ECTI Transactions on Computer and Information Technology
Journal homepage: https://ph01.tci-thaijo.org/index.php/ecticit/
Published by the ECTI Association, Thailand, ISSN: 2286-9131

The Implementation of Discrete-Event Simulation and Demand Forecasting Using Temporal Fusion Transformers to Validate Spare Parts Inventory Policy for The Petrochemicals Industry

Sanit Innuphat[1] and Manachai Toahchoodee[2]

ABSTRACT

One of the important strategic factors of petrochemicals plant maintenance is the spare parts inventory policy, It is significant for the efficiency, reliability, and productivity of the petrochemicals industry. An unsuitable spare part inventory policy will lead to a loss in long engineering machinery downtime due to a shortage of spare parts. To implement the spare parts inventory policy which is able to fulfill the future demand for spare parts, calculation by various statistical theories and working processes is used to custom the spare parts inventory policy. However, to validate whether or not the customized spare parts inventory policy is suitable, the discrete-event simulation library SimPy is used to mimic the actual spare parts inventory system. It must be involved in the performance evaluation process of the customized spare parts inventory policy. The inventory simulation model consists of many events depending on the supply chain system. The crucial event which is the most complex for the simulation of spare parts inventory is the demand event. This work applies the demand forecasting technique to the simulation by using deep learning with a pre-built architecture model called Temporal Fusion Transformers (TFT). The averaged MAE of the point predictions from a global model is 0.4874+/-6.7744 on the validation dataset and 0.6424+/-3.4963 on the test dataset. Our method predicts a quantile forecast of the future demand which is able to handle the stochastic nature of the spare parts demand in the petrochemicals industry and the result from the simulation outcome is more accurate and close to the outcome from an actual inventory system. The information from the analysis of the simulation outcome is used by the inventory management team to make decisions about the custom inventory policy before deploying it to the actual system.

Article information:

Keywords: Spare Parts, Inventory Policy, Time Series, Demand Forecasting, Deep Learning, Temporal Fusion Transformers, Discrete-event Simulation

Article history:
Received: November 27, 2021
Revised: January 14, 2022
Accepted: March 26, 2022
Published: June 18, 2022
(Online)

DOI: 10.37936/ecti-cit.2022163.246900

1. INTRODUCTION

A spare part is a part of engineering machinery, which is stocked to replace the original following the plant maintenance plan and also for replacing bad parts in the engineering machinery from the plant operational activities. The spare parts are stocked in the inventory, and like other inventory, the spare part inventory has to be managed. Spare part inventory management is very important for the petrochemicals industry because unsuitable spare parts inventory management may result in a shortage of necessary maintenance spare parts, long downtime of the engineering machinery, and a significant decrease in efficiency, reliability, and productivity for the petrochemicals industry.

The inventory management described here is like “how to fill the inventory” where the policy is about replenishing the inventory, Inventory Policy has two major topics to consider. One is “When to order?”, and the other is “How much to order?”. The goal is to adjust the inventory level to the right level that will fulfil the future demand of the spare part [1] and re-

[1,2]The authors are with School of Science and Technology, University of the Thai Chamber of Commerce, Thailand, E-mail: isanit@gmail.com and manachai_toa@utcc.ac.th

tain the required Service Level [2]. To customize the inventory policy, the systematic thinking approach for determining the inventory policy includes understanding the inventory policy with its parameters and components such as demand of the spare parts, lead times of spare part delivery, review periods, excess demand, and working process. There are many forms of inventory policy depending on inventory policy parameters, e.g., inventory policy (S, s) where s and S are the minimum and maximum inventory levels respectively. When the on-hand inventory level falls below a minimum level, a replenishment order will be generated that will restore the on-hand inventory level to a target or maximum level of inventory.

The spare part inventory policy is justified as one of the important strategy factors of plant maintenance for the petrochemicals industry that must be implemented and deployed carefully.

In practice, the inventory policy is still very challenging in operational management because a practical inventory policy is difficult to prove effective due to there being possible uncertainty factors that differ from theoretical assumptions, unlike an inventory policy from a mathematical model, in which the theory can be proven under the theoretical assumption that establishing a clear scope of the problem such as the demand consumption follows the normal distribution.

To ensure the custom spare parts inventory policy is the right inventory policy before applying it to the actual spare parts inventory system, the simulation method refers to a collection of steps and processes used to mimic the behavior of the actual system. The manager must be involved in providing the information to decide on the custom spare part inventory policy. The actual inventory system and processes have been designed in the diagram and transformed into the simulation model [3]. The outcome from the simulation model will be evaluated using inventory performance metrics to validate the custom inventory policy. The final result is the information for the inventory management team to help them decide on the spare parts inventory policy improvement or deploy the spare parts inventory policy to the actual inventory system. Nowadays, a discrete-event simulation library is suitable to model the inventory system. We used one developed in Python named SimPy, which is available and widely used to simulate inventory systems.

The inventory simulation model consists of many events that depend on the inventory system's type of supply chain system. One type is the single-echelon supply chain system, which consists of two events: order-arrival events and demand events. The event that is most complex to simulate so it behaves like the actual system is the demand event. The challenge is the stochastic nature of demand [4] that parametric estimation or statistical estimation is unable to handle. The uncertainty and dynamic nature of the demand in the real-world inventory system cannot be implemented clearly. Another approach proposed in this work to obtain better results of the future demand estimation is demand forecasting. This is frequently used to predict accurate future values in a time-series sequence as the demand estimator for the simulation.

There are many different techniques to achieve demand forecasting for time-series analysis applications. The parametric method called the Autoregressive Integrated Moving Averages (ARIMA) model. At the same time, these models are good for examining observations and reaching conclusions but they also have some serious limitations [5]. We consider a new method using deep learning models that have seen great success for time-series forecasting and which better handle the complexity of time-series forecasting and obtain the improved results significantly. They can gain better results with higher accuracy and do not require identifying the autocorrelation structure, trends, and seasonality, and the knowledge of the probability distribution function [6].

As the previous statement proposed, this work will validate the customized spare parts inventory policy, which can fulfill the demand from plant maintenance plans and plant operational activities of the petrochemical industry by using the simulation method that mimics the real-world processes and system of the petrochemicals spare parts inventory by forecasting future demand of spare parts by analyzing historical data of spare parts usage from the real-world system. Our Inventory Management System can then forecast the future demand with a machine learning algorithm based on deep learning. Afterward, using the inventory performance metrics; we can assign a Fill Rate (β service level) to evaluate the performance of the customized spare parts inventory policy from the simulation outcome. The information from analyzing the performance evaluation outcome is used to decide on the customized spare parts inventory policy before deploying the customized spare parts inventory policy to the actual inventory system.

The main objectives of this work are identified as follows: To present the transformation of the actual system into the simulation model that mimics the real-world process and system of the spare part inventory in the petrochemicals industry by using discrete-event simulation, improving the simulation model by applying a demand forecasting technique, and using deep learning using a pre-built architecture model TFT, to forecast the future demand of the spare parts for the simulation model. Our solution must preserve the nature of spare parts demand accurately match to the actual future demand to make the results from the simulation be close to the result from the actual system.

