of cost (labor, materials). Records within the MADW™ are cleaned, standardized, resolved, and reconciled. LMI also applies Studio—an extract, transform, and load analysis and visualization platform—to facilitate rapid

baseline model development. LMI applies a repeatable process for conducting simulation-based studies and analysis.

LMI's Demand Pro models include details from many life-cycle management perspectives: platform configuration, equipment and weapons system performance data, fleet size and composition, reliability, maintainability, supply capability and capacity, logistics constraints, task times, and ongoing programmatic issues, including upgrades, reset, retirements, battle loss, service life extension programs, operations tempo, and aging and degradation. Demand Pro can model programs experiencing data sparsity such as new equipment and weapon systems and programs without robust data collection. Likewise, Demand Pro can ingest volumes of input data from data-rich programs such as aviation fleets with highly detailed transactional maintenance data.

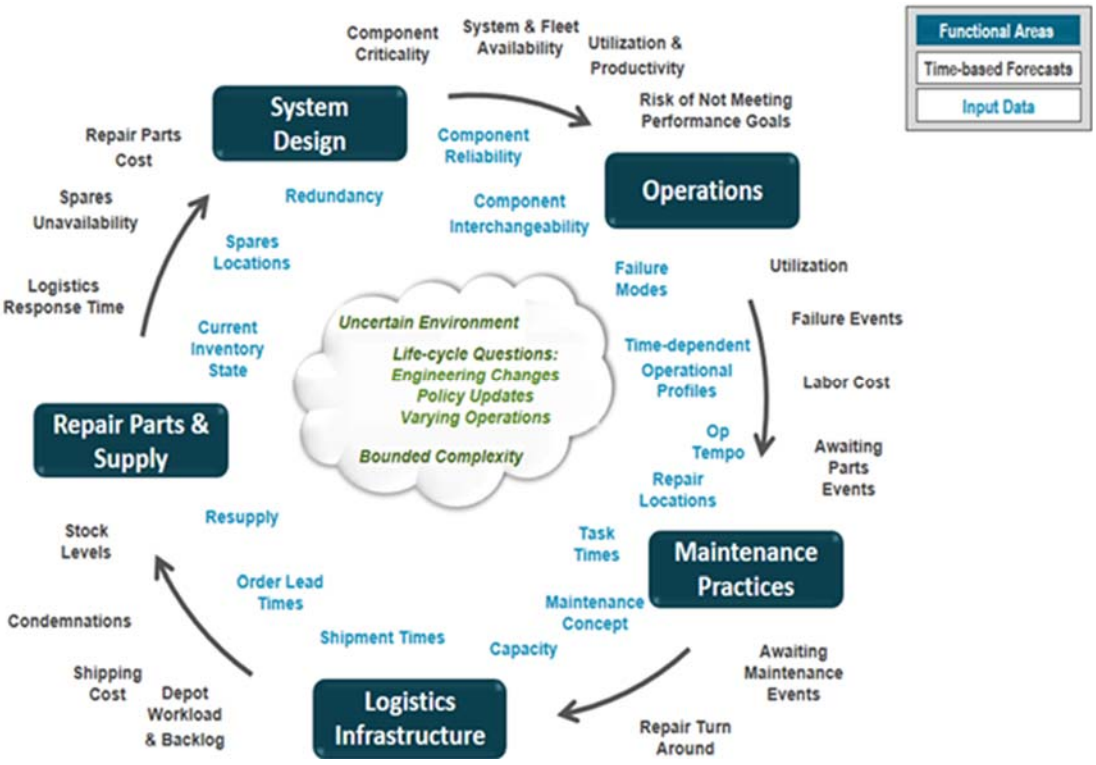
LMI develops comprehensive asset maintenance models spanning fleet and weapons system populations with serialized platforms and indentured BOM components. This model serves as the baseline model for use in determining baseline performance and cost. The baseline model is modified to develop any predictive analysis models used in modeling alternatives.

Once baseline Demand Pro models are developed for an asset fleet, use of the simulation model extends well beyond inventory predictions. LMI's predictive modeling services and technology enables customers to leverage predictive models for capital-intensive fleets to meet key objectives:

- Balance future life-cycle inventory levels as a function of time while controlling cost.
- Measure the effects of uncertainty in terms of future performance and cost.
- Measure and evaluate the drivers of future repair and maintenance cost-per-unit-mile as a function of time.
- Integrate distinct data analysis products and tools—without overlapping existing efforts.
- Predict the future effects of equipment aging.
- Predict future effects of part obsolescence as a function of time in terms of life-cycle cost and asset performance.
- Summarize asset status as a function of time in terms of future performance and cost.

Figure E-5 shows Demand Pro's holistic concept. This asset-focused simulation platform encompasses the operational and logistical aspects of the weapon system fleet and their interrelationships. Each platform's configuration and operational profile are detailed along with component reliability. Maintenance concepts, supply chain elements, and sustainment strategies are also included for a complete, interacting, time-dependent depiction of an asset's life cycle.

Figure E-5. Demand Pro Holistic Concept



Using the input data shown in blue, the Demand Pro model generates many predicted, future metrics, including the following:

- Inventory requirements, by part number, over time and by location
- Spare parts unavailability
- Logistic delays
- Operational availability
- Planned and unplanned maintenance
- Achieved operating hours
- Mean time between failures
- Repairs and condemnations at operating and depot echelons
- Life-cycle costs.

LMI has developed Demand Pro predictive models and delivered life-cycle sustainment analyses for ground programs including the AAV, ITV, HMMWV, M1A1, M1A2, M88, LAV, MTVR, LVSR, MPC, M9 ACE, EFV, M-ATV, Cougar, Buffalo, M777, and TQGs. Aviation Demand Pro models delivered by LMI include the UH-60, AH-1, OH-58, CH-47, and V-22.

Appendix F
Abbreviations

ADF	augmented Dickey-Fuller
AFAST	Aviation Financial Analysis Tool
ANOVA	analysis of variance
BOM	bill of materiel
CBM+	Condition Based Maintenance Plus
CLS	contract logistics support
CNN	convolutional neural network
DBSCAN	density-based spatial clustering of applications with noise
DLA	Defense Logistics Agency
DECKPLATE	Decision Knowledge Programming for Logistics Analysis and Technical Evaluation
DoD	Department of Defense
EDA	exploratory data analysis
EWMA	exponential weighted moving average
FFNN	feed forward neural network
FINISM™	Financial and Inventory Simulation Model™
GCU	Generator Converter Unit
GMM	Gaussian mixture model
GSS	Global Strategic Solutions, LLC
IID	independent and identically distributed
k-NN	*k*-Nearest Neighbors
LASSO	least absolute shrinkage and selection operator
LDA	linear discriminant analysis
MADW™	Maintenance and Availability Data Warehouse™
MAE	mean absolute error
ML	machine learning
MSE	mean squared error
MVP	minimum viable product
NIIN	National Item Identification Number
NMCM	not mission capable maintenance

NMCS	not mission capable supply
PBL	performance-based logistics
PIS	period in stock
PM	program manager
P-R	precision-recall
R&D	research and development
RCB	Reliability Control Board
RNN	recurrent neural network
SD	standard deviation
SE	standard error
ServiceWBS	Service work breakdown structure
TMS	Type/Model/Series
TWG	technical working group
WUC	work unit code

Forecasting Parts Demand Using Condition-Based Maintenance Data and Machine Learning

Volume 2

Daniel A. Hutchinson, Global Strategic Solutions, LLC

J. Luis Hernandez, Global Strategic Solutions, LLC

January 2020

NOTICE:

THE VIEWS, OPINIONS, AND FINDINGS CONTAINED IN THIS REPORT ARE THOSE OF LMI AND SHOULD NOT BE CONSTRUED AS AN OFFICIAL AGENCY POSITION, POLICY, OR DECISION, UNLESS SO DESIGNATED BY OTHER OFFICIAL DOCUMENTATION.

LMI

Forecasting Parts Demand Using Condition-Based Maintenance Data and Machine Learning

January 2020

Executive Summary

Forecasting demand for repair parts to support weapon systems is a challenging task for the Defense Logistics Agency (DLA) and the Services that requires advanced analytics. The demand for repair parts varies greatly, maintenance patterns change, and multiple inventory levels shroud true consumption patterns. Through research and development, DLA wants to create and test a minimum viable product using machine learning (ML) techniques. LMI conducted this project by exploring the application of these techniques on historical data from Service maintenance records and a Condition Based Maintenance Plus (CBM+) program to forecast parts demand. Improved forecasts will enable DLA to better manage the supply chain, enhancing support to retail customers.

Traditionally, DLA has based parts forecasts on historic supply demands and the application of statistical models. These techniques do not account for the actual use (operating environment), actual weapon system reliability, and ad-hoc business processes used by the Services to meet operational demands and tackle readiness issues. As CBM+ becomes the norm for all Services, weapon system maintenance patterns will change and impact supply support. Using Service maintenance records and CBM+ data, with advanced analytics, offers the opportunity to improve forecasting and supply support for weapon systems.

This volume presents an approach using Service CBM+ data from multiple sources to establish the foundation for developing an artificial intelligence (AI)/ML model that best fits the data currently available from the Services and the problem addressed by this project. The method employed by Global Strategic Solutions (GSS) considers the current state (across the Services) in implementing CBM+ and other digital transformation initiatives. We wanted to explore the practical implications of using ML on CBM+ data, in the context of the DLA operational use case for improving the parts forecasting and planning process.

As defined in Department of Defense (DoD) Instruction 4151.22, CBM+ is a collaborative DoD readiness initiative focused on the development and implementation of data analysis and sustainment technology capabilities to improve weapon system availability and achieve optimum costs across the enterprise. CBM+ turns rich data into information about component, weapon system, and fleet conditions to more accurately forecast maintenance requirements and future weapon system readiness to drive process cost efficiencies and enterprise activity outcomes. The DoD CBM+ policy endorses proactive equipment maintenance using system health indications to predict a functional failure

ahead of the event and take appropriate preemptive actions. Adding to the health indications, is the potential use of historical operational, sustainment, and repair data to generate useful information and drive enterprise activity outcomes. The preemptive actions include more efficient maintenance planning, efficient and effective maintenance, and proactive supply support planning. This perspective is used in exploring the use of AI/ML on CBM+ data to improve parts forecasting. Given this perspective, CBM+ data types include the following:

- Onboard and offboard aircraft health management data; system built-in-test (BIT), sensor data, flight profile, health condition indicators, etc.
- Operational, sustainment process, and repair analysis data; curated data sets linking operational performance issues and health state data to data collected throughout the sustainment process, including testing and repair at the intermediate and depot/original equipment manufacturer (OEM) levels of maintenance.
- Planned maintenance event information; information on life limited parts, scheduled maintenance events, planned equipment upgrades, etc.

