

# Machine learning for multi-criteria inventory classification applied to intermittent demand

**Lolli Francesco<sup>a</sup>, Balugani Elia<sup>a</sup>, Ishizaka Alessio<sup>b</sup>, Gamberini Rita<sup>a</sup>, Rimini Bianca<sup>a</sup>,  
Regattieri Alberto<sup>c</sup>**

<sup>a</sup> *Department of Sciences and Methods for Engineering, University of Modena and Reggio Emilia,  
Via Amendola 2 – Padiglione Morselli,  
42100 Reggio Emilia, Italy*

<sup>b</sup> *Centre of Operations Research and Logistics, Portsmouth Business School, University of Portsmouth,  
Portland Street – Richmond Building,  
Portsmouth PO1 3DE, UK*

<sup>c</sup> *Dipartimento di Ingegneria Industriale, University of Bologna, Viale Risorgimento 2, Bologna*

## **Abstract**

Multi-criteria inventory classification groups inventory items into classes, each of which is managed by a specific re-order policy according to its priority. However, the tasks of inventory classification and control are not carried out jointly if the classification criteria and the classification approach are not robustly established from an inventory-cost perspective. Exhaustive simulations at the single item level of the inventory system would directly solve this issue by searching for the best re-order policy per item, thus achieving the subsequent optimal classification without resorting to any multi-criteria classification method. However, this would be very time-consuming in real settings, where a large number of items need be managed simultaneously.

In this paper, a reduction in simulation effort is achieved by extracting from the population of items a sample on which to perform an exhaustive search of best re-order policies per item; the lowest cost classification of in-sample items is therefore achieved. Then, in line with the increasing need for ICT tools in the production management of Industry 4.0 systems, supervised classifiers from the machine learning research field (i.e. support vector machines with a Gaussian kernel and deep neural networks) are trained on these in-sample items to learn to classify the out-of-sample items solely on the basis of the values they show on the features (i.e. classification criteria).

The inventory system adopted here is suitable for intermittent demands, but it may also suit non-intermittent demands, thus providing great flexibility.

The experimental analysis of two large datasets showed an excellent accuracy, which suggests that machine learning classifiers could be implemented in advanced inventory classification systems.

**Keywords:** Multi-criteria inventory classification; Inventory control; Industry 4.0; Machine learning; Intermittent demand

## 1. Introduction

In this paper, we focus on classifying items for inventory control. When a classification framework is applied to inventories, it determines the importance of items and the level of controls placed on the items (Onwubolu and Dube 2006). An appropriate re-order policy needs to be selected for each class (Nenes *et al.* 2010), so that applying a specific re-order policy per class rather than per item simplifies the approach to dealing with every aspect of stock control. However, classifying items through multi-criteria inventory classification (MCIC) approaches is a separate action from finding appropriate strategies for each class (Mohammaditabar *et al.* 2012, Babai *et al.* 2015). The performance of MCIC methods does not necessarily include the performance of the inventory control system and little attention has been paid to the empirical implications of a simultaneous inventory classification and inventory control (Bacchetti *et al.* 2012). The original goal of minimising the total relevant cost is therefore forgotten when MCIC and inventory systems are not carried out jointly.

Clearly, an exhaustive simulation of the inventory system performed at a single item level would avoid recourse to MCIC by optimally classifying all the items *ex post* in an inventory-cost perspective, solving the aforementioned open issue. However, an exhaustive simulation of the whole population of items requires computation efforts that are too high for time-varying settings with thousands of items needing to be managed.

This paper investigates the use of supervised machine learning classifiers as effective ICT tools for MCIC, in particular support vector machines with Gaussian kernel (SVM) and deep neural networks (DNN). Machine learning tools represent a challenge for smart manufacturing, also known as Industry 4.0 - see Wuest *et al.* (2016) for a review on the applications of machine learning in smart manufacturing.

Machine learning needs a large amount of data to work well, and enables complex decision-making processes to be automated. MCIC applied to thousands of items is one of these complex processes, and machine learning tools may therefore also be promising in

MCIC. However, the literature is particularly poor on this specific area of research. From a methodological viewpoint, this two-stage automatic classification approach (i.e. classification via simulation of the in-sample items and classification via machine learning of the out-of-sample items) could be extended to any inventory system. However, the choice of the inventory system depends on the demand pattern to be managed. In this paper, intermittent demand patterns are considered, which require ad hoc inventory systems in terms of forecasting approaches, probabilistic assumptions and subsequent mathematical modelling, but the inventory system selected here is flexible enough to also suit non-intermittent demand. In particular, a periodic order-up-to level inventory control system is adopted, where the decision variable defining the classes, and thus driving the item classification, is the review interval. Hence, the reason to classify the items is different from that used in standard MCIC approaches (see Section 2.2), where a high target cycle service level is assigned to high ranked items. In our case, the higher the criticality of an item, the lower the review interval should be, and vice versa.

Section 2 reviews the main published contributions related to the research. Section 3 contains the notations used, and presents the assumptions underlying the inventory system. Section 4 details the steps involved in the solution. Section 5 describes the experimental analysis performed to investigate the effectiveness of the proposal. Finally, Section 6 outlines the conclusions and directions for further research.

## **2. Literature review**

Given that the literature on the inventory theory is very broad, we focus only on the aspects relevant for our study. Section 2.1 addresses the forecasting and stock control for intermittent demand. Section 2.2 contains various important contributions on MCIC, and particularly on the application of machine learning classifiers to MCIC.

### **2.1. Forecasting and stock control for intermittent demand**

The demand for spare parts is typically intermittent, and the literature has investigated the effectiveness of demand forecasting and stock control methodologies in said context. For a complete discussion of this topic, the reader can refer to the book edited by Altay and Litteral (2011). Cavalieri *et al.* (2008) provided a decision-making framework for managing spare parts, and Driessen *at al.* (2015) suggested a possible roadmap of the future agenda on this stream of research.

Croston's method (1972) represents the seminal contribution on intermittent demand forecasting according to a normally distributed demand size and a Bernoulli probability of a demand occurrence. A simple exponential smoothing is applied to both variables when the demand occurs and then an estimator of the expected value of demand per period is evaluated by the ratio of these smoothing effects. Syntetos and Boylan (2001) showed that this estimator is theoretically biased, and proposed an approximately unbiased modification of Croston's method called Syntetos-Boylan Approximation (SBA).

On the modifications of Croston's method, the reader can refer to Levén and Segerstedt (2004), Boylan and Syntetos (2007), Dhakshayani and Narayanan (2014), Teunter and Sani (2009) and Babai *et al.* (2014). Regattieri *et al.* (2005) and Downing *et al.* (2014) compared other forecasting approaches for intermittent demand.

Artificial intelligence has also been applied for forecasting intermittent demand because of its ability to generalise a non-linear process without requiring any distributional assumptions; see for example Gutierrez *et al.* (2008), Kourentzes (2013), and Lolli *et al.* (2017), who tested several different artificial neural networks. When the intermittent demand patterns also contain seasonal and trend components, SARIMA (Seasonal Autoregressive Integrated Moving Average) modelling shows promising results (Gamberini *et al.* 2010, Lolli *et al.* 2011, Lolli *et al.* 2014a).

Although demand forecasting and stock control contribute equally towards the efficiency of inventory management (Tratar 2015), it should be emphasised that the comparison of forecasting methods was performed by the aforementioned authors in terms of forecasting accuracy, which does not necessarily lead to better performance of the inventory system in terms of total relevant cost.

With regards to the stock control of intermittent demand profiles, the literature can be classified into two main streams.

The first stream provides a wide set of compound distributions for modelling purposes. However, as emphasised by Babai *et al.* (2015), as the data become more erratic, the true demand size distribution may not comply with any standard theoretical distribution. In particular, on the statistical analysis of intermittent demand, Zotteri (2000), Eaves (2002), Teunter *et al.* (2010b), Babai *et al.* (2011), and Gamberini *et al.* (2014). On the interaction between forecasting and stock control, see Syntetos and Boylan (2006).

The second stream relating to stock control focuses on the performance of different inventory systems on a wide range of items taken from industrial case studies. Clearly, this stream of research also matches MCIC (Section 2.2). The effectiveness of inventory

systems in the case of intermittent demand has been explored in several works (e.g. Porras and Dekker 2008, Syntetos *et al.* 2009; Nenes *et al.* 2010, Syntetos *et al.* 2010, Conceição *et al.* 2015), however the item classification is again created before testing the inventory system on the generated classes, and thus appears disjointed.

