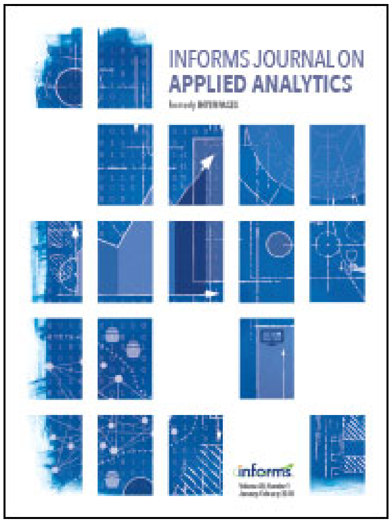


INFORMS Journal on Applied Analytics

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

A Machine Learning-Based System for Predicting Service-Level Failures in Supply Chains

Gabrielle Gauthier Melançon , Philippe Grangier , Eric Prescott-Gagnon , Emmanuel Sabourin , Louis-Martin Rousseau

informs
http://pubsonline.informs.org/journal/inte

INFORMS JOURNAL ON APPLIED ANALYTICS
Vol. 51, No. 3, May–June 2021, pp. 200–212
ISSN 0092-2102 (print), ISSN 1526-551X (online)

A Machine Learning-Based System for Predicting Service-Level Failures in Supply Chains

Gabrielle Gauthier Melançon,[a,b,*] **Philippe Grangier,**[c] **Eric Prescott-Gagnon,**[b] **Emmanuel Sabourin,**[b] **Louis-Martin Rousseau**[a]

[a] Polytechnique Montréal, Montréal, Québec H3T 1J4, Canada; [b] Element AI, Montréal, Québec H2S 3G9, Canada; [c] IVADO Labs, Montréal, Québec H2S 2J9, Canada

*Corresponding author

Contact: gabrielle.gauthier-melancon@polymtl.ca, https://orcid.org/0000-0003-3240-685X (GGM); phi.grangier@gmail.com (PG); ericprescottgagnon@gmail.com (EP-G); emmanuel.sabourin@blueyonder.com (ES); louis-martin.rousseau@cirrelt.net, https://orcid.org/0000-0001-6949-6014 (L-MR)

Received: May 13, 2019
Revised: November 20, 2019; February 20, 2020; July 9, 2020
Accepted: July 13, 2020
Published Online in Articles in Advance: April 23, 2021

https://doi.org/10.1287/inte.2020.1055

Abstract. Despite advanced supply chain planning and execution systems, manufacturers and distributors tend to observe service levels below their targets, owing to different sources of uncertainty and risks. These risks, such as drastic changes in demand, machine failures, or systems not properly configured, can lead to planning or execution issues in the supply chain. It is too expensive to have planners continually track all situations at a granular level to ensure that no deviations or configuration problems occur. We present a machine learning system that predicts service-level failures a few weeks in advance and alerts the planners. The system includes a user interface that explains the alerts and helps to identify failure fixes. We conducted this research in cooperation with Michelin. Through experiments carried out over the course of four phases, we confirmed that machine learning can help predict service-level failures. In our last experiment, planners were able to use these predictions to make adjustments on tires for which failures were predicted, resulting in an improvement in the service level of 10 percentage points. Additionally, the system enabled planners to identify recurrent issues in their supply chain, such as safety-stock computation problems, impacting the overall supply chain efficiency. The proposed system showcases the importance of reducing the silos in supply chain management.

History: This paper was refereed.

Keywords: supply chain management • manufacturing • machine learning • human–computer interface • explainable AI

Supply chain planning typically comprises multiple optimization systems that differ in scope and planning horizon, from strategic sales and operations planning to near-real-time transportation systems. Despite advanced planning and execution systems, manufacturers and distributors tend to observe service levels below their targets, owing to different sources of uncertainty throughout the supply chain. Supply chain risk management (SCRM) is the field dedicated to identifying these risks and mitigating them.

Jüttner et al. (2003) classifies sources of uncertainty in supply chains between environmental, organizational, and network risks. Environmental risks refer to the impact of the environment on the supply chain, such as natural disasters and political factors. Organizational risks include uncertainty within the supply chain, such as delays in transport or production problems. Lastly, network risks refer to issues due to poor interactions between the subparts of the supply chain. Indeed, as supply chain planning requires the collaboration of different teams and systems that need to be tightly integrated, issues in systems configurations may be undetected and lead to suboptimal plans. Jüttner et al. (2003) underscore that network risks are a very impactful, but often neglected, source of risks. As part of SCRM, risk mitigation is an area of research that focuses on building a robust plan that can account for different sources and magnitudes of uncertainty. Yet, it can be too costly to plan for the worst outcomes, and it is impossible to model uncertainty perfectly. Consequently, service-level failures can occur in supply chains.

A steady stream of research applying machine learning (ML) to the field of supply chain recently emerged because of recent advances in ML and the growing availability of data (Nguyen et al. 2018). In this paper, we present a system that uses ML to raise alerts when the supply chain conditions may lead to service-level failures. The alerts need to anticipate issues in time for the planners to take corrective actions, but not so early that the next plan naturally accounts for them. The system focuses the attention of the planners on alerts that are *actionable* (it is possible to avoid the failure), *exclusive* (other systems did not detect the issues), and *significant* (failures concern important items for which performing the corrective action is worthwhile). The system also aims to explain alerts by identifying their underlying causes, so users gain confidence in the results and get the necessary context for potential fixes. We developed the system in cooperation with Michelin, an international tire manufacturer, which provided the business-use case and the data.

In the remainder of this paper, we first present Related Work, followed by the section Michelin Context, introducing some of Michelin's current challenges and how we frame the problem. In the Methodology section, we outline how we model the problem and generate predictions using ML. In the User Interaction with the Model section, we present the workflow and user interface (UI) that we developed so that the system's predictions are useful and explainable. We follow with the Results section. Eventually, we give some perspectives on our work and the remaining challenges for ML to have a deep impact on supply chain management in Perspectives. The appendix contains more details on the model and the data.

Related Work

Nguyen et al. (2018) recently published a survey of big-data analytics for supply chains that classifies the studies by supply chain functions, including demand management, manufacturing, warehousing, and general supply chain management when the study encompasses multiple functions of the supply chain together. In the survey, papers falling in the latter category are either descriptive or prescriptive approaches on topics such as managing sustainability (Papadopoulos et al. 2017) or natural-disaster risk management (Ong et al. 2015), but the authors do not report any predictive approaches. They highlight the increased usage of ML for specific areas of supply chain management, such as demand forecasting and machine maintenance. Our approach, which would fall in the predictive applications for the general supply chain category, thus clearly stands out in the current domain's stream of research. Additionally, to our knowledge, no software vendors are offering a disruption-prediction tool ingesting data from several supply chain components simultaneously, at the time of this study.

