1415 N CHERRY AVE
CHICAGO, IL 60642
(312) 281-6900
DMDII.ORG
DMDII@UILABS.ORG

DMDII
a UI LABS Collaboration
DIGITIZING AMERICAN MANUFACTURING

DMDII FINAL PROJECT REPORT

Enabling Real-Time Supply Chain Visibility Through Predictive Analytics

Principal Investigator / Email Address	Ashis G. Banerjee, Ph.D.; ashisb@uw.edu
Project Team Lead:	University of Washington
Project Designation	DMDII-15-12-02
UI LABS Contract Number	0220160028
Project Participants	General Electric Global Research, ITAMCO, National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign
DMDII Funding Value	$755,110
Project Team Cost Share	$870,310
Award Date	12/23/2016
Completion Date	09/23/2018

SPONSORSHIP DISCLAIMER STATEMENT: This project was completed under the Cooperative Agreement W31P4Q-14-2-0001, between U.S. Army - Army Contracting Command - Redstone and UI LABS on behalf of the Digital Manufacturing and Design Innovation Institute. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Department of the Army.

DISTRIBUTION STATEMENT A. Approved for public release; distribution unlimited.

TABLE OF CONTENTS

I. EXECUTIVE SUMMARY

a. Specific Industry Problem or Challenge Being Addressed

In today's inter-connected economy, original equipment manufacturers (OEMs) procure parts, many of which have lead times of more than a year, from hundreds of globally distributed suppliers. The OEM suppliers, which are often small and medium-scale enterprises (SMEs), obtain parts themselves from many other dispersed suppliers, leading to the creation of complex supply chain networks with several layers of hierarchical dependencies. The networks frequently include multiple bottlenecks in the form of specialized suppliers who act as sole sources of certain critical parts. These characteristics make it *imperative to have a high degree of visibility into the flow of parts through the networks* to facilitate production planning and inventory management for OEMs and SMEs alike. However, the critical information pertaining to operation schedules, production capacities, and material procurement decisions are typically not shared among the stakeholders to maintain competitive benefits. Hence, part flows must be *predicted* based on either supplier promises and historical performances, or demand estimates and buyer preferences. Furthermore, such prediction-driven visibility needs to happen in real-time to afford any form of substantial value to the stakeholders. This project brought together a collaborative and multi-faceted academia-industry team to address certain key aspects of this visibility problem for real-world supply chain networks.

b. Solution Approach

Using both multivariate linear and non-linear decision tree-based regression modeling combined with dimension reduction of categorical variables, moderately large volumes of transactional data were analyzed to predict the delivery times of purchases orders (POs) and fulfillment times of sales orders (SOs) for our partner OEM and SME, respectively. In addition to regular serial processing, the regression models were trained using parallel processing on regular multi-core personal computers to ensure that they could be rapidly updated (re-trained) on an as-per demand basis. The outcomes of the models were visualized in an easy-to-use, web-based dashboard, which was customized based on the feedback from the SME production personnel. As a part of an open-source software tool, the dashboard allowed the users to interact with the model outcomes in real time, select orders of interest, quickly identify the most risky part deliveries, and forecast the delivery dates for potential, future POs.

c. Summary of Project Outcomes and Recommendations

The project led to the development of an open-source predictive analytics-based supply chain visibility tool that yielded significant improvements with respect to all the four key performance indicators (KPIs). Specifically, it resulted in more than 40% improvement in predicting the delivery dates for 90% of the POs as compared to the currently-used supplier estimates, and decreased the number of POs with at least 3 weeks of delivery time prediction errors by more than 50%. Similarly, it led to more than 50% improvement in predicting the fulfillment dates for 90% of the SOs over current heuristic-based estimates, and reduced the number of SOs with at least 3 weeks of fulfillment time prediction errors by 45%. Furthermore, the tool interface (dashboard) was refined based on a structured user study conducted at our partner SME site, and was successfully demonstrated live at multiple DMDII events. Therefore, our recommendations are to promote the deployment of our tool within the existing supply chain solutions of both OEMs and SMEs, and continue further development of the tool in terms of even better usability, additional validation studies, and a larger suite of predictive models to address the complete array of supply chain forecasting and visibility problems.

II. PROJECT REVIEW

a. Project Scope and Objectives

The scope of the project is defined by a combination of business problem (case) and corresponding technical challenges. The business case is that both sourcing and demand fulfillment operations are currently inefficient and non-resilient to market changes or risks. The primary technical reason for this problem is that information flows across supply chains are limited, making it difficult to have satisfactory predictive capabilities on critical decision making considerations such as on-time parts deliveries and production demand forecasts.

Consequently, the specific objective of this project is to leverage the power of supervised machine learning in utilizing historical transactional data to provide real-time information into the predicted flow of parts or materials across supply chain systems for both OEMs and SMEs. The main project deliverable is in the form of a self-contained, open-source software tool that implements the analytics methods and enables interactive visualization of the analysis outcomes with adequate documentation for easy installation and ready use.

b. Technical Approach and Planned Benefits

Our technical approach comprises developing predictive models, specifically for PO delivery dates and SO fulfillment dates, based on the historical transactional data available at partnering OEM and SME organizations, using advanced machine learning methods. The models are then used by an interactive web-based software tool to provide real-time estimates on the flow of POs and SOs across the supply chain networks of OEMs and SMEs, respectively. In the absence of direct information exchange across the supply chain networks, these estimates are expected to provide enhanced visibility in a surrogate manner.

The planned benefits of our approach are listed below:

- It leverages historical data in making predictions using supervised machine learning (regression) models with user-specified options on whether to opt for conservative, optimistic, or nominal decision choices.
- It enables rapid training of the regression models using parallelization, leading to real-time updates of the learned models for newly available data.
- It generates models that are useful in prediction tasks for both OEMs and SMEs.
- It leads to the creation of a visibility tool interface or dashboard that is customized based on specific supply chain user needs and preferences.
- It ensures that the visibility tool is readily deployable on individual desktops and laptops, and can be potentially integrated with existing enterprise data warehouses.

III. KPI'S & METRICS

Metric	Baseline	Goal	Results	Validation Method
Delivery times for 90% of POs	Supplier-provided estimates	50% more accurate than baseline	41% more accurate than baseline	Testing on historical POs over a 1 year period provided by GE
# of POs with > 3 weeks of non-conforming delivery times	Supplier-provided estimates	50% fewer than baseline	52% fewer than baseline	Testing on historical POs over a 1 year period provided by GE
Fulfillment times for 90% of SOs	Heuristics or simple linear model-based estimates	50% more accurate than baseline	50% more accurate than baseline	Testing on historical SOs over a 10 year period provided by ITAMCO
# of SOs with > 3 weeks of non-conforming fulfillment times	Heuristics or simple linear model-based estimates	50% fewer than baseline	45% fewer than baseline	Testing on historical SOs over a 10 year period provided by ITAMCO

IV. TECHNOLOGY OUTCOMES

a. System Overview, Requirements, and Architecture

The visibility tool includes a front-end for data visualization and a securely stored data warehouse in the form of a MySQL database to host the OEM/SME data (automatically pushing the data from an existing source to the database is outside the project's scope). Analytics models, in the form of supervised learning via regression combined with dimension reduction, then pull the data from the database to predict the most likely values of the chosen response variables (e.g., PO delivery times, SO fulfillment times, etc.). In addition to the standard serial mode, it is possible to train the models in parallel using the multiple cores available on the host machine. The visibility tool is implemented using the R programming language and the web-based dashboard is hosted on the R Shiny server. It is made available for ready installation and use as a Docker package. The overall system architecture is shown in Figure 1.

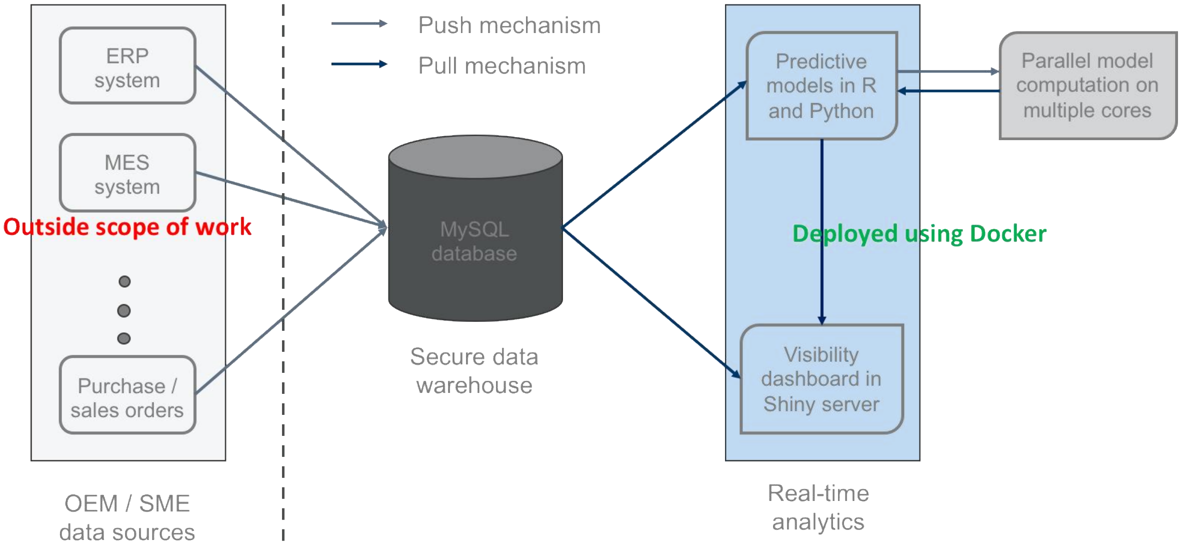


Figure 1. Overall System Architecture of Predictive Analytics-Driven Supply Chain Visibility Tool

b. Solution Features & Attributes

i. Dimension reduction for categorical variables

In machine learning and statistical regression analysis, categorical predictors with a large number of levels are a challenge. By default, the algorithms will expand the categorical predictors into dummy variables during data preprocessing. For categorical predictors with many levels, the algorithms are likely to run into the problem often called the *curse of dimensionality*. To prevent this problem, we leverage the fundamental characteristic of the widely-used principal components analysis (PCA) method and reduce the dimensionality by selecting the key levels for the categorical predictors. However, given the computational complexity of the PCA matrix decomposition for categorical predictors with too many levels, it is necessary to pre-process the data. In summary, the dimension reduction is implemented via the following steps:

- Keep the levels that maintain 80% of the total rows, and combine the remaining levels into a single level.
- Expand the categorical predictors into dummy variables.
- Apply PCA with varimax rotation to ensure that the principal components rely on the least number of variables (levels).
- Extract the loadings for the top *n* principal components, and select the *m* variables (levels) that have the most contributions across all the *n* principal components.