## **2.2. Multi-criteria inventory classification**

A pioneering contribution on MCIC was provided by Flores *et al.* (1992). They applied the Analytical Hierarchy Process (AHP) to classify items in terms of annual usage value (given by the product of the unit cost and the total annual demand), average unit cost, criticality, and the lead-time. This represents one of the early attempts to overcome the weakness shown by the mono-criterion classification on usage value in representing the whole criticality of an item. AHP is also adopted by other authors (e.g. Partovi and Burton 1993, Partovi and Hopton 1994, Cakir and Canbolat 2008, Lolli *et al.* 2014b), and in particular for spare parts classification (Scala *et al.* 2014). Other multi-criteria methods have been used, such as weighted optimisation (e.g. Ramanathan 2006, Zhou and Fan 2007, Ng 2007, Hadi-Vencheh 2010, Chen 2011, Ladhari *et al.* 2015, Ishizaka *et al.* 2018) and case-based reasoning (e.g. Chen *et al.* 2008 and Soylu and Akyol 2014). Since the final goal is to associate the classes with specific inventory control methods, a further open issue to investigate is the correlation between the classification criteria and the efficiency of the inventory system. Only two papers have proposed an inventory classification from an inventory cost perspective. Zhang *et al.* (2001) presented an item classification scheme that is consistent with their cost optimization model. A constrained model is proposed (constrained to the target cycle service level and order frequency) instead of a cost-based model including order and shortage costs, through a continuous review inventory system. They derived an expression for reorder points to obtain an item classification ratio: an item is ranked as high (i.e. with a high target cycle service level) if the demand rate is large, or if the squared holding cost and the replenishment lead time are small. Starting with a total cost model including the shortage cost, Teunter *et al.* (2010a) derived an exact expression of the cycle service level with a different ratio for item classification. An item is ranked high if the demand rate and shortage cost are large or if the holding cost and the order quantity are small. They show empirically that their new classification criterion outperforms all the other methods compared.

Regardless of the classifying criteria and the multi-criteria approach used, item classification requires a sorting method, so that the exercise of ranking items does not

achieve the objective (Ishizaka and Nemery 2013). In other words, the cardinalities of the clusters should be subjectively established by decision-makers based on their expertise (see the criticism in Lolli *et al.* 2014b). It follows that the classes are disjointed in relation to inventory control, with items not being robustly classified.

To the best of our knowledge, only a few contributions have focused on applying machine learning classification algorithms to MCIC, and none have been extended to the inventory system.

The first work on applying machine learning classification algorithms to MCIC dates back to 2002 (Partovi and Anandarajan, 2002), where neural networks (NNs) were used to classify small populations of items by training the networks on a sample of items previously classified by decision-makers on subjective criteria (called 'features' in machine learning theory). However, the inventory system was not considered, and thus the classification obtained is not inventory cost-oriented. Similarly, Yu (2011) experimented with SVM, NN and the k-nearest neighbour algorithm on Reid's famous dataset (1987) of forty-seven items using the approaches of Flores *et al.* (1992), Ramanathan (2006), and Ng (2007) as benchmarks. SVM has been shown to be the most accurate machine learning classifier to replicate these MCIC approaches. However, it does not take into account the cost-oriented optimality of the classifications. Kabir and Hasin (2013) tested single and two-hidden layer NNs on a dataset of 351 items, which had already been classified by a fuzzy AHP priority scoring method. Again the classification is not cost-oriented and the MCIC remains disjointed from the inventory system. López-Soto *et al.* (2016) exploited Logical Analysis of Data, a supervised data mining technique able to generate patterns into classes by a Boolean function, in order to detect biases or inconsistencies in the classification provided by decision-makers. Kartal *et al.* (2016) combined several multi-criteria decision making approaches with machine learning algorithms for MCIC. Multi-criteria approaches are used to classify items into classes, and then machine learning classifiers (i.e. naïve Bayes, Bayesian network, NN, and SVM) are trained on the resulting classifications in order to predict the classes to which the items belong. NN and SVM were shown to have the highest accuracy. However, the accuracy of machine learning classifiers was again evaluated with respect to classifications without considering the inventory system. As a consequence, the selection of classification criteria from an inventory-cost perspective remains an open issue.

Conversely, our proposal is geared towards reducing the human contribution with its potential fallibility. In fact, the lowest cost classification of in-sample items obtained by the

exhaustive simulation of the inventory system, which is used to train the machine learning classifiers, fulfils this goal.

### 3. Notation and framework

Below is the notation adopted in this paper.

*Setting dimensions:*

$K$  = number of classes of items

$J$  = number of classification criteria

$T$  = number of time periods used for simulation

$n$  = number of warm-up periods ( $(T - n)$  is thus the number of collection periods)

*Sets:*

$I$  = population of items

$I'$  = sample of items  $I' \subseteq I$

$C_{k,I}$  =  $k$ -th class of items into  $I$ , with  $k = 1, \dots, K$

$C_k$  =  $k$ -th class of items into  $I'$ , with  $k = 1, \dots, K$

*Item features:*

$D_i$  = demand of item  $i$ , with  $i = 1, \dots, |I|$  (stochastic)

$p_i$  = positive demand probability of item  $i$ , with  $i = 1, \dots, |I|$  (stochastic)

$D_i^+$  = positive demand size of item  $i$ , with  $i = 1, \dots, |I|$  (stochastic)

$Std(D_i^+)$  = standard deviation of the positive demand size of item  $i$ , with  $i = 1, \dots, |I|$

$MSE_i$  = mean square error of the positive demand size of item  $i$ , with  $i = 1, \dots, |I|$   
(calculated over  $T$  periods)

$D_{i,t}$  = demand of item  $i$  in time period  $t$ , with  $i = 1, \dots, |I|$  and  $t = 1, \dots, T$  (events of  $D_i$ )

$RLT_i$  = replenishment lead time of item  $i$ , with  $i = 1, \dots, |I|$  (deterministic)

$tCSL_i$  = target cycle service level of item  $i$ , with  $i = 1, \dots, |I|$  (assigned)

$F_{i,t}$  = forecasted demand of item  $i$  for time period  $t$ , with  $i = 1, \dots, |I|$  and  $t = 1, \dots, T$   
(calculated at the end of period  $(t - 1)$ )

$S_{i,t}$  = order-up-to level of item  $i$  in time period  $t$ , with  $i = 1, \dots, |I|$  and  $t = 1, \dots, T$  (calculated at the end of period  $(t - 1)$ )

$C_i$  = unitary purchasing cost of item  $i$ , with  $i = 1, \dots, |I|$  (deterministic)

$h_i$  = unitary holding cost of item  $i$ , with  $i = 1, \dots, |I|$  (deterministic)

$o_i$  = unitary ordering cost of item  $i$ , with  $i = 1, \dots, |I|$  (deterministic)

*Decision variables:*

$R_i$  = review interval of item  $i$ , with  $i = 1, \dots, |I|$

The demand generation process is obtained using two probability distributions, i.e. a Bernoulli distribution generating the positive demand events with a probability  $p_i$  and a normal distribution generating  $D_i^+$  when a positive demand takes place. The goal is to assign  $K$  different values of the review interval  $R_i$  to the items into the population  $I$ . That is to say,  $R_i$  drives the classification of the items into  $I$  and may vary in a range of  $K$  feasible values for all the population items. This parameter was exhaustively tested by simulating the inventory control system on the sample  $I' \subseteq I$ , reaching by this way the best classification of the  $|I'|$  in-sample items. In fact, the value minimising a total cost function is selected per item into  $I'$ , and thus it follows that each item is classified into the class  $C_k$  associated with its best review interval. These in-sample items are then used to train different machine learning classifiers for comparison, whose goal is therefore to classify the remaining  $(|I| - |I'|)$  items into the  $K$  classes  $C_{k,l}$  without having to resort to an exhaustive search on all of them, but only on the values they show on the  $J$  criteria. A forecast-based periodic inventory control system is implemented, where the inventory status of each item is periodically reviewed and an order is placed to reach the order-up-to level  $S_{i,t}$ . This is dynamically computed with a model originally introduced by Syntetos *et al.* (2010), where the aim is to reach a target service level  $tCSL_i$ , assigned to single items as the probability of not incurring in a backorder during a replenishment cycle.

Some of the features of the analysed inventory system are clarified below:

- Time is treated as a discrete variable on single time units, called periods (e.g. days, weeks, months). All the following time measures are expressed as periods.
- The independence among the items is hypothesised, i.e. a single-item inventory system is proposed as in most of the literature related to intermittent demand (Section 2.1).
- The replenishment lead times are considered deterministic.
- Backorders are allowed for all items, but backordering costs are not available. This implies an inventory system constrained by a measure of service level, in particular the target service level.

#### **4. The solution approach**



This section presents our approach. First, the inventory control system adopted is explained (Section 4.1). It is then exhaustively simulated on a sample  $I'$  of items belonging to  $I$  in order to reach the best re-order policy at a single item level (Section 4.2). Following this simulation, the  $J$  classification criteria are defined (Section 4.3), and the machine learning classifiers adopted (i.e. SVM and DNN) are presented (Section 4.4).

#### 4.1. The forecast-based inventory control system

A periodic review system is employed given its simplicity and compatibility with physical warehouse reviews and periodic orders issued to the suppliers, as in the literature (e.g. Syntetos and Boylan 2006). We chose a  $(R, S)$  periodic review, with review interval  $R$  and order-up-to level  $S$  dynamically computed per item on a forecast basis every  $R$  periods.  $S$  depends on the time period  $t$  and the item  $i$  ( $S_{i,t}$ ), while  $R$  needs to be established for each item  $i$  and drives the inventory classification.

