World Academy of Science, Engineering and Technology
International Journal of Industrial and Manufacturing Engineering
Vol:18, No:3, 2024

PredictionSCMS: The Implementation of an AI-Powered Supply Chain Management System

Ioannis Andrianakis, Vasileios Gkatas, Nikos Eleftheriadis, Alexios Ellinidis, Ermioni Avramidou

***Abstract*—**The paper discusses the main aspects involved in the development of a supply chain management system using the developed PredictionSCMS software as a basis for the discussion. The discussion is focused on three topics: the first is demand forecasting, where we present the predictive algorithms implemented and discuss related concepts such as the calculation of the safety stock, the effect of out-of-stock days etc. The second topic concerns the design of a supply chain, where the core parameters involved in the process are given, together with a methodology of incorporating these parameters in a meaningful order creation strategy. Finally, the paper discusses some critical events that can happen during the operation of a supply chain management system and how the developed software notifies the end user about their occurrence.

***Keywords*—**Demand forecasting, machine learning, risk management, supply chain design.

I. Introduction

PREDICTIONSCMS is an integrated supply chain management system, with the ultimate goal of optimizing the flow of products, avoiding the loss of sales due to lack of available goods and saving resources through limiting excess inventory and automated management of the supply chain. The project is developed along three main pillars, which are a) demand forecasting, b) supply chain design and c) visualizations and risk management.

In terms of demand forecasting, PredictionSCMS is capable of making predictions for the future demand of products. The forecasts are based on historical sales data, taking into account factors such as seasonality, inventory shortages, etc. The system uses machine learning techniques as well as classic statistical methods of time series forecasting, in order to achieve the best possible result, both in terms of the quality of the forecasts and the overall efficiency of the system. At the same time, the user has the opportunity to evaluate the forecasts and modify both the forecasted values and the way they are calculated.

The demand forecasts are then used to drive the supply chain design, aiming at extracting orders to the supplier in a timely manner, so that the available stock maintains a desired level. The system is adaptable to the user needs, via a flexible parameterization that can accommodate multiple products and suppliers, different lead times, supplier constraints, order frequencies, safety stock levels and so on.

All the available functionalities are presented via a modern and user-friendly web interface, that provides precise information about the state of the supply chain through tables and charts. The interface also facilitates the system parameterization by the end user and issues notifications and warnings about critical events that enable the user to take corrective actions that can mitigate the risks involved in the operation of the supply chain.

The supply chain management software industry is a mature industry with an annual turnover in excess of $15bn [1]. The industry includes several established software companies, that develop large scale supply chain management systems such as SAP [2], Oracle [3], Blue Yonder [4] and others. At the same time, a large number of smaller companies are actively developing similar products that attempt to claim a share on the global sales, such as Open Boxes [5], Open Pro ERP [6], Odoo Inventory [7] and others.

The goal of PredictionSCMS is to be a low-cost integrated solution that covers most of the supply chain management needs of small to medium enterprises. Among its key benefits we would list the implementation of state-of-the-art AI and statistical algorithms for demand forecasting, that are presented in an intuitive way and are made accessible to user without a particular forecasting expertise. At the same time, the system offers a flexible parameterization so that it can be adapted to the specific needs of its end users, and can capture abnormalities in the flow of products within the supply chain, thus helping reduce the risks involved in its operation.

The structure of this paper is as follows: in Section II we present the demand forecasting module of PredictionSCMS, detailing the implemented algorithms and key features that enhance its functionality. Section III discusses the system parameterization and the strategy employed in extracting supplier orders. Section IV presents the notification functionality and how this can help minimizing the risks involved in running a supply chain. Finally, Section V concludes the paper.

II. Demand Forecasting

A. General

At the heart of PredictionSCMS lies an AI-powered demand forecasting engine. The demand forecasting engine is key to any supply chain management system, as the precision of demand predictions allows for an accurate design of the supply chain transactions, in terms of the extracted orders, the stock levels, and in avoiding pitfalls such as stock shortages or excesses. The demand forecasting engine of PredictionSCMS implements state of the art forecasting algorithms derived both

Ioannis Andrianakis*, Vasileios Gkatas, Nikos Eleftheriadis, Alexios Ellinidis, and Ermioni Avramidou are with "SGA Sistimata Pliroforikis SA", Aiantos 2a, Kalamaria, Thessaloniki, Greece (*corresponding author Ioa phone: +302310459496, e-mail: info@sga.gr).

from the classical statistical literature but also from more modern machine learning methodologies. The current section describes the demand forecasting algorithms available in the application, along with some key concepts that enhance their usefulness in a supply chain management system. The demand forecasting engine of PredictionSCMS is also available as an open source software, in [8]; more details about the development and usage of the engine can be found in [9].