The remaining part of this work is organized as follows. The literature related to the simulation for decision-making and time-series forecasting is reviewed in Section 2. After that, the experimental setup used to propose the work is presented in Section 3 empirically. Further, the experimental research results for this work are highlighted in Section 4. Finally, the conclusions for this work and the recommendations for future work is presented in Section 5.

2. LITERATURE REVIEW

Simulation methods are widely used in management science and operations research and have been found to be powerful tools for decision-making in many domains of business, such as for supply chain management or optimization of the business processes [7].

Because of to the usefulness of simulations for decision-making in business, simulation methods are used in the automotive industry [8] to simulate inventory systems with various alternative choices in order to evaluate the company activities and then deliver the analysis result of the simulation outcome to the management team. The results are considered to make decisions to increase customer satisfaction and provide better customer service. Discrete-event simulations are utilized to simulate the inventory systems of hospitals [9] to address the questions about the service levels and costs of operating rooms of the hospital that focus on the new method, which is the coordination between the inventory management and the material handling. They compare and analyze the simulation outcome of the observed method with the current method to make a decision on the improvement of the operation room.

Because of the usefulness of the simulation methods, many commercial software packages in the market provide a built-in function to simulate the system. The Arena Simulation Software by Rockwell Automation company [9] is one example. There is a free-of-charge simulation framework available to use also; called SimPy. It is a process-based discrete-event simulation library developed in Python [10] to validate inventory policy performance by simulating a single-echelon inventory system [11] and deploying the inventory policy with the simulation model. This is done by running the simulation model for one year of data and observing the inventory policy performance with its statistics data, then comparing the service level from the outcome of the simulation model to the calculated service level from the closed-form literature model.

In terms of the simulation model for decision making, the variables for the simulation event, such as the arrival time, demand, etc., have to be estimated for the simulation framework demonstrated in Fig. 1. There are various techniques to estimate the values for the simulation, such as assuming the probability distribution for the variable, e.g., the demand followed normal distribution [11] or using parametric or statistical estimation to randomly choose the value of the variable by fitting the probability distribution to the collected historical data [8] [9] [12]. However, to build a simulation model which is close to the actual system that has an uncertain situation from many factors such as trend and seasonality, and others, the parametric estimation or statistical estimation may be suitable for this task because of the dynamic data which defines the fitted probability distribution to the variable of simulation event is not easy to determine due to the fact that the theoretical assumption for the method cannot be implemented clearly. Considering a new approach of value estimation is inescapable to obtain better results in the simulation. The time-series forecasting technique is proposed to improve the simulation model for decision making.

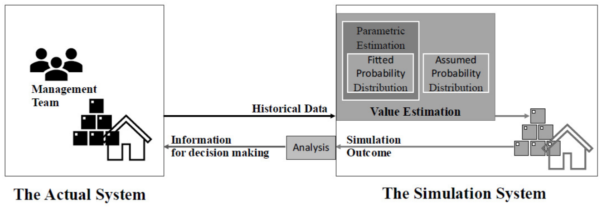


Fig.1: *The simulation framework for decision making.*

Nowadays, there are many algorithms that have been employed for time-series forecasting. One of the traditional algorithms is Autoregressive Integrated Moving Average (ARIMA). It is a parametric method that requires identifying the autocorrelation structure, trends, and seasonality, as well as other explanatory variables. Non-parametric methods using supervised learning algorithms, such as Random Forest Regression, have also been used in time-series forecasting problems [13]. ARIMA and Random Forest Regression have treated each time-series data separately, which is one of the major limitations makes them are incapable of using cross-learning. Meanwhile, over the last few years, the non-parametric methods using deep learning models such as Recurrent Neuron Network(RNNs) have been used . They are suitable for time-series forecasting since the output state not only depends on current input but also considers the previous input of time steps. RNNs have cross-learning capability which allows training the model across a set of time-series. They are frequently used and have had success in solving time-series forecasting problems in a variety of fields, including retail, stock exchange, traffic flow, and so on.

Many research papers have successfully applied deep learning models. They claim that the models outperform the parametric model and can handle the complexities of time-series forecasting better and, thus, obtain significantly improved results. Kasun B. et al. [14] proposed a Long Short-Term Memory Neural Network (LSTMs) model that is one of the RNN extensions which beats the constraints of vanilla RNNs by the capability to learn long-term dependencies in a sequence on sales demand forecasting. The LSTMs model have achieved very competitive accuracy. David S. et al. [15] introduced DeepAR, another variant of RNNs like LSTMs. DeepAR uses probabilistic forecasting based on autoregressive RNNs that estimate future probability distributions instead of future values to compute quantile estimates and thus improves the optimization of business processes. Interestingly, DeepAR is applicable to medium-sized datasets with only a few hundred time-series data values and it works on a wide variety of datasets with little or no hyperparameter tuning. Recently Bryan L. et al. [16] presented Temporal Fusion Transformers (TFT), an attention-based deep neuron networks (DNN) architecture for time-series forecasting. They show significant performance improvements of TFT that surpass DeepAR by 36-69 percent in benchmarks.

Using the capabilities of cross-leaning and output quantile forecast, the simulation framework and the value estimation employing time-series forecasting is presented in Fig. 2.

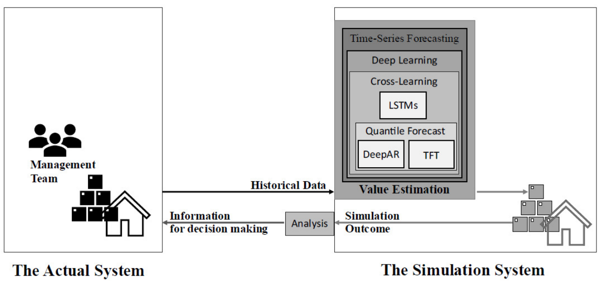


Fig.2: *The simulation framework for decision making with time-series forecasting as the value estimator.*

Because of the TFT's remarkable performance, this work decides to employ TFT as a future demand estimator in a simulation to evaluate the performance of a custom spare parts inventory policy.

3. METHODOLOGY

In this section, the experimental setup used to empirically propose the work is presented in Fig. 3, which includes various components categorized in four domains: 1. inventory management, which consists of inventory system, inventory policy, and inventory performance metrics, 2. the real-world datasets which related to the spare part inventory, 3. the demand forecasting using TFT to predict future demand, and 4. the simulation method for the spare parts inventory system used to perform the experiments. The results obtained from the simulation will be analyzed and sent to the inventory management team to help them make decisions for inventory policy improvement and ultimately for deployment to the actual inventory system.

