Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 11 (2017) 1871 – 1881

Procedia MANUFACTURING

27th International Conference on Flexible Automation and Intelligent Manufacturing, FAIM2017, 27-30 June 2017, Modena, Italy

Decision trees for supervised multi-criteria inventory classification

Francesco Lolli[a]*, Alessio Ishizaka[b], Rita Gamberini[a], Elia Balugani[a], Bianca Rimini[a]

[a]*Department of Sciences and Methods for Engineering, University of Modena and Reggio Emilia, Via Amendola 2 – Padiglione Morselli, 42100 Reggio Emilia, Italy*

[b]*Centre of Operations Research and Logistics, Portsmouth Business School, University of Portsmouth, Richmond Building, Portland Street, Portsmouth PO1 3DE, United Kingdom*

Abstract

A multi-criteria inventory classification (MCIC) approach based on supervised classifiers (i.e. decision trees and random forests) is proposed, whose training is performed on a sample of items that has been previously classified by exhaustively simulating a predefined inventory control system. The goal is to classify automatically the whole set of items, in line with the fourth industrial revolution challenges of increased integration of ICT into production management. A case study referring to intermittent demand patterns has been used for validating our proposal, and a comparison with a recent unsupervised MCIC approach has shown promising results.

Keywords: multi-criteria inventory classification; decision trees; machine learning; inventory control; intermittent demand

1. Introduction and research background

Forecasting, inventory control and MCIC represent strictly interrelated fields of research. When dealing with a huge amount of items, firms are often interested in grouping them with the aim of simplifying their management. Each class is thus managed by means either of the same inventory control system or the same forecasting technique. With regards to classification approaches oriented to the inventory control, inventory managers often associate the

* Corresponding author. Tel.: +39 0522522635
E-mail address: francesco.lolli@unimore.it

doi:10.1016/j.promfg.2017.07.326

same target cycle service level (i.e. the probability of not incurring in a stock-out during a replenishment cycle) to the items belonging to the same cluster for the safety stock calculation or the same target fill rate (the percentage of demand not satisfied) or the same type of re-order policy (e.g. continuous or periodic review systems). However, current methodologies produce classification models based on criteria and threshold values that may not be adequate for the predefined set of re-order policies or cycle service levels associated with the obtained classes. This limitation has been recently underlined by [1]. In general, three ABC classes of ordered importance are defined and then inventory control policies are attached to them. These policies are negotiated with (or imposed by) the suppliers (e.g. deliveries can be done only on Fridays) and are often fixed before the classification of the items. Actually, the best service-cost assignment of items to one of the classes can be obtained by means of an exhaustive simulation search of the best policy at single item level. However, an exhaustive search is highly onerous, especially when the number of items is high. Therefore, companies prefer to use predefined criteria for the classification without running an exhaustive classification search. Historically, items are classified into three classes, ABC, according to a single criterion, which is often the usage value. The assignment is typically based on an arbitrary percentage, for example class A receives 20% of the items with the highest usage value, class B receives the next 30% and class C the remaining 50%. [2] recognised that multiple-criteria would give more precision in the definition of classes by augmenting the item characterisation. Therefore, several multi-criteria methods have been proposed for enriching the inventory classification: AHP and its extensions [3, 4, 5, 6, 7, 8]; TOPSIS [9]; the weighted linear optimisation [10, 11, 12, 13, 14, 15, 16, 17]; fuzzy logic [18]; and case-based reasoning [19, 20]. Artificial intelligence-based methods are applied as well to learn and replicate classical ABC classifications [21] or actual decision of inventory managers [22]. These last methods assume that a classification has already been produced in some way and considered correct. Once the classes have been established, a unique inventory control method (e.g. type of policy, cycle service level, fill rate and etc) is selected for all items of the same class [23]. However, there is no certitude that the criteria used for the classification are appropriate to guarantee the best performance of the inventory method. Indeed, it has been empirically shown that MCIC methods based on the annual dollar usage and the unit cost criteria have a low cost-service performance [1]. Moreover, it may be argued that different MCIC methods reach different classifications [24] when applied to the same dataset, and this trivially proves that these methods are not robust. [24] introduced new procedures for reaching the consensus among different MCIC approaches, but the relationship of the criteria with the inventory system is not even explored, and the class cardinalities are again pre-defined. [25] provided the first contribution devoted to constructing a criterion aimed to minimise the inventory cost. The calculation of this criterion is based on the probability that there is no stock out during the lead time. [26] derived another criterion which incorporates the cost of stock out. In both cases, the items are ranked and the classification is done arbitrarily with the first 20% assigned to group A, 30% to group B and 50% to group C.