Since time is treated as a discrete variable in periodic order-up-to level policies, let  $t$  be a generic time period at the end of a replenishment cycle of  $R$  periods. Supposing that orders are placed at the very end of the replenishment cycles, on the boundary with the next period, and item  $i$  has a replenishment lead time of  $RLT_i$  periods, the chronological order of events in  $t$  is assumed as follows: if an order is placed in  $(t - RLT_i - 1)$ , at the very end of this period, it arrives at the beginning of period  $t$ ; a demand occurs (if it is not null) and  $S_{i,t}$  is re-computed on the forecast basis. The rule for emitting orders is: after every  $R$  periods if the inventory position  $IP_{i,t}$  (composed of net inventory  $NI_{i,t}$  + planned orders - backorders) is below  $S_{i,t}$  an order to reach  $S_{i,t}$  is placed. The inventory position ( $IP_{i,t}$ ) and the net inventory ( $NI_{i,t}$ ) have different meanings: the former contains the balance between issued but not yet delivered orders, and backorder units to be delivered as soon as available, the latter is the physical inventory status. If backorders do not occur and  $R \geq RLT_i + 1$ , i.e. an order always arrives before the next one is issued, the net inventory equals the inventory position. Conversely, if  $R < RLT_i + 1$ , replenishment cycles are allowed to overlap and a new order can be placed before the last one issued arrives. Both Croston's (CR) (Croston 1972) and the SBA method (Syntetos and Boylan 2005) are applied in the experimental analysis as estimators of the mean demand per period. Syntetos *et al.* (2005) establish regions (characterised by the average demand interval and squared coefficient of variation) of superior performance for each estimator based on a theoretically quantified error measure, assuming the demand occurs as a Bernoulli

process (i.e. the inter-demand intervals are geometrically distributed). The forecasts for period  $t$  provided by CR and SBA for the item  $i$  at the end of period  $(t - 1)$  are given respectively by:

$$F_{i,t}(CR) = \frac{\hat{Z}_{i,t}}{\hat{T}_{i,t}} \quad (1)$$

$$F_{i,t}(SBA) = \left(1 - \frac{\alpha}{2}\right) \frac{\hat{Z}_{i,t}}{\hat{T}_{i,t}} \quad (2)$$

where:

$$\hat{Z}_{i,t} = \hat{Z}_{i,t-1} + \beta(Z_{i,t-1} - \hat{Z}_{i,t-1}) \quad (3)$$

$$\hat{T}_{i,t} = \hat{T}_{i,t-1} + \gamma(T_{i,t-1} - \hat{T}_{i,t-1}) \quad (4)$$

are the estimated demand size (Equation 3) and interval (Equation 4), updated by a single exponential smoothing at the end of the periods in which the demand occurs.  $Z_{i,t-1}$  is the actual value of the demand, and  $T_{i,t-1}$  is the actual value of the time between consecutive transactions.

Since a pure cost objective function (composed of holding, backordering and ordering costs) cannot be evaluated due to the unavailability of backordering costs, a constraint perspective is suggested. In this scenario the minimisation of a cost function including the holding and ordering cost is pursued, while satisfying a pre-specified service level. Two common alternatives to define the service level are the Cycle Service Level (*CSL*), i.e. the probability of not incurring a backorder during a replenishment cycle, and the fill rate, i.e. the fraction of demand that is satisfied directly from the stock on hand. For an in depth analysis, see Chopra and Meindl (2004).

We use the *CSL* for safety stock calculation by defining a target *CSL* ( $tCSL_i$ ) for each item  $i$ . For instance, in an assembly process, the base unit is required upstream of the flow, its

criticality being very high since its shortage may affect all the subsequent stages. The higher the criticality of an item, the higher its  $tCSL_i$  assigned by a decision-maker. When applying a forecast-based stock control, the variance of forecast errors can be used to calculate the safety stocks necessary to achieve  $tCSL_i$ . According to Syntetos *et al.* (2010), the single exponential smoothing of the mean squared error ( $MSE$ ) provides an estimator of the variance as follows:

$$MSE_{i,t} = \delta(D_{i,t-1} - F_{i,t-1})^2 + (1 - \delta)MSE_{i,t-1} \quad (5)$$

where  $(D_{i,t-1} - F_{i,t-1})$  is the difference between the actual demand in  $(t - 1)$  and the corresponding forecast (Equations 1 and 2) computed at the end of period  $(t - 2)$ . Syntetos *et al.* (2010) recommend using low smoothing parameters  $\alpha, \beta, \gamma$  and  $\delta$ , a conclusion reached by empirical investigation.

In order to dynamically compute  $S_{i,t}$  at the end of period  $(t - 1)$ , the following equation is applied:

$$S_{i,t} = (RLT_i + R)F_{i,t} + \phi_i^{-1}(tCSL_i)\sqrt{MSE_{i,t}(RLT_i + R)} \quad (6)$$

where  $\phi_i(\cdot)$  is the cumulative distribution function of the demand over  $(RLT_i + R)$ . The first right-hand-side term corresponds to the forecast demand for periods  $(RLT_i + R)$  in a stationary mean model, with  $F_{i,t}$  given either by Equation 1 (CR) or Equation 2 (SBA). The second term is the safety stock required to reach  $tCSL_i$  by applying a safety factor of  $\phi_i^{-1}(tCSL_i)$ .

In summary, given either CR or SBA as the forecasting method, the following parameters depend directly on item  $i$ :

- $RLT_i$  is deterministic and given for each item  $i$ .
- $tCSL_i$  refers to the criticality level of item  $i$ . It can be assigned using an MCIC method associating a unique  $tCLS$  to each item class.
- $\phi_i^{-1}(tCSL_i)$  depending on distributional assumptions in the project.

All the other parameters can be optimised (Section 4.2).

This inventory system can also be applied to non-intermittent demands, in this case Croston's estimator (Equation 1) becomes a single exponential smoothing. Conversely, the SBA contains a corrective factor, introduced ad hoc for intermittent demands by Syntetos and Boylan (2005). Nevertheless, the exhaustive search enables all the predictors to be tested and all factors to be optimized.

#### 4.2. Exhaustive simulation of the in-sample items

The forecast method (CR or SBA) and the parameter settings ( $\alpha, \beta, \gamma, \delta$  and  $R$ ) need to be established per item. Given a time horizon of  $T$  periods,  $\alpha, \beta, \gamma, \delta$  are optimised per item in a warm-up period of the first  $n$  periods by minimising the mean squared error between demands and estimators. The performance of the inventory system obtained by changing  $R$  among  $K$  values is collected for the remaining  $(T - n)$  periods. For a fixed  $R$ , this performance is given by the net inventory  $NI_{i,t,R}$  and the emitted orders  $NO_{i,t,R}$  for each period  $t$  between  $(n + 1)$  and  $T$ , defined as:

$$NO_{i,t,R} = \begin{cases} 1 & \text{If an order of item } i \text{ is emitted in time period } t \text{ with review interval } R \\ 0 & \text{Otherwise} \end{cases} \quad (7)$$

The total relevant cost ( $TRC_{i,R}$ ) per period depends on the review interval  $R$  as follows:

$$TRC_{i,R} = h_i \cdot C_i \cdot \sum_{t=n+1}^T \frac{NI_{i,t,R}}{T-n} + o_i \cdot \sum_{t=n+1}^T \frac{NO_{i,t,R}}{T-n} \quad (8)$$

where the first and second terms are the average holding and ordering costs per time period, respectively.  $h_i$  is the unitary holding cost, defined as the percentage of the unitary purchasing cost  $C_i$  for holding item  $i$  in stock for a certain period, while  $o_i$  is the unit ordering cost.

The  $R$  minimizing  $TRC_{i,R}$  (Equation 8) is selected for each item  $i$ . This is the most efficient  $R$  in terms of  $TRC_{i,R}$  after optimizing all the forecasting parameters.

The  $R$  selected for each item is the driver for classifying the  $|I'|$  in-sample items into  $K$  classes from an inventory perspective, that is all the items  $i \in I'$  are now classified into a class  $C_k$ , with  $k = 1, \dots, K$ . These items are therefore the references used for classifying the

whole population of  $|I|$  items by means of  $J$  classification criteria (Section 4.3). Given Equation 8, it is possible to calculate a theoretically optimal value of  $R$  for each  $i$  by solving Equation 51 (see Appendix A). In Section 5.3 this theoretical classification is compared with those achieved by the machine learning classifiers.

### 4.3. Classification criteria

In this work, the classification is inventory-oriented. The criteria expected to impact on the performance of the inventory control system, and consequently the item classification, are selected. These criteria belong to two different groups, which refer to the statistics of the time series and the characteristics of the items, respectively.

The first group includes the parameters required for setting the demand generation. They refer to the two stochastic variables of an intermittent demand process, i.e. the inter-arrival between successive demands and the positive demand size:

- Expected value of the positive demand probability  $E(p_i)$ , which is calculated as the ratio between the number of periods with positive demands and  $T$ .
- Expected value of the positive demand size  $E(D_i^+)$ , which is calculated as the mean demand (only positive values) over  $T$ .
- Expected value of the standard deviation of the positive demand  $E[Std(D_i^+)]$ , which is calculated as  $\sqrt{MSE_i}$  over  $T$ .