The analysis process employed by GSS included meetings and data exchange with the U.S. Navy (Naval Air Systems Command) and DLA (Center for Planning Excellence, DLA Aviation) stakeholders to gain more insight into the issues, understand the data available for the project, establish a solution hypothesis, and characterize the ML modeling technique that best fits the problem and the available data. Based on the analysis, we hypothesized that *"the Services could employ advanced analytics techniques—using CBM+ operational, maintenance event, and depot repair data—to obtain advance signals (proactive supply alert identifiers) of part demand and point-of-use data for DLA through formal collaboration channels."*

With approval from DLA, the project analyzed the F/A-18 Generator Converter Unit (GCU), which the F/A-18E/F Reliability Control Board classified as a top reliability degrader, with a high not mission capable supply rate. It should be noted that the GCU itself does not have sensors to monitor its health condition other than BIT which is monitored by the aircraft mission computer. Analysis of the GCU data obtained from the Navy, resulted in the following insights:

- For the GCU, the actual health management data is limited to diagnostics (i.e., BIT). From a parts forecasting perspective, it provides the foundation for flagging a degrader component. Tracking and computing the actual component mean time between failures (MTBF) or mean flight hours between failures (MFHBF) and comparing the metrics with the design MTBF/MFHBF provides an early warning of a "reliability degrader."
- Combining weapon system data with GCU maintenance event data: test results, Shop Replaceable Assembly (SRA) callouts, narrative (corrective action, action taken description, and malfunction description), and the GCU and SRA depot repair data is the most promising approach for using ML on CBM+ data to improve the DLA parts forecast and planning process. The idea is to use advanced analytics to identify influential supply demand features or proactive supply alert identifiers (PSAIs). These features can then be used to provide a more efficient means for notifying DLA planners when demand patterns have changed or are changing due to actual use (operating environment), reliability, or

ad hoc business processes used by the Services to tackle operational demands and readiness issues.

- For the GCU, there is an ongoing effort to replace or convert older units (G2/G3) to the latest G4 version. The schedule and rate of conversions at the depot or OEM can be used to develop projected demand signals for parts supplied by DLA.

GSS applied the CBM+ method described in this volume without determining a specific model or set of models to address forecasting improvements. Following our hypothesis, we found that this method can be used to generate a list of PSAIs that can be integrated into the existing DLA forecasting and planning models to define the recommended operation framework to exchange data more efficiently. The findings of the analysis and recommended PSAI classes are detailed in Chapter 4 of this volume.

Using this approach, DLA planners can use the constant streams of PSAI class data to identify demand indicators and trends in projected parts demand. Services could supply to DLA degradation trends, soft-threshold trends, and revised maintenance schedules derived from the CBM+ data analysis. DLA parts planning could use this additional information about Service part repair demand levels to automate collaboration and development of future demand projections. This approach can be used by DLA and the Services to improve forecasting and supply support for DoD weapon systems.

It should be noted that, in executing CBM+ within a weapon system, the Services employ data from multiple data sources throughout their operational and sustainment operations. While data connectivity, data quality, and curation continue to be difficult challenges for the Services, they are moving forward in implementing CBM+. Therefore, with respect to the use of CBM+ data and AI/ML to improve parts forecasting, GSS makes the following recommendations as the next course of action for DLA:

1. Collaborate with the Services in developing the recommended PSAI classes described in Chapter 4, Table 4-1, of this volume. The PSAI classes would serve as a stream of data to be analyzed and sent from each Service to DLA to render a more complete picture of demand for DLA-provided parts. Each PSAI class of information would show trends and a timely picture of the actual condition driving supply support issues with DLA supported weapon systems. The PSAI classes and information would identify top degraders and root causes, reliability and monitoring, logistics life limited parts, service actual usage, and weapon system maintenance scheduling.
2. Collaborate with Services in Automating the generation of PSAI classes of information and the stakeholder collaboration process. Using Services CBM+ data, their integrated data environment, and advanced analytics tools; the Services can derive data streams of PSAI information for DLA parts projection. The Services PSAI classes of data streams can be used by DLA to generate ML models of actual part usage with current repair levels of demand. These models can help identify changes in low demand, not seen on the bill of material (BOM), to high demand parts as needed. With the PSAI classes of data streams from the Services, TensorFlow models can be used by DLA planning to adjust current DLA planning models with actual usage. Automated exchange of actual repair information in addition to BOMs give DLA a more complete picture of Service DLA National Item Identification Number (NIIN) actual usage and demand. ML

algorithms can be developed in collaboration with the Service PSAI data points for specific NIINs to monitor Service trends and adjust DLA parts forecasting and planning.

3. Collaborate with a specific Service and Weapon System Program Office (e.g., F/A-18) to develop a Visualization App of PSAI information for a specific weapon system component (e.g., GCU) for stakeholder collaboration. The idea is to pilot the use of ML analysis against PSAI classes of data to develop neural network perception points of demand signals. Using data analysis and visualization tools, these perception points of demand signals can be merged to develop an overall state of supply health for the GCU NIIN. This would require the development and integration of multiple models, as described in Chapter 4, Figures 4-4 and 4-5.

Contents

Preface ..... **ix**
Chapter 1 Introduction **1-1**
Background 1-1
DLA Needs and Benefits 1-1
Chapter 2 CBM+ Method **2-1**
CBM+ 2-1
CBM+ Data Types 2-1
ML Modeling Technique 2-2
Chapter 3 Technical Concept and Approach **3-1**
Source Data 3-1
Aircraft Subsystem Health Monitoring Data 3-2
Operational, Sustainment Process, and Repair Data 3-3
Planned Maintenance Information 3-4
Chapter 4 Findings and Recommendations **4-1**
Findings 4-1
Recommendation for use of CBM+ Data to Improve Parts Forecasting 4-1
PSAI Classes 4-2
Top Degrader and Root Causes 4-2
Reliability and Monitoring 4-3
Logistics Life Limited Parts 4-3
Service Actual Usage 4-4
Weapon System Scheduling 4-4
Weapon System Health Monitoring 4-4
Recommendation for Automating PSAI Generation and Stakeholder Collaboration 4-4
Recommendation for Visualization App of PSAI Information for Stakeholder Collaboration 4-6
Appendix A CBM+ Method
Appendix B Abbreviations

Figures

Figure 2-1. Characterizing the ML Modeling Technique 2-2
Figure 3-1. F/A-18E/F CBM+ Capability Level Overview 3-2
Figure 3-2. Analysis of Operational, Sustainment Process, and Repair Data 3-3
Figure 3-3. Correlation of Data Across Sustainment Process for Specific GCU Serial Number 3-4
Figure 3-4. G3 to G4 Upgrades in 2019 3-5
Figure 4-1. DLA Business Model Using PSAIs 4-1
Figure 4-2. Navy's RCB Process Overview 4-3
Figure 4-3. CBM+ Notional Points of Presence to Process CBM+ Data and Collaboration 4-4
Figure 4-4. Service Data Processing to Create Single PSAI Neural Network Percept 4-7
Figure 4-5. Notional Application Dashboard—Quick View of Parts Status 4-7

Tables

Table 3-1. DECKPLATE Data 3-2
Table 4-1. Service PSAIs Classes Matrix 4-2

Preface

This research and development project seeks to create and test a minimum viable product using machine learning (ML) techniques on historical data from Service maintenance records to improve parts demand forecasting.

The report has two volumes:

1. Volume 1 addresses the efforts in predictive modeling using maintenance data.
2. Volume 2 discusses use of Condition Based Maintenance Plus (CBM+) data and methods.

In this volume, we highlight the rationale and key considerations behind the method used by Global Strategic Solutions, LLC (GSS), to determine whether using ML on CBM+ data can improve parts forecasting. The GSS method considers the current state across the Services in implementing CBM+ and other digital transformation technologies. The goal is to explore the practical implications of using ML on CBM+ data, in the context of the Defense Logistics Agency operational use case, for improving the parts forecasting and planning process.

Chapter 1
Introduction

The Defense Logistics Agency (DLA) wants to apply a Condition Based Maintenance Plus (CBM+) program to improve its retail customer support through better supply chain management.

This research and development (R&D) project seeks to create and test a minimum viable product using machine learning (ML) techniques on historical data from Service maintenance records to improve parts demand forecasting. The customers for this effort are DLA J3 (J31 Mission Support, National Account Managers, and J34 Process Owners), J6, DLA Land and Maritime, Troop Support, and Distribution. The Service program manager, weapon system, and associated maintenance depot are all key stakeholders.

Background

DLA manages over 2 million unique spare parts. These items are not all stocked, many have no demand, some are shipped directly from suppliers to DLA customers, and some are buy-on-demand. DLA uses two types of forecasting. The first employs statistical models using DLA historical demand data. The second initiates with the customer organization and is finalized through collaboration. Both are based on item supply data and are filtered across multiple levels of inventory. Forecasts that produce too little stock result in backorders and may decrease readiness. Forecasts that produce too much stock consume DLA acquisition funds and depot space, incurring the associated costs of maintaining inventory.

DLA does not exploit information-rich Service maintenance data to develop parts forecasts. The private sector has known the value of this point of consumption data for years. Traditional DLA supply forecasts, based on depot demand information, have errors in types, quantities, timing, and location of required parts. The inclusion of Service historical maintenance records, which have more detailed information, and CBM records, which predict future maintenance failures and subsequent parts needs, can improve the forecast of parts requirements.

DLA Needs and Benefits

The Department of Defense (DoD) is implementing CBM+ across the Services to achieve the target availability, reliability, and operation and support costs of weapon systems and components throughout their life cycles. CBM+ uses a system engineering approach to collect data, enable analysis, and support the decision-making processes for system acquisition, modernization, sustainment, and logistics operations, including supply support.

The anticipatory supply chain management method developed during this project will identify innovative ways to use ML with Service maintenance records and CBM+ data to enable proactive business processes and identify the actionable information the Service

maintenance records and CBM+ system should make available to DLA. This information will enhance DLA's wholesale support and any actions (including R&D) it needs to take to enable those improvements. The actionable information will improve DLA business processes and enable the following:

- Setting and managing the DLA supply chain support strategy for its DoD retail customers
- Improved collaboration and increased visibility through Service maintenance records and CBM+ data exchange across the supply chain
- Enhancement of the automated data exchange between Service enterprise resource planning (ERP) and DLA ERP to directly transmit parts needs associated with depot production plans to DLA
- A consistent set of business solutions that support repeatable, standardized processes for both DLA and Service managers.

DLA wants to explore the value of applying Service historical maintenance records in evaluating whether CBM analysis (which predicts future maintenance failures and subsequent parts needs) can improve the forecast of parts requirements. This R&D project seeks to answer the following question: Does analysis using Service historical data support CBM that improves parts forecasts and resulting supply support?

Chapter 2
CBM+ Method

In general, our method considers the current state across the Services in implementing CBM+ and other digital transformation technologies. We wanted to explore the practical implications of using ML on CBM+ data, in the context of the DLA operational use case, for improving the parts forecasting and planning process.

CBM+

As defined in DoD Instruction 4151.22 (pending revision), CBM+ is a collaborative DoD readiness initiative focused on the development and implementation of data analysis and sustainment technology capabilities to improve weapon system availability and achieve optimum costs across the enterprise.

CBM+ leverages reliability-centered maintenance (RCM) principles to enhance safety, increase maintenance efficiency, improve availability, and ensure environmental integrity. It diminishes life-cycle costs by reducing unscheduled maintenance and enabling predictive maintenance.