B. Demand Forecasting Algorithms

We now describe the demand forecasting algorithms that are accessible through the PredictionSCMS software. These include both classical times series forecasting and machine learning techniques, thereby forming a comprehensive arsenal of prediction algorithms that can suit most demand forecasting needs.

1) Exponential Smoothing

Exponential smoothing is a classical time series forecasting algorithm [10]-[12], that has a very small execution time, and offers plenty of tuning options. These include modeling the seasonality of the data, the existence of a trend, Box-Cox transformations for heteroscedastic data etc. The implementation allows for manual tuning of the above parameters, and there is also the option of automatically learning the above parameters from the training data.

2) ARIMA

Another classical statistical algorithm that is offered is the ARIMA algorithm [13]. The ARIMA algorithm is based on Auto Regressive – Moving Average (ARMA) models, that exploit the correlations existing in time series data such as those of sales for making predictions about the future demand. The implementation offers an automatic tuning of the parameters based on the training data.

3) Prophet

Facebook/Meta has come up with a very flexible time series forecasting algorithm called Prophet [14], which is based on Generalized Linear Models, and has been used extensively by the company for their forecasting needs. It has the ability to model seasonal or trending data and it is claimed to be robust in sudden changes of the data mean values, missing data or outlier values. The implementation also offers an automatic tuning of its parameters, while its computational demands are comparable to those of the two aforementioned algorithms.

4) Recurrent Neural Networks

Recurrent Neural Networks (RNNs) [15] are a more recent addition to the family of time series forecasting algorithms, that are known to excel in cases of large data volumes, albeit at the cost of an increased computational complexity. The implementation offers several variants of the RNN family of algorithms, namely the plain RNN, the Long-Short Term Memory (LSTM) and the Gated Recurrent Unit (GRU). The user has the option to tune some of the key parameters of the model, such as the network's hidden size, the learning rate, the training epochs, the optimization algorithm etc. Some understanding of the role of the above parameters might be required on behalf of the user for a successful tuning of the above parameters, but on the other hand, this allows for the development of very flexible forecasting models. Moreover, the user is free to experiment with different parameter settings and store those that fit its requirements best for use across several products.

5) Tree Based Methods (Boost)

The final algorithm that is offered by the implementation is the Boost variety of the tree-based methods [16]. This technique stacks a large number of decision tree layers, with each layer correcting the prediction shortcomings of the layers that came before it, resulting in a final prediction model that is capable of modelling diverse types of time series data. As with the case of RNNs, the user is offered a large number of tunable parameters, such as number of boosting rounds, maximum tree depth, maximum number of leaves etc., that allow fitting most types of sales data.

C. Prediction Profiles

The available algorithms have a large number of tuning parameters, through which the fit of the model to the training data can be optimized. Some of these parameters can be automatically learned by the training data, but others have to be selected by the end user. The system also supplies fit metrics such as the Root Mean Squared Error (RMSE) the Mean Absolute Scaled Error (MASE) etc., for each fit of an algorithm to the data, so that the end user can evaluate the performance of each algorithm and her choice of the tuning parameters. As we alluded to previously, PredictionSCMS offers the possibility to the end user to store some of their favorite configurations (algorithm plus parameter set) for easy recall and for use across several products. There is also the option to set a default ordering profile for use with all new products or with products that have not had a profile assigned yet.

D. Service Level

All the algorithms of the demand forecasting engine, apart from predictions about the future level of the demand of a certain product, can also make statements about the confidence interval of these predictions, or in other words, about the dispersion (standard deviation) the *true* value of the demand is likely to have around those predictions. Given the confidence intervals and a Gaussian assumption about the distribution of the errors around their mean, it is possible to calculate a demand value that the actual demand will not surpass with a certain probability, e.g. 80%, 90%, 95% etc. Using this demand value in the calculation of the order quantity, we can make a claim that a stock shortage will not occur with the same probability level. This is called the service level, i.e. the probability of not experiencing a stock shortage, and the PredictionSCMS allows the user to set its value for each product independently. The quantity added to the demand in order to satisfy the service level is used in the calculation of the safety stock.