As already underlined, selecting the optimal item classification and the best policy for each class can be done by an exhaustive search, but it is highly time-consuming for thousands of items. Meta-heuristics [27, 28, 29] or exact methods [30] have been proposed to solve this combinatorial problem through simplified assumptions without recurring to the exhaustive solution, but the classification still remains opaque.

In this paper, the classification rules will be generated through supervised classifiers well-established into machine learning field by starting from the exhaustive solution on a subset of items, on which the classifiers are trained. This approach bridges the gap between the theories of MCIC and inventory control when the exhaustive classification is impractical on the whole set of items, and a set of re-order policies is already defined and unchangeable. In particular, decision trees and random forests are compared as effective tools for overcoming the main concerns of MCIC, which are: i) the need for a set of predefined criteria that are not robustly linked with the inventory control system; ii) the predefined cardinalities of the generated classes defined a priori without any justification.

Among the machine learning techniques available for classification purposes, decision trees and random forests have been selected for theoretical simplicity and readability of the results. The connection between input features and obtained results in other methods, such as neural networks and non-linear SVM, is harder to analyse. Examining a decision tree it is possible to rank the splits and obtain a visual representation of what features are the most impacting in the classification process. Said features can be monitored by the management with the controlling unwanted shifts towards categories more expensive to manage. For the intermittent spare parts related to new products, particular attention on the most critical features can drive the design phase. Standardization procedures should be tuned not only to group as many components as possible but also to substitute hard to manage components with those belonging to the least expensive classes. A similar analysis can be carried on in a random forest scenario using a ranking averaged among the different trees composing the forest.

Furthermore, these issues are strongly exacerbated in the case of intermittent demand (e.g. spare parts consumption) due to its intrinsic management complexity. This paper specifically applies the supervised classifiers for items showing intermittence, but can also be used for general profiles of consumption. Machine learning tools have been already adopted for forecasting intermittent demand [31] but, to the best of our knowledge, this is the first contribution on classifiers for this kind of demand patterns. Machine learning goes towards the increased integration of ICT into production of industry 4.0.

The remainder of the paper is organised as follows. Section 2 summarises the proposed approach step by step. Section 3 validates the approach through a case study by following the steps previously explained. Section 4 compares the achieved classification with that provided by a DEA-based classification approach. Section 5 contains the conclusions as well as some suggestions for the further research agenda.

2. Methodology

In the sequel, the proposed multi-criteria inventory classification approach is defined step by step. Given a set of items and an inventory system composed by more re-order policies, the objective is to associate each item to one re-order policy. An exhaustive simulative approach is applied to a subset of items in order to achieve their optimal re-order policies, i.e. the classes to which they belong. The optimal classification for the non-simulated items is subsequently achieved by means of two supervised classifiers, i.e. the decision tree and the random forest.

Step 1: selection of the forecast system

Given an item with a known demand history, this has to be managed by means of an inventory policy. In stochastic systems, the input of the inventory policy is the forecast of the future demand, determined by applying a forecast system to the item demand history. In first that forecast system is selected.

Step 2: selection of the inventory control system

It is selected an inventory control system to be used as a reference. For each item, this inventory control system can be applied with different system parameters, which are item-specific and thus different choices lead to different performance. This paper does not specifically focus on the choice of the inventory control system, being the methodology non-dependent by such a choice. It is to note that the set of parameter values or possible combinations of parameter values determines the number of the classes. For instance, in an inventory system constrained by a target service level $tCSL$ (i.e. when backordering/lost sales costs are unavailable), supposing that $tCSL$ is the only system parameter, if three values of $tCSL$ are chosen (e.g. 99%, 95%, and 90%) then three classes of items have to be generated.