The number of extracted principal components and the number of levels kept in each categorical predictor are determined by both computational efficiency and prediction accuracy. In the implementation process, R packages reshape2 is used to conduct PCA on the dataset. Finally, 5 principal components are extracted and 15 levels are kept in the categorical variables.

ii. Random Forest and Quantile Regression Forest

Random Forest and Quantile Regression Forest are both ensemble learning methods for supervised machine learning. They are constructed by a multitude of decision trees at the training time, which aim to robustly capture the underlying nonlinear trends and consider different measures of central tendency and statistical dispersion. Furthermore, they usually provide accurate predictions of unobserved data, prevent overfitting to training samples, and avoid undue influence of the outliers in the datasets.

The difference between the two methods lies in their outputs: for Random Forest, it is the mean prediction, whereas, for Quantile Regression Forest, it is the quantile prediction of the individual trees. Both the methods are built upon the underlying decision tree model.

There are two types of decision trees, classification tree and regression tree, of which the latter one is of interest here. Regression tree model uses a tree structure to partition the dataset recursively and compute specific regression values at the leaf node for different conjunctions of predictor values. The regression tree model uses variance reduction to identify the most suitable variable for splitting any interior node into two branches. By recursively computing the variance reduction for each split, the learning goal is successfully achieved by identify the splitting variable with the optimal split of the dataset. Therefore, the regression tree model is gradually expanded until there is no variance reduction in any split.

Given the regression tree model, a random forest model is built on it consisting of multiple simple regression tree models. The terminology "Random" often refers to two sources: randomly sampled training dataset by applying Bootstrap on the original dataset, and randomly selected subset of features to choose the best split in each tree model. It is a simple but effective mechanism to aggregate many simple models to tackle a complex prediction task.

The prediction of a single regression tree may be highly sensitive to noise and outliers, but the average of many trees is not, as long as the trees are not strongly correlated. Bootstrap is the way of preventing correlations in the trees by randomly sampling with replacements from different training sets. However, the practical implementation of random forest model is not exactly the same as the above procedure. In order to gain better computational and learning performance, each tree is trained using only a subset of the predictors in the training dataset. This process is another way of confirming that the trees are uncorrelated. Since the predictors are not guaranteed to be independent, using the highly correlated features to train the regressing tree will cause serious correlation issues even using bootstrap samples. Therefore, training with randomly sampled predictors ensures the uncorrelation property of the trees.

As a generalization of Random Forest, Quantile Regression Forest gives the predictions of the conditional quantiles instead of the conditional mean. In statistical analysis, the goal is to infer the relationship between the response variable and the predictors. The standard regression analysis is trying to develop an estimate of the conditional mean of the response variable. After successfully training the Quantile Regression Forest, it is used to predict any quantile of the condition distribution of the response variable.

In the implementation process, R packages randomForest and quantregForest were employed for implementing Random Forest and Quantile Regression Forest, respectively.

iii. Parallelization of dimension reduction and regression model training

We found out that R can run on the Windows system smoothly either in a serial mode or in a parallelized mode. We realized that, unlike the Linux platform which supports both "SOCKET" and "FORKING", R's parallelization on Windows only uses the mechanism of "SOCKET", but not "FORKING"; this difference restricts the usability of some R packages that rely on "FORKING". We also noticed that Windows' hyper-thread has no effect for parallelization. The optimal degree of parallelization on the Windows platform depends on the number of cores equipped, but not on the number of hyper-threads. After intensive experiments, we also concluded that R's parallelization on the Windows platform has significant overhead. The performance of parallelization is determined by the rate of speed-up offset by the overhead.

As presented in Figure 2, both 2-way and 4-way parallelization ran slower than serial processing for a dataset with 50,000 data points. This indicates that the overhead introduced by parallelization overshadowed the performance speed-up. However, by increasing the data points to 100,000 or 150,000 in the dataset to prolong the computation time, a significant runtime reduction was observed by parallel processing. This demonstrated that, for a lengthy and complex computation, the performance escalation becomes more prominent than the overhead generated by parallelization. Also, as shown in Figure 2, 4-way parallelization did not provide any performance improvement over 2-way parallelization. Since this Windows system is a dual core with hyper-thread, this confirms our finding that hyper-thread does not play any role in parallelization.

This work is inspired by the fact that many companies are using Windows and MAC computers, and most computers are equipped by multi-cores. Therefore, like the Linux platform, Windows and MAC should be able to take the advantage of multi-core architecture to support parallel computing. We have, therefore, endowed our tool with two special features: (1) working on multiple platforms; (2) powered by parallelization techniques.

All the implementation and experiments were carried out on two computers, a Windows machine and a MAC machine.

The Windows machine is equipped with:

- Windows 10 Enterprise
- Intel Core i7-4770 CPU @3.40GHz, 4 Cores
- RAM 16.00 GB

The MAC machine provides:

- MacOS Sierra
- Intel Core i7 CPU @3.20GHz, 4 Cores
- Memory 16 GB

We have performed extensive benchmark experiments on these two computers by running the Random Forest algorithm. We compared the system performance between serial and parallel processing. We studied the impact of the number of cores on parallel computing. We also investigated the effect of number of data points on performance. Figure 2 shows the benchmark result from the Windows machine, and Figure 3 demonstrates the result from the MAC machine.

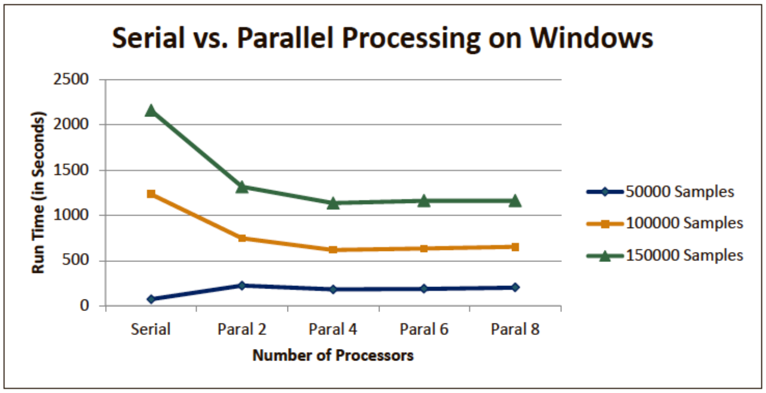


Figure 2: Serial vs Parallel Training of Random Forest Model on a Windows Machine

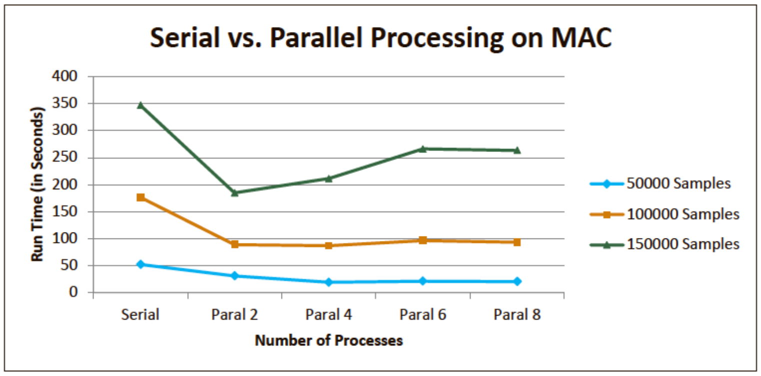


Figure 3: Serial vs Parallel Training of Random Forest Model on a MacOS Machine

We then implemented two new algorithms for the project: parallelized Quantile Regression Forest (QRF) and Principal Components Analysis (PCA).

Quantile regression is used to quantify prediction confidence and certainty by estimating an interval into which future observations will fall with a given probability. A general way for finding quantile regression for decision tree based algorithms is to use QRF. As an extension of the Random Forest (RF) algorithm, the QRF algorithm expands the tree fully so that each leaf has exactly one value. Then a prediction returns individual response variables from which the distribution can be calculated.

Since QRF needs to construct the trees down to a single leaf, more intensive computation than RF is required. To meet this computational challenge, we have implemented a parallelized QRF algorithm. The trees in the forest are first partitioned into *N* groups, each group of trees is then constructed at a single core in a parallel fashion, and finally the results from the *N* cores are summarized and quantile regression is calculated.

Our experiments have proved that parallelization reduces the runtime QRF requires. As shown in Figure 4, comparing to serial processing where a single core is used, 2-way parallelization resulted in more than 50% runtime reduction, and 4-way parallelization caused additional 25% runtime reduction. But we have also observed some overhead introduced by parallelization. With more cores added into parallelization, the performance boosting gradually reduced and eventually disappeared. However, based on these benchmark experiments, we expect our tools can run efficiently on the Windows platform and the MacOS platform. On these two platforms, utilizing two- or four-cores can maximize the benefit, but avoid the overhead of parallelization.

Our experiments also proved that parallelization does not deteriorate the accuracy of QRF. As demonstrated in Figure 5, all the measurements in QRF under different degrees of parallelization are similar. This result ensure us that we can efficiently perform the analysis using the parallelization technique without sacrificing the accuracy.