In the SCRM literature, most papers focus on the identification of risks, such as Heckmann et al. (2015) and Kumar et al. (2010), or on their mitigation, such as Schmitt (2011) and Paul et al. (2017). Still, a few papers combine both. Simchi-Levi et al. (2015) propose methods to identify and mitigate risks in the automotive supply chain context at a tactical decision level. Some papers, such as Garvey et al. (2015) and Ojha et al. (2018), use Bayesian networks to model risk propagation on simulated data. In contrast, our approach predicts failures on a short-term horizon on real live data. Sharma et al. (2018) propose a similar technique, yet the study focuses on predicting failures among last-mile pickup and delivery services. Our work is centered around detecting failures in supply chain segments that are upstream of that last delivery step.

For more information on supply chain planning, we refer the reader to Stadtler and Kilger (2002), and to Khojasteh (2017) for SCRM specifically.

Michelin Context

Michelin is an international manufacturer that produces and sells tires for a vast range of vehicles, from cars and motorcycles to tractors and aircraft. Michelin produces roughly 200 million tires per year and has a commercial presence in 170 countries, reaching 13.7% of the global tire market in 2014. For the car-tires segment, Michelin distinguishes two channels: one for orders placed well in advance (typically large quantities, for car manufacturers or large retailers) and one for orders placed only a few days ahead (typically for local mechanics), called *store-and-sell*.

Service-Level Failure

In this study, we focus on the store-and-sell channel for car tires in Europe. For this channel, Michelin has a catalog of products, each with a given target delivery window. For example, a 48-hour delivery window means that a local garage can place an order and expect delivery within 48 hours. For the corresponding Michelin distribution center (DC), this translates into a deadline by which the items need to be available for delivery. Michelin considers a supply chain failure the inability to meet this deadline. Michelin tracks the performance of its supply chain with key performance indicators aggregating orders' service level over different scopes, such as product groups, regions, and periods. In this study, we focus on service levels aggregated by item and DC at a weekly frequency and consider a service-level failure to be a situation in which this aggregation is below Michelin's target.

Challenges and Opportunities

In an ideal world, Michelin's planners would continuously monitor the supply chain data and adjust its parameters when they detect situations or patterns that could lead to service-level failures. However, they are usually unable to monitor the data at a granular level because of the high volume of data and the complexity of supply chains. To illustrate, for the store-and-sell channel in Europe only, Michelin has around 4,000 different types of items, produced in 10 plants and stored in 15 DCs. Additionally, the planners' responsibilities are typically siloed between the different supply chain segments. As such, the ownership of the service level's performance is shared, increasing the complexity in identifying issues and mitigating them. For these reasons, planners adopt a proactive approach for their most important products only and resort to a corrective approach for the vast majority of their items—that is, adjusting the parameters only if the service level drops significantly below the target. This approach is reasonable because it is too costly to monitor all items. Yet, having a system that can predict situations at risk that require planners' attention would help in improving the supply chain performance. Moreover, such a system could detect problems that human planners would miss.

Sources of Service-Level Failure

Different sources of uncertainty in the supply chain cause problems that Michelin typically categorizes as execution issues or planning issues. *Execution issues* refer to situations in which the plan is adequate to fulfill the orders on time, but an event creates a disruption leading to a failure. It can be due to environmental and organizational risks, such as delays in shipments and machine failure. Execution issues are typically challenging to foresee and prevent. By *planning issues*, we mean that the plan is inadequate in fulfilling the right amount of items on time. Planning issues can be imputable to uncertainty in demand that the plan does not account for, or inadequate systems configurations. By *configurations*, we refer to the parameters and rules of the system that create the supply chain plans, such as safety-stock targets, demand-forecast settings, and master-plan parameters. Additionally, planning issues can be due to poor interaction between the different subparts of the supply chain—that is, network risks—as manufacturers tend to leverage heterogeneous systems with limited integration, either from different vendors or built in-house. Typically, it is possible to detect planning issues a few weeks in advance, and, as such, they can be predictive of future service-level failures.

In the store-and-sell channel, the flow of material starts at raw-material procurement and ends when retailers receive the tires. In previous internal studies, Michelin identified that planning and execution issues impacting the service level occur mostly at the stage of production, in internal logistics, and at DCs. As such, in this study, we ignore raw-material procurement, upstream logistics, and channel logistics, as shown in Figure 1. We, hence, measure service-level failures at the shipping door of the DCs. This has the added benefit of limiting the data effort.

Methodology

In this section, we present how we model the task of service-level-failure prediction from an ML perspective and describe our methodology to train such a model.

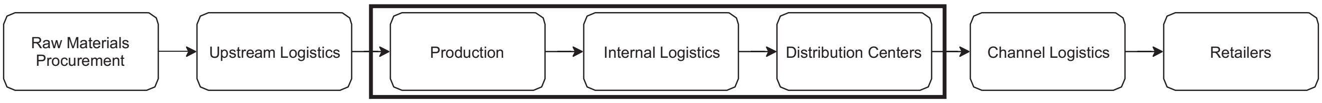
The Task

We define a *service-level failure* as a situation in which the service level for an item at a DC for a given week is below Michelin's service-level target, resulting in a binary variable. The model predicts the likelihood of a service-level failure. We call *alert* each model's prediction. Given the context of Michelin's supply chain, we use a prediction horizon of 14 days; that is, we generate predictions every week on Mondays for the week starting 14 days later for each item–DC pair. We use the mean service level in the training data set, 87.5%, as the model's baseline. This corresponds to 12.5% failures.

Features

We engineer features from the raw data to represent the state at each segment of the supply chain included in the rectangular box in Figure 1. Generally, the features compare the actual and planned values for a few periods before the moment of prediction so that they can measure uncertainty and deviations from the plan. Yet, historical data are not available for all possible factors impacting the service level, limiting us in our ability to model the situation at any given time precisely. Thus, we add features that are good proxies in estimating the uncertainty at each step of the supply chain. As an example, we do not have access to machine-failures data, but we can model if the output of the plants matches the plan, a good estimation for production-execution performance. Additionally, we incorporate the service level of the preceding weeks as features because they are good indicators of the global health of the supply chain. We also include as features some projections known at the moment of prediction that informed the current supply chain plan and indicate if it is likely to meet the projected need. We detail the features that we use in the appendix, as well as potential ones we would have liked to have used if data had allowed.

Figure 1. Flow of Materials in the Supply Chain



Note. As part of this study, we only include in the model the following segments: production, internal logistics, and distribution centers.

Model

We selected Gradient Boosted Decision Trees (GBDTs) (Friedman 2001), as implemented in XGBoost (Chen and Guestrin 2016), after some initial comparison with logistic regression, random forests, and neural networks. They offer the best performance overall, are easy and fast to train, and handle missing values out of the box. GBDTs are an ensemble of decision trees built iteratively, in which each tree's target is to correct the cumulative error made by preceding trees. The first tree generally trains on the delta between the targets and a baseline, typically the average value of the training set. Additionally, because they are trees-based, ML scientists usually consider GBDT models as being explainable, as detailed in the section User Interaction with the Model.

Evaluation Metrics

For unbalanced classes, such as failure prediction, performance metrics are typically areas under the receiver operating characteristics or precision–recall curves. To evaluate our models, we use the area under the precision–recall curve, which we obtain by plotting the precision (ratio of true positives versus all predicted positives) against the recall (ratio of true positives versus all positives). The precision–recall curve helps to directly quantify two metrics that Michelin cares about: ″How often is the system correct when it predicts a problem?″ (precision); and ″how many of the problems is it capturing?″ (recall).