Step 3: exhaustive simulation search

An exhaustive search is performed by varying the adopted system parameter (e.g. the cycle service level, the review interval etc.) on a randomly chosen subset of items. Each item in the subset is simulated over its historical demand, applying the inventory control system defined in Step 2 on the basis of the forecasts obtained with the forecast system defined in Step 1. For each simulation a performance measure related to the system costs or service level (depending on the designer objectives) is calculated. This leads to the identification, for each analyzed item, of the system parameter value optimizing the performance measure. Each analyzed item is then put into the class characterized by the system parameter or combination of system parameters capable of optimizing the inventory control system performance for the items in said class.

Step 4: choice of classification criteria

A set of criteria able to characterize the items under analysis is selected. For example, in the field of intermittent demand, several criteria have been proposed in literature for classification purposes, especially for spare parts management. Some criteria refer to the demand pattern, like the coefficient of variation and the average demand interval [32], while others to the items features (e.g. criticality). Each criterion is a dimension the decision tree can split along, in order to classify the items into the classes defined before.

Step 5: training folds definition

The subset of items defined in Step 3 is divided into m folds. Such subdivision is generated casually while respecting the class frequencies in each folder. This last strategy is taken in order to add variability to the folds while coping with significant frequencies differences among classes, otherwise some folds might not contain all the classes or only few elements of the last represented classes might be selected.

Step 6: decision tree

A decision tree is trained on each fold, and each tree node can binary split the items on a single criterion among those presented in Step 4. The split is performed by following a predetermined splitting rule (e.g. Gini index). Each decision tree is tested against the items not belonging to its training fold, and the test is performed both for the fully-grown tree and for all its pruned versions. For each fold and pruning level a performance in terms of percentage of items correctly classified in the test phase is memorized. For each pruning level, its average performance among the folds is calculated and the pruning level leading to the best performance is selected. The items belonging to all the folds are then used to train a final tree, pruned up to the best pruning level.

Step 7: random forest

Instead of a decision tree, a Random Forest approach can be applied to the same folds. A set of Random Forests is trained on each fold and tested on the items not belonging to the fold at hand. A set of Random Forests is designed by varying the number of trees in the forests and the number of criteria randomly chosen for splitting the items at each node. Each random forest yields a classification performance in the test phase, thus the performance of the combination number of trees and number of criteria, in terms of percentage of items correctly classified, can be identified. Given a combination the average performance among all the folds is calculated and the combination yielding the best performance is selected. A final Random Forest in trained, using the items belonging to all the folds with the best combination of number of trees and number of criteria for splitting the items at each node.

Step 8: classification of the remaining items

At the end of Steps 6 and 7, a decision tree and a Random Forest with optimal parameters trained on all the simulated items is available. The items not used for training are those not simulated in Step 3; such items are classified by means of these classifiers. Each class is tied to a set of parameters, as defined in Step 2, and the classification leads to the identification of the best possible parameters for each item to be used in that item inventory control system.

3. Experimental validation

3.1. Experimental validation overview

The methodology proposed in Section 2 is applied to items characterized by intermittent demand after a brief outline of the experimental environment.

A firm producing make-to-stock electric resistances through a multi-stage assembly system has thousands of items delivered by its suppliers. Among these items, a high percentage (more than one thousand) shows weekly intermittent consumptions due to the high heterogeneity of the final customers and the large portfolio of final products. These items cannot be managed in a push system through the Material Requirement Planning (MRP) because their replenishment lead times are higher than the maximum allowed by the constraints of the assembly process and the time-to-market.

A representative sample of one hundred and four items is selected, and the demand history data have been collected over 150 weeks (3 years).