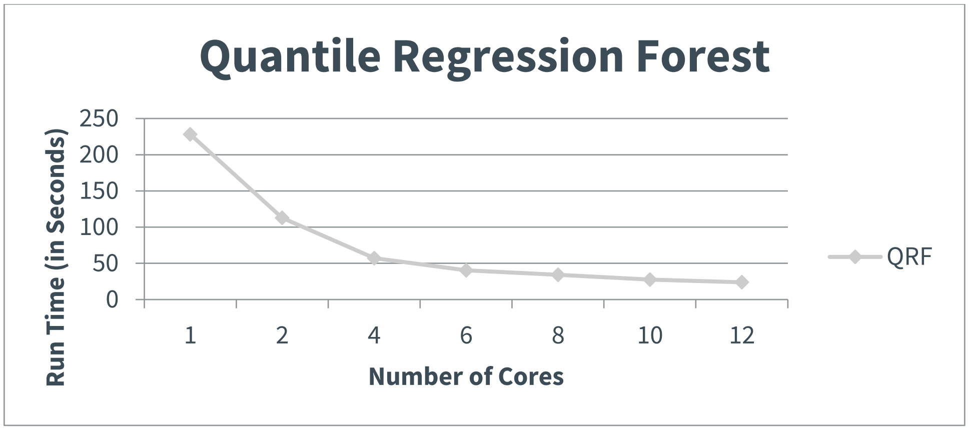


Figure 4. Impact of Parallelization on the Performance of Quantile Regression Forest (QRF) Model

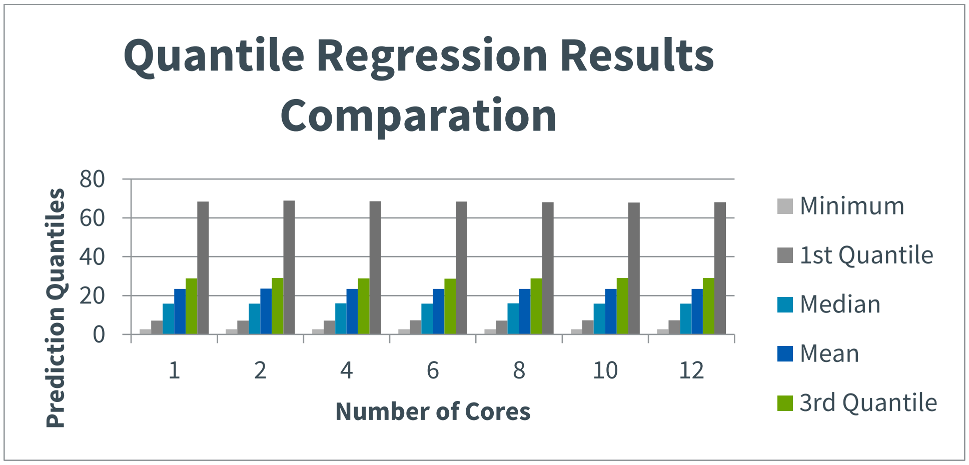


Figure 5. QRF Predictions under Different Levels of Parallelization

We also implemented a parallelized Principle Component Analysis (PCA) algorithm. PCA is a statistical procedure that explains the variance-covariance structure of a set of variables through linear combinations. It is often used as a dimensionality-reduction technique. Since our project needs to handle very large datasets with many variables, we use PCA to identify the principal components. In particular, we have improved the **dudi.pca** function in the package of **ade4** by using the parallelization technique. Our implementation has the following workflow:

The data set is first horizontally partitioned to *N* subsets, where *N* is the number of computer cores to be used. A PCA process is applied to each data partition to identify a list of Principal Components (PCs). *N* of the PCA processes are run simultaneously on *N* computer cores, and each core is responsible for one PCA. Results from *N* of the PCA processes are merged, ranked and filtered to generate the final list of PCs. Figure 6 shows the workflow for parallelized PCA.

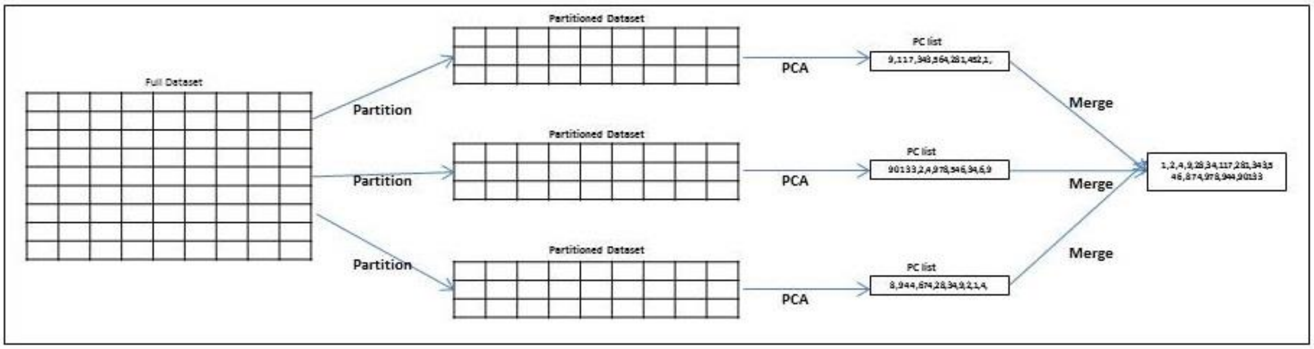


Figure 6. Workflow of Parallelized Principal Components Analysis (PCA)

In our benchmark experiments, we observed significant performance improvement after PCA is parallelized. This is due to two reasons. First, the partitioned data set makes the computed matrix that holds the covariance and eigen values to become smaller. And second, each data partition is processed by a dedicated computer core, and more computation resources are involved in distributing the work. We also observed that parallelization introduces overhead due to the tasks for "job coordination", "network communication", and "result consolidation". Figure 7 shows that runtime substantially reduced when using 2- and 4-way parallelization, but the speed-up gradually faded off after more cores were added to the process.

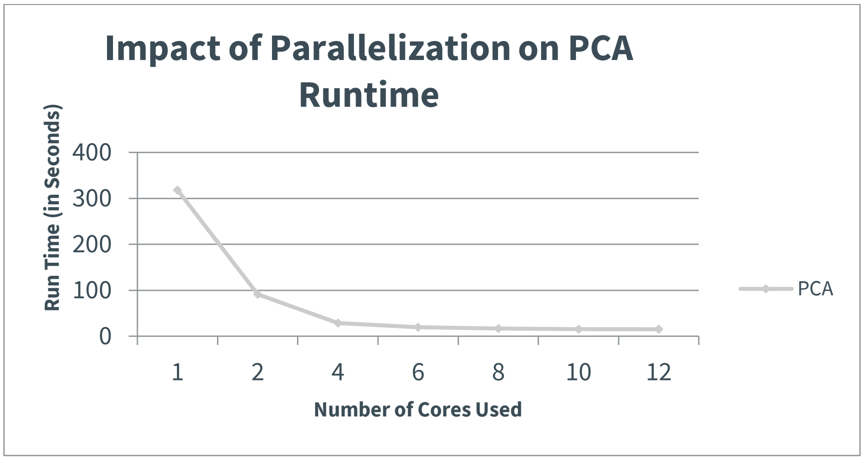


Figure 7. Impact of Parallelization on PCA Runtime

To study the impact of parallelization on the quality of PCA, we compared the PCs identified by the serial processing and by the parallel processing. In general, the PCs identified by the parallel processing is the subset of the one generated by the serial processing. Depending on the degree of parallelization and the nature of the data, the reduction on the number of PC varied from 5% to 50%. This PC reduction has a negative correlation with the degree of parallelization, i.e. the more computer cores are used, the less number of PCs are identified. Figure 8 displays the number of PCs on four different variables under various degrees of parallelization.

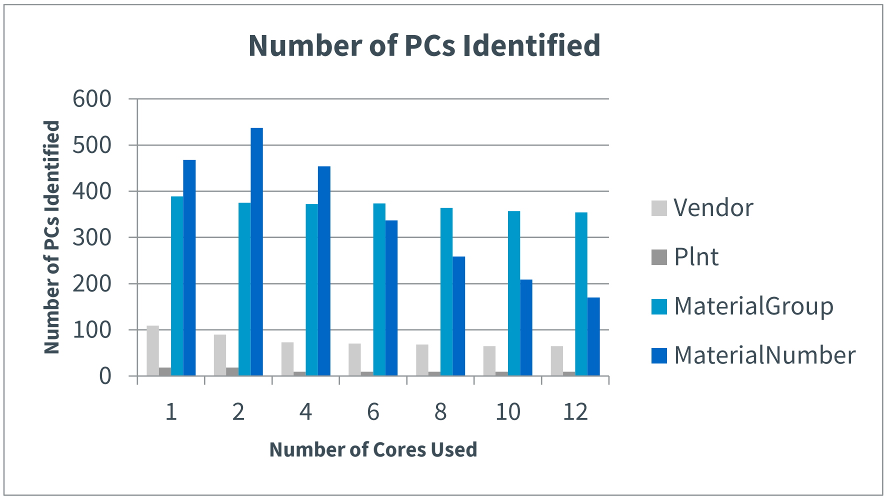


Figure 8. Number of PCs Identified Using Parallelization

c. Modes of Operation

The modes of using our software system are explained later in Section V on Accessing the Technology.

d. Software Development Documentation

A detailed software development documentation is provided in the Appendix Section of this report.

e. Users, Use Cases, and Validation Results

We considered two kinds of business or end users and, correspondingly, two different use cases in this project. They are listed below.

- *DMDII membership use case*
  As a sourcing manager at an OEM, I can use the readily installable supply chain visibility tool to accurately estimate the delivery dates for open purchase orders based on machine learning (regression) models trained on historical transaction data. This capability will enable me to rely less on supplier-provided estimates, and get updated predictions in real-time as and when new data becomes available.

  Results:

  The historical transactional dataset for analyzing is a purchase order (PO) dataset provided by our OEM partner. It contains 53,105 closed POs and 3,533 open POs. The time span is from April 2016 to May 2017. We choose 7 predictors, including 5 categorical and 2 continuous variables, to predict the

PO delivery time in days. The regression models were trained using two continuous variables (quantity ordered and planned delivery time) and one categorical variable (high-level part classification as ABC based on functional requirement). Further, with the dimension reduction method, four more categorical predictors (plant ID, vendor ID, material group ID and material number ID) with a large number of levels are added as predictors.