E. Linked Products

The demand forecasting algorithms create predictions about

future demand based on historical sales data for each product. In some cases, such data may not be available, as for example in the case of a new product. For these cases, PredictionSCMS offers the possibility of linking products: the user can select a product that is believed to have a similar sales profile with the product for which there is no sales data available, and create sales forecasts using the history of the linked product. This allows the creation of demand forecasts for new products, or for products that for some reason their sales history is not available.

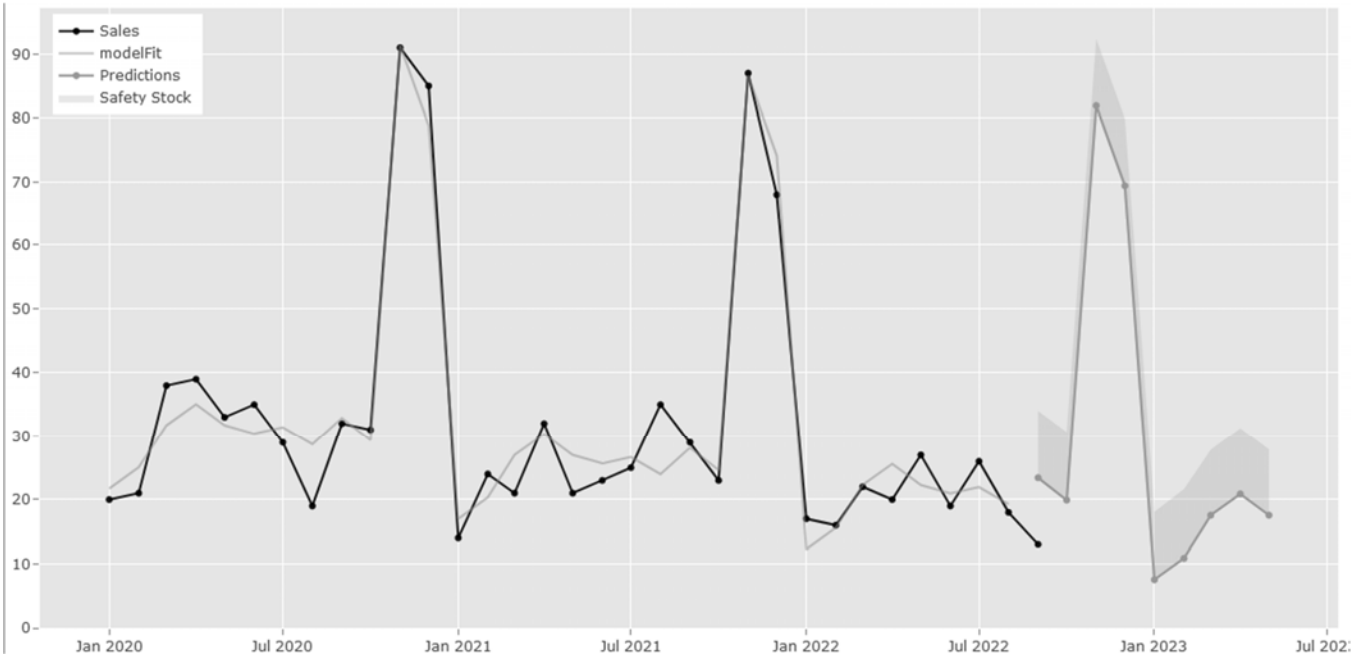

Fig. 1 Example of demand forecasting

F. Out-of-Stock Days

Another important aspect of the demand forecasting engine is the treatment of out-of-stock days. The sales history of a product can reflect the actual demand, when the warehouse has sufficient stock of this product at all times. In cases of stock shortage however, a drop in sales does not necessarily indicate a lower demand, but can instead be attributed to the lack of available stock for sale. The demand forecasting engine can compensate for such a drop in sales by considering the days in a month that a product is not available for sale. Specifically, the actual demand is calculated using the following formula:

$$s_f = \frac{30}{30 - \min(d_{oos}, 20)} s_a \quad (1)$$

where s_a is the actual number of sales, d_{oos} is the number of days the product was not available for sale in a given month and s_f is the estimated number of sales had there been no stock shortage.