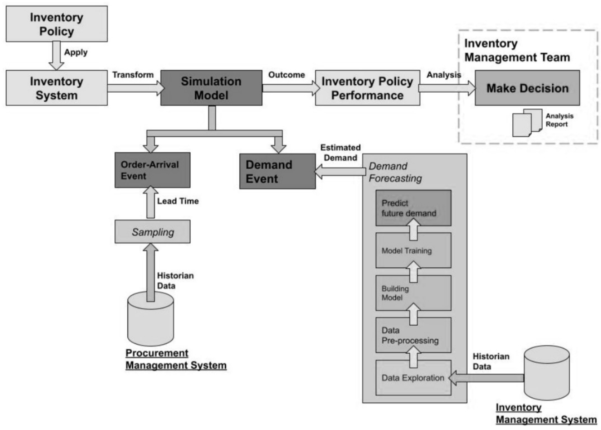


Fig.3: *The components of the methodology used.*

The following sections describe each of these components in detail.

3.1 Inventory Management

The spare parts inventory system considered in this work is a real-world inventory system. It is based on a single-echelon supply chain network. The echelon means the number of layers between the supplier and consumer or customer. In this work, the consumers, or the customers, are maintenance engineers who withdraw the spare parts from the spare parts inventory to maintain the petrochemicals plant following plant maintenance orders.

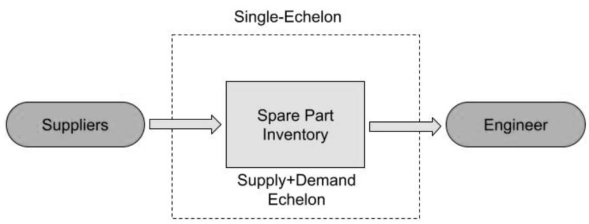


Fig.4: *The single-echelon supply chain for the spare parts inventory system.*

The spare parts inventory for the petrochemicals industry is shown in Fig. 4. The inventory is a combination facility of supply echelon (in contact with an external supplier) and demand echelon (directly facing the demand from plant maintenance orders).

The spare parts inventory policy is validated in this work is denoted as inventory policy (s, S), which is demonstrated in Fig. 5. The inventory policy parameters s and S are the reorder point (ROP) or trigger level and the order-up-to level or the number to which the inventory level is restored. When the on-hand inventory level has fallen below the reorder point (ROP), a replenishment order will be generated to restore the on-hand inventory level to the order-up-to level.

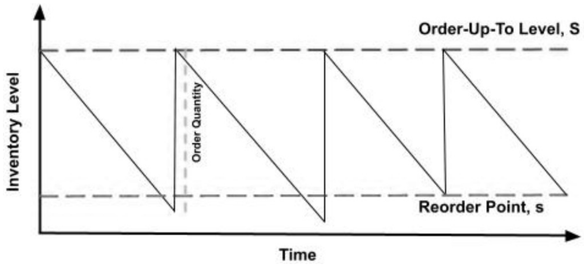


Fig.5: *The inventory policy (s, S).*

The inventory performance metric Service Level is used to validate the custom spare pare inventory policy. There are two types of Service Level that are most frequently used: Service Level Type-I: Cycle Service Level (α Service Level), which measures the number of inventory cycles in which stockout did not occur, and Service Level Type-II: Fill Rate (β Service Level) which is used in this work and measures the fraction of the amount of demand met from the inventory. The Fill Rate is given in Eq. 1.

$$\text{Fill Rate} = \frac{\text{Number of item that meet the demand}}{\text{Total items of all demand}} \tag{1}$$

3.2 Real-world Datasets

The datasets for this work consist of 2 sources of data from one of the petrochemicals industries in Thailand: the historical inventory data and the historical purchasing data collected from the real-world software. These sources use data from the Inventory Management System and Procurement Management System, respectively.

The historical inventory data describes the spare parts demand for plant maintenance and operational activities for the petrochemicals plants. It also has the inventory policy properties of the spare part such as the reorder point (ROP), order-up-to level, and the required Service Level.

The historical purchasing data describes the lead time of the spare parts purchasing from the replenishment order-created date until the goods-received date sampled as the lead time for the simulation of the spare parts inventory system.

3.3 Demand Forecasting using TFT

The demand forecasting approach provides the future demand for the simulation of the spare parts inventory system using the pre-built time-series forecasting architecture TFT from the open-source Python package PyTorch Forecasting [17] [18] as the demand forecasting model of the spare parts. The proposed demand forecasting framework for this work is composed of five constituents: data exploration, data pre-processing, model building, model training, and future demand prediction.

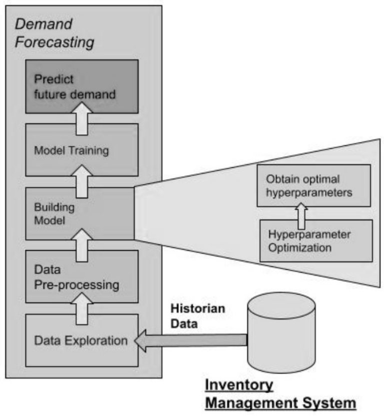


Fig.6: *Demand forecasting framework.*

Fig. 6 gives a schematic overview of the proposed demand forecasting framework. Initially it performs several data exploration and data pre-processing techniques to transform raw data from the inventory management system into time-series data that is ready to train the TFT model. Afterwards, the TFT model is built and tuned to assign the hyperparameters, then it is trained in order to obtain the output quantile forecast of the spare parts demand. The details of each constituent of the proposed demand forecasting framework are given in the following section.

Data Exploration

The demand forecasting framework evaluates the dataset from the real-world Inventory Management System that consists of historical data from all spare parts inventory in the petrochemicals industry. First, a subset of the historical data into the experimental data by collecting a single inventory consisting of 4,956 different spare parts. Second, the experimental data gathered, which spans over eleven years and covers two turned-around petrochemicals plants processes is separated into training + validation datasets (ten years) and a test dataset (one year). The data exploration of the spare part usage is illustrated in Fig. 7.

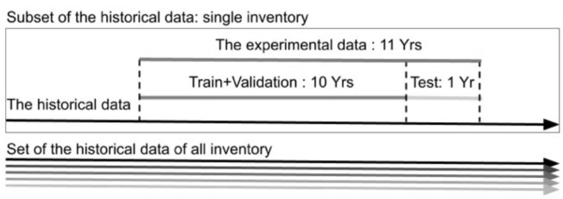


Fig.7: *Data exploration of the historical data.*

Data Pre-processing

The experimental data is the transaction data from the real-world Inventory Management System, so the data is already validated by the system and ready for analysis. The experimental data must be transformed into a time-series dataset which refers to the data collected over a time period and captures a series of the data points where every data point is equally spaced over the time period. The data points for this work are for months, so the experimental data must aggregate the demand consumption from all of the record into the demand consumption by month and stamp the time index for each month step in the intervals of time. Afterwards, to train the TFT model, PyTorch Forecasting also provides a built-in function to convert the time-series dataset into the training dataset and the validation dataset using the same label encoders and data normalization so that they are ready to be fed to the model training step. Following the data exploration step, the training dataset has 108 points in time. The validation dataset is created to predict 12 points in time for each series. The framework of the data pre-processing is shown in Fig. 8.