CBM+ turns rich data into information on component, weapon system, and fleet conditions to more accurately forecast maintenance requirements and future weapon system readiness to drive process cost efficiencies and enterprise activity outcomes.

In summary, the DoD CBM+ policy endorses proactive equipment maintenance using system health indications to predict functional failures and take appropriate preemptive actions. In addition, historical operational, sustainment, and repair data can be used to generate useful information and drive enterprise activity outcomes. The preemptive actions include more efficient maintenance planning, efficient and effective maintenance, and proactive supply support planning. We took this perspective in exploring the use of ML on CBM+ data to improve parts forecasting.

CBM+ Data Types

When assessing what can go wrong in a weapon system, it is intuitive to monitor the operational performance of individual components that contribute to the most critical functions. It also makes sense to focus on components with a history of failures and those that undergo accelerated usage or excessive duty cycles, have costly repair or maintenance profiles, lack redundancy within the platform, or are difficult to replace. Some suppliers provide “health-ready components” with embedded sensors for detecting failures, while others offer comprehensive monitoring strategies associated with their equipment’s critical failure modes.

The assessment of individual equipment health within a weapon system can be further enhanced by embracing the use of historical operational, sustainment, and repair data to generate actionable information for driving enterprise activity outcomes. Maintenance and Logistics enterprise operations can leverage advanced analytics to predict future

states of the weapon system, optimize maintenance and logistics processes, and drive weapon system readiness—thus enhancing maintenance and logistics enterprise operations. Advanced analytics can produce actionable intelligence regarding availability, quality, sustainability, and reliability of a weapon system. This information is also useful for establishing proactive supply support strategies that strive to keep the weapon system operational.

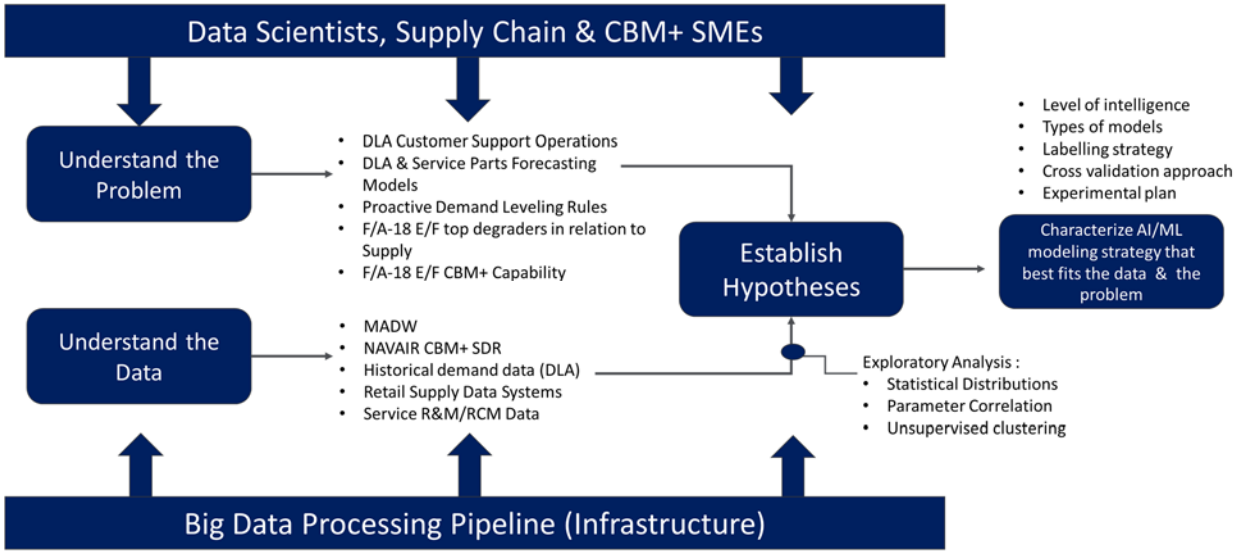
The Services are in the process of implementing CBM+ using system health indications to predict a functional failure ahead of the event and take appropriate preemptive actions. Adding to the health indications, is the use of historical operational, sustainment, and repair data to generate useful information and drive enterprise activity outcomes. The preemptive actions include more efficient maintenance planning, efficient and effective maintenance, and proactive supply support planning. Given this perspective of CBM+, we explore three general CBM+ data types:

1. Onboard and offboard aircraft subsystem health monitoring data
2. Operational, sustainment process, and repair data
3. Planned maintenance event information.

ML Modeling Technique

Figure 2-1 provides an overview of the analysis process employed by Global Strategic Solutions, LLC to understand the problem, understand the data available for the project, establish a solution hypothesis, and characterize the ML modeling technique that best fits the problem and the available data. As shown in the diagram, the project team included data scientists and supply chain and CBM+ subject matter experts. Using the company's cloud-based infrastructure and application development environment, the team conducted the exploratory modeling and analysis of the source data obtained for this project.

Figure 2-1. Characterizing the ML Modeling Technique



Note: AI = artificial intelligence; MADW™ = Maintenance and Availability Data Warehouse™; NAVAIR = Naval Air Systems Command; R&M = reliability and maintainability; SDR = Standard Data Repository.

Chapter 3
Technical Concept and Approach

We met with the DLA Center of Planning Excellence (CoPE) staff to review the project objectives and get an understanding of their perspective on the problem. We found the following:

- DLA has many forecasting tools and has tried many others. Stakeholders view the development and value of new forecasting models with skepticism.
- One or more of the following improvements will constitute success:
  - Providing better data from the Services to DLA models with AI curation
  - Helping the Services provide better collaboration input
  - Creating a more effective means for notifying DLA planners when demand patterns have changed or are changing.

The collaboration data contain much noise:

- How can they determine what's important?
- What Service data (features) are important to improve planning?
- Planners need to account for important exceptions, changes in assumptions, etc.
- How can they know what "they don't know"?

The underlying message is, "DLA wants to project what its customers will buy in the future, instead of forecasting what the demand will be in the future." More effective collaboration between the Services and DLA is key in accounting for the actual use (operating environment), reliability, and ad hoc business processes used by the Services to meet operational demands and tackle readiness issues. Historical demand-based forecasting and planning is not enough to accurately project what the Services will buy in the future. DLA needs better situational awareness.

Source Data

The project analyzed the F/A-18 Generator Converter Unit (GCU), which the F/A-18E/F Reliability Control Board (RCB) classified as a top degrader. We validated this classification by analyzing Navy data from the decision knowledge programming for logistics analysis and technical evaluation (DECKPLATE) in the MADW™. Our analysis showed the GCU suffered from high not mission capable supply (NMCS) hours. Historically, it has experienced low reliability and has gone through three design upgrades. We collaborated with DLA and Navy stakeholders to obtain DECKPLATE source data for the project (Table 3-1).

It should be noted that the GCU has not specific sensors assigned. The data is captured from by the onboard test computer.

Table 3-1. DECKPLATE Data

Data source	Data description
Navy	Service demand change request example
Navy	F/A-18 E/F RCB analysis report of readiness top degraders
Navy	GCU Maintenance actions (2011 to present)
Navy	GCU Depot data: 2016 to present full GCU and Shop Replaceable Assemblies (SRAs) repairs
Navy	Flight records related to GCU repairs and corresponding Maintenance Status Panel (MSP) Codes
Navy	Specific GCU maintenance actions: (Jan 1, 2019 to Oct 7, 2019), National Item Identification Number (NIIN) History of Failure (HOF): 015459351, 014793739, 014708688, 015452665, 016432784, 014793815, 014938781, 015452661 Circuit Card Assembly
Navy	Specific GCU maintenance actions: Jan 1, 2019 to Oct 3, 2019) and NIIN HOF: 015664393, 015997663, 016270932, 014708681, 015452670, 014553692 Generator

Aircraft Subsystem Health Monitoring Data

Using the industry standard Society of Automotive Engineers JA6268, Figure 3-1 provides an overview of the F/A-18E/F CBM+ capabilities, highlighting the capability level 1 functions in red. The Navy uses built-in test (BIT) and MSP codes to drive maintenance actions on critical components and is implementing a predictive analytics function on the Environmental Control System. For the GCU, the actual health monitoring capability is limited to diagnostics. GCU BIT/MSP codes are captured and downloaded to the ground station. The GCU test station results are captured in DECKPLATE; SRA callouts, malfunction (MAL) code, and narrative. Flight records are downloaded from the aircraft memory unit.

Figure 3-1. F/A-18E/F CBM+ Capability Level Overview

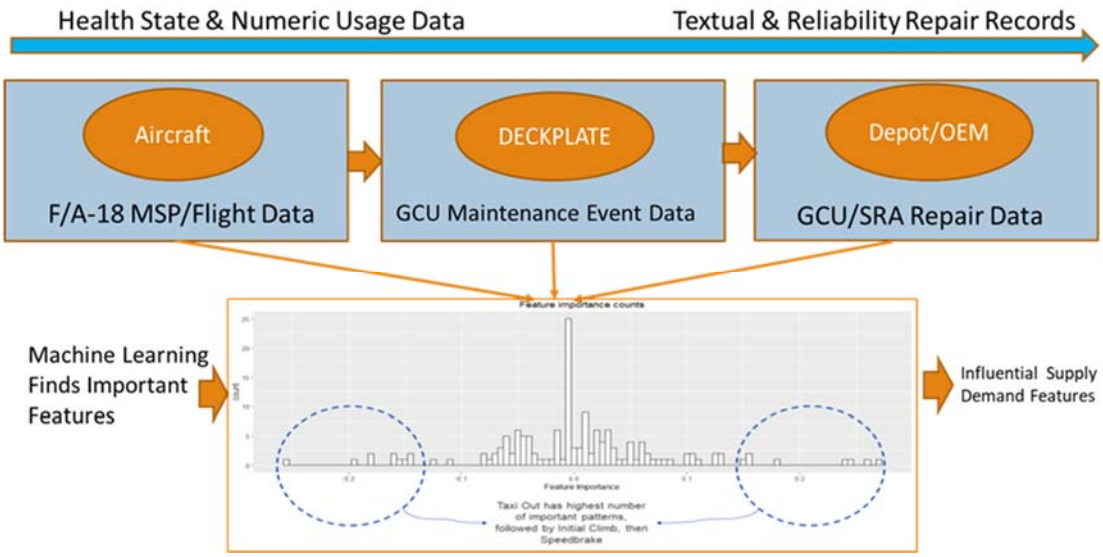
Note: IDE = Integrated Data Environment; PLR = performance life remaining.

Weapons system diagnostic data help in assessing failures once they occur and in reducing the time to reliably replenish (TRR) and the number of parts ordered per repair. TRR is the time required to troubleshoot and repair failed equipment and return it to normal operating condition. From a parts forecasting perspective, it provides the foundation for flagging a degrader component. Tracking and computing the actual component mean time between failures (MTBF) or mean flight hours between failures (MFHBF) and comparing the metrics with the design MTBF/MFHBF provides an early warning of a "reliability degrader." Using advanced analytics on BIT/MSP data across the fleet may render insight into the underlying root cause.