User Interaction with the Model

In this section, we describe important features of the workflow and supporting UI that we developed for users to interact with alerts. A classical spreadsheet approach containing the model's predictions does not answer users' needs around trust, context, and explainability. The UI, in Figure 2, serves as a dashboard, where we display the outputs of the model, the raw data (such as stock levels, production plans, and logistics delays), and some additional data sources that can help identify the right resolution for the alert. By grouping all the information, users do not have to access multiple systems to get the necessary information and can save time to assess and resolve the case. It also increases the users' confidence in the model's results. We will present three essential features from the workflow and UI.

Supply Chain Health Check

First, as an entry point to explore all alerts, a heatmap near the top displays each item–DC combination, where the gradation represents the likelihood of failure. Items are on the x axis grouped by plant, whereas DCs are on the y axis grouped by region. At a glance, managers can assess the health of the supply chain and investigate correlations.

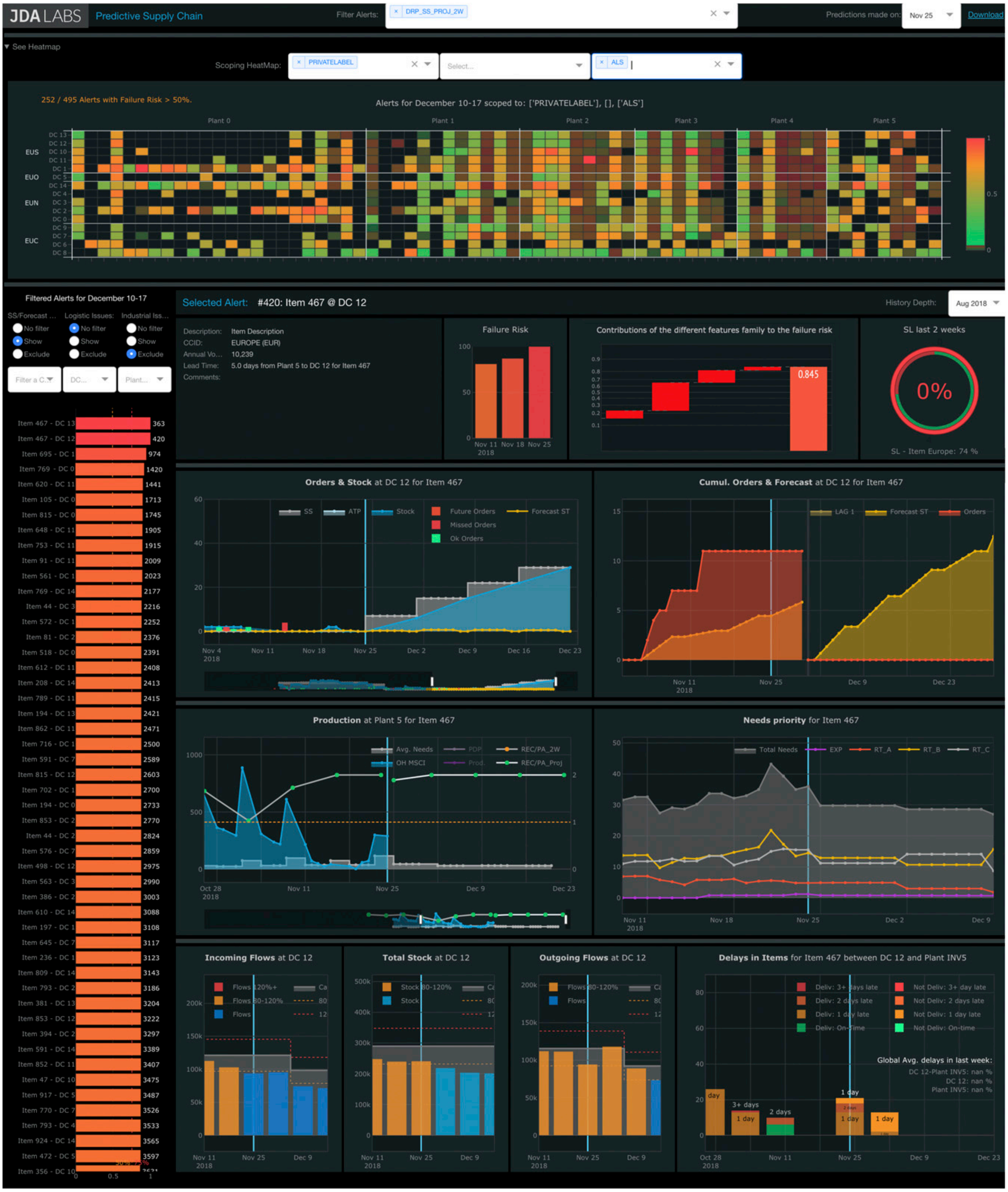
Focusing on Useful Alerts

Second, because of the number of alerts that the system generates, users need the ability to filter alerts to focus on the ones where they can be impactful. Based on Michelin's feedback on which alerts were useful for them, we narrowed down the properties of a useful alert to three aspects. We define the notion of a *useful* alert as being exclusive, actionable, and significant situations. By *exclusive*, we mean that the system should generate alerts that are not obvious or detected by other tools. For example, planners are usually aware of production-capacity problems and quality issues. By *actionable*, we mean that alerts should identify situations in which the planners can attempt to avoid the failure event. Potential actions include changing the forecast, adjusting the safety-stock levels, and adding transportation options. The prediction horizon should be long enough to ensure a minimum number of available actions. For example, anticipating a stockout for the next day when the replenishment lead time is one week is not helpful. By *significant*, we mean that alerts need to concern items and locations that are important (e.g., in terms of volume or strategy), so that the corrective action is worth the effort and cost. Nevertheless, alerts that are not exclusive, actionable, or significant may still allow supply chain planners to forewarn customers of delays and highlight recurrent problems that structural changes in planning processes could alleviate. In the UI, we enable planners to filter out unuseful alerts through dropdown filters (in the white box at the top). The planners could add their own filters via a configuration file that would apply to specific items, DCs, weeks, or a combination of those.