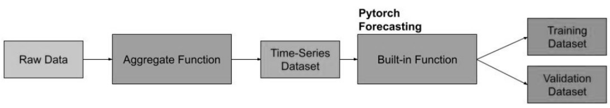


Fig.8: *Data pre-processing framework.*

Building Model

In this work, the pre-built time-series forecasting architecture of the TFT model is loaded from Pytorch Forecasting and tuned with the hyperparameters listed in Table 1 by feeding it the training and validation datasets to a built-in hyperparameter optimizer which Pytorch Forecasting provides to find the optimal hyperparameters for the TFT model. These hyperparameters are used for the model training.

Table 1: *The hyperparameters tuned for the TFT model.*

Hyperparameter Name	Range	
	Min	Max
gradient_clip_val_range	0.01	1.00
hidden_size_range	8	128
hidden_continuous_size_range	8	128
attention_head_size_range	1	4
learning_rate_range	0.001	0.1
dropout_range	0.1	0.3

Model Training

A builder function of Pytorch Forecasting is used to train the model in this work. Observe that the optimal hyperparameters of the TFT model have been obtained from the hyperparameters optimizer. The accuracy of the TFT model during training is determined based on the accuracy of prediction values that provide the quantile loss, which is a value that is summed across all quantile outputs.

Afterward, the trained TFT model has evaluated by measuring the error between the point prediction values, the validation dataset, and the test dataset by calculating the mean absolute error (MAE) with Eq. 2.

$$MAE = \frac{1}{N}\sum\nolimits_{i=1}^{N}|y_i - \hat{y}| \tag{2}$$

Predicting Future Demand

All of the available experimental data is utilized for training the TFT model. The data pre-processing and model building steps wre explained in the previous sections. The trained TFT model is used to predict each spare parts demand for the next 12 coming months.

In addition to providing the output point demand, the trained TFT model also provides output quantile forecasts of the spare parts demand in the future where the set of output quantiles is 0.02, 0.1, 0.25, 0.5, 0.75, 0.9, 0.98, which is the range of likely demand of the spare parts and is different in each month. The output quantile forecasts are shown in Fig. 9.

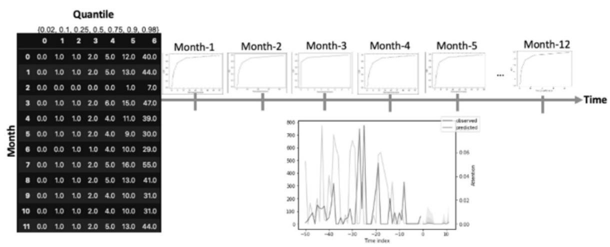
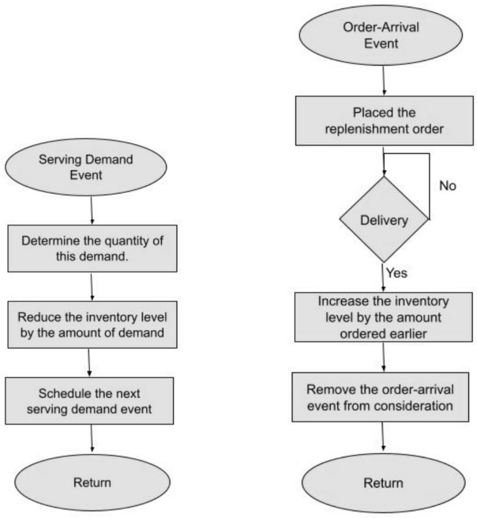


Fig.9: *Predicted spare parts demand as output quantile forecast for 12 coming months.*

3.4 Simulation Models

For this work, discrete-event simulation (DES) is well suited for simulating the single-echelon supply chain of the spare parts inventory system using SimPy, which is a free-of-charge process-based discrete-event simulation library based on standard Python. It is used to assess the performance of the custom spare part inventory policy. The single-echelon supply chain of the spare parts inventory system consists of two different events, which are transformed into routines in the simulation model of the spare parts inventory system. These two routines are Serving Demand Event and Order-Arrival Event.



(a) (b)

Fig.10: *Flowchart of method for serving demand routine (a) and order-arrival routine (b).*

Serving Demand Event

This is the event where a spare part is withdrawn from the spare parts inventory to serve the plant maintenance orders and operational activities. The routine for serving demand events is shown in Fig. 10 (a) and turns into a simulation procedure.

```
Procedure: servingDemand(Spare Part Number)
Data: Inventory Policy Properties of the Spare Part
Input: Spare Part Number
Output: Net Inventory Level, On-hand Inventory Level,
        Total Demand, Total Meet Demand,
        Total Backorder and Total Unit Short
        get updated.

ROP = Inventory Policy Properties of the Spare Part:
 ROP of Spare Part Number
Spare Part Number = Spare Part Number
Demand 12 Coming Months [] = Demand Forecasting
 (Spare Part Number)
Demand 365 Coming Days [] = Monthly Demand to
 Daily Demand(Demand 12 Coming Months [])
Total Demand = 0
Total Meet Demand = 0
Total Backorder = 0
Total Unit Short = 0
While (t <= 365) Do
   Demand = Demand 365 Coming Days[t]
   Total Demand = Total Demand + Demand
   Number of Withdrawn Spare Part = Min(Demand +
   Total Backorder, On-hand Inventory Level)
   Number of Backorder = Demand - Number of
   Withdrawn Spare Part
   Total Meet Demand = Total Meet Demand +
   Number of Withdrawn Spare Part
   Total Backorder = Total Backorder + Number of
   Backorder
   Total Unit Short = Total Unit Short + Max(0,
   Number of Backorder)
   On-hand Inventory Level = On-hand Inventory
   Level - Number of Withdrawn Spare Part
   Net Inventory Level = Net Inventory Level –
   Number of Withdrawn Spare Part
   If Net Inventory Level <= ROP Then
     call(OrderArrival())
   t = t + 1
   End
```

Order-Arrival Event

This is the event where the spare part inventory places the replenishment order of a spare part to refill the spare part inventory to the level following the custom inventory policy. Each replenishment order will wait for the supplier to deliver the spare part. The time duration from creating the replenishment order until receiving the spare part is called lead time. The routine for the order-arrival events is shown in Fig. 10 (b) and turns into a simulation procedure.

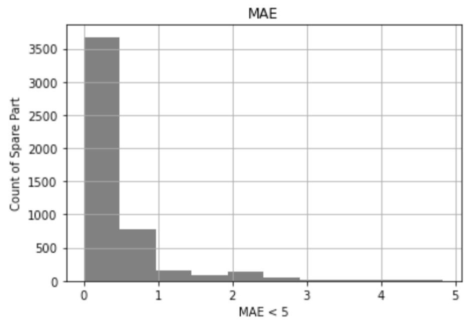
```
Procedure: OrderArrival()
Input:  Inventory Policy Properties of the Spare Part,
        The Historical Data of Spare Part Purchasing
Output: Net Inventory Level, On-hand Inventory Level
        get updated.,

Order-up-to Level = Inventory Policy Properties of
 the Spare Part: Order-up-to Level of Spare Part
 Number
Lead Time = Sampling (The Historical Data of Spare
 Part Purchasing of Spare Part Number)
If Total Backorder > Order-up-to Level Then
  Order Quantity = Total Backorder + Order-up-to
  Level
Else
  Order Quantity = Order-up-to Level - Net Inventory
  Level
Net Inventory Level = Net Inventory Level + Order
 Quantity
Waiting (Lead Time)
On-hand Inventory Level = On-hand Inventory Level
 + Order Quantity
```

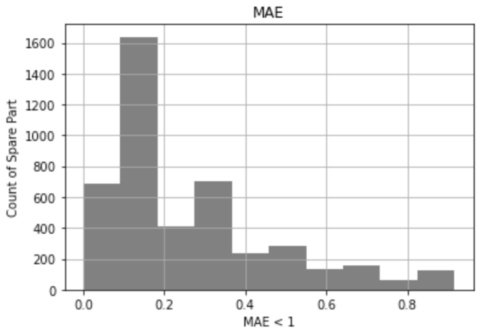
Finish running the simulation, the Fill Rate of each spare part is calculated, and the tracking data are loaded into the outcome table for analysis.