Fig. 1 shows an example of the demand forecasting output of PredictionSCMS. The historic sales data for the period from January 2020 up to September 2022 are shown in the dark dotted line. The light black line in the same time period shows the model fit. The demand forecast for the period from September 2022 up to May 2022 is shown with the dotted light black line. Finally, the shaded area in the same period indicates the maximum value the sales are expected to take, according to the specified service level.

III. Ordering

In this section, we are describing the ordering engine of the PredictionSCMS platform. By ordering we mean the process by which the system calculates when to issue an order for a specific supplier, and the quantities of each product that the order will include. The ordering process depends on the results of the demand forecasting module of the system, which estimates the demand for each product over the planning period. In the next step, the ordering engine, based on the available and the expected stock for each product suggests a quantity that will be ordered. This quantity is then passed to the rounding module, that rounds up the quantity to the units used by the supplier, e.g. boxes, pallets etc., and applies any ordering constraints that are in place, such as the minimum order quantity, warehouse capacity and so on. The ordering process relies on a number of parameters that allow the user to fit the ordering engine to the specific needs of their supply chain. We start off by describing the parameterizations available in PredictionSCMS, continuing with a presentation of the ordering strategy and finishing off with a description of the rounding process.

A. Ordering Parameterization

The ordering parameters allow fitting the ordering calculations of the system to the needs of the end user. The parameters can be broadly split into timing and quantity parameters. The former mainly determines the time that a new order will be issued, and the latter affects the quantities that will be included in each order. The core parameters required for

describing the ordering process are given next.

The following is a list of the key timing parameters:

- Ordering Frequency (OF): how often the user wants to issue an order for a specific supplier
- Ordering Period (OP): the amount of time that the quantity on order is expected to cover.
- Lead Time (LT): the time required for the products to arrive on the shelf, after an order has been issued
- Delivery Off Dates (DOD): set of dates on which the supplier is not available to deliver goods.

The key quantity parameters are as follows:

- Supplier Units (SU): the units in which the supplier accepts the orders, and can be either items, boxes or pallets
- Minimum Order (MO): the minimum quantity of an order so that it is accepted from a supplier. It is expressed in the supplier units.
- Boxes per Pallet (BP): the number of boxes in a pallet; this parameter is used in the rounding process
- Items per Box (IB): the number of items per box; also used in the rounding process.

Based on the above parameters, the system calculates a set of key dates for the ordering process. These are given next:

- First Delivery Date (FDD): the date the goods are expected to arrive if the order is issued today. If the FDD falls on a day that is not available for delivery, according to the DOD, this date is shifted to the next first available date.
- Second Delivery Date (SDD): the second delivery opportunity if an order is issued OF days after today. The SDD must satisfy the DOD constraint, in the same manner as the FDD.
- Planning Horizon (PH): OP days after the day the order is issued.

B. Ordering Strategy

The ordering process for a specific supplier begins by checking whether OF days have elapsed since the last order for the same supplier. If this condition is true, the system requests from the demand forecasting engine predictions until the latest of the SDD and PH dates.