Model Explainability

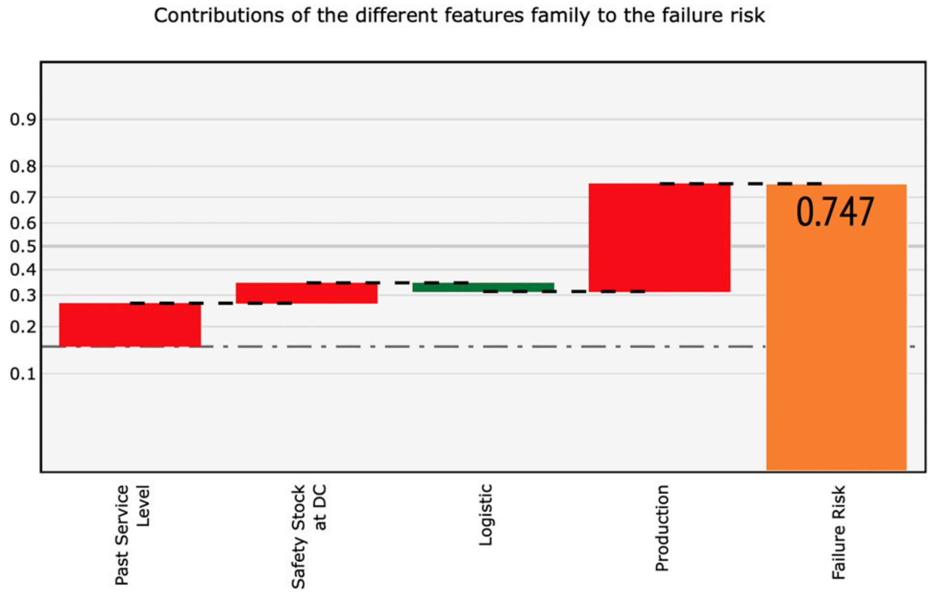
Lastly, in the waterfall graph in the top third of the screen (for readability, we present in Figure 3 a larger version on a white background), we provide some explainability around the predictions. In general, explaining results helps users gain confidence in an ML system. Additionally, identifying which supply chain conditions lead to a failure prediction can help planners find the appropriate resolution. One way to explain a model's output is to identify the most important features. Additive feature-attribution methods (Lundberg and Lee 2017) are approaches that assign a contributing value to each feature for each prediction, such that the sum of all contributions is equal to the model's output. Through game theory, we can consider this value-attribution problem as a cooperative game, by viewing features as different players. The solution to this problem is solved

Figure 2. (Color online) Interactive UI



Notes. We developed an interactive UI that planners can use to have an overview of all alerts and to explore each prediction individually, providing context and explanations to identify potential failure resolutions. The UI uses the library Dash by Plotly. It connects to the model output and additional data sources.

Figure 3. (Color online) Contributions of the Different Features in a Waterfall Graph



Notes. The contribution of the different features for a given prediction can be approximated with methods such as Tree SHAP through a waterfall graph, where the sum of their contributions is equivalent to the difference between the model's output (here, 0.747) and the model's baseline (here, 0.151). In this example, the main underlying cause is a production problem. The service level in the last few weeks and features representing the safety stock at the DC are also factors that increase the failure risk. Favorable logistic conditions slightly lower it.

with Shapley values (Shapley 1953). The intuition behind those values is to compare the prediction with and without each feature. Unfortunately, computing Shapley values has exponential complexity, and, as such, they typically need to be approximated. We use a method called Tree SHAP (Lundberg et al. 2018), which estimates Shapley values for decision trees in pseudo-polynomial time. As Shapley values are additive, we can display them in a waterfall graph, such as in Figure 3. In the context of GBDTs, Shapley values sum to the difference between the prediction (failure risk) and the model's baseline—in our case, the average service level in the training set (y intercept). In this example, the rightmost bar displays the model prediction. Through a waterfall graph, the other bars illustrate the positive (when going down) or negative (when going up) contributions, as estimated by Tree SHAP, for each feature's family on the model's prediction. Table A.2 in the appendix indicates the mapping between features and features' family that we considered. Note that Shapley values share the same units as the model's output—here, log-odds ratio—and, as such, they represent log-odds contributions. The graph uses a nonlinear (logit) y axis so that we can linearly compare the contributions of the bars.

Results

In this section, we present the experiments and discuss the results from both a statistical and business perspective.

Experiments Overview

The project was carried out in four phases. In the first, we gathered data and developed the model, by focusing on about 100 17″ summer tires that Michelin identified as adequate representatives of the general situation in their supply chain. Once we validated the performance of the model, we moved to the second phase, where we performed live tests for 10 weeks. This phase's focus was on users' adoption of the system, as we developed a workflow and UI to convert predictions into actionable alerts that could bring business value. Users experimented dynamically with the tool for a few weeks, during which they identified pertinent issues in their supply chain. As planners found the tool useful, we moved to a third phase, where we tested whether our model could generalize to an extended range of products. As such, we tested the 17″ summer tires model on the complete range of Michelin car tires in Europe and obtained satisfactory results. Thus, we extended to a fourth phase, where planners used the system dynamically once more, for six weeks, and performed corrective actions at scale to measure the impact on the service level.

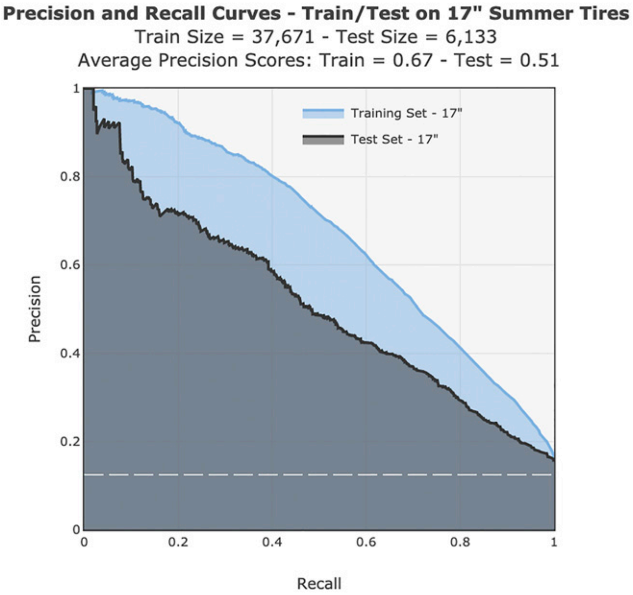
First Phase: Initial Performance

For the first phase, targeting 17″ summer tires, we had access to more than 43,000 data points covering 14 months and describing the supply chain conditions of 95 items, produced at 10 plants, and stored in 16 DCs. All tires are monosourced; that is, only one plant produces them at a time. These orders represent more than 23,000 customers. We used the first 12 months as the training set and performed nested cross-validation (CV) (as is standard in time-series-based predictions) and Grid Search for hyperparameters' tuning. In the appendix, we list the selected hyperparameters and the CV parameters. We used the remaining two-and-a-half months as a test set. Figure 4 shows the precision–recall curve of the model. The mean service level is around 87.5%. As such, a random classifier would correspond to a horizontal line at $y = 0.125$, as the dashed line represents. Our curves are significantly above this, which indicates the predictive value of the proposed model.

Second Phase: Business Validation with Users

In the second phase, we developed a UI so planners could interact with the model's predictions. They used the system dynamically for a few weeks, and we iterated on the UI so that it could support a manageable workflow in which planners could quickly identify useful alerts and have an impact. As such, we verified the model's performance on a subset of useful alerts only, with the exception that we included alerts concerning tires of all significance (big runners, medium runners, and long-tail products). Hence, the subset contains exclusive and actionable alerts only, as defined by planners. As shown in Figure 5, the performance is slightly lower than for all the alerts, and, in particular, the system does not reach a high precision level. Because users tend to filter out easy-to-find problems captured by other systems—that is, not exclusive alerts—we expected a drop in performance. Nonetheless, results are still satisfactory.