4. RESULT

The result of this work is the validation of the customized spare parts inventory policy performance, which is a comparison between the Service Level which can fulfill the demand from plant maintenance plans and plant operational activities of the petrochemical industry which is known as the required Service Level. The Service Level from the simulation where the spare parts demand for the 12 coming months is the prediction from demand forecasting using the TFT model, a pre-built architecture based on a deep neural network. The TFT builder function trained a global model across all the experimental data and the MAE of the point predictions was compared to the validation dataset. The test dataset has values for what of 0.4874+/-6.7744 and 0.6424+/-3.4963. The histogram of MAE is illustrated in Fig. 11 and shows the accuracy of the trained TFT model on the test dataset is great and covered most of the spare parts in the experimental data.



(a)



(b)

Fig.11: *: MAE between the prediction and test data, which is less than 5 (a) and less than 1 (b).*

The observed spare parts are selected from the experimental data by selecting the top-five most frequently used spare parts in each group of Frequency Usage Analysis (FF-vary fast move, F-fast move, S-slow move). A total of 15 spare parts were used to consider and demonstrate the accomplishment of the objective for this work.

The demand forecasting model has many machine learning algorithms available to handle this task. This work has compared the performance of supervised learning models Random Forest Regression, from the open-source Python package Scikit Learn [19] to the TFT model. In Table 2, TFT shows better performance in most of the observed spare parts in groups F and S of Frequency Usage Analysis.

Table 2: *The comparison of MAE between the actual and predicted demand from the demand forecasting models using Random Forest Regression and TFT.*

Spare Part No.	Frequency Usage Analysis	List of Demand in 12 Coming Months			MAE	
		Random Forest Regression	TFT	Actual	Random Forest Regression	TFT
1	FF	[46 33 36 49 44 56 32 41 25 31 30 18]	[74 0 72 0 76 70 33 77 0 55 0 80]	[0 0 95 0 0 0 0 0 0 0 0 0]	37.00	40.67
2	FF	[5 2 4 4 2 4 3 3 3 3 2 3]	[3 3 2 3 3 3 3 3 2 2 3 2]	[0 0 6 2 0 0 2 0 2 0 0 0]	2.33	2.33
3	FF	[78 113 71 122 123 176 72 85 120 173 169 148]	[15 33 41 28 15 18 15 19 23 11 24 19]	[159 149 238 55 109 108 111 238 137 108 324 108]	92.67	131.92
4	FF	[2 2 2 2 2 2 2 2 2 2 2 2]	[3 3 2 3 3 3 3 3 2 2 3 2	[7 4 0 3 0 2 2 3 0 4 0 7]	1.58	2.00
5	FF	[113 103 82 90 108 92 88 88 97 99 90 94]	[16 27 37 24 16 15 16 18 22 11 29 27]	[0 0 100 150 50 100 50 100 0 150 50 200]	51.67	68.50
6	F	[14 3 0 17 -15 6 -6 -1 24 5 7 -10]	[0 2 0 0 0 0 0 0 2 0 0 2]	[0 0 0 16 0 0 20 0 0 32 0 0]	7.83	6.17
7	F	[2 6 5 4 2 6 3 4 10 6 10 13]	[0 0 0 0 0 0 2 0 0 0 0 2]	[0 0 0 0 0 50 0 0 0 50 20 0]	12.92	10.33
8	F	[2 1 2 1 1 0 5 1 3 1 4 1]	[0 0 0 0 0 0 0 2 0 0 0 2]	[0 0 0 0 0 0 20 0 4 0 16 0]	4.83	3.67
9	F	[2 1 3 0 2 1 3 0 4 1 2 2]	[0 0 0 0 0 0 0 2 0 0 0 1]	[0 0 0 0 0 0 20 0 4 0 16 0]	3.58	3.58
10	F	[2 4 5 2 4 4 3 4 3 2 3 4]	[0 0 0 0 2 0 0 0 0 0 0 2]	[0 20 0 0 0 0 0 0 0 0 0 0]	3.33	2.00
11	S	[1 0 1 1 1 1 1 0 0 0 1 0]	[0 2 0 0 0 0 0 0 0 1 0 2]	[0 0 0 0 0 0 0 0 0 0 0 0]	0.50	0.42
12	S	[5 2 12 3 6 1 1 2 1 4 2 0]	[0 0 7 0 0 0 0 0 0 0 0 4]	[0 0 0 0 0 0 0 0 0 0 0 0]	3.25	0.92
13	S	[2 0 3 0 0 0 2 0 0 3 1 1]	[0 0 0 0 0 0 0 0 0 0 0 2]	[0 0 0 0 0 0 0 0 0 0 0 0]	1.0	0.17
14	S	[1 2 2 2 2 2 1 2 1 2 2 1]	[0 0 0 0 0 1 0 0 2 0 0 2]	[0 0 0 0 0 0 0 0 0 0 0 0]	1.67	0.42
15	S	[2 2 2 2 2 3 2 3 3 1 2 2]	[0 0 0 0 0 0 0 0 0 0 0 0]	[0 0 0 24 0 0 0 0 0 0 0 0]	3.83	2.00

To measure the improvement of the simulation for the spare parts inventory using the demand forecasting model, the observed spare part historical data was tested with the Kolmogorov-Smirnov Test to find the fitted distribution [20]. It used random numbers for the demands of 12 coming months then calculated MAE with the test dataset. The comparison of MAE between the demand forecasting model and the fitted distribution is listed in Table 3, It shows most of the observed spare parts have lower MAE in the demand forecasting model using TFT. TFT better handles the stochastic nature of the spare parts demand, especially the observed spare parts in groups F and S.