The second group of criteria derives from the inventory control system as well as from the approach adopted for the selection of the best policy per item:

- Relative unitary cost  $\frac{o_i}{c_i h_i}$ .
- Safety factor  $\phi_i^{-1}(tCSL_i)$ .

To generalise the subsequent sections, it is assumed that  $J$  criteria have been assessed for the item classification, and that an item  $i$  shows a value  $x_{ij}$  on criterion  $j$ , with  $i = 1, \dots, |I|$  and  $j = 1, \dots, J$ .

### 4.4. Machine learning classifiers

Two machine learning classifiers are used:

- Support vector machine (SVM) with radial basis function kernel (Vapnik, 1999).
- Deep neural network (DNN) (Bishop, 2006).

Both algorithms can be very effective in non-linear contexts and present a limited number of meta-parameters to be optimized.

For the SVM, the meta-parameters are:

- Box constraint ( $C$ ), penalty applied to incorrectly separated items.
- Kernel scale ( $scale$ ), the radial basis function kernel is multiplied by this scale factor.

While for the DNN, the relevant meta-parameters are:

- Number of hidden layers ( $N_{hidden}$ ).
- Number of neurons per hidden layer ( $N_{neurons}$ ).

For each algorithm and meta-parameter, a tenfold cross-validation is applied to assess the combination performance. A fold uses nine tenths of the items available (training set) to train the method and the remaining one tenth to predict the classes (validation set). The overall performance is evaluated by pooling the validation set predictions and reconstructing the original dataset, these pooled predictions are evaluated against the true classes of items and the ratio of correct predictions is measured. An optimal set of meta-parameters for each algorithm is identified as the one maximizing the prediction performance.

This cross-validation strategy enables the meta-parameters to be optimized while leveraging the entire dataset and limiting the impact of the single fold division on the measured performance.

#### 4.4.1 SVM with radial basis function kernel

The standard SVM is a two-class linear classifier dividing the features space with a hyperplane. The hyperplane is designed to maximize the margin ( $m$ ), double the distance between the hyperplane and its closest items in the features space, whilst maintaining the correct class division.

The distance between the hyperplane ( $\vec{w}^T \vec{x} + b$ ) and item  $i$  is:

$$r_i = \frac{\vec{w}^T \vec{x}_i + b}{\|\vec{w}\|}$$

(9)

If  $i$  is one of the closest items, then  $\vec{x}_i$  is a support vector and  $r_i$  equals  $\frac{m}{2}$ .

An item can be classified into class 1 or class -1 according to its relative position to the hyperplane:

$$class_i = sign(\vec{w}^T \vec{x}_i + b).$$

(10)

Following this notation, the optimal hyperplane is not well defined as it can be expressed by infinite combinations of  $\vec{w}$  and  $b$ . To obtain an unambiguous optimal solution, the following notational constraint is imposed:

$$|\vec{w}^T \vec{x}_{SV} + b| = 1$$

(11)

where  $\vec{x}_{SV}$  is any support vector.

This constraint does not change the optimal solution but enables  $m$  to be expressed in a simple form:

$$m = 2 \frac{\vec{w}^T \vec{x}_{SV} + b}{\|\vec{w}\|} = \frac{2}{\|\vec{w}\|}$$

(12)

The constraints bounding the items to their correct class are also expressed in simple terms. The original equations of the constraints are:

$$\frac{\vec{w}^T \vec{x}_i + b}{\|\vec{w}\|} \leq \frac{m}{2} \quad \text{if } y_i = -1$$

(13)

$$\frac{\vec{w}^T \vec{x}_i + b}{\|\vec{w}\|} \geq \frac{m}{2} \quad \text{if } y_i = 1$$

(14)

where  $y_i$  is the true class of item  $i$ , and becomes:

$$\vec{w}^T \vec{x}_i + b \leq 1 \quad \text{if } y_i = -1$$

(15)

$$\vec{w}^T \vec{x}_i + b \geq 1 \quad \text{if } y_i = 1$$

(16)

The optimization problem can be expressed with quadratic programming as:

$$\min \|\vec{w}\|^2$$

(17)

s.t.

$$y_i(\vec{w}^T \vec{x}_i + b) \geq 1 \quad \forall i = 1, \dots, |I'|$$

(18)

This optimization problem can be rewritten with the Wolfe Dual as:

$$\max \sum_{i=1}^{|I'|} \alpha_i - \frac{1}{2} \sum_{i=1}^{|I'|} \sum_{l=1}^{|I'|} \alpha_i \alpha_l y_i y_l \vec{x}_i^T \vec{x}_l$$

(19)

s.t.

$$\sum_{i=1}^{|I'|} \alpha_i y_i = 0$$

(20)

$$\alpha_i \geq 0 \quad \forall i = 1, \dots, |I'|$$

(21)

where the solution to the primal problem is:

$$\vec{w} = \sum_{i=1}^{|I'|} \alpha_i y_i \vec{x}_i$$

(22)

$$b = y_l - \sum_{i=1}^{|I'|} \alpha_i y_i \vec{x}_i^T \vec{x}_l \quad \forall l: \alpha_l \neq 0$$

(23)

It is possible to implement the SVM in cases when the dataset is not linearly separable. In these cases, the system is allowed to incorrectly classify some examples, thus paying a price in the objective function. The incorrect classification penalty is a box constraint  $C$  multiplied by the classification error, this SVM is called soft margin SVM.

$$\min \|\vec{w}\|^2 + C \sum_{i=1}^{|I'|} \varepsilon_i$$

(24)



s.t.

$$y_i(\vec{w}^T \vec{x}_i + b) \geq 1 - \varepsilon_i \quad \forall i = 1, \dots, |I'|$$

(25)

Its Wolfe Dual is:

$$\max \sum_{i=1}^{|I'|} \alpha_i - \frac{1}{2} \sum_{i=1}^{|I'|} \sum_{l=1}^{|I'|} \alpha_i \alpha_l y_i y_l \vec{x}_i^T \vec{x}_l$$

(26)

s.t.

$$\sum_{i=1}^{|I'|} \alpha_i y_i = 0$$

(27)

$$C \geq \alpha_i \geq 0 \quad \forall i = 1, \dots, |I'|$$

(28)

With the primal problem solution:

$$\vec{w} = \sum_{i=1}^{|I'|} \alpha_i y_i \vec{x}_i$$

(29)

$$b = y_l(1 - \varepsilon_l) - \sum_{i=1}^{|I'|} \alpha_i y_i \vec{x}_i^T \vec{x}_l \quad \forall l: \alpha_l \neq 0$$

(30)

This SVM classifier can be made non-linear by substituting the features' internal product  $\vec{x}_i^T \vec{x}_l$  with a non-linear kernel  $k(\vec{x}_i, \vec{x}_l)$ . This is equivalent to a non-linear expansion of the features space. What was non linearly separable in the original space might be linearly separable in the new one. The use of a kernel (called the kernel trick) prevents the computational costs associated with the calculation of the new features space.

The kernel used in this paper is the radial basis function:

$$k(\vec{x}_i, \vec{x}_l) = e^{-scale \cdot \|\vec{x}_i - \vec{x}_l\|^2}$$

(31)

The radial basis function is a well-established kernel, capable of achieving a significant performance while dealing with non-linear datasets.

The SVM classifier can be used in contexts with more than two classes by training a classifier for each pair of classes. When a new item  $i$  needs to be classified, it is fed to each SVM and the results are compared using an error-correcting output code model (ECOC) (Escalera *et al.* 2009). In a three-class model, the ECOC is:

$$ECOC = \begin{pmatrix} 1 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & -1 \end{pmatrix} \quad (32)$$

The item  $i$  is associated with the class minimizing:

$$\min \sum_{j=1}^J \sum_{l=1}^J \frac{\max(0, 1 - ECOC_{i,l} \cdot s_i)}{2} \quad (33)$$

where  $\vec{s}_i$  is a logical vector representing the class that item  $i$  is classified in. In a three-class model:

$$\vec{s}_i = (1,0,0) \quad \text{represents the first class} \quad (34)$$

$$\vec{s}_i = (0,1,0) \quad \text{represents the second class} \quad (35)$$

$$\vec{s}_i = (0,0,1) \quad \text{represents the third class} \quad (36)$$

#### 4.4.2 Deep neural network

A DNN is a multiclass non-linear classifier that passes the item features  $\vec{x}_i$  through a set of layers and outputs a vector  $\vec{s} \in R^K$ , where  $s_k$  (i.e. the  $k$ th component of  $\vec{s}$ ) is the probability that  $i$  belongs to a certain class  $C_{k,I}$ .

As shown in Figure 1, each layer is a vector connected to the next one through a matrix of weights  $W_l$  and a set of non-linear functions. In this structure, we consider  $\vec{x}_i$  and  $\vec{s}$  as the input and output layers and  $\vec{z}_l$  with  $l = 1, \dots, N_{hidden}$  as hidden layers. Alongside  $W_l$ , the

non-linear functions in layer  $l$  are obtained by a vector of biases  $\vec{b}_l$  not represented in Figure 1. Each element of bias  $b_{lj}$  enters one of the functions preceding layer  $l$ . The last function is linked to layer  $N_{hidden}$  by a matrix of weights  $W_{softmax}$  and a vector of biases  $\vec{b}_{softmax}$ .