Figure 4. (Color online) Precision–Recall Curves: Initial Performance on Training and Test Sets



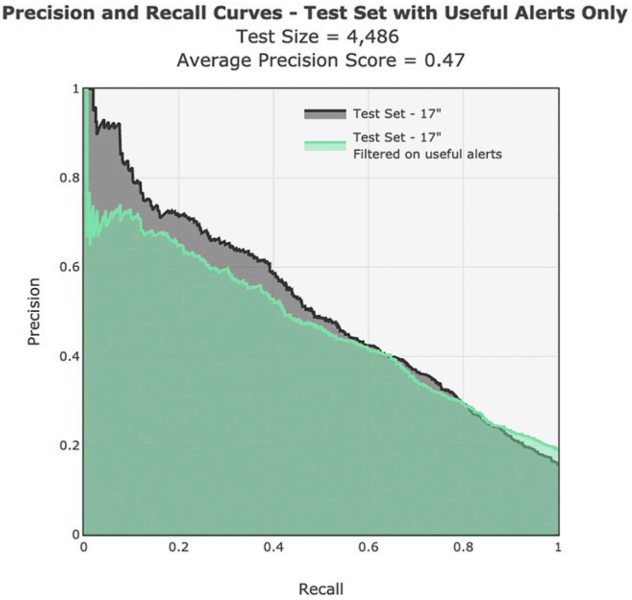
Notes. Curves are displayed for 17″ summer tires on the training set (larger area) and the test set (smaller area). The training set spans the first 12 months of data—more precisely, from September 1, 2016, to September 20, 2017. The test set encompasses the following two-and-a-half months of data for the same tires, from September 21, 2017, to December 12, 2017. The horizontal dashed line displays the mean service level in the data set.

While using the system, planners identified recurrent issues in the supply chain imperceptible in high-level metrics. Below, we present three such situations identified during the second phase.

First, the system generated multiple similar alerts that enabled Michelin to detect a problem in the computation of the safety stock for their small-volume tires, affecting around 25,000 units on that week. To temporarily fix the problem, the planners manually changed the safety-stock targets of the item–DC combinations for which the system predicted service-level failures. After a few weeks, to everyone's surprise, planners discovered that the safety-stock overwriting process was faulty as well, so the manual changes had no impact. By bringing visibility to these issues, planners were able to improve the safety-stock target computations and fix the overwriting process. A month after discovering these issues, these changes reduced the number of items affected by the problem to around 2,800 tires, a decrease of 89%.

Second, the system detected multiple situations in which the underlying cause was a forecasting issue,

Figure 5. (Color online) Precision–Recall Curves: Test Set with Useful Alerts Only



Notes. Curves are shown on the subset of useful (exclusive and actionable) alerts only (smaller area), as compared with the performance on the full original test set (larger area). The larger area represents the same test set as in Figure 4.

highlighting that the forecasting algorithm was not sufficiently dynamic. The system identified individual cases of underforecasting and raised awareness of these issues, motivating further work at Michelin on forecast improvements.

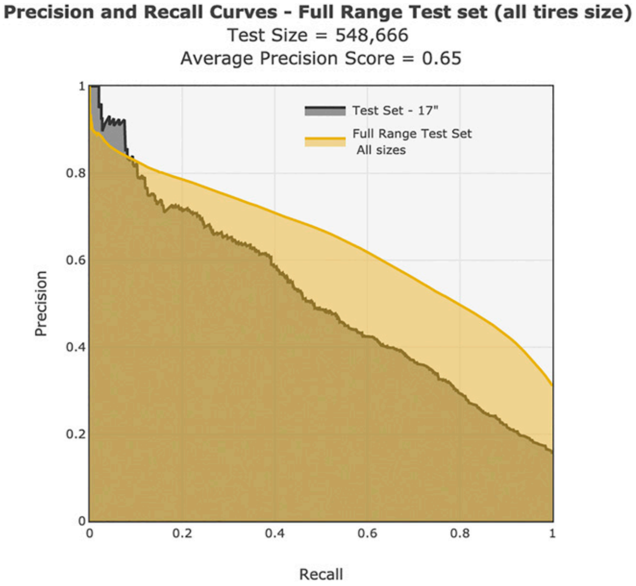
Third, the system detected multiple situations in which the supply chain plan phased out too aggressively an item soon to be discontinued, given the customer demand. The planners reached out to the appropriate team to address this issue.

Overall, these issues resulted from supply chain systems that are wrongly configured, no longer adapted to the current supply chain dynamics, or not interacting well with one another. By raising Michelin's awareness of these issues so they could fix the problems at their source, our alerting system already had a positive impact beyond the tires in the scope of this phase.

Third Phase: Full Scope of Tires in Europe

For the third phase, we had access to the data for most of the car tires sold by Michelin in Europe (around 4,000 items), ranging from 13″ to 22″ and beyond. The data spread about 11 months and represent about 500,000 points. Because we had access to less than a year of data, we decided, together with Michelin, not to retrain the initial 17″ model and test it directly on the extended scope. Figure 6 shows that the extended performance is slightly better than its initial one on 17″ tires. Differences in the nature of the product, such as the presence of long tails (tires with a low volume of sales), which may be more failure-prone, could explain the small performance improvement. Still, for predictions associated with a low recall (top left of the plot), the model is performing at higher precision on the 17″ test set. These results show that the model generalizes well.

Figure 6. (Color online) Precision–Recall Curves: Full-Range Test Set



Notes. Curves are shown on a test set containing the full range of tires (smoother and larger area) as compared with the original test set (smaller area). The smaller area is the initial test set as shown on Figure 4. The new test set includes all sizes of tires (around 4,000 items for 11 months of data). The performance is similar, indicating that the model generalizes well to new items.

Fourth Phase: Dynamic Phase on Full Scope

The focus of phase four was on testing the system dynamically on the full scope of tires and performing corrective actions at scale to measure the impact on the service level.

Today, owing to Michelin's internal processes, planners only have access to adjusting the safety-stock target as a corrective action, as we discuss later in Managerial Insights. To that end, during the last three weeks of the phase, planners implemented a process that would automatically overwrite the safety-stock target for certain item–DC combinations. However, as increasing safety stocks can be costly, the automatic process needs to only apply to situations in which increasing the safety stock can result in avoiding failures. As an example, if an alert is due to production or logistics problems, increasing safety stocks would not result in any gain. As such, to identify the relevant situations, planners use the model explainability, as shown in Figure 3. The automatic safety-stock adjustments only apply to item–DC combinations concerned by an alert with "safety stock at DC" as the main underlying cause. Over the course of three weeks, planners acted on 23% of the store-and-sell volume based on the ML predictions. In Table 1, we present the detailed results observed on the impacted tires and compare them with tires on which planners did not intervene. We break it down by volume of tires. Big runners represent typically around 6% of the tires and 60% in terms of sales volume; medium runners, respectively, 12% and 20%; and, finally, long tails, 82% and 20%. Overall, we witnessed in three weeks a gain of 10 percentage points (p.p.) in the service level of impacted tires. To measure that, planners measure the service level for each item–DC combinations in the week in which the ML model produces the prediction and three weeks later (i.e., the prediction horizon). To put the gain in context, we also provide numbers for the same period on the variation of demand and days of coverage—that is, coverage of the stock as compared with the demand. Admittedly, it could be easy to observe an increase in the service level only because of a decrease in the demand, or at the expense of an increase in the total inventory and storage cost. As shown in the table, this is not the case. First, the decrease in demand is steeper on tires not affected by any corrective action (−10%