Table 3: *The comparison of MAE between the actual and predicted demand from the fitted distribution and the demand forecasting model using TFT.*

Spare Part No.	Frequency Usage Analysis	List of Demand in 12 Coming Months			MAE	
		Fitted Distribution	Demand Forecasting Using TFT	Actual	Fitted Distribution	Demand Forecasting Using TFT
1	FF	[66 78 37 59 51 56 94 -18 69 27 33 30]	[74 0 72 0 76 70 33 77 0 55 0 80]	[0 0 95 0 0 0 0 0 0 0 0 0]	53.25	40.67
2	FF	[0 4 4 5 3 5 5 -4 3 6 5 2]	[3 3 2 3 3 3 3 3 2 2 3 2]	[0 0 6 2 0 0 2 0 2 0 0 0]	3.17	2.33
3	FF	[92 7 53 -47 22 67 95 52 95 -18 44 51	[15 33 41 28 15 18 15 19 23 11 24 19]	[159 149 238 55 109 108 111 238 137 108 324 108]	110.92	131.92
4	FF	[3 1 2 0 0 1 -1 3 2 2 2 5]	[3 3 2 3 3 3 3 3 2 2 3 2	[7 4 0 3 0 2 2 3 0 4 0 7]	2.00	2.00
5	FF	[55 82 45 115 20 -13 -50 -12 43 -12 77 100]	[16 27 37 24 16 15 16 18 22 11 29 27]	[0 0 100 150 50 100 50 100 0 150 50 200]	76.17	68.50
6	F	[14 3 0 17 -15 6 -6 -1 24 5 7 -10]	[0 2 0 0 0 0 0 0 2 0 0 2]	[0 0 0 16 0 0 20 0 0 32 0 0]	11.17	6.17
7	F	[-67 49 -63 32 62 20 -28 -3 9 11 11 14]	[0 0 0 0 0 0 2 0 0 0 0 2]	[0 0 0 0 0 50 0 0 0 50 20 0]	33.75	10.33
8	F	[8 4 5 -4 4 3 4 4 -4 2 -1 2]	[0 0 0 0 0 0 0 2 0 0 0 2]	[0 0 0 0 0 0 20 0 4 0 16 0]	6.42	3.67
9	F	[-2 9 -2 -2 3 -3 0 -2 9 6 2 0]	[0 0 0 0 0 0 0 2 0 0 0 1]	[0 0 0 0 0 0 20 0 4 0 16 0]	5.67	3.58
10	F	[1 -12 5 4 0 -3 1 5 0 21 -3 -2]	[0 0 0 0 2 0 0 0 0 0 0 2]	[0 20 0 0 0 0 0 0 0 0 0 0]	6.42	2.00
11	S	[3 2 7 3 -4 0 0 13 10 9 2 0]	[0 2 0 0 0 0 0 0 0 1 0 2]	[0 0 0 0 0 0 0 0 0 0 0 0]	4.42	0.42
12	S	[15 -8 42 86 -40 -28 -7 86 44 138 78 30]	[0 0 7 0 0 0 0 0 0 0 0 4]	[0 0 0 0 0 0 0 0 0 0 0 0]	50.17	0.92
13	S	[1 -6 38 -20 14 -8 6 -5 -7 -7 29 6]	[0 0 0 0 0 0 0 0 0 0 0 2]	[0 0 0 0 0 0 0 0 0 0 0 0]	12.25	0.17
14	S	[-5 5 -2 4 -5 0 2 -6 3 1 -6 -4]	[0 0 0 0 0 1 0 0 2 0 0 2]	[0 0 0 0 0 0 0 0 0 0 0 0]	3.58	0.42
15	S	[-9 4 -8 0 1 -1 12 8 9 -2 8 5]	[0 0 0 0 0 0 0 0 0 0 0 0]	[0 0 0 24 0 0 0 0 0 0 0 0]	7.58	2.00

Following the methodology of this work, the future demand prediction of the observed spare parts is fed into the simulation model of the spare parts inventory to obtain the inventory policy performance in terms of Fill Rate (β Service Level) for each spare part. Following Table 4, the Fill Rate from the simulation model of the observed spare part is compared with the required Service Level and the actual Fill Rate. The results of the validation of the Fill Rate is that all matched in the group of Frequency Usage Analysis F and S.

Table 4: *Fill Rate of the observed spare parts.*

Spare Part No.	Frequency Usage Analysis	Required Service Level	Fill Rate	
			Simulation	Actual
1	FF	0.99	1.00	1.00
2	FF	0.95	1.00	1.00
3	FF	0.99	1.00	0.64
4	FF	0.99	1.00	0.94
5	FF	0.99	1.00	0.88
6	F	0.99	1.00	1.00
7	F	0.99	1.00	1.00
8	F	0.99	1.00	1.00
9	F	0.99	1.00	1.00
10	F	0.99	1.00	1.00
11	S	0.95	1.00	1.00
12	S	0.99	1.00	1.00
13	S	0.99	1.00	1.00
14	S	0.99	1.00	1.00
15	S	0.99	1.00	1.00

For the observed spare parts in the group of Frequency Usage Analysis FF, the observed spare part Fill Rate is matched, similar to, or different. Focusing on this group by running 100 replicas of the simulation with demand as a quantile forecast to yield the best case and the worst case of Fill Rate yields the results shown in Table 5. For the observed spare part numbers 1, 3, and 5, the average Fill Rate is in the range of the required Service Level.

Table 5: Fill Rate of the observed spare parts in the group of Frequency Usage Analysis FF from running 100 replicas of the simulation with demand as a quantile forecast.

Spare Part No.	Frequency Usage Analysis	Required Service Level	Fill Rate		Fill Rate: 100 replicas with quantile forecasted demand			
			Simulation	Actual	AVG	SD	Best	Worst
1	FF	0.99	1.00	1.00	0.89	0.10	1.00	0.54
2	FF	0.95	1.00	1.00	0.47	0.08	0.75	0.27
3	FF	0.99	1.00	0.64	0.99	0.03	1.00	0.80
4	FF	0.99	1.00	0.94	0.36	0.08	0.57	0.16
5	FF	0.99	1.00	0.88	0.98	0.04	1.00	0.84

For spare part numbers 2 and 4, the best Fill Rate is 0.75 and 0.57, and the worst Fill Rate is 0.27 and 0.16, respectively. The averaged Fill Rate of both spare parts is also lower than the Fill Rate from the simulation with a point prediction. The output quantile forecast of the demand for the spare part number 2 is shown in Fig. 12, and the corresponding value for number 4 is shown in Fig. 13, which is far over the maximum demand of the historical usage data, which impacted the spare parts demand of the simulation model that is sampled from these quantiles.

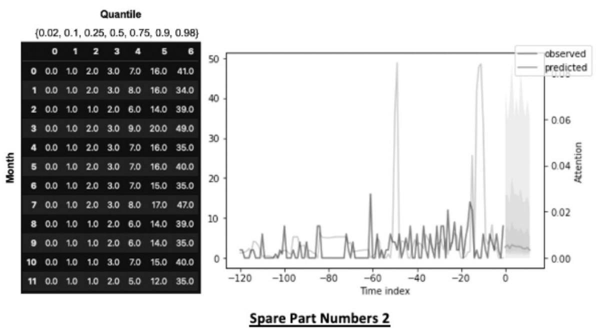


Fig.12: The output quantile forecast of spare part demand for the observed spare part number 2.