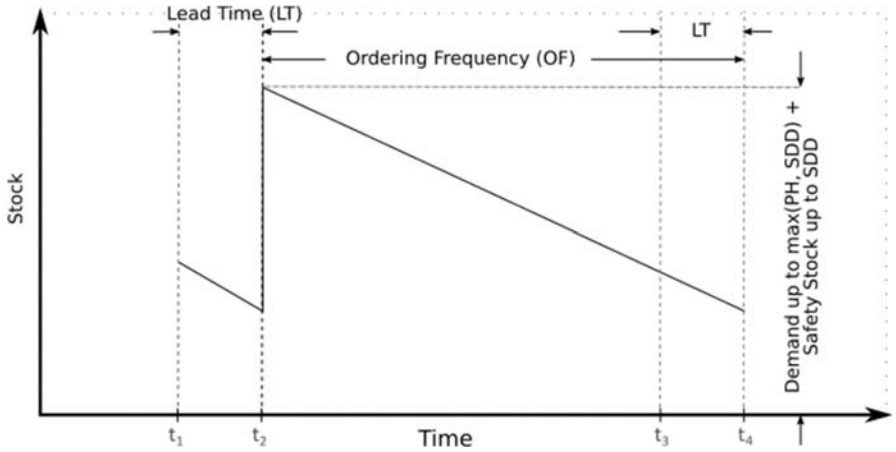

Fig. 2 Timing diagram of the ordering strategy

In the next step, the demand is calculated for each product of the supplier as the sum of the demand forecasts until either the SSD or the PH, whichever is the latest. Based on the service level, the system calculates the safety stock until the SSD. This step attempts to ensure that the stock will last until the second delivery opportunity, with a probability that is defined by the service level. The sum of the demand and the safety stock is the total demand.

The system then checks the available stock for each product, as well as the stock that is expected to arrive within the planning horizon. Finally, the suggested order quantity for each product is calculated as

$$Q_s = D_t - S_a - S_e \quad (2)$$

where Q_s is the suggested order quantity, D_t is the total demand S_a is the available stock and S_e is the expected stock.

The timing of the ordering process is shown in Fig. 2. In the figure, t_1 is the time the order is executed and t_2 is the time the order is expected to arrive. t_3 and t_4 are the respective time points for the next order according to the OF parameter. The system tries to make the stock available on t_2 to equal the demand for the period up to the largest between SDD and PH and the safety stock up to the SDD.

C. Rounding

In the final step, the suggested order quantity Q_s is subjected to the rounding process in order to yield the final order quantity Q_f. The rounding process relies on the supplier units (SU), which can be pallets, boxes or items, in this order of hierarchy. That is, a pallet contains multiple boxes and a box contains multiple items. The first step of the rounding process is the rounding up of the suggested order quantity Q_s to the nearest integer of the supplier units.

The next step involves the application of the supplier constraints to the order quantities. Each supplier has a minimum order quantity (MO), expressed in the supplier units, that an order has to match or exceed for the order to be processed. The system initially checks if Q_s, expressed in SU matches or exceeds the MO. If this is true the ordering process continues

normally. If on the other hand, Q_s is less than half of the MO, the order is cancelled and the procedure stops. Finally, if Q_s is between 50% and 100% of the MO, the system increases the quantities of the individual order products, so that the final quantity reaches the MO. The additional product quantities are calculated as follows.

The system determines what fraction of the total order each product takes up. Next, it finds out how many more units are needed to reach the MO by subtracting Q_s from MO; this amount is called Q_r. Then, it allocates the extra quantity for each product based on its proportion in the initial order times Q_r and rounds it to the closest whole number. Finally, it adds the extra quantities to the original order to get a final order of MO units.

Table I shows an example of applying the supplier minimum order constraint. We suppose that the MO is 100 units and the suggested quantities are 30 units for product A, 20 for product B and 10 for product C, which add up to 60 units. The fractions for each product are 50%, 33.3% and 16.6% respectively. We have to add 40 more units to reach the MO of 100 units. We distribute the 40 units to each product according to its fraction in the initial order, which gives us extra quantities of 20, 13 and 7 units for products A, B and C respectively. This process updates the order to 50, 33 and 17 units for each product, making a total of 100 units for the order, which matches the minimum order quantity set by the supplier.

TABLE I
ROUNDING UP ORDER TO MINIMUM QUANTITY

	Initial Quantity	(%)	Additional Quantity	Final Quantity
Product A	30	50	20	50
Product *B*	20	33.3	13	33
Product *C*	10	16.6	7	37
Total	60	100	40	100

IV. NOTIFICATIONS

A key feature of any supply chain management system is the ability to inform the end user about the supply chain's health and create notifications about events that might affect the regular flow of goods throughout the chain. PredictionSCMS creates various notifications that inform the user about critical events that may occur, the addressing of which could help mitigate risks involved in the day-to-day operation of the supply chain. The main notifications created by the system are described below.