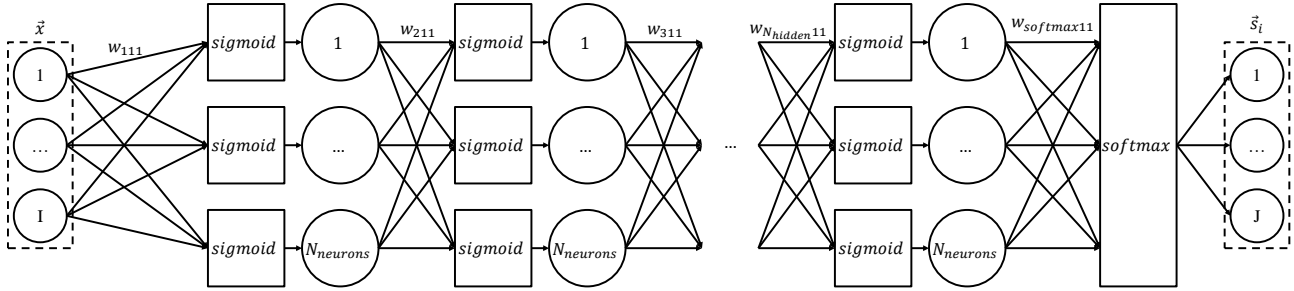


Figure 1. Deep neural network structure.

$\vec{z}_1$  is linked to  $\vec{x}_i$  by the equation:

$$\vec{z}_1 = \text{sigmoid}(W_1 \cdot \vec{x}_i + \vec{b}_1) \quad (37)$$

where the *sigmoid* function of  $W_1 \cdot \vec{x}_i + \vec{b}_1$  is:

$$\vec{z}_1 = \frac{\vec{1}}{\vec{1} + e^{-W_1 \cdot \vec{x}_i - \vec{b}_1}} \quad (38)$$

The same logic applies to each layer  $\vec{z}_l$  with  $l = 1, \dots, N_{hidden}$ .

$$\vec{z}_l = \frac{\vec{1}}{\vec{1} + e^{-W_{l-1} \cdot \vec{z}_{l-1} - \vec{b}_{l-1}}} \quad (39)$$

$\vec{s}_i$  is linked to  $\vec{z}_{N_{hidden}}$  by the *softmax* function:

$$s_{ik} = \frac{e^{\bar{w}_{softmaxk}^T \bar{z}_{N_{hidden}} + b_{softmaxk}}}{\sum_{j=1}^J e^{\bar{w}_{softmaxk}^T \bar{z}_{N_{hidden}} + b_{softmaxk}}} \quad (40)$$

From Equation 39, each layer is linearly combined using weights and biases to produce the next one. If only linear equations were applied, the final result would be a multivariate logistic regression, i.e. a linear classifier using  $\vec{x}_i$  as an input. This is due to the fact that a combination of linear functions is a linear function. In order to add non-linearity, the *sigmoid* function must be used after each linear combination. The *sigmoid* collapses each vector component in the range  $(-1,1)$ , making each  $\vec{z}_i$  a non-linear combination of the previous layer components. From Equation 40,  $\vec{s}_i$  is a multivariate logistic regression of  $\vec{z}_{N_{hidden}}$ , a linear classification of the new features as they emerge from the last hidden layer. The use of a linear classifier in the final layer is common to DNN and SVM with the radial basis function kernel; both algorithms expand the features space and use linear classifiers to obtain non-linear classifications in the original space.

The performance of DNN can be measured by a cross-entropy ( $cross_i$ ) function:

$$cross_i = -\vec{y}_i^T \cdot \log(\vec{s}_i) \quad (41)$$

The  $cross_i$  function is minus the log-likelihood of the multivariate logistic regression classifier used in the output layer, the performance of DNN can be optimized by minimizing the overall cross-entropy of the training set:

$$cross = \sum_{i=1}^{|I'|} cross_i \quad (42)$$

This optimization is carried out by modifying the weights and biases of each layer, including the output one. In order to know how to modify weights and biases, the gradient of  $cross$  is obtained. Using the multivariable chain rule it is possible to calculate the derivatives of  $cross$  over  $W_l$  and  $\vec{b}_l$  for the layer  $l$  from the gradient of  $cross$  over  $\vec{z}_{l+1}$ . This facilitates an efficient reuse of calculations with a considerable computational time saving. Once each derivative has been computed, a gradient descend algorithm is applied to

move the value of *cross* downwards, thus increasing the DNN performance on the training set. This procedure is called backpropagation.

In this paper, part of the training set (30%), as defined in Section 4.4, is used as a holdout. It is not directly fed to the backpropagation algorithm for the DNN training, but is kept aside to check the *cross* performance on it as an unseen dataset. This prevents overfitting since the backpropagation is stopped after reaching the lowest *cross* on the holdout set. In practice weights and biases of the best performing DNNs are memorized as they emerge and the backpropagation algorithm is stopped after a certain number of suboptimal results. In this paper, the optimization algorithm used is a scaled conjugated gradient descent backpropagation due to its effectiveness and not excessive computational complexity.

## 5. Experimental analysis

The procedure outlined in Section 4 is applied to two datasets generated in Section 5.1. The first dataset contains items with a small relative standard deviation in terms of the positive demands  $(\frac{E[Std(D_i^+)]}{E(D_i^+)})$ . This feature produces fewer intermittent time series, since the main source of variability relates to the appearance of positive demands rather than the quantities of the positive demands themselves. The second dataset contains items with higher values of  $(\frac{E[Std(D_i^+)]}{E(D_i^+)})$ , and the resulting time series are more intermittent since the positive demand quantities are less predictable.

Each time series is obtained using two probability distributions, i.e. a Bernoulli distribution generating the positive demand events and a normal distribution generating the quantities demanded when a positive demand takes place. When in the series generation, a positive demand assumes values lower than zero, as a result of the unrestricted normal distribution domain, this demand is approximated to zero in the first dataset, whereas it is recalculated in the second one.

In Section 5.2, both datasets are simulated and the classification methods introduced in Section 4.4 are applied leading to 12 experiments, in Section 5.3 the classification results are analysed.

### 5.1 Experimental setting

Two experimental datasets are generated, the first one composed of 28,350 time series and the second one containing 39,690 time series. Each time series is randomly generated on a time horizon of  $T = 40,000$  periods with a warm-up of  $n = 20,000$  periods.

The following features, uniformly distributed between different ranges, generate each time series:

- $E(p_i)$  between 0.2 and 0.7 with linear steps of 0.1.
- $E(D_i^+)$  between 100 and 500 with linear steps of 50.
- $\frac{E[Std(D_i^+)]}{E(D_i^+)}$  between 0.1 and 0.3 with linear steps of 0.05 for the first dataset and between 0.7 and 1 with linear steps of 0.05 for the second one.

Non statistical features:

- $h_i \cdot C_i$  between 2 and 14 with linear steps of 4.
- $RLT_i$  equal to 2.
- $\phi_i^{-1}(tCSL_i)$  between 1 and 4 with linear steps of 0.5.

The unitary ordering cost  $o_i$  is obtained defining the theoretical  $R_{th,i}$  equal to 3, 5 and 7 and applying Equation 50 in Appendix A.

## 5.2. Exhaustive simulation and classification

Given a dataset, each item time series is exhaustively simulated through the inventory system detailed in Sections 4.1 and 4.2. A warm-up of  $n = 20.000$  periods is used to optimize, at the single item level, the smoothing coefficients  $\alpha, \beta, \gamma$  and  $\delta$ , thus minimising the mean squared error between forecast and demand. The parameters  $\alpha, \beta$ , and  $\gamma$  are optimized continuously, while  $\delta$  is fixed at 0.25. Three values of  $R$  (equal to 3, 5, and 7 periods) are tested for each time series and the value leading to the minimum  $TRC_{i,R}$  is selected.

At the end of this procedure, each item has both a set of classification criteria defined in Section 4.3 and an experimentally optimal value of  $R$ , named  $R_{opt,i}$ . The objective is to measure the performance of the classification methods outlined in Section 4.4. Therefore, six classification experiments are conducted on each dataset. Each experiment involves a different combination of numbers of cross-validation training folds and classifiers. Since the folds are used for training purposes, the higher the number of folds, the lower the ratio of items used in each training set.

Dataset 1			
	Simulation 1	Simulation 2	Simulation 3
SVM	50 folds	20 folds	10 folds

DNN	50 folds	20 folds	10 folds
Dataset 2			
SVM	50 folds	20 folds	10 folds
DNN	50 folds	20 folds	10 folds

Table 1. Number of classification experiments.