These predictors are selected based on prior knowledge from the OEMs that they are likely to be significant predictors in PO delivery time estimation. For example, different Item IDs have different procurement cycle times, since various materials require different process times, queue times, and procurement times from suppliers. For Quantity, given limited supplier resources, larger order quantities may require longer supplier process times; conversely, in other supplier-OEM arrangements, larger order quantities may be tied to improved prioritization and shorter supplier process times. In case of the ABC Indicator, inventory prioritization at an OEM impacts purchase requisition processes, order priority, and other transactional processes, given that A parts are more critical to an OEM's operations than C parts. The planned delivery time entered into the Enterprise Resource Planning (ERP) system drives several processes, including prompts to planner-buyers to create purchase orders, replenishment orders, or Vendor Managed Inventory (VMI) orders; it is a critical baseline predictor of the generally acceptable purchase order cycle time for a specific item. Plant ID and vendor ID are the geographic features of the suppliers, which typically affect the PO cycle times. Material number and material group are also considered to be important variables in the production process.

For dimension reduction, the top 5 principal components were extracted and 15 levels were kept for each categorical variable. All the models were built using 120 trees in the forest, 2 predictors randomly sampled as candidates at each split node, and a minimum of 5 data points in the terminal nodes. 10-fold cross validation was implemented to select the best models.

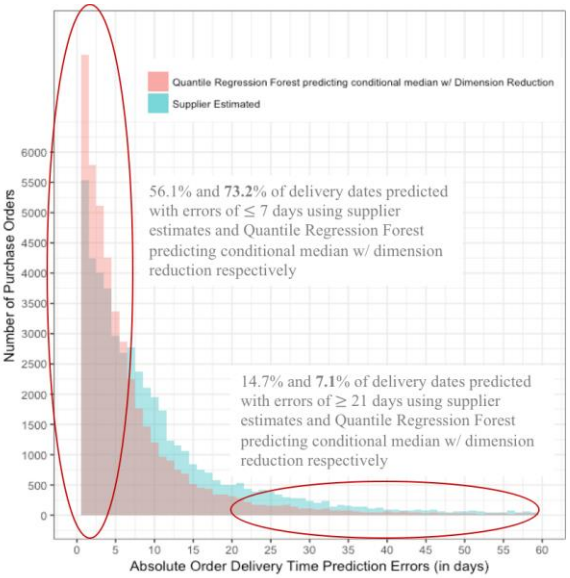
The prediction results are reported in Table 1 as the mean absolute errors in days for several percentiles of the POs. For example, the 25th percentile of supplier estimates means that 25% of the POs have prediction errors of less than 2 days using their estimates. The predictions are compared to the supplier estimated delivery dates and a simple linear regression model with the same predictors.

Table 1 shows promising performance of our models in comparison to the supplier estimates and the linear regression model. Compared to the supplier estimates, our models are able to provide more accurate estimates for all percentiles. Quantile Regression Forest (QRF) predicting conditional median with or without dimension reduction methods provides best estimates for small errors (less than 6 days corresponding to the 50th PO percentile) are provided by the suppliers. It suggests categorical variables are less significant in predicting small errors. For large errors (more than 13 days corresponding to the 75th PO percentile but less than 41 days corresponding to the 95th percentile), the best model is QRF predicting conditional median. Under the extreme circumstances (more than 41 days), QRF predicting conditional mean has the best predicting performance. This results show that our model outperformed the supplier estimates in all percentiles. Quite unsurprisingly, simple linear regression is unable to capture the complex interactions involved in determining the actual PO delivery dates.

Table 1. PO Delivery Times Prediction Errors Using Supplier Estimates and Different Regression Models

Approach		Absolute prediction error (in days)				
		25th percentile	50th percentile	75th percentile	90th percentile	95th percentile
Supplier estimates		**2.00**	6.00	13.00	27.00	41.00
Without dimension reduction	Linear regression	3.62	7.64	13.25	23.02	35.93
	RF	2.63	5.89	11.25	20.67	31.70
	QRF conditional mean	2.38	5.36	10.10	18.01	26.45
	QRF conditional median	**1.00**	**4.00**	8.50	17.00	27.00
	QRF conditional 1st quantile	1.00	4.25	11.00	23.00	35.00
	QRF conditional 3rd quantile	2.00	6.00	12.00	20.75	29.00
With dimension reduction	Linear regression	2.90	6.28	12.06	21.65	32.62
	RF	2.28	5.14	9.98	18.40	28.08
	QRF conditional mean	2.17	4.87	9.37	17.16	**25.54**
	QRF conditional median	**1.00**	**4.00**	**8.00**	**16.00**	26.00
	QRF conditional 1st quantile	1.00	4.00	10.00	22.00	34.00
	QRF conditional 3rd quantile	2.00	6.00	11.00	19.50	28.25

Without dimension reduction, the models provides higher prediction errors as compared to the corresponding models with dimension reduction. In general, dimension reduction method is effective in providing more accurate predictions as we move toward higher PO percentiles. Our best model improves the prediction accuracy by more than 41% for large lead time POs corresponding to unreliable supplier estimates. To summarize, QRF predicting conditional median with dimension reduction has the best overall performance. All our models perform better than supplier estimates and linear regression, even for the POs where the supplier estimates are relatively reliable.



56.1% and **73.2%** of delivery dates predicted with errors of ≤ 7 days using supplier estimates and Quantile Regression Forest predicting conditional median w/ dimension reduction respectively

14.7% and **7.1%** of delivery dates predicted with errors of ≥ 21 days using supplier estimates and Quantile Regression Forest predicting conditional median w/ dimension reduction respectively

Figure 9. Histogram of PO Delivery Times Prediction Errors (for Less than 60 Days) Using Supplier Estimates and QRF Predicting Conditional Median with Dimension Reduction

These trends are further reinforced in the prediction errors histogram plots shown in Figure 9. The distributions show that both our models completely encompass the supplier estimates until we reach POs with prediction errors of more than a week. Correspondingly, the supplier estimates show heavy-tailed distributions with substantially more PO delivery dates being predicted inaccurately, including almost 15% with more than 3 weeks of prediction errors.

- *Commercialized solution use case*
  As a sales order fulfillment manager at an SMM, I can purchase this off-the-shelf supply chain visibility software at a low cost, and integrate it conveniently with my current ERP/MES data warehouses to better forecast the production times of parts. This capability will give me enhanced visibility into shop floor production schedules, and lead to better inventory management and reduced WIP.

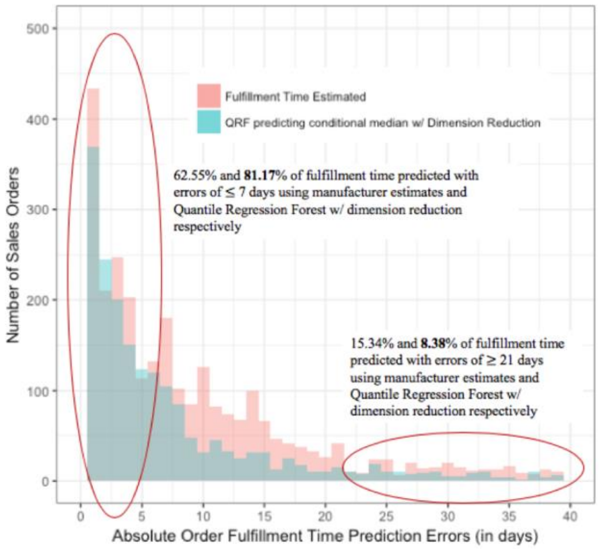
  Results:

  We also analyzed sales order (SO) data for our SME partner. The dataset comprised 4,021 orders after we removed the orders with missing fields. The orders were placed over a rather large time span from April 2006 to August 2017. We considered four predictor variables, namely, SO description, quantity, remaining quantity (portion that had not yet been fulfilled), and expected fulfillment time (in days). SO description was a categorical variable, whereas, all the other variables were continuous. Basic text mining was performed on SO description to reduce the categorical levels from 128 to 20. The (estimated) SO fulfillment time was the response variable.

Table 2. SO Fulfillment Times Prediction Errors Using Heuristic Estimates and Different Regression Models

Approach		**25th percentile**	**50th percentile**	**75th percentile**	**90th percentile**	**95th percentile**
Fulfillment estimates		1.00	4.00	13.00	33.00	63.00
Regression model	**Linear Regression**	6.90	14.66	26.01	50.87	87.79
	RF	4.16	9.60	21.72	45.31	74.86
	QRF conditional mean	3.23	7.06	14.82	34.55	57.82
	QRF conditional median	**0.00**	**1.00**	**5.00**	**16.50**	**34.00**
	QRF conditional 1st quantile	1.00	7.00	15.00	34.25	61.00
	QRF conditional 3rd quantile	1.00	6.00	14.00	28.00	40.75

As reported in Table 2, the QRF model with condition median substantially outperforms all the other models, and most importantly, the currently-used fulfillment time estimates. Dimension reduction is used for all the models. Quantitatively, QRF with conditional median yields 50% improvement in prediction accuracy as compared to the current estimates for 90% of the SOs, thereby, reducing the absolute prediction error to a little over two weeks from nearly five weeks. Interestingly enough, linear regression, random forest, and the other QRF variants do not perform well in this dataset. In fact, they are often worse than the current estimates. This finding further highlights the importance of developing a fairly sophisticated machine learning model that captures the appropriate quantile of the conditional distributions of the response and predictor variables in a way that is robust to the outliers in the data. Figure 10 illustrates this benefit further. Our model results in 45% less SOs with at least three weeks of fulfillment time prediction errors, and a substantially higher proportion of SOs with less than one week or two weeks of prediction errors.



62.55% and **81.17%** of fulfillment time predicted with errors of ≤ 7 days using manufacturer estimates and Quantile Regression Forest w/ dimension reduction respectively

15.34% and **8.38%** of fulfillment time predicted with errors of ≥ 21 days using manufacturer estimates and Quantile Regression Forest w/ dimension reduction respectively

Figure 10. Histogram of SO Fulfillment Times Prediction Errors (for Less than 60 Days) Using Supplier Estimates and QRF Predicting Conditional Median with Dimension Reduction

After constructing the models, to enable the real time visibility of the supply chain operations, we developed an easy-to-use website using R. The users can use the model by selecting appropriate parameters in the website and training the model. The visibility tool can facilitate the decision making of the users by providing various helpful information and analysis results.