A. Stock Shortage

The system is aware of the stock available in the warehouse as well of the stock that is expected to arrive within the planning horizon. At the same time, it has at its disposal an estimate of the expected sales within the planning horizon in terms of their mean value and the maximum value they are expected to take based on the user defined service level. With this information in hand, the system is capable to assess the expected stock level over the planning horizon as well as the minimum value the stock might take, based on the maximum sales defined by the safety stock. When the projected value of the minimum stock drops below zero, the system creates a stock shortage notification, such that the user can review their ordering plan an make any additional orders as needed.

The system also generates a stock projection as shown in Fig. 4. The figure shows the available stock for the period between January 2020 and September 2022. The light black dotted line from September 2022 up to March 2024 shows the expected stock position based on the forecasted demand. Note that new stock is expected to arrive in May 2023, as shown by the sudden increase in the stock level on that month. The lower edge of the shaded region indicates the minimum expected stock quantity, which is found by taking into account the safety stock as calculated by the service level. The minimum available stock is expected to drop below zero for the first time in March 2023. In this case, the system notifies the user about the pending stock-out, who is then invited to take appropriate actions in order to mitigate the risk.

B. Unusual Sales Levels

The system is also capable of detecting unusual sales patterns from the historic sales data available. We suppose that we have the sales data for the past n months at our disposal and we wish to estimate whether the last month's sales show some deviation from the overall pattern. Using the n-1 oldest sales data, the system calculates their mean and standard deviation. The absolute value of the difference between the calculated mean value and the last month's sales is then compared to the calculated standard deviation. If the difference is found to be more than 3 times the standard deviation a notification is issued to the user, so that they become aware that sales in the last month have taken an unusual value. This could be due to either large sales, or a drop in sales or even some error in the way the warehouse management system records the sales data.

The system also offers a visualization of the historic sales data, so that the user can assess visually the change in the sales pattern. An example of this visualization is shown in Fig. 3. The figure shows the sales pattern for the period between January 2020 and September 2022, with a dark dotted line. According to the data, the mean value of the sales for the period up to August 2022 is around 20 units, and the standard deviation is approximately another 5 units. A value of sales in the region of 70 during the month of September, prompts the system to notify the user that sales during that month have been unusually high.

C. System Errors

Finally, the system issues notifications when errors have occurred in the ordering process, which resulted in failure to calculate the necessary order quantities. These errors can include missing or erroneous system parameterization, e.g. missing lead times, sales data, supplier units and so on, the convergence failure of a demand forecasting algorithm, and general errors that can occur during the order calculation process. In all these cases, the user is notified with details of the event, such as the type of error, the order/product/supplier it concerns, and is invited to take corrective actions.