The main features of each item are summarised in the sequel:

- RLT_i is replenishment lead time of 2, 4 or 6 weeks.
- C_i is the unitary purchasing cost varying among the 104 items between 0.10 € and 9.08 €.
- h_i is the unitary holding charge fixed at 1% of C_i per week.
- o_i is the ordering cost fixed at 4.4 € per order for any item.

Table 1 reports the main descriptive statistics of the demand history over the 150 weeks:

- The minimum and maximum mean ($\bar{p}$) and variance (Var_p) of the demand size among all times series.
- The minimum and maximum mean ($\bar{D}$) and variance (Var_D) of the demand per period (both positive and null).
- The minimum and the maximum ratio of null demands ($\%Null$).

It is to notice the great heterogeneity of the items, which underlines the need for their classification and for the subsequent differentiation of their inventory system parameters.

Table 1. Descriptive statistics of the dataset.

Extreme values	$\hat{p}$	Var_p	$\hat{D}$	Var_D	$\%Null$
Min	0.297531	0.068025	0.063899	0.029109	0.12
Max	2555.409	3672179	2244.544	3922768	0.95

The experimental steps are explained in the following sections follow the theoretical ones reported in Section 2.

In order to measure the performance of decision trees and Random Forests in this scenario, all the items are exhaustively simulated and a leave-one-out approach is used. For each item, all the remaining items are used to generate the final decision tree (Step 6), and the final Random Forest (Step 7). The item not used in the training step is then classified with both classifiers (Step 8), and its predicted class is memorized. At the end of the experimental validation, a predicted class for each methodology is attached to each item. These classes can be compared with those defined by the simulation (Step 3), and performance measures can be calculated accordingly.

In order to assess the performance increase determined by the simulation step, that makes possible the use of supervised classifiers as decision trees and the Random Forests, the proposed methodology is compared with a unsupervised MCIC approach, that is not involving the simulation step.

3.2. Selection of the forecast system

As outlined in Section 3.1, the analyzed items present an intermittent consumption. Said historical demand pattern is managed in this experiment by applying two forecast systems specifically designed for intermittency:

- Croston's Method (CR), as designed by [33].
- Synthetos-Boylan Approximation (SBA), as designed by [34].

The forecast for period t provided by CR and SBA for the item i at the end of period $t-1$ is respectively given by:

$$FR_{i,t}^{CR} = \frac{\hat{Z}_{i,t}}{\hat{T}_{i,t}} \tag{1}$$

$$F_{i,t}^{SBA} = (1 - \frac{\alpha_i}{2}) \frac{\hat{Z}_{i,t}}{\hat{T}_{i,t}} \tag{2}$$

Where:

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

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

are the estimated demand and interval size respectively, 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 two consecutive transactions.

The exponential smoothing of the Mean Squared Error (MSE) of forecasts is used as estimator of the demand variance, in accordance with Syntetos et al. (2010), as follows:

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

where $\left(D_{i,t-1} - F_{i,t-1}\right)$ is the difference between the actual demand in $(t-1)$ and the corresponding forecast computed at the end of period $(t-2)$ with Equations (1) or (2), while α_i, β_i, γ_i and δ_i are different smoothing coefficients for each item i.

The historical demand is used to assess, for each item, which forecast system performs better and to define its smoothing coefficients. α_i, β_i, γ_i and δ_i can take two values: 0.1 and 0.2, because higher values are not recommended in the literature [35].

All the sixteen combinations are tested for both CR and SBA on a warm-up period of the first half of time series (i.e. 75 observations). $F_{i,t}$ is initialised with the averages of the first three observations per each time series. It is then chosen for each item the combination of forecast system and smoothing coefficients leading to the lowest MSE at the end of the warm-up period.

3.3. Selection of the inventory control system

A forecast-based periodic order-up-to level policy (R,S) is adopted for all the items.

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

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

Where:

- R_i is the review interval.
- $\phi_i(.)$ is the cumulative distribution function of the demand in $(RLT_i + R_i)$.
- $tCSL_i$ is the target cycle service level required.