Operational, Sustainment Process, and Repair Data

Figure 3-2 shows the classes and sources of data in this approach. As shown, we combine the weapon system data with the GCU maintenance event data: test results, SRA callouts, narrative (corrective action, action taken description, and malfunction description), and the GCU and SRA depot repair data. From the advanced analytics perspective, this framework is the most promising for using ML on CBM+ data to improve the DLA parts forecast and planning process. The idea is to use advanced analytics to identify influential supply demand features. These features can then be used to provide better collaboration input: a more efficient means for notifying DLA planners when demand patterns have changed or are changing due to actual use (operating environment), reliability, or ad hoc business processes used by the Services to tackle operational demands and readiness issues.

Figure 3-2. Analysis of Operational, Sustainment Process, and Repair Data



Note: OEM = original equipment manufacturer.

Connecting the three classes of data shown in Figure 3-2 provides the ground truth for advanced analytics using ML. The GCU features extracted from the flight data can be correlated with the maintenance event and depot repair data. Once the data classes are correlated, we use advanced analytics algorithms. The algorithms analyze all relevant information, including flight data and maintenance event data, plus structured and unstructured data such as technician notes and depot repair records. Analysis of the GCU data obtained for this project can render key insights and benefits, including being

able to leverage large amounts of historical data to inform engineering, maintenance, and supply support planning. Figure 3-3 provides an example illustrating the correlation of data across the sustainment process for a specific GCU serial number.

Figure 3-3. Correlation of Data Across Sustainment Process for Specific GCU Serial Number

We used the following analytics and modeling techniques on DECKPLATE data of known GCU NIINs:

1. Clean data:
   - Remove duplicate data fields
   - Remove observations with too many missing values
   - Eliminate uninformative variables.
2. Select features on the basis of
   - residual sum of square,
   - adjusted R squared,
   - Mallow's Cp, and
   - Bayesian information criterion.
3. Model using
   - linear regression,
   - weighted regression,
   - robust regression,
   - decision tree, and
   - random forest.

Planned Maintenance Information

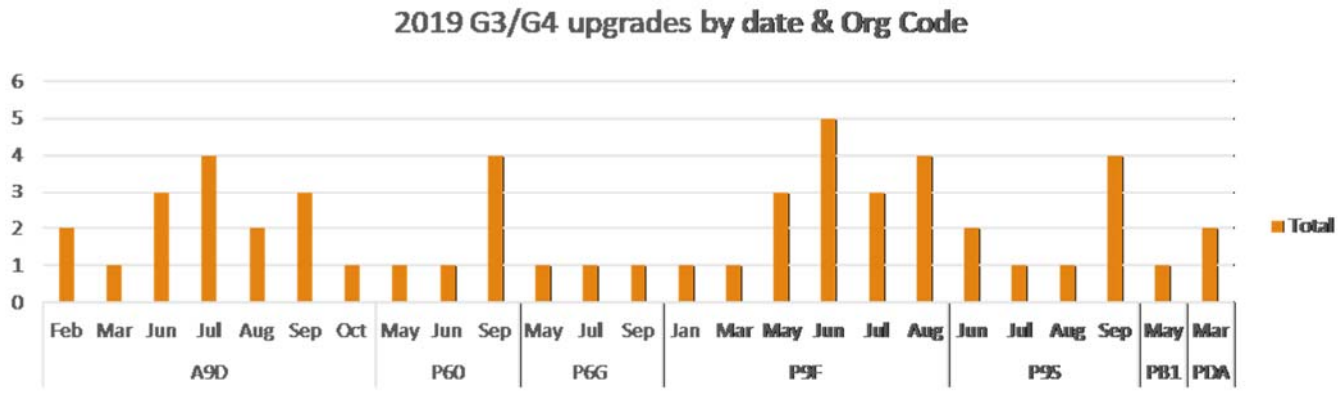
Every weapon system has a predetermined schedule for maintenance events based on its depot-level integrated maintenance concept (IMC)/preventive maintenance interval (PMI). The RCM analytical process determines the appropriate failure management

strategy, including preventive maintenance (PM) requirements and other actions warranted to ensure safe operation and cost-wise readiness of the weapon system.

In addressing readiness issues, the Navy is shifting from fly-to-failure strategies to planned proactive removals. This new soft-time threshold maintenance strategy is based on the commercial best practice used in the civil aviation sector (first used by Delta). After removal, the assets will be inducted for depot maintenance to restore the level of reliability. The depot will do this restoration in accordance with a comprehensive build specification, instead of the "inspect and repair as needed" concept. To implement this strategy, the Services will have to develop a schedule of removals and servicing of the components. The Services and DLA planners can use these schedules and build specifications to develop projected demands for parts supplied by DLA.

Using DECKPLATE data and MAL codes, ML can render insights into demands for future parts. For example, in the case of the GCU, there is an ongoing effort to replace or convert older units (G2/G3) to the latest G4 version. Figure 3-4 shows G3 to G4 upgrades done in 2019 as recorded in DECKPLATE. As in the case of soft-time threshold removals, the schedule and rate of conversions at the depot or OEM can be used to develop projected demands for parts supplied by DLA.

Figure 3-4. G3 to G4 Upgrades in 2019

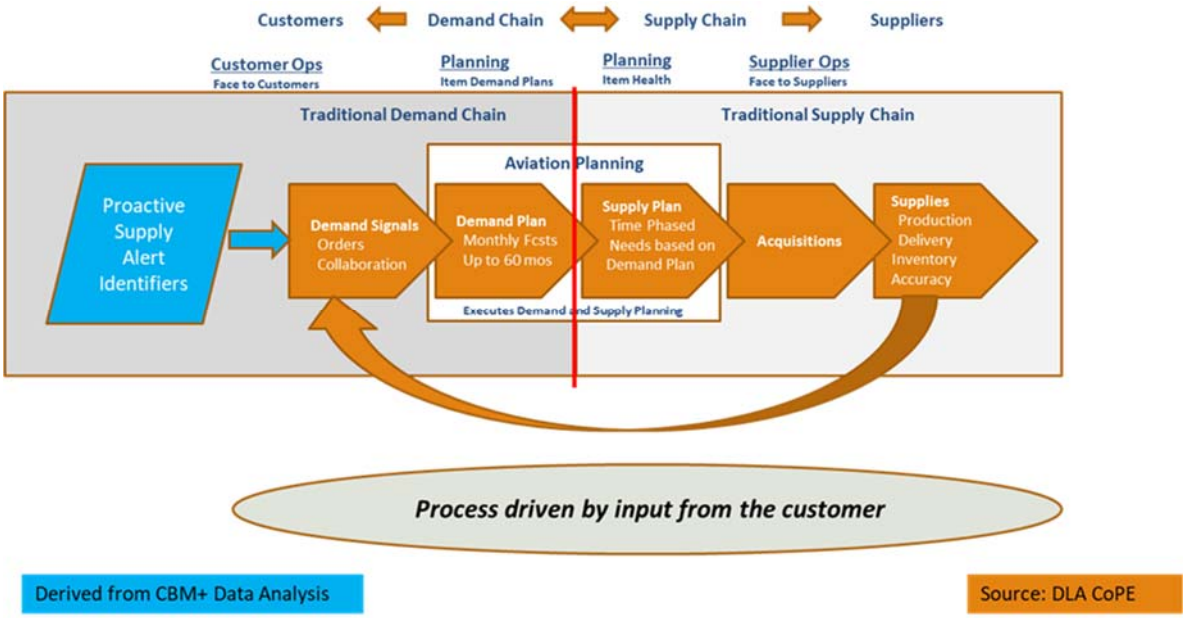


Chapter 4
Findings and Recommendations

Findings

We applied the CBM+ method without determining a specific model or set of models to address forecasting improvements. We found that application of this method can generate a list of proactive supply alert identifiers (PSAIs) that can be integrated into the existing DLA forecasting and planning models to define the recommended operation framework to exchange data more efficiently. Figure 4-1 illustrates the DLA business model with the integration of the PSAIs.

Figure 4-1. DLA Business Model Using PSAIs



Using this approach, DLA planners can use the constant streams of PSAI class data to identify demand indicators and trends in projected parts demand. Services could supply to DLA degradation trends, soft-threshold trends, and revised maintenance schedules derived from the CBM+ data analysis. DLA parts planning could use this additional information about Service part demand levels to automate collaboration and development of future demands.

Recommendation for use of CBM+ Data to Improve Parts Forecasting

The recommended PSAI classes are provided in Table 4-1. The PSAIs are defined in the context of the F/A-18 GCU, but they can be generalized and applied to any weapons system component supported by DLA.

Table 4-1. Service PSAIs Classes Matrix

PSAI class	Service organization	Data sources	PSAI resultants	Benefit
RCB Top Degrader Program	NAVAIR	RCB top degrader list	Top degrader list with root causes, courses of action, and help needed sections	Quick identification of overall problem systems and DLA-specific issues
R&M—actual vs. design reliability	NAVAIR	DECKPLATE, AFAST, and FAME (BIT/HAT and MSP tests)	NIIN MTBF, MFHBF, and LRU cannibalization trends; correlation of entries with NIIN; and flight data analysis	Current NIIN usage and early projection alerts of NIIN failure; correlation to other parts in failing system
Life limited parts (logistics)	NAVAIR	Time-sensitive item maintenance at the O/I/D levels; AFAST/DECKPLATE systems	Specific category of NIIN, listing time on wing, RTAT, installation, expected shelf-life, and service area installed	DLA supply and planning demand schedule of expiring parts
Usage BOM trends (logistics)	NAVAIR FRC	O/I/D levels, AFAST/DECKPLATE, and FAME	IMC/PMI BOM, number of parts ordered per repair, correlation of same part on different systems by TEC, NIIN, and MAL code	Rate of replacement for specific part
Scheduled maintenance (RCM/PM)	NAVAIR	RCB HAT, DECKPLATE, and AFAST	Weapon system IMC/PMI maintenance schedule, NMCM scheduled-to-unscheduled trends, NAVAIR soft-time threshold conversion analysis and part replacement/servicing, degrader upgrade or replacement schedule	Time for DLA to order parts for availability; if the part is obsolete, time to adjust spare SOH parts

Note: AFAST = Aviation Financial Analysis Support Tool; BOM = bill of material; FAME = F-18 F/A-18 Automated Maintenance Environment; FRC = Fleet Readiness Center; HAT = Hornet Asset Tracker; LRU = line replacement unit; NMCM = not mission capable maintenance; O/I/D = organizational, intermediate, and depot; RTAT = repair turnaround time; SOH = stock on hand; TEC = type equipment code.