The tool has implemented the above-mentioned predictive models and basic data visualization plots. In order to customize the design of the interface according to the specific needs of the end users, we performed a structured usability testing. The study is conducted on a designed website which contains 3 basic components including a landing page, practice scenarios and experimental study scenarios. The landing page includes the consent form and trust-related questions. The participants can securely log into the tool with their names and predefined passwords. The practice scenarios give a quick overview of how to use the visibility tool. And the actual experimental study scenarios consist of six different testing views that have different combinations of information content and information complexity. It is based on the method of design of experiments. All the views consist of different elements of data table, graphs, inputs and outputs from the predictive models. The general procedure of this study will be to first provide the details of the purpose of the study through the consent form, then allow the participants to practice using the different scenarios, and finally record their interactions and satisfaction levels via six different experimental scenarios and a series of follow-up questions.

After building the usability study tool, ten supply chain personnel at our SME partner were recruited to evaluate the usefulness and usability of the visibility tool. The results including the operation details of the participants, correct answers and user answers, time stamps of all the operations, and the responses to the follow-up questions are recorded using Google sheets. Multivariate Analysis of Variance (MANOVA) was conducted to analyze the recorded data.

Table 3 presents the accuracy for different displays of usability testing corresponding to different combination of data complexity and data volume. Significant difference was observed for different displays with varying data complexity. It shows that low data complexity can better facilitate the decision making of the users.

Table 3. Accuracy for Different Visibility Tool Usability Testing Scenarios

Data complexity / data volume	Accuracy
Low / Low	100 %
Low / Medium	100 %
Low / High	100 %
High / Low	85 %
High / Medium	81 %
High / High	81 %

We also analyzed the response times of the users using different displays. The box plot of response times is shown in Figure 11. Similar to the trend of accuracy, the scenarios with low data complexity have significantly shorter response time compared to high data complexity scenarios. The results did not indicate any significant difference across different data volume.

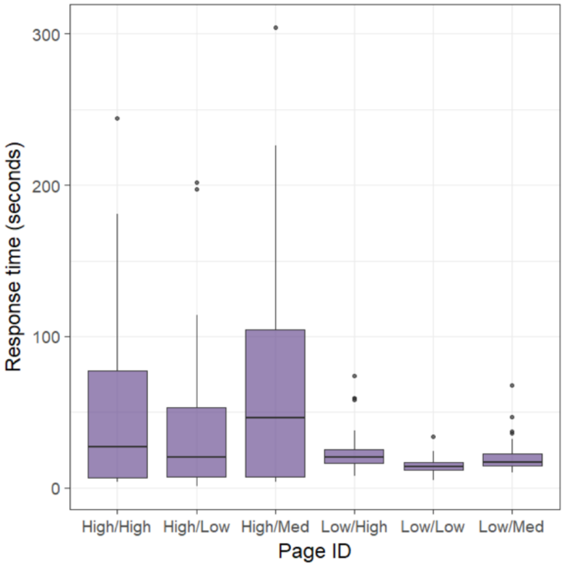


Figure 11. Box Plot of Response Times for Different Visibility Tool Usability Testing Scenarios

For the follow-up questions after the usability testing, the questions are designed to evaluate the usability and trustworthiness of the visibility tool. Figure 12 shows the results for questions related to usability of the tool. We observe that a large number of participants had neutral views for high complexity displays, and a majority of participants had favorable views for low complexity displays.

However, when comparing low data complexity across the different levels of volume of data, there were no clear indicators of which views were the best. While analyzing the trust survey questions, the results show that a majority of the users had a high or neutral trust across the three different views as shown in Figure 13.

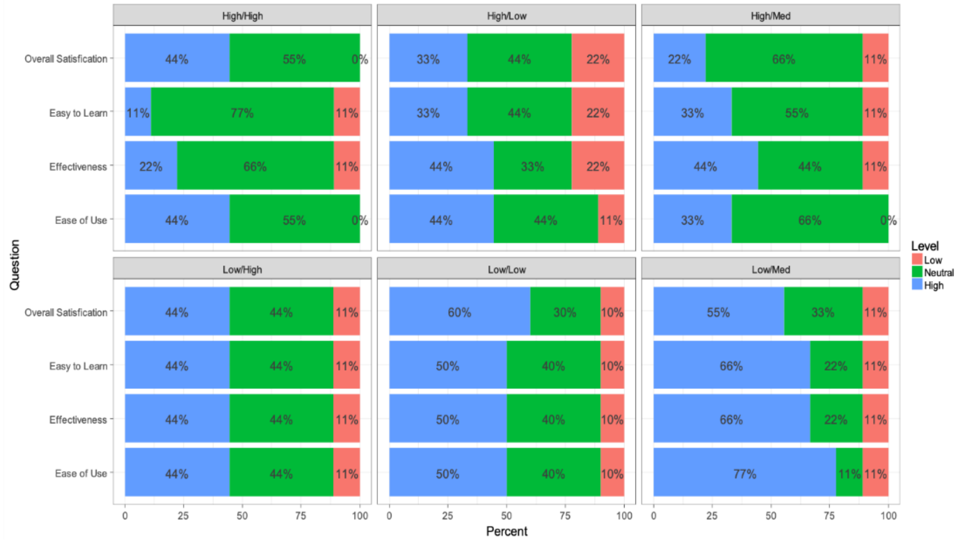


Figure 12. Summary Results of SME Sales Order Personnel Opinions on the Usability of Various Visibility Dashboard Interfaces

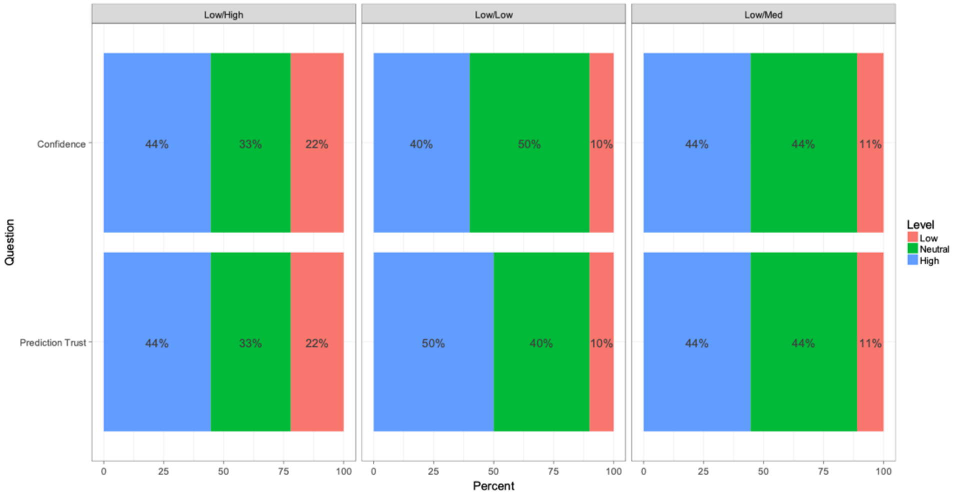
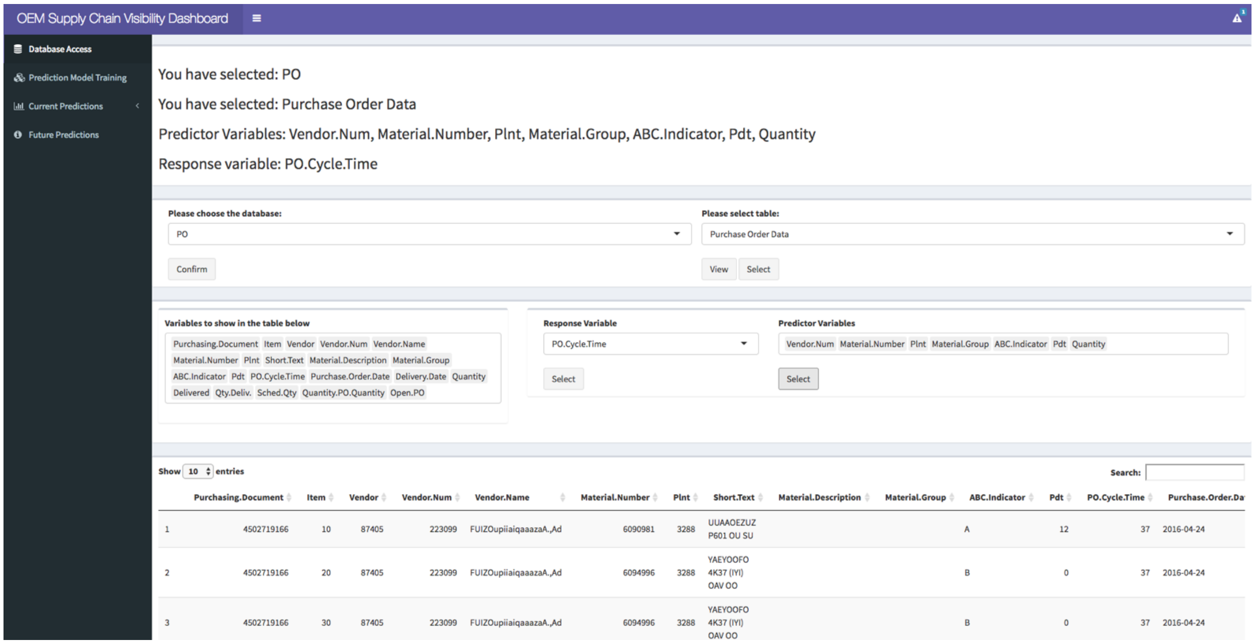


Figure 13. Summary Results of SME Sales Order Personnel Opinions on the Trustworthiness of Various Visibility Dashboard Interfaces

V. ACCESSING THE TECHNOLOGY

The entire software is made available free of cost as a readily installable Docker package to all the DMDII members. The source code is open source under the MIT License. It has been shared as a .zip file with the Program Manager who would distribute or share it as required. The package includes a README file for the installation instructions and a documentation file to provide details on the source code functions and visibility dashboard capabilities. The software is built using only open-source programming languages and platforms; the users would need to install docker, dock-compose, R, and MySQL in their local machines before running the software. Technical support would only be provided in the form of answering questions regarding installation issues.