The first term of the equation corresponds to the forecast demand for the period $(RLT_i + R_i)$ in a stationary mean model, with $F_{i,t}$ given either by Equation (1) or (2).

The second term is the safety stock for reaching a target cycle service level $tCSL_i$ by the safety factor $\phi_i^{-1}(tCSL_i)$, which represents the probability of not incurring in a stock-out during a replenishment cycle.

A pure cost-based model is not suitable due to the uncertainty of backorder costs, and therefore a service constrained model is preferred.

3.4. Exhaustive simulation search

In the model selected in Section 3.3, the review interval is the only system parameter to optimize by means of an exhaustive search, and the company managers are interested in renegotiating R_i with the suppliers for each item i. The review interval is tied to contractual agreements because it determines how often a replenishment order can be issued. An order is issued every time, in a review period, the inventory position falls under $S_{i,t}$.

In the current experimentation R_i can be set to 2, 4 or 6 weeks; these values are exhaustively tested with the already optimised smoothing coefficients by means of a discrete-event simulation working on weekly periods. The simulation operates from the first observation, but results are collected only on last 75 observations (collection period), i.e. half the time series.

For each observation in the collection period relevant measures are memorized:

- $NI_{i,t}$ is the net inventory in stock during the period t.
- $NO_{i,t}$ is a flag with value 1 if an order is issued during the period t and 0 otherwise.

At the end of the simulation a cost-oriented performance measure is computed:

$$TRC_i = h_i \cdot C_i \cdot \frac{\sum_{t=76}^{150} NI_{i,t}}{75} + o_i \cdot \frac{\sum_{t=76}^{150} NO_{i,t}}{75} \tag{7}$$

The first term is the average weekly holding cost for keeping the net inventory in stock while the second term is the average weekly ordering cost for emitting orders.

All the other simulation parameters not defined in Section 3.1 are exogenously assigned:

- $tCSL_i$ relates to the criticality level of the item i, depending on several aspects (e.g. technological and economical). In this case the managers assign a $tCSL_i$ equal to 90% to all the items.
- $\phi_i(.)$ is a standardised normal distribution, and the safety factor $\phi_i^{-1}(tCSL_i)$ is thus also known.

3.5. Choice of classification criteria

The criteria chosen for classification are quantitative and impact directly the inventory control system functionality:

- The unitary purchasing cost (C_i); given the fixed h_i, only C_i affects the total inventory cost.
- The replenishment lead time (RLT_i); this feature is specific for each item and depends on its supplier. The higher RLT_i, the more critical the item.
- The mean positive demand ($\hat{p}_i$); for the Bernoulli process characterising the intermittent demand, when an event occurs, the positive demand follows a stochastic variable with mean $\hat{p}_i$. Hence, this criterion refers to the demand pattern.
- The number of null demands during the simulation period ($Null_i$); this criterion is related to the probability that an event occurs, and therefore depends on the demand pattern as well.

Among these criteria p_i and $Null_i$ are calculated only with the last 75 observations of the simulation. The purpose is to avoid distorted relationships of the re-order policy with p_i and $Null_i$. In fact these values may significantly change over time, while the inventory policy has to overcome the warm-up period.

Other criteria like ordering cost and holding charge are not taken into account because they are the same for all the items, and thus they do not affect the classification.

3.6. Training folds definition

As defined in Section 3.1 each item is analyzed with a leave-one-out approach. Given a reference item, the remaining items are divided into three balanced folds of 34, 35 and 35 items respectively in order to optimize and train both a decision tree and a Random Forest.

As defined in Step 5, the frequency of each class is approximately respected in each fold. The original dataset is composed by 104 items, 74 of which belong to class 2, 20 to class 4 and 10 to class 6. The characteristics of each fold are outlined in table 2.

Table 2. Number of items in each fold divided by class.

Fold	Number of items in each class			
	2	**4**	**6**	**Total**
Fold 1	24	7	3	34
Fold 2	25	7	3	35
Fold 3	25	6	4	35

3.7. Decision tree

Given a fold as defined in Section 3.6, its decision tree is grown with the following parameters:

- The criterion chosen for splitting at each node is the Gini Index.
- The minimum number of items in a branching node is 2.
- The pruning sequence follows the classification errors.