PSAI Classes

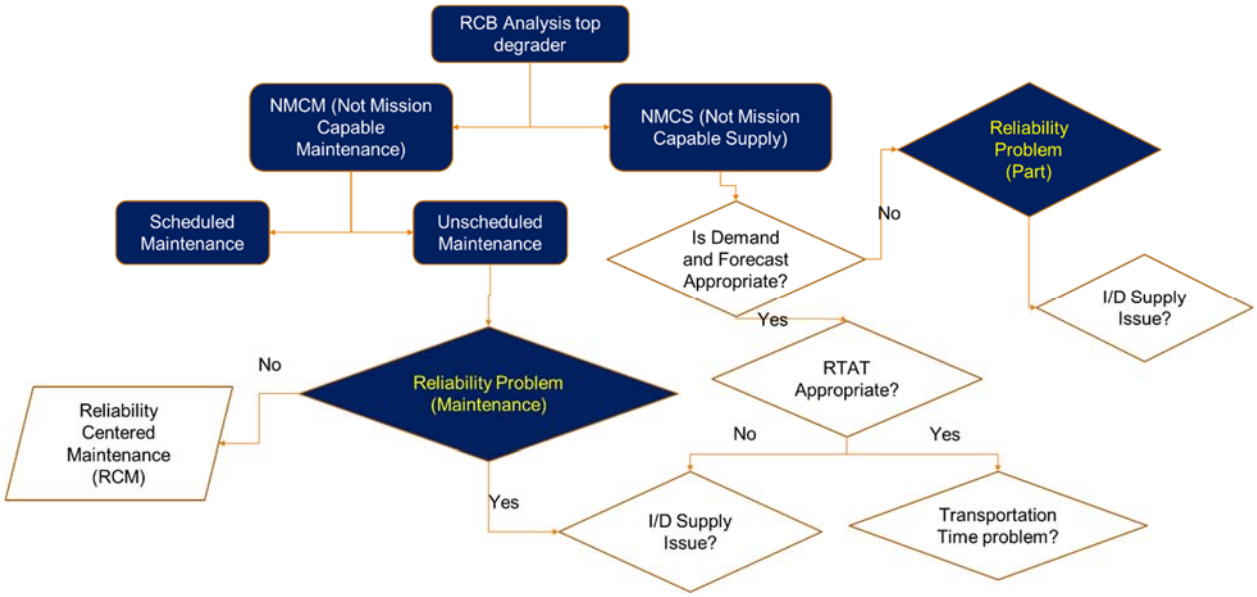
Service-provided PSAI classes would serve as a stream of data to be analyzed and sent from each Service to DLA to render a more complete picture of demand for DLA-provided parts. Each PSAI class of information would show trends and a timely picture of the actual health of F/A-18E/F weapons system. The following subsections describe the PSAI classes and information Services can gather and automate for DLA.

Top Degrader and Root Causes

This PSAI class uses Services' current analysis and centralizes the Service top degrader list and root causes for F/A-18E/F weapon systems. Services generate this report annually by monitoring DECKPLATE and other Service maintenance system trends. The annual report on the F/A-18E/F top 20 degraded systems includes root cause analysis, remedies (current trends), and help requests from outside the Service if needed. These reports give DLA a PSAI watch list to begin monitoring and actual requests to DLA for help in policy and procedures. Over time, these reports show the historical trend of bad actors in the F/A-18E/F top degradation list.

For example, using an RCB, the Navy employs trend analysis to determine maintenance top degraders using data from DECKPLATE and other NAVY systems. The RCB trend analysis shows the F/A-18 E/F GCU has been a top degrader system for several years. The GCU was also verified as a top degrader system based on NMCS hours computed from MADW™ records. Historically, the GCU has experienced low reliability and has gone through three design upgrades. There are two GCUs on each F/A-18 E/F weapon system. Figure 4-2 illustrates the process to determine the root cause of top degraders.

Figure 4-2. Navy's RCB Process Overview



The RCB top degrader information on the GCU could be used to generate (automatically) this type of PSAI class information for each DLA supplied NIIN. DLA planners can use the constant streams of this PSAI class data to identify demand indicators and trends in projected parts demand. Services could supply degradation trends, actual reliability parameters, and NIIN's involved to DLA derived from analysis of the GCU CBM+ data. DLA parts planning could use this additional information to determine future GCU parts projections.

Reliability and Monitoring

This PSAI class includes MTBF and MFHBF from the Navy reliability and monitoring system. Services have done this analysis with the estimated systems hours before GCU failure. This class also includes cannibalization trends, flight record analysis, and maintenance trends. Because analysis has been done for entire systems (GCU), Services should monitor information related to all parts consumed in repairs to give a better picture of demand to DLA. Creating scripts from commands (listed in Appendix A) and filtering on a specific NIIN, the rate of cannibalization of parts is generated by organization or aircraft tail number, including all DLA-provided GCU parts.

Logistics Life Limited Parts

This PSAI class comprises the Navy logistics life limited parts, a subset of the repairs and service FRC inventory and verified with Service DECKPLATE repairs. The Services

provide install date, DECKPLATE repair data, purchase date, and time in use (AFAST flight information) for each part.

Service Actual Usage

This PSAI class identifies actual usage, including the count of repairs that used the part across all Services, depot-level repairs, and special agreements where parts are supplied to Services from the DLA OEM. These data are consolidated from each of the Service's maintenance systems and include Service analysis of repair frequencies or issues.

Weapon System Scheduling

This PSAI class includes current scheduled vs. unscheduled trends, IMC/PMI-scheduled modifications computed from Service maintenance data trends, and whether or not Service analysis was performed and determined to be advantageous to create a soft-threshold part replacement. The actual repairs scheduled in DECKPLATE are sent to DLA.

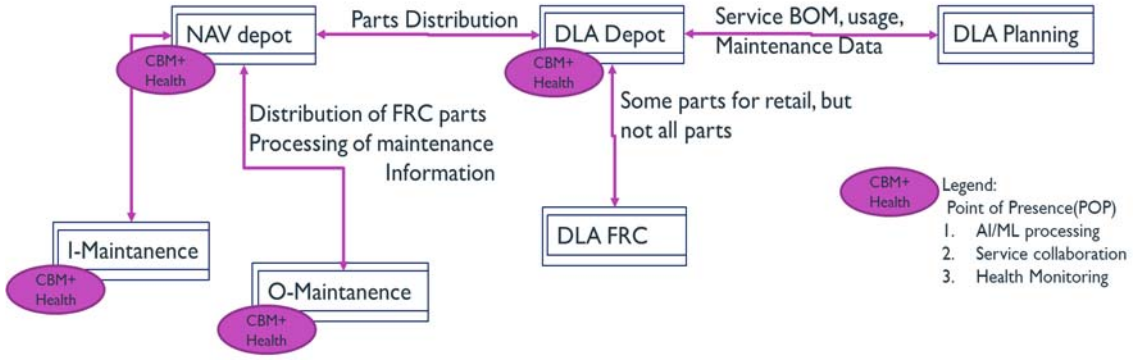
Weapon System Health Monitoring

For this PSAI class, Services continue documentation of RCB trend data to give top degrading overall health of weapon systems using flight data (BIT/Health Ready Components and Condition Indicator/Health Condition Indicator trends), flight analysis, actual DECKPLATE and AFAST maintenance data, and estimated residual performance life of the weapon system. This provides a comprehensive compilation of previous PSAI class data to develop a DLA overall weapon system parts tracking health monitor.

Recommendation for Automating PSAI Generation and Stakeholder Collaboration

Figure 4-3 illustrates the potential introduction of points of presence in support of the collaboration process.

Figure 4-3. CBM+ Notional Points of Presence to Process CBM+ Data and Collaboration



Services and DLA would need to identify POPs at maintenance locations and DLA FRC to consolidate the data and generate PSAI classes of information. These are placeholders to process data and do not necessarily have large data stores.

Services currently use CBM+ data to more efficiently identify GCU G2/G3 remedy repair by maintenance locations, F/A-18E/F usage, and GCU redesign. Using the technique outlined in this study, Services can provide part PSAI classes to DLA. DLA can combine current DECKPLATE and AFAST CBM+ data to develop a repair baseline of demand from the PSAI classes above. This repair Service CBM+ data would be combined with DLA/Service ordering systems to develop more accurate NIIN usage. From Service analysis, maintenance data scripting, and data classifications, the Services would provide streams of PSAI classes to DLA health monitoring systems.

Services have known specific maintenance data points of interest that scripting can develop into streams of data to help in the procurement of parts. Services can start by automating these pivot tables and maintenance data charts to give a stream of data to DLA on a specific point of interest for PSAI class of data. DLA would be able to automate the CBM+ data stream to generate a repair trend for any weapon system RCB degradation systems. We started this process by creating Python scripts of DECKPLATE malfunction code 815 (cannibalization due to part not on hand) on a specific GCU NIIN DLA-supplied part that is out of stock at time of repair.

From Service analysis, we identified common repair kits by DECKPLATE corrective action and narrative description. Performing pattern matching of text searching these fields for specific SRA and MSP code, Python scripts can be generated to identify most common repairs of GCU NIINs. For example, a weeks' worth of DECKPLATE data enable analysis of the frequency of specific repairs.

From text mining of MSP/SRA flight data codes, specific DECKPLATE hits for SRA and MSP codes can be filtered and merged, and count summaries of repairs generated as summaries of DECKPLATE maintenance information. This, combined with DECKPLATE information of cannibalization rate, gives a repair trend for a specific part. This stream of data would be provided to both Services and DLA planning for parts investigation/future procurement. This is a PSAI class to inform DLA and Services to investigate root causes listed earlier and the potential change demand signal of current purchasing from Services to DLA.

Python scripts can analyze results for new trends in CBM+ data, previous MSP problem codes, and soft-threshold NIINs. Previous Service research has observed differences between DECKPLATE maintenance repairs and actual BOM requests. These data can be correlated with actual GCU repair usage and SOH buffering.

In summary, using CBM+ data, the Services can derive data streams of PSAI information for DLA. The Services PSAI classes of data streams can be used by DLA to develop TensorFlow models that correlate overall demand needs of GCU NIINs to determine F/A18-E/F weapon system parts demand over time, by maintenance location and developing usage trend. DLA would grab specific information resultants (PSAIs) from the Services to assist in collaboration in parts needed as well as changes in low demand to high demand parts as needed. With the PSAI classes of data streams from the Services, TensorFlow models using TensorBoards Graphical User Interfaces can be used to adjust DLA planning model proficiencies in supply data streams with actual usage compared with purchasing. Automated exchange of actual repair information in addition to BOMs give DLA a more complete picture of Service DLA NIIN actual usage and demand. ML algorithms can be developed in collaboration with the Service PSAI

data points for specific NIINs to monitor Service trends and adjust DLA parts planning forecasting.

Recommendation for Visualization App of PSAI Information for Stakeholder Collaboration

One example of obtaining a PSAI class data point walkthrough that would assist DLA planning immediately is the creation of cannibalization rate trends of DLA-supplied NIINs (GCU NIIN Generator/Alternator used). The Navy RCB program has already established that GCU has been listed as top degrader in the RCB top degrader report. A list of GCU parts for each version (GCU G2, GCU G3, GCU G4, and hybrid) was created. Analysis of GCU NIINs reveals an increasing trend in cannibalization for specific parts needed in overhaul and upgrades of the GCU. Simple Excel spreadsheet pivot tables of DECKPLATE repair data show a growing trend of GCU NIIN demand for repairs. Because cannibalization rates are not captured in current BOMs supplied to DLA by the Services, part identification of low to high demand is not known from the Services until parts are not available and a Service demand change letter is created.