The visibility tool, accessible directly at https://dmdii.shinyapps.io/oemdashboard/ with any username and *test* as password, has been developed with ease-of-use as one of the primary considerations (also indicated by the usability study discussed earlier). It includes the following five views or pages: 1) *database access* to select the appropriate MySQL database hosting the data to be analyzed along with the desired predictor and response variables (see Figure 14); 2) *prediction model training* to learn the desired regression models with tunable parameters either in serial or parallel mode (see Figure 15); 3) *delivery times predictions* to interactively visualize and drill down into the outcomes of the trained regression models (see Figure 16); 4) *delay risk identification* to quickly detect the extents of potential delays in receiving certain open POs (see Figure 17); and 5) *future orders predictions* to estimate the delivery times of parts corresponding to POs that are still in the planning stage (see Figure 18).



You have selected: PO

You have selected: Purchase Order Data

Predictor Variables: Vendor.Num, Material.Number, Plnt, Material.Group, ABC.Indicator, Pdt, Quantity

Response variable: PO.Cycle.Time

Please choose the database: PO

Please select table: Purchase Order Data

Variables to show in the table below: Purchasing.Document Item Vendor Vendor.Num Vendor.Name Material.Number Plnt Short.Text Material.Description Material.Group ABC.Indicator Pdt PO.Cycle.Time Purchase.Order.Date Delivery.Date Quantity Delivered Qty.Deliv. Sched.Qty Quantity.PO.Quantity Open.PO

Response Variable: PO.Cycle.Time

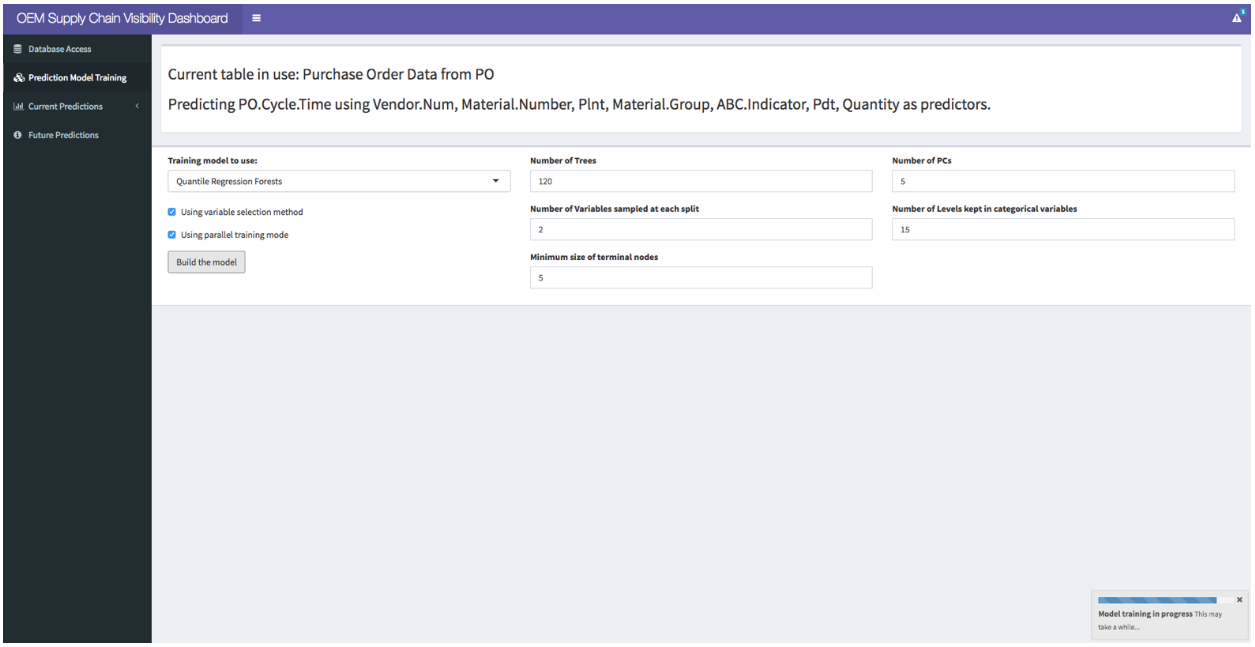
Predictor Variables: Vendor.Num Material.Number Plnt Material.Group ABC.Indicator Pdt Quantity

Show 10 entries

Search:

	Purchasing.Document	Item	Vendor	Vendor.Num	Vendor.Name	Material.Number	Plnt	Short.Text	Material.Description	Material.Group	ABC.Indicator	Pdt	PO.Cycle.Time	Purchase.Order.Da
1	4502719166	10	87405	223099	FUIZOupiiaiqaaazaA.,Ad	6090981	3288	UUAAOEZUZ P601 OU SU			A	12	37	2016-04-24
2	4502719166	20	87405	223099	FUIZOupiiaiqaaazaA.,Ad	6094996	3288	YAEYOOFO 4K37 (IYI) OAV OO			B	0	37	2016-04-24
3	4502719166	30	87405	223099	FUIZOupiiaiqaaazaA.,Ad	6094996	3288	YAEYOOFO 4K37 (IYI) OAV OO			B	0	37	2016-04-24

Figure 14. Visibility Dashboard: Database Access


OEM Supply Chain Visibility Dashboard
Database Access
Prediction Model Training
Current Predictions
Future Predictions

Current table in use: Purchase Order Data from PO

Predicting PO.Cycle.Time using Vendor.Num, Material.Number, Plnt, Material.Group, ABC.Indicator, Pdt, Quantity as predictors.

Training model to use: Quantile Regression Forests

☑ Using variable selection method

☑ Using parallel training mode

Build the model

Number of Trees: 120

Number of Variables sampled at each split: 2

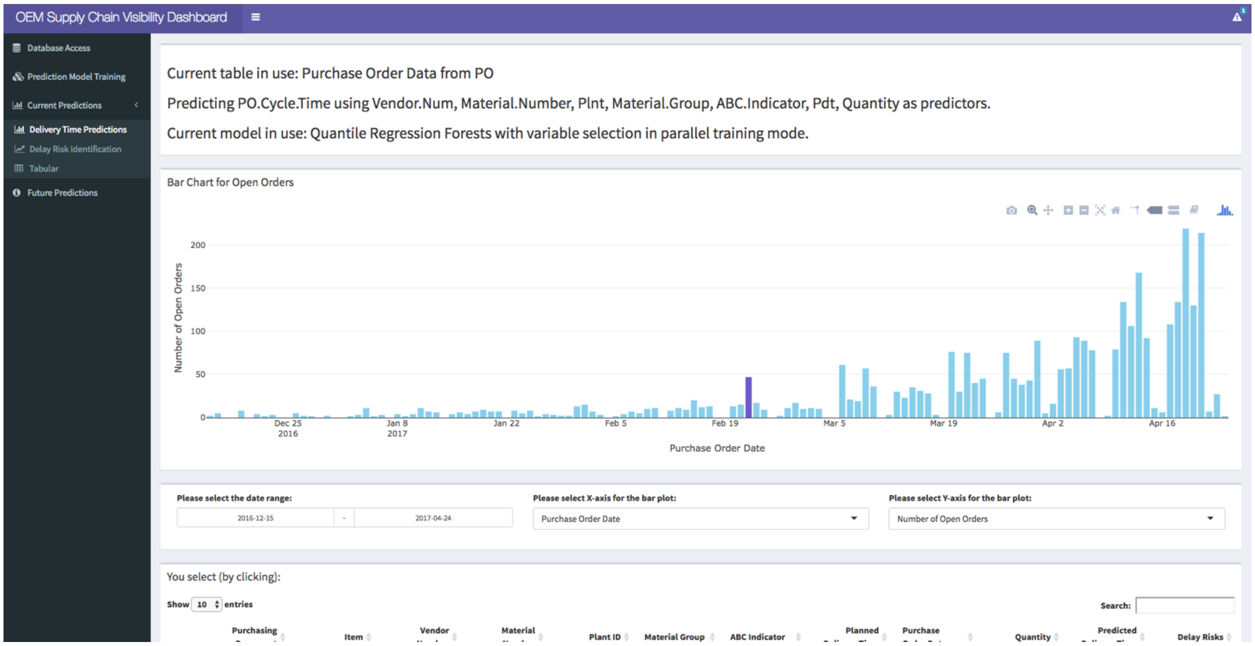
Minimum size of terminal nodes: 5

Number of PCs: 5

Number of Levels kept in categorical variables: 15

Model training in progress This may take a while...

Figure 15. Visibility Dashboard: Prediction Model Training



Current table in use: Purchase Order Data from PO

Predicting PO.Cycle.Time using Vendor.Num, Material.Number, Plnt, Material.Group, ABC.Indicator, Pdt, Quantity as predictors.

Current model in use: Quantile Regression Forests with variable selection in parallel training mode.