Table 1 outlines all the experiments, for instance the 20 fold SVM test divides a dataset into 20 folds. In this case 5% of the dataset items (training set) are used to train a radial basis function SVM which is then used to classify the items outside the fold (95% of the dataset items, i.e. test set), the resulting classes are compared with the real test set classes to obtain performance measures. This procedure is repeated for each fold and, at the end, the single performance measures are averaged.

Two performance measures are obtained for each experiment:

- The average correct classification ratio.
- The average confusion matrix.

The correct classification ratio is the ratio of correctly classified items in the test set over the total number of items in the test set. The average correct classification ratio is the average of these classification ratios over the number of folds.

The confusion matrix is a  $K \times K$  matrix containing a ratio of the test set items in each cell. The cell  $i, j$  for instance contains the number of items that the simulation sorted in class  $i$  and the classifier classified in class  $j$ , divided by the total number of items in the test set. The trace of the confusion matrix equals the correct classification ratio. The confusion matrix belonging to different folds is averaged cell-wise, weighting for the number of items in each test set, to obtain the average confusion matrix.

The meta-parameters analysed in each training fold to train a SVM are:

- $C$  distributed between  $10^{-5}$  and  $10^5$  with geometrical steps of magnitude 10.
- $scale$  distributed between  $10^{-5}$  and  $10^5$  with geometrical steps of magnitude 10.

The meta-parameter analysed in each training fold to train a DNN are:

- $N_{hidden}$  distributed between 1 and 3 with linear steps of 1.
- $N_{neurons}$  distributed between 10 and 100 with linear steps of 10.

This meta-parameter optimization follows the procedure outlined in Section 4.4 and is carried out inside each training fold without using any test fold sample.

### 5.3. Results

Table 2 contains the average correct classification ratio for each experiment, Table 3 contains the average confusion matrixes. The sum of probabilities exceeding 1 result from the rounding process.

	Dataset 1			Dataset 2		
SVN	0.972	0.977	0.982	0.972	0.980	0.983
DNN	0.954	0.972	0.978	0.963	0.975	0.981

Table 2. Average correct classification ratio.

		Dataset 1											
		Simulation 1			Simulation 2			Simulation 3					
		Predicted			Predicted			Predicted					
		A	B	C	A	B	C	A	B	C			
SVN	True	A	0.538	0.006	0	True	0.539	0.006	0	True	0.540	0.004	0
		B	0.010	0.227	0.006		0.009	0.230	0.004		0.009	0.232	0.003
		C	0	0.006	0.207		0	0.004	0.208		0	0.002	0.210
DNN	True	A	0.528	0.014	0.003	True	0.537	0.007	0	True	0.539	0.006	0
		B	0.011	0.222	0.010		0.009	0.227	0.007		0.009	0.231	0.004
		C	0	0.009	0.203		0	0.005	0.207		0	0.003	0.209
		Dataset 2											
		Predicted			Predicted			Predicted					
		A	B	C	A	B	C	A	B	C			
SVN	True	A	0.670	0.008	0	True	0.673	0.005	0	True	0.674	0.005	0
		B	0.008	0.196	0.006		0.006	0.200	0.004		0.005	0.201	0.004
		C	0	0.006	0.106		0	0.005	0.107		0	0.004	0.108
DNN	True	A	0.667	0.011	0	True	0.671	0.007	0	True	0.673	0.005	0
		B	0.011	0.191	0.008		0.007	0.1972	0.005		0.005	0.200	0.005
		C	0	0.007	0.105		0	0.005	0.107		0	0.005	0.107



Table 3. Average confusion matrix.

Three phenomena can be assessed by analysing Table 2:

1. The average correct classification ratio increases as the percentage of items in the training set increases, these increments are small but are present in each combination of dataset and classification methodology.
2. The second dataset, which is the most intermittent, achieves a better performance than the first. This is true for each combination of classification methodology and training percentage.
3. The SVM outperforms the DNN in each combination of dataset and training percentage, the differences are small but always in favour of the SVM.

Figure 2 plots the results of Table 2 against the number of items in the average training set. It shows that the first and second phenomenon listed above are generated by a single cause: the average correct classification ratio scales with the number of items in the average training set. The differences between datasets alone are not obvious from the plot and the performance seems to scale more consistently with the absolute number of items in the average training set than with its relative one. Figure 2 shows how the performance gap between SVM and DNN depends on the number of items in the average training set, and the gap shrinks as the number of items increases. Despite this trend, SVM always outperforms DNN.

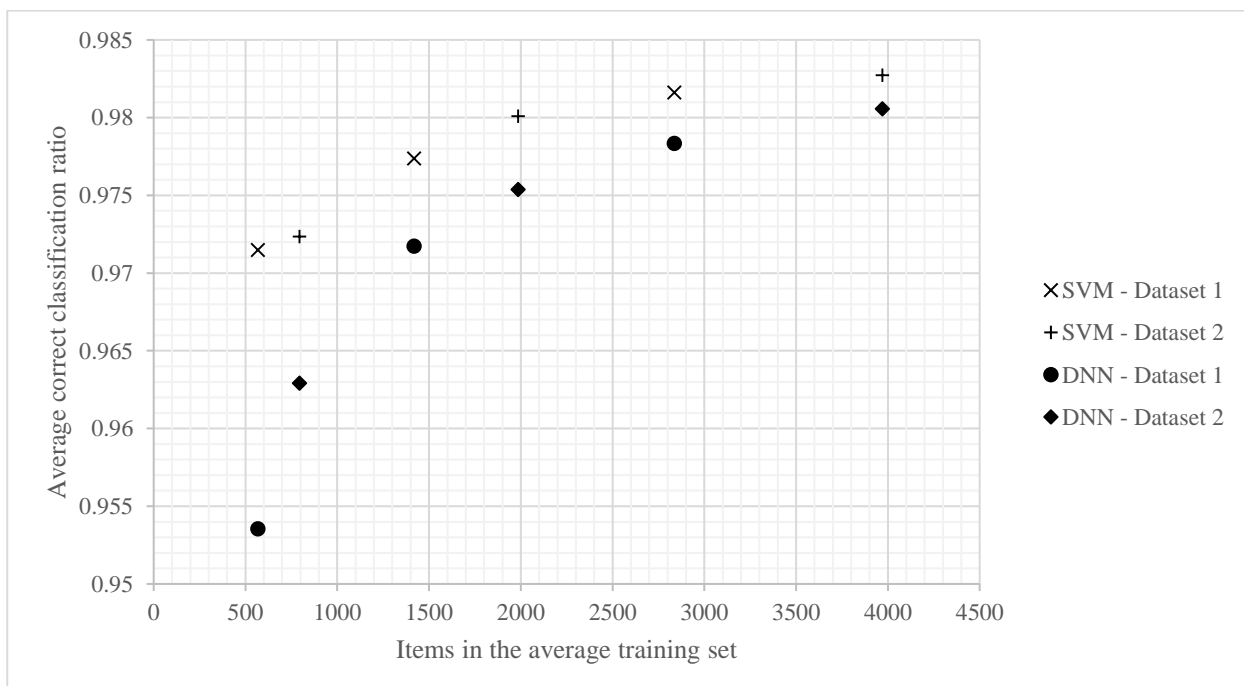


Figure 2. Average correct classification ratio over the number of items in the training set.

The average confusion matrices in Table 3 show that, despite the imbalance between classes, the classifiers do not overfit the most common class ( $R = 3$ ). This finding is consistent throughout the datasets, classifiers and training percentages. Since no preventive measures have been taken beforehand to prevent overfitting phenomena, the classifiers appear robust in dealing with unbalanced classes.

Dataset 1	0.725
Dataset 2	0.561

Table 4. Theoretical classification ratio.

Table 4 shows the theoretical correct classification ratio for both datasets. This is obtained by numerically solving Equation 51 in Appendix A, comparing the theoretical classes with those emerging from the simulations. The theoretical model performance in both datasets is worse than that achieved by the supervised classifiers and decreases as the data become more intermittent. This is exemplified by the 0.164 performance drop between the first and second datasets. The theoretical confusion matrixes in Table 5 show how as the items become more intermittent, they shift away from their theoretical classification towards lower values of the best  $R$ . The classification algorithms are able to adapt and catch this drift, with the subsequent increased class imbalance, while the theoretical classification does not.

Dataset 1					Dataset 2				
		Predicted					Predicted		
		A	B	C			A	B	C
True	A	0.333	0	0	True	A	0.333	0	0
	B	0.154	0.180	0		B	0.218	0.115	0
	C	0.058	0.064	0.212		C	0.127	0.094	0.112

Table 5. Theoretical confusion matrix.

## 6. Conclusions and further research

Managing a large number of items involves both multi-criteria inventory classification methods and inventory control theory. A common solution is to classify similar items first

and then define a unique inventory control policy for all the items belonging to a class. The tasks of classifying and finding appropriate control policies for the classes are generally kept separate, and as a result the original objective is often forgotten. When the items demonstrate intermittent consumption, e.g. spare-parts or multi-stage assembly with a high level of differentiation and customer heterogeneity, this issue is further exacerbated. In this paper, given a forecast-based periodic review inventory system, an exhaustive search on a sample of items is performed to obtain in the first stage, their best classification from an inventory perspective. This step is coupled with SVM and DNN algorithms to classify the out-of-sample items, using the aforementioned classification as an optimum reference along with the relevant set of criteria from an inventory control perspective.