Using GCU NIINs for Generator/Alternator, we developed a Python script to automate, filter, and create pivoted tables and charts of DECKPLATE data search of a specific NIIN. The DECKPLATE pivot information would have NIIN HOF counts (year and month from Comp Date Time), type/model/series, malfunction code, and action organization code. This is PSAI class 2, reliability and monitoring of current GCU NIIN parts. Using this generic NIIN HOF script, we ran all DECKPLATE repair data of F/A-18E/F repairs to generate tables and charts of output. This generates the PSAI of cannibalization rate of specific NIIN HOF information to supply to DLA planning. (Appendix A contains the Python script.)

Because the Python scripts are generic enough to run any DECKPLATE data but specific enough to identify the cannibalization rate of all GCU NIIN parts, Services would provide a total GCU NIIN cannibalization rate to DLA. Services would provide these trend rates for each GCU NIIN part to give a demand PSAI signal to DLA on base NIIN parts used to correlate with Services current BOMs. Services would also give this GCU NIIN cannibalization rate for analysis of why cannibalization is occurring. DLA currently tracks Services' orders but not cannibalization. Therefore, whole GCU SRAs are taken out of Service with no supply tracking to order new parts.

Service CBM+ data from Service maintenance data (DECKPLATE, AFAST, Aviation Store Keeper Information Tracking moved into [FAME], etc.) would be processed, filtered, and merged to create single PSAI class data points of interest to generate a single actual maintenance demand data point of interest (Figure 4-4). From analysis with pivot tables, CBM+ data (DECKPLATE, FAME, and AFAST) can document the cannibalization trend of single GCU NIIN. The training model of data to give DLA a neural network perception point of data for DLA would use ML to give stream of dates, counts, and action organizations of all repair hits of specific streaming maintenance data instants. Using data analysis and visualization tools, we can merge these perception points and develop them into a DLA NIIN PSAI, the answer (stream of repair data rate over time) that supply's trend of GCU NIIN demand (PSAI) over time and schedule points of when to replace. This would require development of multiple models to develop an overall F/A-18E/F system health of specific degraded part.

Figure 4-4. Service Data Processing to Create Single PSAI Neural Network Percept

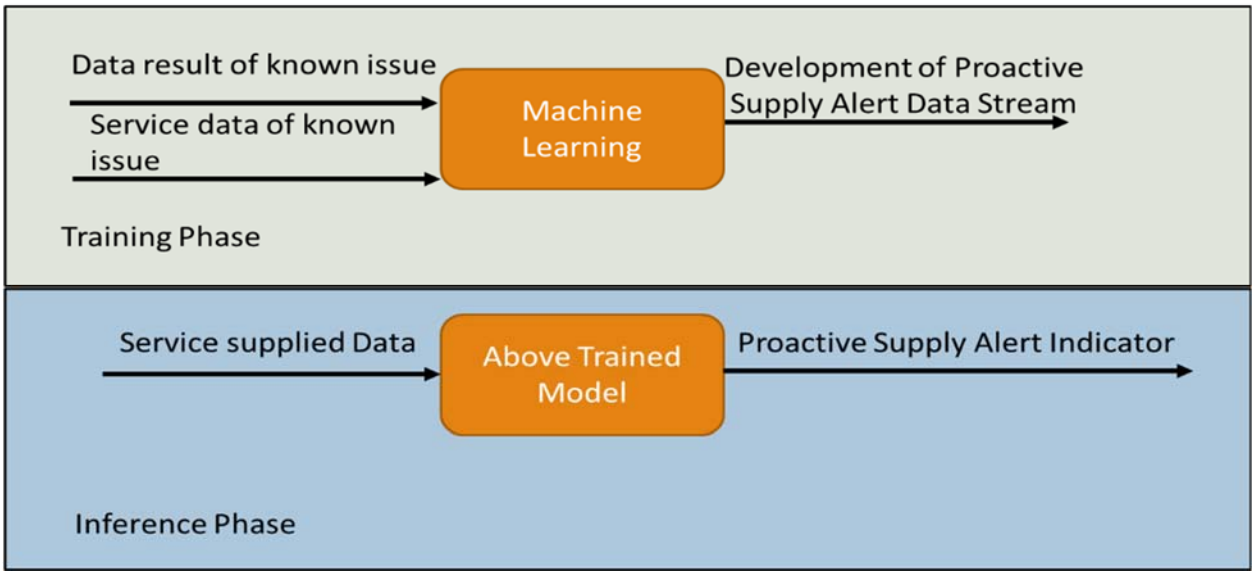
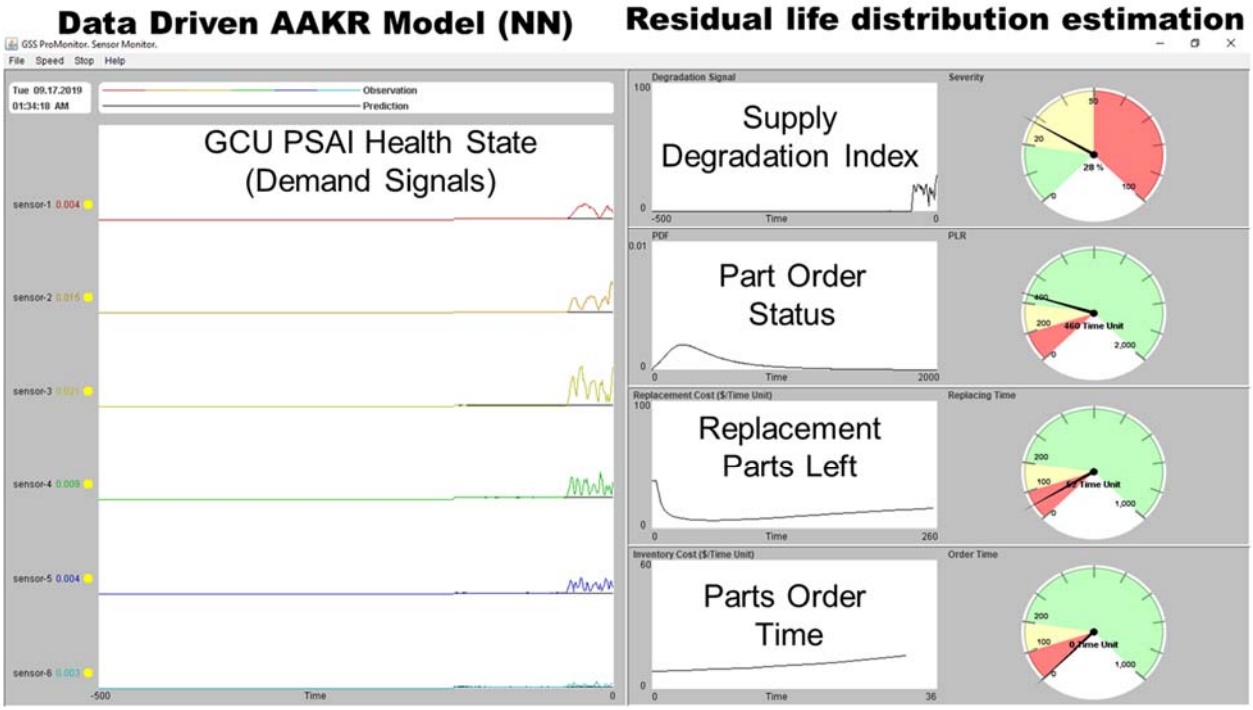


Figure 4-5 shows the RCB top degrading system parts demand dashboard. It gives an overview of DLA Service part demand ML models based on PSAI classes of data. This analysis could easily be repeated for GCU parts repairs using the existing generic scripts.

Figure 4-5. Notional Application Dashboard—Quick View of Parts Status



Appendix A
CBM+ Method

This appendix provides detailed technical information related to the application of the CBM+ method in this study.

Determining Cannibalization Rates

To test the applicability of these methods, we sought to determine cannibalization rates for specific NIINs within the GCU. The steps included the following:

- Create scripts to filter and automatically generate pivot tables for specific NIINs (starting with known or expected "bad actors"); information would include
  - type/model/series,
  - action organization code,
  - NIIN HOF counts, and
  - completion date (year, month).
- Run the scripts with DECKPLATE data by date ranges to create trend data.

Using Service analysis techniques and table of data sources from DECKPLATE, analysis shows an increasing trend in the cannibalization rate and past scheduled maintenance of specific DLA GCU NIINs (015664393, 015997663, 016270932, 014708681, 015452670, and 014553692). Figure A-1 shows the cannibalization rate for these NIINs by month and year. Figures A-2 and A-3 illustrate cannibalization based on malfunction code. In Figure A-3, the cannibalization rates dramatically increase from 2017 into 2018 and 2019 for malfunction code 815—Cannibalization–Part Carried But Not On-Hand in Local Supply System. In other words, this part was cannibalized due to a not-in-stock.

Figure A-1. Total Cannibalization Rate of Known NIINs

Figure A-2. Cannibalization Rate for Malfunction Codes 812-814

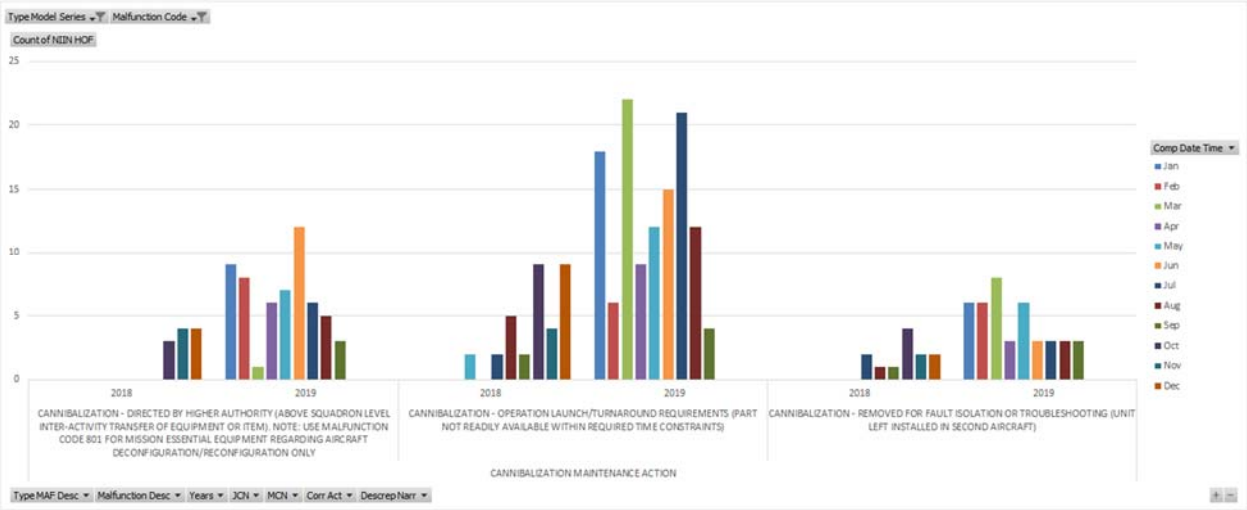
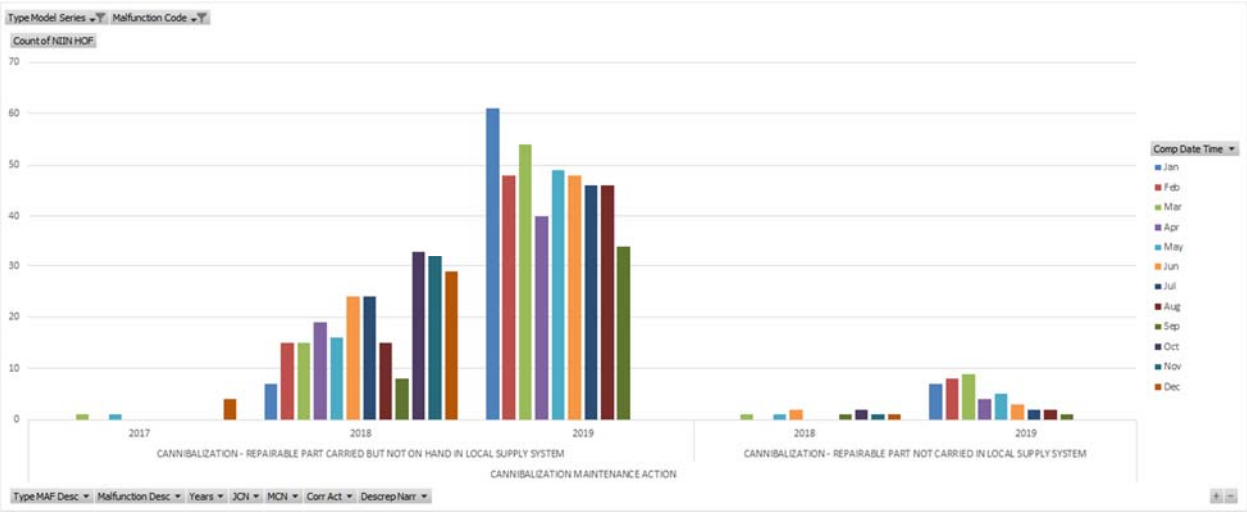