Table 1. Changes Observed over Three Weeks of Corrective Actions in Service Level (in Percentage Points), Demand (%), and Days of Coverage, by Volume of Tires

	Changes in					
	Service level (p.p.)		Demand (%)		Coverage (days)	
Volume of tires	Action	No action	Action	No action	Action	No action
Big runners	+4	−4	0%	−3%	+1	+6
Medium runners	+14	+3	−7%	−10%	+2	+9
Long tails	+14	+5	−2%	−21%	+4	+15
Total (weighted by volume)	+10	0	−2%	−10%	+2	+8

Notes. We measure the values before and after the prediction horizon (i.e., three weeks apart). The columns *Action* display aggregated results on all items affected by an automatic safety-stock adjustment based on an ML alert. The columns *No action* display aggregated results for the remaining items (i.e., those for which the supply chain plan remained as is). The total is a weighted average according to the volume of tires in each category.

versus −2%), without resulting in any gain in service level. This demonstrates that the overall increase of 10 p.p. in the service level is not only due to the negative trend in demand. If it were the case, tires not impacted by any action would have a natural increase in their service level as well. Second, the days of coverage increased only by a small margin (two days) for impacted tires, as compared with an increase of eight days for not-impacted tires. This can be due in part to the decrease in demand. Finally, in addition to these results, Michelin also stated that improving the service level leads to lowering the order-cancellation rate, thus increasing sales and revenue.

Perspectives

In this section, we will discuss some challenges that we faced during our work, such as data availability and making alerts actionable within Michelin's current processes. We also discuss opportunities for our system to facilitate a holistic view of the supply chain. Finally, we mention challenges related to deploying failure-prediction systems, such as concept drift.

Available Data

It is not yet standard practice, to our knowledge, to archive all of the supply chain data at the finest granularity. For example, at the beginning of our case study, Michelin did not archive the forecast at the item–DC level or the *Available To Promise*, a value derived from the fulfillment plan indicating how many items at a DC on a given day are not yet allocated. This lack of history led us to discard some data that we believe could improve the model's predictions. Before ML can significantly enter this space, archiving must become the default policy for supply chain data. Our initiative influenced Michelin's perspective on the importance of archiving data at the most granular level, and, in response, they implemented a data lake during the project.

Managerial Insights

The ML approach for failure predictions brings new opportunities for planners to intervene in the supply chain. As such, Michelin needs to adapt its current process so that planners can perform manual corrections easily. Today, the available corrective actions for planners are still limited, with the main one being adjusting the safety stock at the DC. For example, because Michelin does not perform its forecasting process at the most granular level, it is yet impossible for planners to change the individual forecast of a specific instance. Adjusting safety stock is the most practical lever planners have to reallocate tires or increase production. Adding new corrective actions could allow planners to react to deviations and other issues more readily.

Supply chain management is complex, and planners typically own only one segment, such as safety-stock specialists, forecasting experts, and logistics planners. As such, it is challenging to gain a product or customer view of the supply chain. At Michelin, our system was the first one to offer a quick, holistic view of the supply chain conditions affecting one particular item. It provided an opportunity for different specialists to collaborate and discuss global supply chain mechanisms, as well as specific issues, broadening their understanding of the supply chain. Indeed, the tool instigated regular meetings between planners, who would not interact as frequently otherwise. Our experiments suggest that the tool could also help with the training of new employees because it gives just the right level of information about every stage of the supply chain affecting the service level, and the available levers to react to different disruptions.

Long-Term Viability

As manufacturers deploy failure-prediction systems at scale, the supply chain performance will tend to improve. The system will catch systemic issues and help refine supply chain planning tools, such as making better forecasts. From a statistical perspective, as fewer failures will occur, the system will face a shift in the distribution of the data, often referred to as concept drift in the literature. As such, the model will need continuous retraining. This could imply dropping from the model historical periods that are too different from the current distribution or weighting the data points differently according to their relevance in today's dynamics. We believe that designing such a retraining loop will be one of the main challenges for the long-term viability of failure-prediction systems.

Conclusion

We developed a system that uses ML to predict service-level failures in a supply chain. Early on in this project, it has been clear that a well-performing algorithm is usually not sufficient to ensure users' adoption. It needs to be paired with a system that builds confidence and understanding in the model's predictions. Our performance results and the level of engagement from planners show the potential of ML systems to complement existing systems for supply chain management. We believe that this type of approach can be applied to more complex supply networks and other areas, such as production planning. As this system gets used in production over an extensive period, it will probably trend toward identifying less structural changes for more fine-grained issues. A natural extension would be to perform corrective actions based on the predicted failures automatically so that the supply chain becomes a *self-learning* entity, dealing with deviations in *autopilot* mode. As the availability of data improves, such initiatives will lead the way to a new era in supply chain management.

Acknowledgments

This work was performed at JDA Software, now Blue Yonder, where all authors were working during the time of this study. The authors thank Emmanuel Cadet of Michelin for his constant support throughout this project: It led to valuable learning on both sides.

Appendix. Data and Parameters

Data Format

For each m historical instances, we compute a set of n features (f_n) at the item, DC, and week level. We present the complete list of features in the Features Set section. To determine failures, we compute aggregated service levels with the mean of the corresponding orders weighted according to the product quantities. We then compare these aggregated service levels to the global service-level target and convert them into a binary variable, y. If the service level is below the target, it is considered a failure (1), else a 0. Table A.1 shows the data format. Note that we do not use the product, location, and week (gray columns) as features explicitly; hence, the model generalizes to new locations and products.

Table A.1. Data Format Used in the Model

	Product	Location	Week	f_1 $f_2 \ldots f_n$	Failure	
$\mathbf{x_1}$	*AXP*	9272	*W* 1		y_1	1
$\mathbf{x_2}$	*BFR*	875	*W* 1		y_2	0
$\mathbf{x_3}$	*AXP*	9272	*W* 2		y_3	0
...	...	...	...		...	...
$\mathbf{x_m}$	*GFT*	7654	*W* 700		y_m	1

Note. Gray columns were dropped out of the model, so the system can produce alerts for new products and locations.

Features Set

Below, we first detail general features that could be useful for the task of predicting service-level failure. Following, we present the features that were available for our project and that we used in our final model. In Table A.2, we list the specific features, and in Table A.3, statistical measures for each feature.

General Useful Features

First, production problems can be good indicators of future failures because the lead time between the plant and the DCs delays their impact, allowing the system to foresee potential issues and delays. Relevant features include (1) production plan versus actual production; (2) inventory on hand compared with latest forecast; and (3) percentage of production capacity achieved.

Second, between the plant and the DC or between the DC and the ultimate consumer, logistics problems may occur in the transportation network or the loading and unloading, delaying the shipment of the items. Although essential to understand past failures, transportation disruptions are less likely to predict future ones. For example, a delay may be weather-related and likely to be resolved within the time horizon. Still, useful features could include (1) average logistics delays in the last period and (2) percentage of logistics capacity achieved.