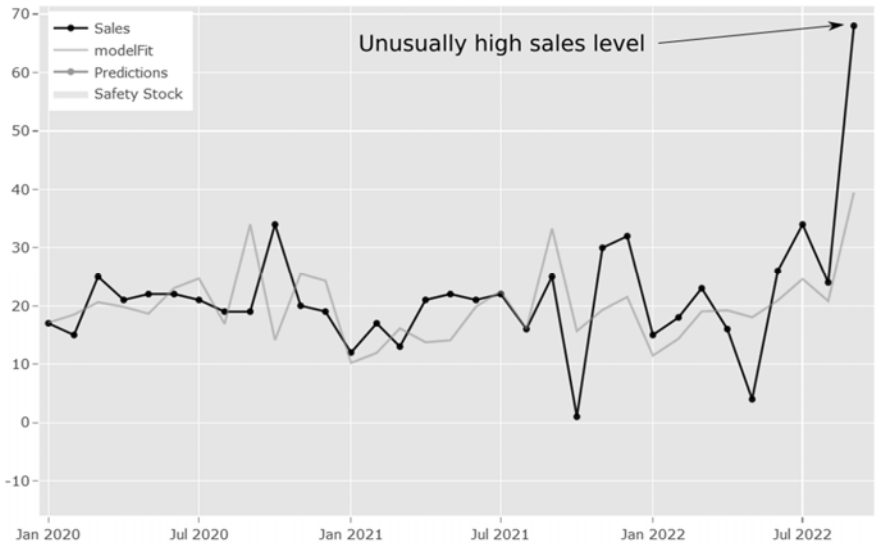

Fig. 3 Unusual sales pattern

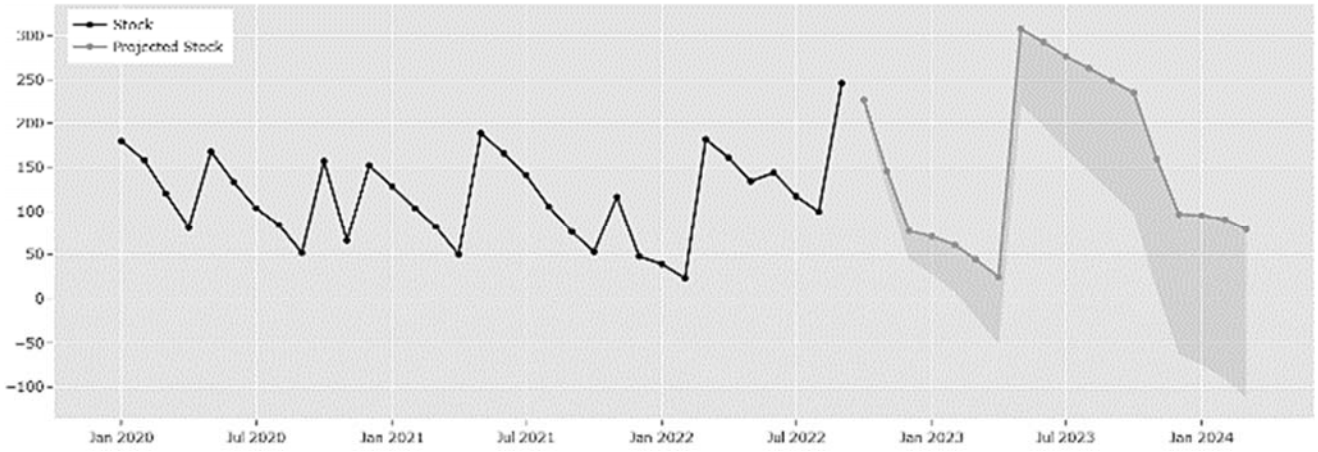

Fig. 4 Projected Stock

V. Conclusion

The paper provided some insights on the main aspects involved in the development of a supply chain management system, using a development software named PredictionSCMS as an example. Apart from a description of the statistical and AI algorithms used for demand forecasting, the paper described the key parameters involved in designing a supply chain system and incorporated these in a comprehensive strategy for extracting supplier orders. It also reviewed critical events that can affect the operation of a supply chain and the respective notifications that can help mitigate the related risks.

Acknowledgment

Co-financed by the European Regional Development Fund of the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH– CREATE - INNOVATE (project code:T2EDK-01197, PredictionSCMS).

References

[1] B. Abbabatulla, N. Gupta and A. Goyal, "https://www.gartner.com/en/documents/3986284/market-share-analysis-supply-chain-management-software-w," 2020. (Online).
[2] SAP. (Online). Available: https://www.sap.com/products/scm.html.
[3] Oracle. (Online). Available: https://www.oracle.com/scm/.
[4] Blue Yonder. (Online). Available: https://blueyonder.com/.
[5] Open Boxes. (Online). Available: https://openboxes.com/.
[6] Open Pro. (Online). Available: https://openpro.com/.
[7] Odoo. (Online). Available: https://www.odoo.com/app/inventory.
[8] SGA, "pscmsForecasting," (Online). Available: https://github.com/sgasa/pscmsForecasting.
[9] I. Andrianakis, V. Gkatas, N. Eleftheriadis, A. Ellinidis and E. Avramidou, "pscmsForecasting: a demand forecasting python web service," in preparation.
[10] R. G. Brown, Statistical forecasting for inventory control, McGraw/Hill, 1959.
[11] C. E. Holt, "Forecasting seasonals and trends by exponentially weighted averages," 1957.
[12] P. R. Winters, "Forecasting sales by exponentially weighted moving averages," Management Science, vol. 6, p. 324–342, 1960.
[13] G. E. P. Box and G. M. Jenkins, Time series analysis: Forecasting and control, Holden-Day, 1970.
[14] Facebook, "FBProphet," (Online). Available: https:/facebook.github.io/prophet.
[15] I. Goodfellow, Y. Bengio and A. Courville, Deep Learning, MIT Press,

2016.
[16] T. Hastie, R. Tibshirani and J. Friedman, The Elements of Statistical Learning, Springer, 2008.