Please select the date range: 2016-12-15 - 2017-04-24

Please select X-axis for the bar plot: Purchase Order Date

Please select Y-axis for the bar plot: Number of Open Orders

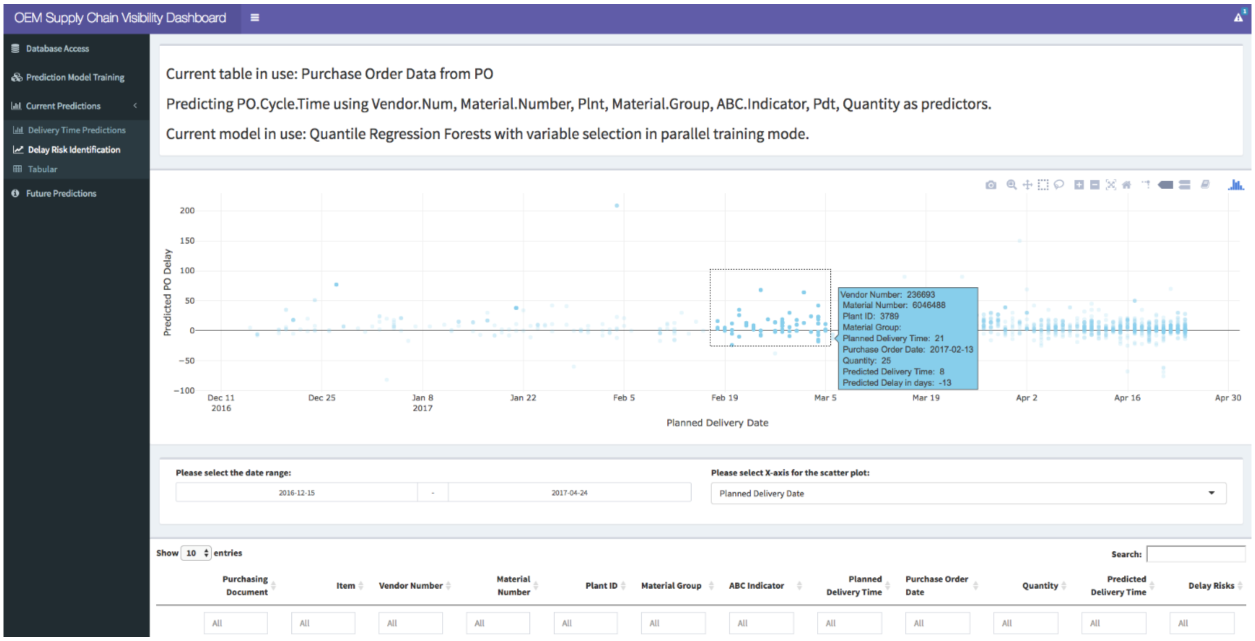
You select (by clicking):

Show 10 entries

Search:

Purchasing	Item	Vendor	Material	Plant ID	Material Group	ABC Indicator	Planned	Purchase		Quantity	Predicted	Delay Risks

Figure 16. Visibility Dashboard: Delivery Times Predictions



Current table in use: Purchase Order Data from PO

Predicting PO.Cycle.Time using Vendor.Num, Material.Number, Plnt, Material.Group, ABC.Indicator, Pdt, Quantity as predictors.

Current model in use: Quantile Regression Forests with variable selection in parallel training mode.

Vendor Number: 236693
Material Number: 6046488
Plant ID: 3789
Material Group:
Planned Delivery Time: 21
Purchase Order Date: 2017-02-13
Quantity: 25
Predicted Delivery Time: 8
Predicted Delay in days: -13

Please select the date range: 2016-12-15 - 2017-04-24

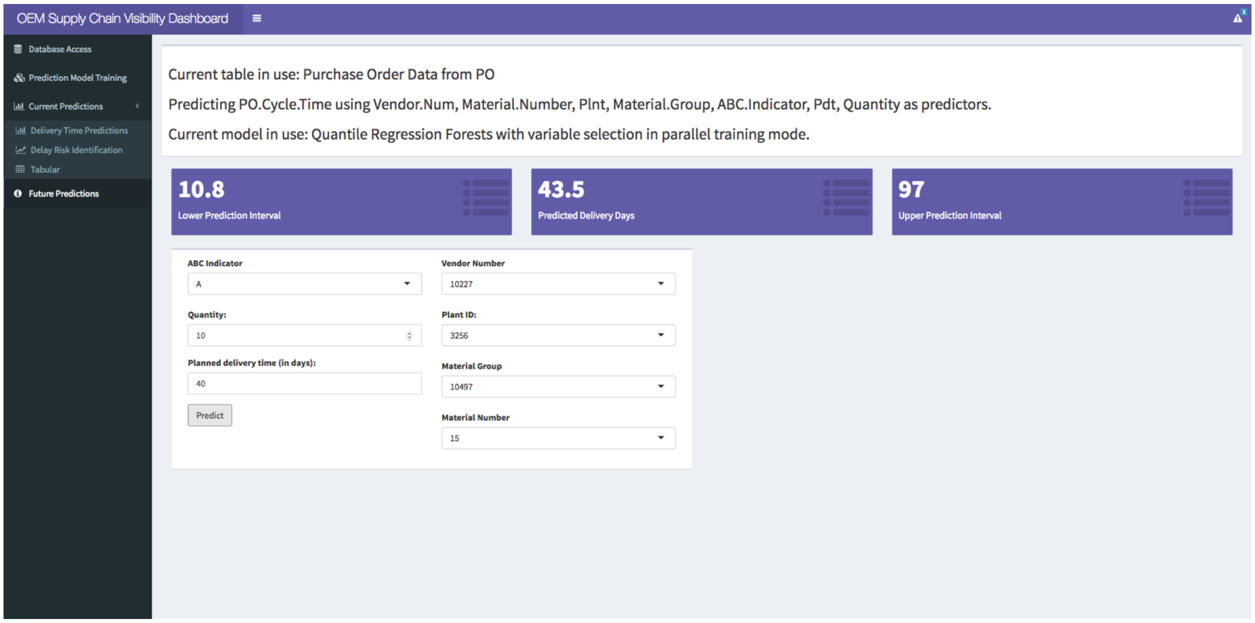
Please select X-axis for the scatter plot: Planned Delivery Date

Show 10 entries

Search:

Purchasing Document	Item	Vendor Number	Material Number	Plant ID	Material Group	ABC Indicator	Planned Delivery Time	Purchase Order Date	Quantity	Predicted Delivery Time	Delay Risks
All	All	All	All	All	All	All	All	All	All	All	All

Figure 17. Visibility Dashboard: Delay Risk Identification


OEM Supply Chain Visibility Dashboard
Database Access
Prediction Model Training
Current Predictions
Delivery Time Predictions
Delay Risk Identification
Tabular
Future Predictions
Predict

Current table in use: Purchase Order Data from PO

Predicting PO.Cycle.Time using Vendor.Num, Material.Number, Plnt, Material.Group, ABC.Indicator, Pdt, Quantity as predictors.

Current model in use: Quantile Regression Forests with variable selection in parallel training mode.

10.8 Lower Prediction Interval

43.5 Predicted Delivery Days

97 Upper Prediction Interval

ABC Indicator: A

Quantity: 10

Planned delivery time (in days): 40

Vendor Number: 10227

Plant ID: 3256

Material Group: 10497

Material Number: 15

Figure 18. Visibility Dashboard: Future Orders Predictions

VI. INDUSTRY IMPACT & POTENTIAL

a. Specific Market Impact

Our software tool is specifically targeted toward two business segments: a) sourcing and inventory organizations with large OEMs, particularly in the aerospace industry; and b) sales and production units within SMEs particularly engaged in manufacturing heavy-duty aerospace and ground transportation equipment. If adopted for daily use, we expect our tool to result in (tens of) millions of dollars in savings annually for both the target segments, with the amount of savings roughly proportional to the net worth of the parts being procured or supplied.

b. Usefulness in Other Industries

Our tool is fundamentally not limited in its use to any specific industry type. In fact, it would be of value to any OEM or SME that is engaged in procuring or supplying a substantial number of parts with large lead times of the order of several months, regardless of where and how the parts would be used in the final assemblies.

c. Next Steps Based on Other Use Potential

The next steps based on the other potential uses of our supply chain visibility tool are listed below:

- Development of additional capabilities for inventory management and predicting parts flows across tiered supply chain networks; such development would be greatly facilitated by the availability of data, shared in a secure manner without providing proprietary information using a blockchain system, for example, from all the organizations involved in the supply chain networks
- Integration of the visibility tool with organizational data warehouses, such as ERP/MES systems, to push data for predictive analysis as and when required

VII. TECH TRANSITION PLAN & COMMERCIALIZATION

a. Future Plans

The project team has no commercialization plan. Instead, it has made the software tool available in open-source format, both for direct use or testing and further development or integration with existing supply chain management (SCM) solutions, to all the DMDII members. Detailed installation instructions and software documentation have been provided, which should enable all the interested users to start adopting this tool within their own organizations. Future technical support would, however, be limited to answering installation-related questions only. The project team does not assume any responsibility for debugging or upgrading any of the software capabilities based on the specific requirements of any adopting organization. The team is, however, open to the possibility of further developing the tool as a part of a sponsored project, either or an organizational basis, or for the DMDII consortium, or for the government.

b. Identified Barriers to Adoption

There are two primary barriers to widespread adoption of our software tool. They are listed below:

- A substantial implementation effort is required to develop an automated mechanism to push supply chain transactional data from existing data warehouses, which may be multiple, disparate sources with varying levels of stored data quality, into a MySQL database with standardized data fields whenever analysis is required.
- Significant engagement efforts are needed to train the end users (sourcing managers, production engineers, etc.) on using the software tool regularly to facilitate their own day-to-day decision making in terms of both planning and forecasting.

c. Additional Information to Consider

The project team is willing to partner with other commercial organizations, both manufacturing and technology, to build upon the capabilities of the developed supply chain visibility tool and convert it into a plug-and-play tool that works with a majority of ERP/MES and SCM software systems.

VIII. WORKFORCE DEVELOPMENT

As a part of workforce development, we developed and shared tutorials on how to design and conduct structured user studies to evaluate the usefulness of software systems.

In addition to the mandatory project quarterly and final presentations to all the DMDII stakeholders, we gave three conference presentations on the design, development, and validation of our supply chain predictive analytics tool. While one of the conferences was geared toward the high performance computing community, the other two conferences, organized by the Institute for Industrial and Systems Engineers (IISE) and the American Society of Mechanical Engineers (ASME), respectively, were two of the most widely-attended forums by the broad product design and manufacturing community. Our presentations were very well received at all these forums, and we got several follow-up emails requesting us to share related documentation and provide the link to the interactive visibility tool.

Last but not the least, we gave a live demonstration of our tool at the June 2018 DMDII Supply Chain Exhibit and Symposium. Once again, we received positive feedback and many similar requests to provide additional documentation and software link.