Lastly, forecasting deviations and stock problems at the DC may indicate that the plan is no longer meeting the demand. Relevant features include (1) cumulative forecasts of the last few periods compared with the customer orders, indicating overforecasting or underforecasting; and (2) inventory on hand versus safety-stock target.

Available Features

In Table A.2, we detail the complete list of features used in the final model. Each feature type corresponds to an item, DC, plant, or combination of those, as noted in the column *Aggregation*. We compute each feature type for different time steps, from the three weeks preceding the instance to the three weeks after (for some projections), as detailed in column *Time*. The variable $t0$ corresponds to the *day* on which the prediction is done. $t + n$ corresponds to values that were projected for the week n relative to $t0$. As an example, $t + 1$ means projection for the entire week starting

Table A.2. Features Set Used in the Final Model, According to Data Availability

Feature types	Aggregation	Time	Nb	Underlying cause
(1) Production plan vs. actual production	Item–plant	*t*0	1	Production
(2) Inventory and production plan vs. needs	Item–plant	*t*0:*t*+2	9	Production
(3) Inventory vs. safety stock	Item–plant	*t*−2:*t*+2	5	Production
(4) Average logistic delays	Item–plant–DC	*t*−3:*t*0	4	Logistic
(5) Total stock (all items) vs. capacity	DC	*t*0	1	Logistic
(6) Inventory vs. safety stock	Item–DC	*t*0:*t*3	4	Safety stock at DC
(7) Projected safety stock vs. current safety stock	Item–DC	*t*1:*t*3	3	Safety stock at DC
(8) Past service level (volume tires)	Item–DC	*t*−3:*t*0	4	Unknown
(9) Past service level (count orders)	Item–DC	*t*−3:*t*0	4	Unknown

Notes. For feature type (2), three different needs projections were used for each time step, resulting in 3 × 3 features in total. Nb, number.

Table A.3. Distribution of Features in the Training Set

				Percentiles					%	
Type	Time	Mean	Variance	Minimum	25	50	75	Maximum	NaN	Inf
1	*t*0	−0.6	9.5	−324.0	−0.1	0.0	0.3	4.0	0.5	0.0
2	$t0_{low}$	463.1	2,013.5	0.0	2.3	8.7	28.0	9,999.0	0.3	0.0
2	$t0_{med}$	348.9	1,788.6	0.0	1.8	4.5	11.6	9,999.0	0.3	0.0
2	$t0_{high}$	320.8	1,738.7	0.0	1.6	3.1	6.8	9,999.0	0.3	0.0
2	$t1_{low}$	479.7	2,050.9	0.0	2.3	8.7	27.8	9,999.0	0.4	0.0
2	$t1_{med}$	356.2	1,808.2	0.0	1.8	4.5	11.6	9,999.0	0.4	0.0
2	$t1_{high}$	329.8	1,763.5	0.0	1.6	3.1	6.9	9,999.0	0.4	0.0
2	$t2_{low}$	486.2	2,064.1	0.0	2.3	8.8	27.3	9,999.0	8.4	0.0
2	$t2_{med}$	373.1	1,851.8	0.0	1.8	4.6	11.7	9,999.0	8.4	0.0
2	$t2_{high}$	344.6	1,798.2	0.0	1.6	3.1	7.1	9,999.0	8.4	0.0
3	*t*−2	116.4	526.6	0.0	5.8	27.2	74.2	31,680.0	0.3	0.0
3	*t*−1	110.9	524.6	0.0	5.6	26.6	72.0	31,802.0	0.3	0.0
3	*t*0	109.2	519.2	0.0	5.4	25.7	70.6	31,802.0	0.3	0.0
3	*t*1	107.6	486.3	0.0	5.4	25.2	69.8	31,802.0	0.4	0.0
3	*t*2	119.0	820.7	0.0	5.3	24.2	66.7	31,680.0	8.4	0.0
4	*t*−3	−1.4	2.9	−22.0	−2.8	−0.9	0.0	28.0	0.3	0.0
4	*t*−2	−1.4	3.0	−22.0	−2.9	−0.9	0.0	28.0	0.3	0.0
4	*t*−1	−1.4	3.0	−22.0	−2.9	−0.9	0.0	28.0	0.3	0.0
4	*t*0	−1.4	3.0	−22.0	−2.8	−0.9	0.0	28.0	0.3	0.0
5	*t*0	0.8	0.1	0.3	0.7	0.8	0.9	1.2	0.3	0.0
6	*t*0	1.3	8.4	−471.4	0.3	0.9	1.5	772.0	0.0	0.1
6	*t*1	1.2	7.7	−471.4	0.3	0.9	1.5	772.0	0.0	0.1
6	*t*2	1.4	7.0	−463.6	0.3	0.8	1.6	386.0	0.0	0.1
6	*t*3	1.5	7.2	−458.8	0.3	0.8	1.6	386.0	0.0	0.1
7	*t*1	1.0	0.2	0.0	1.0	1.0	1.1	3.0	0.1	0.0
7	*t*2	1.1	0.4	0.0	1.0	1.0	1.1	8.0	0.1	0.0
7	*t*3	1.1	0.5	0.0	1.0	1.0	1.2	10.0	0.1	0.0
8	*t*−3	0.9	0.3	0.0	1.0	1.0	1.0	1.0	0.4	0.0
8	*t*−2	0.9	0.3	0.0	1.0	1.0	1.0	1.0	0.4	0.0
8	*t*−1	0.9	0.3	0.0	1.0	1.0	1.0	1.0	0.4	0.0
8	*t*0	0.9	0.3	0.0	1.0	1.0	1.0	1.0	0.3	0.0
9	*t*−3	0.9	0.3	0.0	1.0	1.0	1.0	1.0	0.4	0.0
9	*t*−2	0.9	0.3	0.0	1.0	1.0	1.0	1.0	0.4	0.0
9	*t*−1	0.9	0.3	0.0	1.0	1.0	1.0	1.0	0.4	0.0
9	*t*0	0.9	0.3	0.0	1.0	1.0	1.0	1.0	0.3	0.0

Notes. The columns *Type* and *Time* map to Table A.2 columns. The *low*, *med*, and *high* subscripts for feature type 2 refer to different projections. The two last columns quantify the percent of nan and infinite (inf) values for that feature.

Table A.4. Cross-Validation Splits Used to Train and Test the Model

Start date	End date	Number of rows
Train splits		
—2016-09-01	2017-04-06	21,396
—2016-09-01	2017-05-31	27,415
—2016-09-01	2017-07-31	33,335
—2016-09-01	2017-09-20	37,671
Test splits		
—2017-04-07	2017-05-31	6,019
—2017-06-01	2017-07-31	5,920
—2017-08-01	2017-09-20	4,336
—2017-09-21	2017-12-12	6,133

Note. The test set from the last split is what we called our test set for phase 1.

on *t*0. Even though the time indicator points that these values are in the "future," these are values that are known at *t*0, as they are projections made for these weeks, such as the forecast at the DC. The column *Nb* indicates the number of features as a result of the different time steps for each feature type. Lastly, the column *Underlying Cause* indicates the mapping between feature types and the feature families. In particular, we use these mappings to produce the cumulative contributions graph in Figure 3, as well as to categorize the alerts by cause in the UI to accelerate their resolution. In total, our model uses 35 features.