The Gini index for splitting and the classification errors for pruning are chosen as common methodologies for the decision trees. The number of items in branching nodes is kept low in order to let the trees grow unconstrained, and the reduction of the trees size is left to the pruning phase optimized among the folders.

3.8. Random Forest

Given a fold as defined in Section 3.6, its Random Forest is grown with the following parameters:

- The criterion chosen for splitting at each node is the Gini Index.
- The minimum number of items in a branching node is 2.
- No pruning is performed on the forest trees.

The trees belonging to a forest are left unpruned, and the overfitting tendencies of the individual trees is countered by the number of trees balancing each other biases and by the random choice of characteristics at each split. Both these parameters are optimized among the folders (Step 7).

3.9. Classification of the remaining items

Each decision tree and Random Forest classifies the item not used for its training, leading to the confusion matrix reported in Table 3.

Table 3. Supervised methods confusion matrices.

Classification methodology		Predicted		
		2	**4**	**6**
Decision tree	**2**	66	6	2
	4	9	10	1
	6	2	2	6
Random Forest	**2**	68	4	2
	4	12	8	0
	6	3	2	5

4. Comparison with an unsupervised MCIC approach

These results reported in Table 3 are compared with those obtained with the LBL-model proposed by [24], and presented in Table 4.

Table 4. Unsupervised method confusion matrix.

Classification methodology		Predicted		
		2	**4**	**6**
LBL	**2**	61	9	4
	4	12	6	2
	6	1	5	4

The LBL-model is unsupervised, and its class cardinalities are to be established a priori. In order to compare its best performance with the results achieved by the decision trees and the Random Forests, the optimum cardinalities reached through the exhaustive search are fixed, i.e. 74-20-10 respectively for the 2-4-6 class.

The LBL-model is a DEA-based optimization approach that considers both a good and a bad index for each item and can be solved without an optimization solver. It needs a composite index to be set, defined in the current case by assigning equal weights to the good and bad indexes, and a ranking of the classification criteria. In order to obtain the LBL-model best performance, all the twenty four possible rankings of criteria (i.e. 4!) are tested and only the one leading to the best classification is considered.

The degree of agreement between different classification methods is established by means of the number of misclassified items from one class to another. A single misclassification is verified if a 2-class and a 4-class item is respectively classified into class 4 and class 6, and vice versa. A double misclassification arises when an item of class 2 is classified into class 6, and vice versa. The total number of misclassifications represents an accuracy measure for the methods under comparison (see Table 5).

Table 5. Misclassification performance comparison.

	Decision tree	Random Forest	LBL
Misclassifications	26	28	38
Misclassifications percentage compared to LBL	68%	74%	100%
Items incorrectly classified	22	23	33
Percentage of items incorrectly classified	21%	22%	32%

The number of misclassifications reached by decision trees and Random Forests is lower than those achieved by the best possible case of LBL model usage, obtaining respectively 32% and 26% less misclassifications. These findings are case sensitive, but show promising results for decision trees and machine learning techniques adoption into MCIC field. Table 5 presents also the same results from a different perspective showing for each methodology the raw percentage of items uncorreclty classified, without the misclassification weighting.

Both perspectives show similar results for the decision tree and the random forest methodologies, while the LBL technique presents significantly worse performance. This comparison suggests that the implemented machine learning methodologies impact mostly the items misclassified by the LBL technique with a re-classification capable of finding their correct place, the effect is not limited to a re-classification of said items in slightly less incorrect classes.

5. Conclusions and further research agenda

When managing a large number of items, MCIC is widely used to classify similar items and associate to each class a specific inventory control model. Different cycle service levels, typologies of re-order policies as well as lengths of review interval may be used for differentiating the inventory control system among the generated classes. However, the item classification still remains opaque if not jointly optimised with the inventory system. If MCIC is tackled separately, the classification criteria and the cardinalities of the classes have to be established a priori without any robust assessment from an inventory cost perspective.