The overall approach was validated through a large experimentation with satisfactory results, both classifiers lead to significant improvements in the classification accuracy in comparison with a theoretical approach.

We believe that our proposal is an interesting attempt to bridge the gap between inventory theory and multi-criteria inventory classification since they are strictly interrelated research fields. The decision to apply this methodology to intermittent demand is due to the highest analytical and operative criticality of this type of demand pattern, however the adopted inventory system is flexible and the overall framework is absolutely general and thus applicable to other inventory systems and demand patterns.

It is worth highlighting the importance of adopting an exhaustive search in order to fix a set of reference items as inputs for the classification procedures. This is in line with case-based reasoning for multi-criteria inventory classification, however the reference items are optimally classified here and not subjectively established by decision-makers. In other words, the classification of the in-sample items, which affects the overall classification, is free from human error. This is in line with Industry 4.0 guidelines, which underline the increasing need for ICT solutions within manufacturing. Our experimental research confirms the positive impact of the exhaustive search and highlights the importance of this phase.

Our proposal has some limitations that could be the subject for further research:

- This paper adopts a single-item forecast-based inventory system. In the case of items used in an assembly process, a multi-item inventory system would be more appropriate. The forecasts could be made for the end products and then propagated to the components according to the bills of materials.

- The analysed datasets were generated in order to better control the experimental procedure. However, real data might enrich the validation of this approach.
- Other inventory systems could be evaluated and compared with the one implemented here.

## Appendix A

This appendix outlines the calculation of a theoretically optimal value for  $R$  given the inventory system and replenishment policy under analysis. The results obtained act both as a benchmark for the machine learning algorithms presented and as a tool for the simulation setup.

Simulating a random interval of features could in fact lead to a strong imbalance between the classes, with one class encompassing the vast majority of the series, while in real world applications, different codes require different review intervals.

The total cost  $TRC_{i,R}$  can be minimized over  $R$  by rewriting it as:

$$TRC_{i,R} = h_i \cdot C_i \cdot E(NI_{i,R}) + o_i \cdot E(NO_{i,R}) \quad (43)$$

where  $E(NI_{i,R})$  is the expected value of the net inventory in a period and  $E(NO_{i,R})$  is the expected number of orders in a period. Both these variables depend on  $R$ .

$E(NI_{i,R})$  represents both a component related to the inventory cycle and a component related to the safety stock  $SS_i$ .

$$E(NI_{i,R}) = \frac{1}{2} O_i + SS_i \quad (44)$$

The cyclic component is maximized after each lead-time, when the ordered replenishment of quantity  $O_i$  arrives. After this replenishment, the cyclic inventory decreases and eventually drops to zero just in time for a new order to arrive, thus the average stock in a random period is half the maximum stock. This assumption is on average correct for a normally distributed non-intermittent demand and is applied here as an approximation. The safety stock component  $S$  is constant on average.

The order quantity is designed to cover  $RTL_i + R$  demand periods, thus:

$$O_i = E(D_i) \cdot p_i \cdot (RTL_i + R) \quad (45)$$

On the other hand, the safety stock is related to the safety factor and the single period variance:

$$SS_i = \Phi^{-1}(tCSL_i) \cdot \sqrt{MSE_i \cdot (RTL_i + R)} \quad (46)$$

where the single period demand  $MSE_i$  is equal to:

$$MSE_i = p_i \cdot (1 - p_i) \cdot E(D_i)^2 + p_i \cdot Var(D_i) \quad (47)$$

This leads to a  $TRC_{i,R}$  of:

$$TRC_{i,R} = \frac{1}{2} \cdot h_i \cdot C_i \cdot E(D_i) \cdot p_i \cdot (RTL_i + R) + h_i \cdot C_i \cdot \Phi^{-1}(tCSL_i) \cdot \sqrt{MSE_i \cdot (RTL_i + R)} + \frac{O_i}{RTL_i} \quad (48)$$

Its derivative over  $RTL_i$  minimizes the total relevant cost:

$$E(D_i) \cdot p_i - \frac{2 \cdot O_i}{h_i \cdot C_i \cdot R^2} + \frac{\Phi^{-1}(tCSL_i) \cdot MSE_i}{\sqrt{RTL_i + R}} = 0 \quad (49)$$

Substituting  $MSE_i$ ,  $O_i$  is given by:

$$O_i = \frac{1}{2} \cdot h_i \cdot C_i \cdot R^2 \cdot E(D_i) \cdot \left( p_i + \Phi^{-1}(tCSL_i) \cdot \sqrt{\frac{p_i \cdot \left( 1 - p_i + \left( \frac{Std(D_i)}{E(D_i)} \right)^2 \right)}{RTL_i + R}} \right) \quad (50)$$

Equation 50 is used in the dataset generation in Section 5.1 to calculate  $o_i$  given the other features.

The best  $R$  can be computed numerically, calculating the real roots of the polynomial:

$$R^5 \cdot D_i^2 p_i^2 + R^4 \cdot D_i^2 p_i \cdot \left( p_i R L T_i - \phi^{-1}(t C S L_i)^2 \cdot \left( (1 - p_i) + \left( \frac{Std(D_i)}{E(D_i)} \right)^2 \right) \right) - 4R^3 \cdot D_i p_i \cdot \frac{o_i}{h_i \cdot c_i} - 4R^2 \cdot D_i p_i \cdot \frac{R L T_i o_i}{h_i \cdot c_i} + 4R \left( \frac{o_i}{h_i \cdot c_i} \right)^2 + 4 \cdot R L T_i \left( \frac{o_i}{h_i \cdot c_i} \right)^2 = 0$$

(51)

Equation 51 is used in the results analysis in Section 5.3 to compare theoretical and simulated classes.

## References

- Altay, N. & Litteral, L.A., 2011. *Service parts management: demand forecasting and inventory control*. Springer-Verlag, London, UK.
- Babai, M.Z., Jemai, Z. & Dallery, Y., 2011. Analysis of order-up-to-level inventory systems with compound poisson demand. *European Journal of Operational Research*, 210 (3), 552-558.
- Babai, M.Z., Ladhari, T. & Lajili, I., 2015. On the inventory performance of multi-criteria classification methods: empirical investigation. *International Journal of Production Research*, 53 (1), 279-290.
- Babai, M.Z., Syntetos, A.A & Teunter, R., 2014. Intermittent demand forecasting: An empirical study on accuracy and the risk of obsolescence. *International Journal of Production Economics*, 157, 212-219.
- Bacchetti, A., Plebani, F., Saccani, N. & Syntetos, A.A., 2013. Empirically-driven hierarchical classification of stock keeping units. *International Journal of Production Economics*, 143 (2), 263-274.
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
- Boylan, J.E. & Syntetos, A.A., 2007. The accuracy of a modified Croston procedure. *International Journal of Production Economics*, 107 (2), 511-517.

- Cakir, O. & Canbolat, M., 2008. A web-based decision support system for multi-criteria inventory classification using fuzzy ahp methodology. *Expert Systems with Applications*, 35 (3), 1367-1378.
- Cavalieri, S., Garetti, M., Macchi, M. & Pinto, R., 2008. A decision-making framework for managing maintenance spare parts. *Production Planning & Control*, 19 (4), 379-396.
- Chen, J.-X., 2011. Peer-estimation for multiple criteria abc inventory classification. *Computers & Operations Research*, 38 (12), 1784-1791.
- Chen, Y., Kevi, W.L., Kilgour, D.M. & Hipel, K.W., 2008. A case-based distance model for multiple criteria ABC analysis. *Computers & Operations Research*, 35 (3), 776-796.
- Chopra, S. & Meindl, P., 2004. Supply chain management: strategy, planning, and operation. Prentice Hall, Inc.
- Conceição, S.V., Da Silva, G.L.C., Lu, D., Nunes, N.T.R & Pedrosa, G.C., 2015. A demand classification scheme for spare part inventory model subject to stochastic demand and lead time. *Production Planning & Control*, 26 (16), 1318-1331.
- Croston, J., 1972. Forecasting and stock control for intermittent demands. *Operational Research Quarterly*, 23 (3), 289-303.
- Dhakshayani, E. & Narayanan, S., 2014. Demand forecasting model for slow moving items. *International Journal of Logistics Systems and Management*, 19(2), 245-260.
- Doumpos, M., & Zopounidis, C., 2004. A multicriteria classification approach based on pairwise comparisons. *European Journal of Operational Research*, 158 (2), 378-389.
- Downing, M., Chipulu, M., Ojiako, U. & Kaparis, D., 2014. Advanced inventory planning and forecasting solutions: a case study of the UKTLCS Chinook maintenance programme. *Production Planning & Control*, 25 (1), 73-90.
- Driessen, M., Arts, J., Van Houtum, G.-J., Rustenburg, W. & Huisman, B., 2015. Maintenance spare parts planning and control: A framework for control and agenda for future research. *Production Planning & Control*, 26 (5), 1-20.
- Eaves, A., 2002. *Forecasting for the ordering and stock holding of consumable sparepart*. Lancaster University. Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
- Escalera, S., Pujol, O., Radeva, P. (2009). Separability of ternary codes for sparse designs of error-correcting output codes. *Pattern Recognition Letters*, 30(3), 285–297.