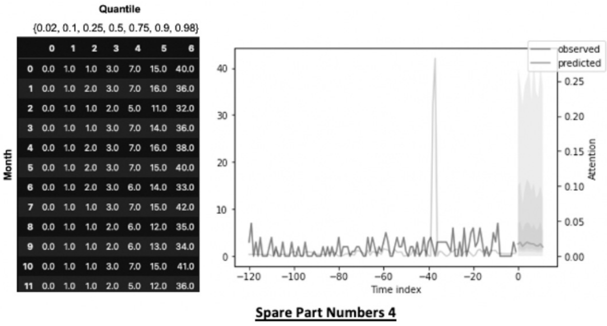


Fig.13: The output quantile forecast of spare part demand for the observed spare part number 2.

Take a look at the simulation outcome in case of the best Fill Rate of the observed spare part numbers 2 shown in Fig. 14. The list of predicted demand of 12 coming months sampled from the quantile forecast is [9, 7, 2, 0, 7, 4, 4, 0, 3, 13, 0, 12]. Some monthly demand is over the up-to level and that means a shortage of the spare parts occurred in this case. And the worst case of Fill Rate, which has a list of demand [26, 24, 5, 1, 1, 31, 26, 16, 1, 8, 26, 10], is shown in Fig. 15. The demand is over the up-to level is not the only cause of the shortage of spare parts. The high demand that is over the maximum level of the spare parts demand [26,24] and [31,26] has occurred in a very short time while the spare part purchasing orders have not been delivered yet.

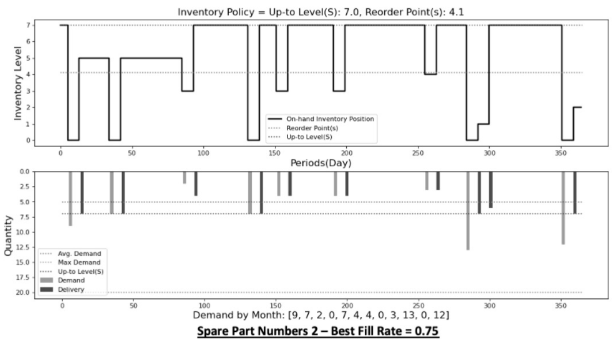


Fig.14: The simulation outcome in case of the best of Fill Rate for the observed spare part number 2.

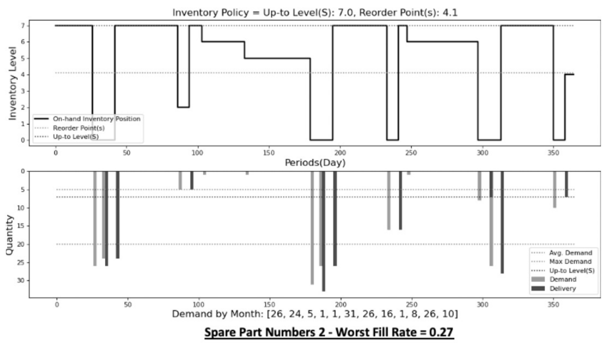


Fig.15: The simulation outcome in case of the worst of Fill Rate for the observed spare part number 2.

Finally, the analysis report, which is the information from the analysis of the simulation outcome, will be used by the spare part inventory management team to help them make a decision on the custom spare part inventory policy before deploying it to the actual system. An example of the analysis report in Microsoft Excel format is demonstrated in Fig. 16. It contains the result of the validation of the custom spare parts inventory policy: "Matched" - the Fill Rate from the simulation is matched with the required Service Level, "Acceptable with conditions" - the range of Fill Rate (Average+/-SD) from the simulation meets the required Service Level, and "Verifying is required" - the range of Fill Rate (Average+/-SD) from the simulation does not meet the required Service Level. In the best case for 100 replicas of the simulation it is lower than the required Service Level, and we need to verify the output quantile forecast of spare parts demand. The full details of the simulation outcome are shown in Fig. 17 to allow discovering the root cause of these results.

Spare Part No.	Inventory Policy		Required Service Level	Fill Rate: Simulation	Fill Rate: 100 replicas of the simulation with quantile forecasted demand						Validation of Inventory Policy
	Up-to Level(S)	Reorder Point(s)			Average	SD	Best	Best RepNo.	Worst	Worst RepNo.	
1	175	110.1	0.99	1.00	0.89	0.10	1.00	6	0.54	60	Acceptable with conditions
2	7	4.1	0.95	1.00	0.47	0.08	0.75	65	0.27	33	Verifying is required
3	194	173.1	0.99	1.00	0.99	0.03	1.00	2	0.80	67	Acceptable with conditions
4	6	5.1	0.99	1.00	0.36	0.08	0.57	60	0.16	39	Verifying is required
5	166	105.1	0.99	1.00	0.98	0.04	1.00	1	0.84	61	Acceptable with conditions
6	74	70.1	0.99	1.00	1.00	0.00	1.00	0	1.00	0	Matched
7	212	180.1	0.99	1.00	1.00	0.00	1.00	0	1.00	0	Matched
8	33	22.1	0.99	1.00	1.00	0.00	1.00	0	1.00	0	Matched
9	39	27.1	0.99	1.00	1.00	0.00	1.00	0	1.00	0	Matched
10	49	35.1	0.99	1.00	1.00	0.00	1.00	0	1.00	0	Matched
11	62	51.1	0.95	1.00	1.00	0.00	1.00	0	1.00	0	Matched
12	317	300.1	0.99	1.00	1.00	0.00	1.00	0	1.00	0	Matched
13	86	67.1	0.99	1.00	1.00	0.00	1.00	0	1.00	0	Matched
14	36	25.1	0.99	1.00	1.00	0.00	1.00	0	1.00	0	Matched
15	43	30.1	0.99	1.00	1.00	0.00	1.00	0	1.00	0	Matched

Fig.16: *An example of the analysis report in Microsoft Excel format.*

Item No.	ReplicationN	Inventoty Level(Initial)	Net Inventory Level(initial)	Demand	Inventoty Level(final)	Net Inventory Level(final)	Lost Sales	Purchase	Lead Time	Delivered
2	65	7	7	0	7	7	0	0	0	0
2	65	7	7	0	7	7	0	0	0	0
2	65	7	7	0	7	7	0	0	0	0
2	65	7	7	0	7	7	0	0	0	0
2	65	7	7	0	7	7	0	0	0	0
2	65	7	7	0	7	7	0	0	0	0
2	65	7	7	9	0	7	2	7	8	0
2	65	0	7	0	0	7	0	0	0	0
2	65	0	7	0	0	7	0	0	0	0
2	65	0	7	0	0	7	0	0	0	0
2	65	0	7	0	0	7	0	0	0	0
2	65	0	7	0	0	7	0	0	0	0
2	65	0	7	0	0	7	0	0	0	0
2	65	0	7	0	0	7	0	0	0	0
2	65	7	7	0	5	5	0	0	0	7

Fig.17: *An example of the full details of the simulation outcome (100 replicas of the simulation with quantile forecasted demand.*

5. CONCLUSIONS

In this article, the SEC-DEC code scheme is modified using a product code to enhance bit error correction performance. In addition, a weight reduction code is proposed to decrease the errors caused by writing '1' bits. This work presented a discrete-event simulation of a spare parts inventory system with demand forecasting used to validate the customized spare parts inventory policy. The demand forecasting model was trained on the historical data of the spare parts usage obtained from a real-world system. The trained demand forecasting model predicts the 12 coming months' demand for the spare parts for the simulation model.