In Table A.3, we display the distribution for each feature in the training set (i.e., on 17″ summer tires for the first 12 months of data (September 1, 2016, to September 20, 2017)). We show the mean and variance, important percentiles, and the percentage of NaN (not a number) and infinite values.

Algorithm

We selected the following hyperparameters for the GBDT using XGBoost: *n* estimators = 75, learning rate = 0.1, and max depth = 5. These are the parameters of the best-performing model according to the mean average performance over the training splits detailed hereafter. We kept the other hyperparameters at their default value.

For CV parameters, we used three splits. In Table A.4, we detail the dates that we used to create the splits.

References

Chen T, Guestrin C (2016) Xgboost: A scalable tree boosting system. *Proc. 22nd ACM SIGKDD Internat. Conf. Knowledge Discovery Data Mining* (Association for Computing Machinery, New York), 785–794.

Friedman JH (2001) Greedy function approximation: A gradient boosting machine. *Ann. Statist.* 29(5):1189–1232.

Garvey MD, Carnovale S, Yeniyurt S (2015) An analytical framework for supply network risk propagation: A Bayesian network approach. *Eur. J. Oper. Res.* 243(2):618–627.

Heckmann I, Comes T, Nickel S (2015) A critical review on supply chain risk—definition, measure and modeling. *Omega* 52:119–132.

Jüttner U, Peck H, Christopher M (2003) Supply chain risk management: Outlining an agenda for future research. *Internat. J. Logist. Res. Appl.* 6(4):197–210.

Khojasteh Y (2017) *Supply Chain Risk Management* (Springer, Singapore).

Kumar SK, Tiwari M, Babiceanu RF (2010) Minimisation of supply chain cost with embedded risk using computational intelligence approaches. *Internat. J. Prod. Res.* 48(13):3717–3739.

Lundberg SM, Lee SI (2017) A unified approach to interpreting model predictions. *Adv. Neural Inform. Processing Systems* 30:4765–4774.

Lundberg SM, Erion GG, Lee SI (2018) Consistent individualized feature attribution for tree ensembles. Preprint, submitted February 12, https://arxiv.org/abs/1802.03888.

Nguyen T, Li Z, Spiegler V, Ieromonachou P, Lin Y (2018) Big data analytics in supply chain management: A state-of-the-art literature review. *Comput. Oper. Res.* 98:254–264.

Ojha R, Ghadge A, Tiwari MK, Bititci US (2018) Bayesian network modelling for supply chain risk propagation. *Internat. J. Prod. Res.* 56(17):5795–5819.

Ong JBS, Wang Z, Goh RSM, Yin XF, Xin X, Fu X (2015) Understanding natural disasters as risks in supply chain management through web data analysis. *Internat. J. Comput. Comm. Engrg.* 4(2):126–133.

Papadopoulos T, Gunasekaran A, Dubey R, Altay N, Childe SJ, Fosso-Wamba S (2017) The role of Big Data in explaining disaster resilience in supply chains for sustainability. *J. Cleaner Prod.* 142:1108–1118.

Paul SK, Sarker R, Essam D (2017) A quantitative model for disruption mitigation in a supply chain. *Eur. J. Oper. Res.* 257(3):881–895.

Schmitt AJ (2011) Strategies for customer service level protection under multi-echelon supply chain disruption risk. *Transportation Res. Part B Methodological* 45(8):1266–1283.

Shapley LS (1953) A value for n-person games. *Contributions Theory Games* 2(28):307–317.

Sharma M, Glatard T, Gelinas E, Tagmouti M, Jaumard B (2018) Data models for service failure prediction in supply chain networks. Preprint, submitted October 20, https://arxiv.org/abs/1810.09944.

Simchi-Levi D, Schmidt W, Wei Y, Zhang PY, Combs K, Ge Y, Gusikhin O, Sanders M, Zhang D (2015) Identifying risks and mitigating disruptions in the automotive supply chain. *Interfaces* 45(5):375–390.

Stadtler H, Kilger C, eds. (2002) Supply Chain Management and Advanced Planning: Concepts, Software, Models, and Case Studies. 4th ed. (Springer, Berlin).

Verification Letters

Suresh Acharya, Chief Scientist, JDA Software, Scottsdale, Arizona 85260, writes:

"This is to attest that the research work conducted and described in the paper 'A Machine-Learning-Based System for Predicting Service Level Failures in Supply Chains' by Gabrielle Gauthier-Melançon et al. was done at JDA Software, Montreal. The authors did this work in collaboration with Michelin."

Emmanuel Cadet, Supply-Chain Engineer, Manufacture Française des Pneumatiques Michelin, 63040 Clermont-Ferrand Cedex 9, France, writes:

"With this letter, we confirm that the approach discussed in 'A Machine-Learning-Based System for Predicting Service Level Failures in Supply Chains' by Gauthier-Melançon et al. was developed and used as described in the paper. In total, the project took 2 years, from the data gathering to the final dynamic tests. The results and benefits outlined in the paper were achieved as stated."

Gabrielle Gauthier Melançon completed an MSc degree in applied mathematics at Polytechnique Montreal in 2019. Her thesis focuses on quantifying uncertainty in complex AI systems to increase trust and leverage a human-in-the-loop. Prior to her master's, she did 5 years of consulting work in supply chain and retail as part of a machine learning research laboratory at JDA Software. She now works at Element AI, where her work combines both explainability and uncertainty in machine learning.

Philippe Grangier completed a PhD in operations research at IMT Atlantique in 2015. Part of his thesis work was awarded the 2017 Practice Prize Award at CORS. He was later a MITACS Elevate postdoctoral scholar at Polytechnique Montréal and JDA Software. Following this, he joined an R&D laboratory, and he is currently an applied research scientist at IVADO Labs. His current research focuses on the combination of machine learning and optimization for short-term decision making in supply chain.

Eric Prescott-Gagnon completed a PhD in operations research at Polytechnique Montréal in 2011. After his PhD, he went to work in the industry, accumulating now more than 10 years of experience doing applied research and development on many aspects of optimization in retail and supply chain planning. He is currently working as an applied research scientist at Element AI in Montreal, working on combining machine learning and operations research to solve novel problems.

Emmanuel Sabourin joined a nascent supply chain planning software industry in 1998 after obtaining an MSc degree from the Virginia Polytechnic Institute in CAD/CAM, 3D printing, and numerical simulation. Over a tenure of 20+ years, he has amassed an innovation track record from core algorithmic, integration and data management platforms, and user interface as well as solution marketing and implementation practice. He now leads the applied research practice in machine learning at Blue Yonder.

Louis-Martin Rousseau joined Polytechnique Montréal in 2003 after completing his PhD in computer science and operations research at Université de Montréal. He was one of the first researchers to investigate the hybridization of classical operations research methods and constraint programming. His current research focuses on logistics, scheduling, and resource optimization in healthcare. He holds the Canada Research Chair in Healthcare Analytics and Logistics.