Figure A-3. Cannibalization Rate for Malfunction Codes 815-816



Navy Service Supplied CBM+ Data

Table A-1 provides the data description from the Navy DECKPLATE system.

Table A-1. DECKPLATE Data Description

Data description
Service demand change request example
F/A-18E/F RCB analysis report of readiness top degraders
GCU maintenance actions (2011 to present)
GCU depot data: 2016 to present full GCU and SRA repairs
100 to 1,000 flights records related to GCU repairs and corresponding MSPs
Specific GCU maintenance actions (Jan 1 to Oct 7, 2019) and NIIN HOF: 015459351, 014793739, 014708688, 015452665, 016432784, 014793815, 014938781, 015452661 Circuit Card Assembly

Table A-1. DECKPLATE Data Description

Data description
Specific GCU maintenance actions (Jan 1 to Oct 3, 2019) and NIIN HOF: 015664393, 015997663, 016270932, 014708681, 015452670, 014553692 Generator

On the basis of data from DECKPLATE, the Navy RCB analyzes the root causes of trends on the top degraders. Table A-2 identifies the top three degraders and root cause analysis for the F/A-18E/F.

Table A-2. RCB Top Degrading F/A-18E/F Parts with Root Cause

Rank	Degrader name	Root cause
1	Outboard Leading Edge Flap Lower Fairing	Maintenance induced damage, over-torque during removal and replacement. Compounded by demand forecast transfer from Naval Supply Systems Command to DLA.
2	GCU	G2/G3 not designed to withstand alternating current non-linear electrical loads. Erroneous removal while troubleshooting wiring discrepancies.
3	Arresting Tailhook Assembly	Nicks, gouges, and corrosion driving excessive RTAT at scheduled overhaul. Compounded by inaccurate forecast model, Quality Engineer at FRC, and carcass constraints.

Figure A-4 illustrates Navy RCB top degrader trending data.

Figure A-4. Navy RCB Top Degrader Trending Data

Total Rank & Trend		Action Area	Degrader Action Area	Total Degrader Score	Current Activity Score	Rank	Demand Impact & Trend	Rank	Maint. Impact & Trend	Rank
1	Up	42A1E	GEU 34/A AIRCRAFT (GCU)	1.00	0.66	3	0.96	2	0.65	3
2	uP	73x51	ID-2582/A DISPLAY UNIT (MPCDR)	0.81	0.23	8	1.00	1	0.19	13
3	-	27D00	F414-GE-() ENGINE	0.72	1.00	1	0.36	5	0.47	4

Figure A-5 illustrates Navy root cause analysis for GCU G2 and G3 repair issues.

Figure A-5. GCU G2 and G3 Repair Issues

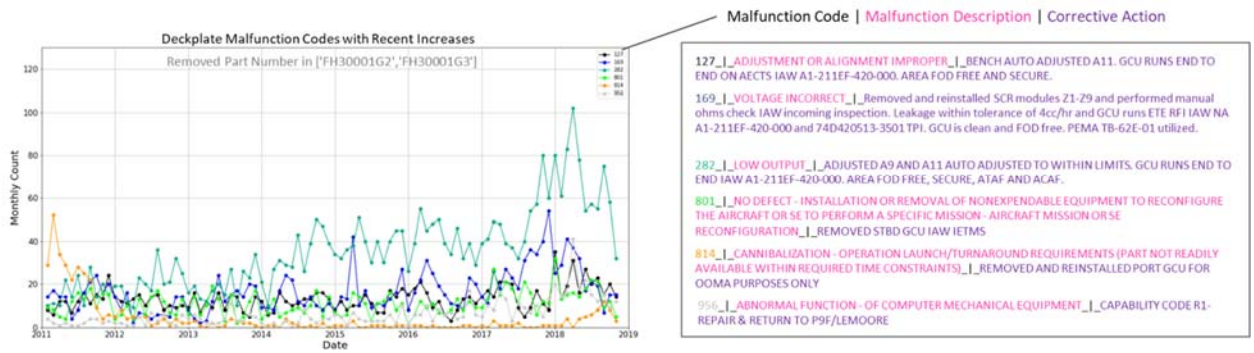


Figure A-6 illustrates the findings that most repair data information is in the Malfunction Description, Narrative Description, and Corrective Action fields of DECKPLATE. These list the MSP codes of F/A-18E/F flight data to identify the GCU component repaired and any common components. Service analysis identify that MSP code 870 and 871 were most common and the corrective action taken would be to remove and replace SCR component, Integrated Product Team, A11, A13, and adjust SRA components A2 and A9. Service analysis of GCU G2 and G3 found 19 common SRA parts repaired due to overheat and overload.

Figure A-6. Data from DECKPLATE—Malfunction Description, Narrative Description, and Corrective Action Fields

Summary of Last Altered Date Time: On or after Jan 1, 2019 12:00:00 AM AND NIIN HOF: 015664393, 015997663, 016270932, 014708681, 015452670, 014553692

By Type Model Series

EA-18G	FA-18	FA-18E	FA-18F	N EA-18G
1310	23	2675	2408	7

By Maintenance Level Description

Depot Level	Intermediate Level	Organizational Level
3	4579	4579

By Action Taken Code

0	1	2	7	8A	B	C	D	F	N	P	R	T
1	121	1	115	32	26	7	1341	175	2	121	771	2286	1424

By Malfunction Type #66 rows with missing values

C(Cannibalization)	U(Repairs)	N/V(Upgrade)
2239	4118	66

By Work Center Code #59 rows with missing values

020	05A	110	120	140	220	320	62A	62E	62L	X50	X55	No Value(N/V)
82	278	4354	3	54	1	7	1	1513	30	10	31	59

The graphs in Figure A-7 illustrate the obvious noise within the plot. Moving average methods can be applied.

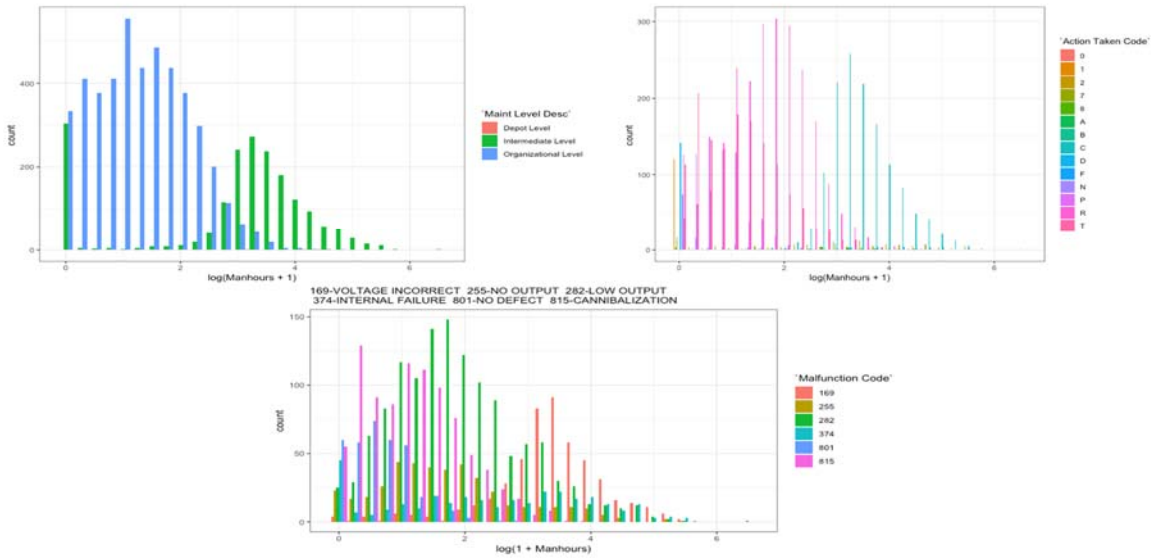
The tradeoff between cost and benefit is as follows:

- Suppose X is the number of units provided in a store.
- If X is larger than the actual number, X_a, needed in the store, then $X - X_a$ units will be unsalable. Suppose the cost of one unit is C. Then we will lose $\max[(X - X_a) \cdot C, 0]$.
- If X is smaller than the actual number, X_a, then $X_a - X$ customers will face the shortage problem, and the store will probably lose customers, which will negate potential benefits. Suppose the benefit gained from one customer is B and the probability of losing one customer is P. Then we lose $\max[(X_a - X) \cdot B \cdot P, 0]$.
- Thus, the function of total benefit should be

$$f(X) = B \cdot X_a - \max[(X - X_a) \cdot C, 0] - \max[(X_a - X) \cdot B \cdot P, 0].$$

- It is obvious that when $X = X_a$, this store gets the maximum total benefit. Most of the time, we have errors in prediction, so our goal is to maximize the mean of total benefit given the distribution of X.

Figure A-7. Data from DECKPLATE—Areas of Maintenance, Action Code Counts, and Top Six Malfunction Code Types



We investigated and plotted malfunction code 815 to discern trends in repairs performed where parts are not available. This is a specific indicator from Service maintenance data that a GCU part is not available at the action organization for repair due to GCU NIIN not available. While six cannibalized GCU NIINs in 2017 documented from DECKPLATE data this may be okay to keep F/A-18E/F in readiness. When 2018 has 309 Generator Alternator parts cannibalized for repairs due to lack of parts and 2019 has 696 parts cannibalized with 2019 September seeing a decrease cannibalization rate, DLA would have advance indicator that not enough Generator Alternators are being procured or Service supplying to area using Root Cause analysis. This doesn't give a total project of GCU NIIN parts but will produce a base demand.