The comparison of the performance of Random Forest Regression and TFT shows the model performance is similar. The main benefit of TFT is that it is capable of using cross-learning to train a global model for all of the spare parts in the dataset. It can perform multi-horizontal forward predictions with high prediction accuracy and that allows it to deal with the dynamics of the spare parts demand.

TFT not only provides point prediction but also provides quantile forecast prediction, which is the range of likely spare parts demand that are different in each month. In contrast, with a parametric estimation, the spare parts demand for all 12 coming months will follow only one distribution function.

Because of the capability of the quantile forecast of TFT, the spare parts inventory simulation model with demand forecasting using TFT is simulated with a demand variable that is closer to the stochastic nature of demand from the actual spare part inventory system and yields the best-case and the worst-case simulation outcomes. The result of the validation of the custom spare parts inventory from the simulation between the demand from the actual data and the demand from the forecasting model is matched in most of the observed spare parts. However, the spare parts that have quantile forecast predictions that are far over the average demand have to verify both the demand forecasting prediction and the simulation outcome to find out the root cause. This is required to find the solution to narrow down the error of the simulation outcome and create a specific demand forecasting model for these spare parts.

Future work should focus on three areas:

1. Clustering the spare parts with similar demand behavior, then training a specific model for each cluster to close the gap of actual demand and quantile forecast prediction to improve the simulation outcome.

2. Try a multivariate demand forecasting model, to find out the leading indicator affecting the spare parts demand, such as the number of the year after the turn-around process of the petrochemicals plant, the predictive maintenance of the engineering machinery, etc.

3. The spare parts inventory policy using an accurate simulation model will allow managers to optimize the inventory policy parameters ROP and Order-up-to Level. At the same time, the required Service Level can still met.

References

[1] P. Chaovalitwongse, Determining Inventory Policy: Theories and a Systematic Thinking Approach, BKK: ChulaPress, 2018.

[2] R. Kripunyapong , E. Shayan and F. Ghotb, "Supply Chain Management Problems: A Review," in *Proceedings of the 2nd International*

Conference on Operations and Supply Chain Managemen, BKK, 2007.

[3] S. Al-Fedaghi and N. Al-Huwais, "Conceptual Modeling of Inventory Management Processes as a Thinging Machine," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 9, no. 11, 2018.

[4] F. Zhang and Z. Guan, "Inventory management for spare parts on engineering machinery," *2015 Chinese Automation Congress (CAC)*, pp. 1951-1955, 2015.

[5] E. Spiliotis, S. Makridakis, A.-A. Semenoglou and V. Assimakopoulos, "Comparison of statistical and machine learning methods for daily SKU demand forecasting," *Operational Research*, 2020.

[6] A. Oroojlooyjadid, L. Snyder and M. Takac, "Applying Deep Learning to the Newsvendor Problem," 6 March 2018. [Online]. Available: `https://arxiv.org/abs/1607.02177`.

[7] A. M. Law, Simulation Modelling and Analysis, New York: McGraw-Hill Education, 2015.

[8] H. Susanto, M. N. Almunawar, M. S. Aksoy and Y. C. Tuan, "A Simulation Approach Paradigm: An Optimization and Inventory Challenge Case Study," 28 March 2012. [Online]. Available: https://arxiv.org/abs/1204.0225.

[9] A. Bhosekar, S. Eksioglu, T. Isık and R. Allen, "A Discrete Event Simulation Model for Coordinating Inventory Management and Material Handling in Hospitals," 11 June 2020. [Online]. Available: https://arxiv.org/abs/2006.06764.

[10] T. SimPy, "SimPy Documentation," 15 April 2020. [Online]. Available: `https://simpy.readthedocs.io/_/downloads/en/latest/pdf/`.

[11] A. Agarwal, "Validation of Inventory models for Single-echelon Supply Chain using Discrete-event Simulation," Chicago, 2018.

[12] A. Agarwal, "Multi-echelon Supply Chain Inventory Planning using Simulation-Optimization with Data Resampling," Chicago, 2019.

[13] G. Sharma, V. Tripathi, M. Mahajan and A. K. Srivastava, "Comparative Analysis of Supervised Models for Diamond Price Prediction," *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pp. 1019-1022, 2021.

[14] K. Bandara, P. Shi, C. Bergmeir, H. Hewamalage and Q. Tran, Sales Demand Forecast in E-commerce using a Long Short-Term Memory Neural Network Methodology.

[15] D. Salinas, V. Flunkert and J. Gasthaus, "DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks," 22 Febuary 2019. [Online]. Available: `https://arxiv.org/abs/1704.04110`.

[16] B. Lim, S. O. Arık, N. Loeffb and T. Pfister, "Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting," 27 September 2020. [Online]. Available: `https://arxiv.org/abs/1912.09363`.

[17] J. Beitner, "PyTorch Forecasting Documentation," 20 September 2020. [Online]. Available: `https://pytorch-forecasting.readthedocs.io/en/latest/`.

[18] E. Stevens, L. Antiga and T. Viehmann, Deep Learning with PyTorch, NY: Manning Publications Co., 2020.

[19] L. Buitinck, G. Louppe, M. Blonde, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. Vanderplas, A. Joly, B. Holt and G. Varoquaux, "API design for machine learning software: experiences from the scikit-learn project," 2013. [Online]. Available: `https://arxiv.org/abs/1309.0238`.

[20] M. Choy and M. Cheong, "Identification of Demand through Statistical Distribution Modeling for Improved Demand Forecasting," 1 October 2011. [Online]. Available: https://arxiv.org/abs/1110.0062.

Sanit Innuphat received his Master's degree from the School of Science and Technology, University of the Thai Chamber of Commerce (UTCC). His Bachelor's degree from the faculty of engineering, King Mongkut's University of Technology North Bangkok (KMUTNB). He is currently a data scientist at one of the largest petrochemical industries in Thailand. Nowadays, his fields of interest area are the application of machine learning, deep learning, text mining, and natural language processing for industrial supply chain management and optimization.

Manachai Toahchoodee is a Professor in the Computer Science Department at the University of the Thai Chamber of Commerce. Prior to joining UTCC, he was a Senior Analyst at IBM Solutions Delivery Co., Ltd., Thailand. He obtained his Ph.D. from Colorado State University under the supervision of Professor Indrakshi Ray. His Master's degree in Information Technology is from King Mongkut's University of Technology Thonburi, Bangkok, Thailand. His Bachelor of Science degree in Mathematics is from King Mongkut's Institute of Technology Thonburi, Bangkok, Thailand.

Dr. Toahchoodee's research interests include security and privacy, ubiquitous computing and formal methods in software engineering. He has served in various capacities for journals and conferences. He was the Session Chair of ICDMAS 2014 and ASTC 2014, and General Chair of DICTAP 2012 and DICTAP 2014. He has served on the program committees of various conferences including ASTeM and HEABPRO.