- Flores, B., Olson, D. & Dorai, V., 1992. Management of multicriteria inventory classification. *Mathematical and Computer Modelling*, 16 (12), 71-82.
- Gamberini, R., Lolli, F., Regattieri, A. & Rimini, B. 2014. Dynamic re-order policies for irregular and sporadic demand profiles. *Procedia Engineering*, 69, 1420-1429.
- Gamberini, R., Lolli, F., Rimini, B. & Sgarbossa, F., 2010. Forecasting of sporadic demand patterns with seasonality and trend components: an empirical comparison between Holt-Winters and (S)ARIMA methods. *Mathematical Problems in Engineering*, Article ID 579010.
- Gutierrez, R.S., Solis, A.O. & Mukhopadhyay, S., 2008. Lumpy demand forecasting using neural networks. *International Journal of Production Economics*, 111 (2), 409-420.
- Hadi-Vencheh, A., 2010. An improvement to multiple criteria abc inventory classification. *European Journal of Operational Research*, 201 (3), 962-965.
- Ishizaka, A., Lolli, F., Balugani, E., Cavallieri, R. & Gamberini, R., 2018. DEASort: Assigning items with data envelopment analysis in ABC classes. *International Journal of Production Economics*, 199, 7-15.
- Ishizaka, A. & Nemery, P., 2013. *Multi-criteria decision analysis: methods and software*. John Wiley&Sons Ltd, Chichester, UK.
- Kabir, G. & Hasin, M.A.A., 2013. Multi-criteria inventory classification through integration of fuzzy analytic hierarchy process and artificial neural network. *International Journal of Industrial and Systems Engineering*, 14 (1), 74-103.
- Kartal, H., Oztekin, A., Gunasekaran, A. & Cebi, F., 2016. An integrated decision analytic framework of machine learning with multi-criteria decision making for multi-attribute inventory classification. *Computers & Industrial Engineering*, 101, 599-613.
- Kourentzes, N., 2013. Intermittent demand forecasts with neural networks. *International Journal of Production Economics*, 143 (1), 198-206.
- Ladhari, T., Babai, M.Z. & Lajili, I., 2015. Multi-criteria inventory classification: new consensual procedures. *IMA Journal of Management Mathematics*.
- Lengu, D., Syntetos, A.A. & Babai, M.Z., 2014. Spare parts management: linking distributional assumptions to demand classification. *European Journal of Operational Research*, 235 (3), 624-635.
- Levén, E. & Segerstedt, A., 2004. Inventory control with a modified Croston procedure and Erlang distribution. *International Journal of Production Economics*, 90 (3), 361-367.



- Lolli, F., Gamberini, R., Regattieri, A., Balugani, E., Gatos, T. & Gucci, S., 2017. Single-hidden layer neural networks for forecasting intermittent demand. *International Journal of Production Economics*, 183, 116-128.
- Lolli, F., Gamberini, R., Regattieri, A. & Rimini, B., 2014a. Application of Tramo-seats automatic procedure for forecasting intermittent demand patterns. OPT-i, Kos Island, Greece.
- Lolli, F., Gamberini, R., Regattieri, A., Rimini, B., Grassi, A. & Belluti, P., 2011. Application of tramo-seats automatic procedure for forecasting sporadic and irregular demand patterns with seasonality. 13th Int. Conference on Harbor, Maritime and Multimodal Logistics Modeling and Simulation, HMS 2011, Held at the International Mediterranean and Latin American Modeling Multiconference, I3M 2011, 214-220.
- Lolli, F., Ishizaka, A. & Gamberini, R., 2014b. New ahp-based approaches for multi-criteria inventory classification. *International Journal of Production Economics*, 156 (0), 62-74.
- López-Soto, D., Yacout, S. & Angel-Bello, F., 2016. Root cause analysis of familiarity biases in classification of inventory items based on logical patterns recognition. *Computers & Industrial Engineering*, 93, 121-130.
- Mohammaditabar, D., Hassan Ghodsypour, S. & O'brien, C., 2012. Inventory control system design by integrating inventory classification and policy selection. *International Journal of Production Economics*, 140 (2), 655-659.
- Nenes, G., Panagiotidou, S. & Tagaras, G., 2010. Inventory management of multiple items with irregular demand: A case study. *European Journal of Operational Research*, 205 (2), 313-324.
- Ng, W., 2007. A simple classifier for multiple criteria abc analysis. *European Journal of Operational Research*, 177 (1), 344-353.
- Onwubolu, G. & Dube, B., 2006. Implementing an improved inventory control system in a small company: A case study. *Production Planning & Control*, 17 (1), 67-76.
- Partovi, F. & Anandarajan, M., 2002. Classifying inventory using an artificial neural network approach. *Computers & Industrial Engineering*, 41 (4), 389-404.
- Partovi, F. & Burton, J., 1993. Using the analytic hierarchy process for abc analysis. *International Journal of Operations & Production Management*, 13 (9), 29-44.
- Partovi, F. & Hopton, W., 1994. The analytic hierarchy as applied to two types of inventory problems. *Production and Inventory Management Journal*, 35 (1), 13-19.

- Porras, E. & Dekker, R., 2008. An inventory control system for spare parts at a refinery: an empirical comparison of different re-order point methods. *European Journal of Operational Research*, 184 (1), 101-132.
- Ramanathan, R., 2006. Abc inventory classification with multiple-criteria using weighted linear optimization. *Computers & Operations Research*, 33 (3), 695-700.
- Regattieri, A., Gamberi, M., Gamberini, R. & Manzini, R., 2005. Managing lumpy demand for aircraft spare parts. *Journal of Air Transport Management*, 11 (6), 426-431.
- Reid, R. A., 1987. The ABC method in hospital inventory management: A practical approach. *Production and Inventory Management Journal*, 28 (4), 67-70.
- Scala, N.M., Rajgopal, J. & Needy, K.L., 2014. Managing nuclear spare parts inventories: a data driven methodology. *IEEE Transactions On Engineering Management*, 61 (1), 28-37.
- Soylu, B. & Akyol, B., 2014. Multi-criteria inventory classification with reference items. *Computers & Industrial Engineering*, 69, 12-20.
- Syntetos, A.A., Babai, M.Z., Davies, J. & Stephenson, D., 2010. Forecasting and stock control: a study in a wholesaling context. *International Journal of Production Economics*, 127 (1), 103-111.
- Syntetos, A.A. & Boylan, J.E., 2001. On the bias of intermittent demand estimates. *International Journal of Production Economics*, 71 (1-3), 457-466.
- Syntetos, A.A. & Boylan, J.E., 2005. The accuracy of intermittent demand estimates. *International Journal of Production Economics*, 21 (2), 303-314.
- Syntetos, A.A. & Boylan, J.E., 2006. On the stock control performance of intermittent demand estimators. *International Journal of Production Economics*, 103 (1), 36-47.
- Syntetos, A.A., Boylan, J.E. & Croston, J., 2005. On the categorization of demand patterns. *Journal of the Operational Research Society*, 56 (5), 495-503.
- Syntetos, A.A., Keyes, M. & Babai, M.Z., 2009. Demand categorisation in a european spare parts logistics network. *International Journal of Operations & Production Management*, 29 (3), 292-316.
- Teunter, R., Babai, M.Z. & Syntetos, A.A., 2010a. Abc classification: Service levels and inventory costs. *Production and Operations Management*, 19 (3), 343-352.
- Teunter, R. & Sani, B., 2009. On the bias of Croston's forecasting method. *European Journal of Operational Research*, 194 (1), 177-183.

- Teunter, R., Syntetos, A.A. & Babai, M.Z., 2010b. Determining order-up-to levels under periodic review for compound binomial (intermittent) demand. *European Journal of Operational Research*, 203 (3), 619-624.
- Tratar, L.F., 2015. Forecasting method for noisy demand. *International Journal of Production Economics*, 161, 64-73.
- Vapnik, V. N., 1999. *The Nature of Statistical Learning Theory*. Berlin: Springer.
- Wuest, T., Weimer, D., Irgens, C. & Thober, K.-D., 2016. Machine learning in manufacturing: advantages, challenges, and applications. *Production & Manufacturing Research*, 4 (1), 23-45.
- Yu, M.-C., 2011. Multi-criteria abc analysis using artificial-intelligence-based classification techniques. *Expert Systems with Applications*, 38 (4), 3416-3421.
- Zhang, R.Q., Hopp, W.J. & Supatgiat, C., 2001. Spreadsheet implementable inventory control for a distribution center. *Journal of Heuristics*, 7 (2), 185-203.
- Zhou, P. & Fan, L., 2007. A note on multi-criteria ABC inventory classification using weighted linear optimization. *European Journal of Operational Research*, 182 (3), 1488-1491.
- Zotteri, G., 2000. The impact of distributions of uncertain lumpy demand on inventories. *Production Planning & Control*, 11 (1), 32-43.