Python script to parse DECKPLATE data can create quick processing the data and send results to DLA planning.

First, Python would need to import the following modules:

Import pandas as pd

Import numpy as np

Import matplotlib.pyplot as plt

Import datetime

Data = pd.read_excel(r'<Directory of deckplate file>\<name of deckplate file>.xlsx')
#Used to import deckplate search file

```
index =
['col1','col2','col3','col4','col5','col6','col7','col8','col9','col10','col11','col12','col13','col14','
col15','col16','col17','col18','col19','col20','col21','col22','col23','col24','col25','col26','col2
7','col28','col29','col30'] #Used to rename column names for processing

column = ['TEC','Type Model Series','Action Org Code','BCM','RFI Ind','Comp Date
Time','Last Altered Date Time','JCN','MCN','Maint Level','Maint Level Desc','Action
Taken Code','Action Taken Desc','Malfunction Code','Malfunction Desc','Malfunction
Type','Work Center Code','Repair Net Price','NIIN HOF','COG','SMR Code','DLR
Cost','Unit Price','Corr Act','Descrep Narr','Manhours','Type MAF Code (OOMA)','Type
MAF Desc','NIIN HOF Desc','Bu/SerNo'] #column names of Deckplate

df = pd.DataFrame(Data, columns= column) #data imported from deckplate

df2 = df.set_axis(index,axis=1, inplace=False) #data imported from deckplate with
column name substitute

i = Data.index.stop # find last deckplate entry

col6date = pd.DataFrame.to_numpy(df2['col6']) #obtaining and indexing Comp Date
Time

col6dateclean = pd.DatetimeIndex(data=col6date, start=0, end=i)

compdateyear ={'Year':col6dateclean.year} # create year dictionary list

compdatemonth ={'Month':col6dateclean.month} # create month dictionary list



compdateday ={'Day':col6dateclean.day} #create day dictionary list

compdatetime ={**compdateyear, **compdatemonth, **compdateday}

df3 = pd.DataFrame.from_dict(compdatetime) # create year, month, day Dataframe



pivot = df.pivot_table(index=['Type Model Series'], values=['NIIN HOF'],
aggfunc=np.count_nonzero) # Simple pivot table to give counts of NIIN for each F/A-
18E/F type

df4 = pd.concat([df3,df2], axis=1) ] # filters to apply to data to get total cannibalization of
F/A-18E/F specific systems (may need addition filters on addition a columns to identify
specific parts)

df4 =
df4[df4.col19.isin(['015664393','015997663','016270932','014708681','015452670','0145
53692'])]

df4 = df4[df4.col1 != 'NMAK']
```

```
df4 = df4[df4.col1 != 'AMAK']

df4 = df4[df4.col1 != 'AMA9']

df4 = df4[df4.col14.isin(['812','813','814','815'])]pivot2 = df2.pivot_table(index=['Type Model Series'], values=['NIIN HOF','Comp Date Time'], aggfunc='count') #pivot table of filtered data

index2 = ['Year','Month','Day']+column

df4 = df4.set_axis(index2, axis=1, inplace=False) #filtering and create datapoints of interest for excel spreadsheet export

df4['Month'] = df4['Month'].apply(lambda x: calendar.month_abbr[x])

df4['Month'] = pd.Categorical(df4['Month'], categories=months, ordered=True)

pivot = df4.pivot_table(index=['Year','Month','Type Model Series','Action Org Code'],values=['NIIN HOF'], aggfunc = ['count'])

pivot2 = df4.pivot_table(index=['Year','Month','Type Model Series'],values=['NIIN HOF'], aggfunc = ['count'])

pivot3 = df4.pivot_table(index=['Year','Action Org Code'],values=['NIIN HOF'], aggfunc = ['count'])

writer=pd.ExcelWriter(r'C:\Users\danhu\Downloads\testdata2.xlsx')

df4.to_excel(writer,sheet_name='MalFunction_Code',index=False)

pivot.to_excel(writer,sheet_name='pivot_table1')

writer.save()

plt.subplot = df.pivot_table(index=['Type Model Series'], values=['NIIN HOF'], aggfunc='count').plot(kind='bar') # plot simple pivot chart

plt.xlabel('Type Model Series')

plt.ylabel('NIIN HOF Count')

plt.subplot = df2.pivot_table(index=['Type Model Series'], values=['NIIN HOF'], columns=['Comp Date Time'], aggfunc='count').plot(kind='bar') #plot filtered pivot chart

plt.xlabel('Type Model Series')

plt.ylabel('NIIN HOF Count')

print(df2) #full deckplate data chart of filtered results.
```

Another trend discovered in DECKPLATE data is scheduled maintenance tasks in G3/G4 upgrades. Using number of GCU G3/G4 upgrades by creating pivot table to Type MAF Desc = 'TECHNICAL DIRECTIVE MAINTENANCE ACTION' and using NIIN HOF Desc = 'GENERATOR,ALTERNATI'.

With above Python scripting, trending of DECKPLATE data can process all F/A-18E/F rates of upgrades and compare with Service provided-estimates to get the trend of GCU NIIN parts needed to repair GCU G4 while GCU G2 and G3 parts that are not common with other equipment could be phased out of purchase.

With Service assistance in analysis trending, automated pivot tables creating streams of data and plotting charts give data points of low demand changes to high demand. Figure A-8 shows the DECKPLATE columns usable to create pivot tables.

Figure A-8. DECKPLATE Columns Usable to Create Pivot Tables

Data Element	Data Type	Data Desciption	Values	Unit of Measure	Source System Name
TEC	Type Equipment Code	F/A - 18 E NIIN type part	AMAH		DECKPLATE
Type Model Series		F/A -18E description	F/A--18E		DECKPLATE
Action Org Code		Organization preforming maintenance	A9D		DECKPLATE
BCM	Boolean	Maintenance repair code	Y/N		DECKPLATE
RFI Ind	Boolean		Y/N		DECKPLATE
Comp Date Time	date	Date repair started	date		DECKPLATE
Last Altered Date Time	date	Date repair completed or jcn moved to next level			DECKPLATE
JCN	9 digit field	Job Control Number	A8D066123		DECKPLATE
MCN	7 digit field	Material Control Number	A9DVU8J		DECKPLATE
Maint Level	1 digit field	Intermediate, Organizational, Depot	1,2,3		DECKPLATE
Maint Level Desc	Text	Description of Intermediate, Organizational, Depot	Intermediate		DECKPLATE
Action Taken Code	1 digit field	Action of Repair taken	C		DECKPLATE
Action Taken Desc	Text	Description of Action of Repair	Cannibalization		DECKPLATE
Malfunction Code	3 digit field	Code of maintenance action	815		DECKPLATE
Malfunction Desc	Text	Description of maintenance action	Cannibalization		DECKPLATE
Malfunction Type	1 digit field	Type of maintenance action	C/U		DECKPLATE
Work Center Code	3 digit field	organization maintenance center	62E		DECKPLATE
Repair Net Price	price	Price of repair	$		DECKPLATE
NIIN HOF	9 digit field	unique part identifier	16270932		DECKPLATE
COG	2 digit field		2 digit		DECKPLATE
SMR Code	5 digit field		text		DECKPLATE
DLR Cost	price	Labor price	$		DECKPLATE
Unit Price	price	Unit price	$		DECKPLATE
Corr Act	text	Description of corrective action taken	text		DECKPLATE
Descrep Narr	text	Description of NIIN part problems and MSP/SRA codes	text		DECKPLATE
Manhours	number	hours worked repair	number		DECKPLATE
Type MAF Code (OOMA)	2 digit field		text		DECKPLATE
Type MAF Desc	text	Description of actions performed	text		DECKPLATE
NIIN HOF Desc	Text	Description of NIIN part	text		DECKPLATE
Bu/SerNo	number	Serial Number	number		DECKPLATE

Table A-3 shows the MSP and SRA codes for 1 week of data against all F/A-18E/F repairs from DECKPLATE.

Table A-3. SRA Words in the "Corrective Action" Column

Word	Frequency
R/R	29
A1	263
A2	235
A3	293
A4	158
A5	57
A6	32

Table A-3. SRA Words in the "Corrective Action" Column

Word	Frequency
A7	150
A8	397
A9	455
A10	7
A11	10
A12	3
A13	1
A14	21

Table A-4 summarizes the frequency of certain MSP hit words used in the "Description Narrative" column merged with Air Division MSP definitions for descriptions.

Table A-4. MSP Words in the "Description Narrative" column

MSP code	Description form NAVAIR MSP code definition	Frequency
041	RADAR TRANSMITTER FAIL	48
074	L FUSELAGE DECODER FAIL	18
076	R FUSELAGE STATION DECODER FAIL	12
112	DMC FAIL	46
120	XXX	147
159	MU/AMU FAIL	49
191	RATE SENSOR ASSY B FAIL	7
195	READ BLIN CODES	22
544	CTR EXTERNAL TANK FUEL QUANTITY PROBE	3
678	R ENGINE OIL PRESSURE SIGNAL FAIL	48
813	L ENGINE ANTI-ICE FAIL	8
M	RLCS DOOR OPERATION FAIL	19
871	RIGHT GENERATOR FAIL	24

Appendix B
Abbreviations

AFAST	Aviation Financial Analysis Support Tool
AI	artificial intelligence
BIT	built-in test
BOM	bill of materiel
CBM+	Condition Based Maintenance Plus
CoPE	Center of Planning Excellence
DECKPLATE	decision knowledge programming for logistics analysis and technical evaluation
DLA	Defense Logistics Agency
DoD	Department of Defense
ERP	enterprise resource planning
FAME	F/A-18 Automated Maintenance Environment
FRC	Fleet Readiness Center
GCU	Generator Converter Unit
GSS	Global Strategic Solutions, LLC
HAT	Hornet Asset Tracker
HOF	History of Failure
IDE	Integrated Data Environment
IMC	integrated maintenance concept
LRU	line replacement unit
MADW™	Maintenance and Availability Data Warehouse™
MAL	malfunction
MFHBF	mean flight hours before failures
ML	machine learning
MSP	Maintenance Status Panel
MTBF	mean time between failures
NAVAIR	Naval Air Systems Command
NIIN	National Item Identification Number
NMCM	not mission capable maintenance
NMCS	not mission capable supply

OEM	original equipment manufacturer
O/I/D	organizational, intermediate, and depot
PLR	performance life remaining
PM	preventive maintenance
PMI	planned maintenance interval
POP	point of presence
PSAI	proactive supply alert identifier
R&D	research and development
R&M	reliability and maintainability
RCB	Reliability Control Board
RCM	reliability-centered maintenance
RTAT	repair turnaround time
SDR	Standard Data Repository
SOH	stock on hand
SRA	Shop Replaceable Assembly
TEC	type equipment code
TRR	time to reliably replenish