IX. CONCLUSIONS/RECOMMENDATIONS

To conclude, it would be fair to say that our project was overall successful. While it required quite a bit of coordination and negotiation among the different partners, at the end, we accomplished most of the objectives laid out at the beginning of the project. It was particularly satisfying to validate our basic hypothesis that historical data-driven machine learning models can, indeed, accurately predict the delivery times of current POs and fulfillments times of active SOs, thereby, providing an opportunity to *use the models outcomes as surrogate measures of parts and materials availabilities* to alleviate the issue of limited information exchange in supply chain systems. It was also gratifying to integrate the models within an interactive visualization tool that was refined based on the feedback of end users and demonstrated live to the various supply chain stakeholders.

Of course, we learned a fair number of lessons and made quite a few changes to our original scope of work, which are described in the next Section. Our recommendations, therefore, would be two-fold: a) facilitate and promote deployment of our tool within the existing SCM solutions of both OEMs and SMEs; and b) continue supporting this line of research and development after streamlining the effort based on the tool users experiences and taking necessary steps to prevent recurrences of our encountered problems. We firmly believe that our open-source software would prove invaluable in saving millions of dollars in production time, inventory, and human resources for large enterprises and small manufacturers alike.

X. LESSONS LEARNED

a. Problems Encountered

We encountered a few problems in executing this project. They are listed below:

- It was challenging to figure out a suitable method for selecting the key levels for the categorical variables in our supply chain datasets.
- Even though training time is reduced with no corresponding loss in prediction accuracy for moderate parallelization of the supervised learning models, we realized that it is not always beneficial to parallelize model construction fully just because the processor has multiple cores.
- It took a substantial amount of time to acquire useful OEM and SME data, which delayed the process of developing the visibility dashboard.
- It was very challenging to create a good quality dataset linking material flows across suppliers and buyers in multi-tier supply chain networks with different contractual relationships; ultimately, this effort did not work out within the project time frame.

b. Plan/Scope of Work/Proposal Claim Deviations

Based on the encountered problems and useful feedback received from potential end users of our predictive analytics-based visibility tool, we made the following changes to our original scope of work:

- We implemented simple text mining methods to extract some useful information from the qualitative descriptions of the parts and materials.
- We developed dimension reduction methods to select the most important levels for the categorical variables in our datasets.
- We adopted random forest and quantile regression forest instead of a hybrid combination of stepwise linear regression and Bayesian implementation of non-Gaussian multivariate distribution fitting as our prediction models.
- We deployed the visibility tool using Docker instead of a DMC service.
- We parallelized the process of constructing the prediction models on personal computers (both Windows and MAC OS) instead of using high performance computing (HPC) resources to enable more widespread adoption of the models in the business community

- We could not run the final visibility tool user study with our OEM partner due to time and budgetary constraints; as a result, we could not validate, in a statistically significant way, whether the dashboard would be more trustworthy and more likely to be used on a daily basis than the existing systems
- We could not deploy the visibility tool on the SME shop floor due to budgetary constraints

c. Risks Realized

We were successful in realizing the following risky propositions stated in our work plan:

- We achieved close to or slightly more than the proposed goal of 50% improvement for certain KPIs (orders deliveries and completion times prediction accuracy).
- We demonstrated real-time PO delivery times predictions using an interactive supply chain visibility tool.
- We showed that low information complexity displays would result in high accuracy and favorable opinions among the end users (sales order personnel) of our visibility tool.

XI. DEFINITIONS

What follows are a set of definitions, terms, and acronyms used in this document. These definitions are gathered from various source including the internet, reference papers, standards organizations, and the authors of these document.

- OEM: Original equipment manufacturer
- SME/SMM: Small and medium-scale enterprise/manufacturer
- LR: Linear regression
- PCA: Principal component analysis
- RF: Random forest
- QRF: Quantile regression forest
- KPI: Key performance indicator
- PO: Purchase order
- SO: Sales order
- ERP: Enterprise resource planning
- MES: Manufacturing enterprise system
- DMC: Digital manufacturing commons
- WIP: Work in progress
- SCM: Supply chain management

XII. APPENDICES

User manual for supply chain visibility dashboard software tool

- Available Features

Database Access View

In this view, the users can specify the database and available tables within it to be used by the training process. After choosing the database, an authentication step is needed. The users can also view the specific table by choosing the column names in the multiple selection box. The number of rows to be displayed is adjustable, and the users can search for useful information in the table. Finally, the users need to specify the response variable and predictor variables. At the top of the page, all current information is displayed to help the users keep track of the dashboard status.

Key functions: Specify the database and table, view the table, specify the response variable and predictor variables.

Basic operations:

1. Use the dropdown selection box to choose the desired database to use.
2. Press confirm to submit your selection and put in your name and password for authentication.
3. After successful authentication, the table selection box will pop up and you should use it to identify the table you want to analyze.
4. The two selections, View and Select, are available for the selected table. Click on View first and the data table will show up with further options.
5. Use “Variable to show in the table below” box to select multiple variables that you want to view in the table below. You can order the table by different column and do advanced search use the search box in the upper right corner with the data table area. Additionally, you can select how many rows to be shown in the data table.
6. After you have decided the table to use, click on Select to confirm your selection.
7. Finally, you should use the response variable selection box and the predictor variables selection box to specify the response and predictors in your prediction model. Note that you are forced to select only one response and any number of predictors.

Prediction Model Training View

The users can select the training model to use and all the parameters used by the model. Additionally, the users can choose whether to use parallel training mode and variable selection method. If the variable selection method is selected, the users need to specify number of principal components to be extracted using principal component analysis method and number of levels to be kept in each categorical variable. After training the model, the detailed information about the model is shown on the top of the page. To let the users keep track of the training process, a progress bar on the bottom right will indicate the model training progress.

Key functions: Specify the parameters of the model, choose whether to use parallel training mode and variable selection method or not, show training progress.

Basic operations:

1. In this view, you will train the model and set different parameters. It is suggested to use the default settings. First select the desired prediction models to train in the dropdown selection box.

2. Specify whether to use variable selection method and parallel training mode.
3. For Random Forest and Quantile Regression Forest, set the number of trees to grow, number of variables sampled at each split and minimum size of terminal nodes.
4. For variable selection method, set the number of Principle Components to be exacted from the dataset and number of levels kept in each categorical variable.
5. Finally, click on build the model to train it.

Current Predictions View

Delivery Time Predictions Page

A bar chart for open orders is shown on this page. The users can hover on each bar using their mouse, and more information about that bar will pop up as hover info. Additionally, the users can click on each bar. The bar will change color and the detailed information will be shown in the data table below. The users can select different time range, x-axis and y-axis for the bar plot.

Key functions: View current open orders, select parameters for the bar plot, multiple interactive methods.

Basic operations:

1. Select the date range you want to view.
2. Select the x-axis and y-axis using the two dropdown selection boxes.
3. Click on a bar, then the detailed information will show up in the table below.
4. Sort or search using the table advanced options.

Delay Risk Identification Page

A scatter plot shown delay risks is used to help the users to identify the most risky open orders. Delay risk is defined as the absolute difference between predicted delivery time and planned delivery time. The users can specify the date range and x-axis of the plot. Similar to the bar plot, the user can hover on each dot to view more information and select a rectangular area to show detailed information in the data table below.

Key functions: View delay risks, select parameters for the bar plot, multiple interactive methods

Basic operations:

1. Select date range you want to view.
2. Select the x-axis for the scatter plot above.
3. Select a rectangular are using the cursor in the scatter plot, then the detailed information of the selected area will pop up in the data table below.
4. Sort or search using the table advanced options.

Tabular Page

The users can view open purchase orders and closed purchase orders using this page. The detailed information of each PO will be shown in the table format. The users can search and specify different parameters for the table.

Key functions: View open/closed POs.

Basic operations:

1. Select Closed POs or Open POs using the dropdown menu.
2. Sort or search using the table advanced options.

Future Prediction View

This view is to help users make predictions on future purchase orders. The users can specify the order details using the input area. The inputs correspond to the parameters chosen in the model training view. The results will be shown as a confidence interval and a prediction value.

Key functions: Make predictions on future purchase orders.

Basic operations:

1. Based on the model, different parameters needed to be set using the selection box.
2. Select each purchase order settings using the dropdown menus.
3. Click on predict, then the results will show up in the top boxes.

- Implementation Details

The installation steps are in the README file. Please follow the steps in the documentation to install the whole package to your local machine. The whole software resides in a docker container. Hence, please make sure your local machine has docker installed.

MySQL Database

The software uses MySQL database to store all the data. Please make sure your local machine is compatible with MySQL server. The current example software has a test MySQL database installed. If you want to use your own database, please import your data to MySQL database and configure the corresponding connection in the web files which will be explained in detail later.

R Files

global.R

This file contains login information of the database and Javascript functions for rendering the plots. Please change the database settings to connect to your local database correspondingly. However, do not change the rendering functions without specific needs.

ui.R

This file contains all the front-end visualization functions. Header menu is defined in header function which includes title and tile width, etc. Side bar menu is defined in sidebar function which defines all the tabs and subtabs of the dashboard. Finally, the body function defines each page layout individually.

server.R

All the major front-end and back-end interactions are defined in this file.

Line 2-74 define the database authentication function. If login successfully, the following operations will be granted.

Line 140-405 define the core functions of model training process. Please do not change this part of codes without special assistant. The built-in models are Linear Regression, Random Forest and Quantile Regression Forest. The default model is already set to QRF predicting conditional median using parallel training mode. If you would like to change the default model, please refer to the ui.R file. Note that the parallel training mode require multi-thread programming. Please make sure your operation system has multi-core and multi-thread support.

Line 466-522 define the bar plot of current prediction visualization. Line 554-600 define the scatter plot of current prediction visualization. These functions have already defined the whole visualization process. Please do not change this part without special assistant.

Line 632-681 define the future prediction tab. The model used in the prediction function is already defined in the model training function. If you want to change the width of confidence interval, please modify ‘what’ parameter in line 643 and line 652. To predict other quantiles, please also modify this parameter.