In this paper, decision trees and Random Forests are adopted to joint optimize MCIC and inventory systems. To generate the needed training set, an exhaustive simulative approach is performed on a sample of items while the trained decision tree and Random Forest indicate for each non-simulated item the class to which it belongs.

A real case study referring to a firm producing electric resistances is used for validating this proposal on a set of 104 items with intermittent demand. The exhaustive classification adopts a well-known periodic inventory control system [35] where the classes are defined on the basis of the length of the review intervals. Four classification criteria are then taken into account (the replenishment lead time, the unitary purchasing cost, the number of null demands and the mean positive demand), and the most accurate tree and Random Forest is searched by a tenfold approach. The comparison with a recent unsupervised MCIC method [24] confirms the better performance of the proposed methodology.

It is to be noted that the performance of the compared unsupervised model is measured at its peak while the decision tree and Random Forest performance refer to standard working conditions. The correct class cardinalities are provided to the LBL model, but not to the decision tree or the Random Forest, being unknown in real settings. The LBL criteria ranking is also chosen in order to optimize the LBL performance according with the simulation results obtained for all the items. Actually, such optimization is unfeasible in real settings just because the exhaustive simulation is impracticable, which makes it necessary to adopt unsupervised classification methods.

The experimental investigation on other inventory control systems for intermittent demand, e.g. the binomial approach [36], as well as other machine learning tools, e.g. neural networks and support vector machines, are considered as a part of the further research agenda.

References

[1] M.Z. Babai, T. Ladhari, I. Lajili. On the inventory performance of multi-criteria classification methods: empirical investigation. *International Journal of Production Research* 2015;53(1):279-290.

[2] B. Flores, C. Whybark. Multiple criteria ABC analysis. *International Journal of Operations & Production Management* 1986;6(3):38-46.

[3] B. Flores, D. Olson, V. Dorai. Management of multicriteria inventory classification. *Mathematical and Computer Modelling* 1992;16(12):71-82.

[4] F. Partovi, J. Burton. Using the analytic hierarchy process for ABC analysis. *International Journal of Operations & Production Management* 1993;13(9):29-44.

[5] F. Partovi, W. Hopton. The analytic hierarchy as applied to two types of inventory problems. *Production and Inventory Management Journal* 1994;35(1):13-19.

[6] O. Cakir, M. Canbolat. A web-based decision support system for multi-criteria inventory classification using fuzzy AHP methodology. *Expert Systems with Applications* 2008;35(3):1367-1378.

[7] G. Kabir, M. Hasin. Multiple criteria inventory classification using fuzzy analytic hierarchy process. *International Journal of Industrial Engineering Computations* 2012;3(2):123-132.

[8] F. Lolli, , A. Ishizaka, R. Gamberini. New AHP-based approaches for multi-criteria inventory classification. *International Journal of Production Economics* 2014;156:62-74.

[9] A. Bhattacharya, B. Sarkar, S. Mukherjee. Distance-based consensus method for ABC analysis. *International Journal of Production Research* 2007;45(15):3405-3420.
[10] R. Ramanathan. ABC inventory classification with multiple-criteria using weighted linear optimization. *Computers & Operations Research* 2006;33(3):695-700.
[11] P. Zhou, L. Fan. A note on multi-criteria ABC inventory classification using weighted linear optimization. *European Journal of Operational Research* 2007;182(3):1488-1491.
[12] W. Ng. A simple classifier for multiple criteria ABC analysis. *European Journal of Operational Research* 2007;177(1):344-353.
[13] A. Hadi-Vencheh. An improvement to multiple criteria ABC inventory classification. *European Journal of Operational Research* 2010;201(3):962-965.
[14] J.-X Chen. Peer-estimation for multiple criteria ABC inventory classification. *Computers & Operations Research* 2011;38(12):1784-1791.
[15] S. Torabi, S. Hatefi, B. Saleck Pay. ABC inventory classification in the presence of both quantitative and qualitative criteria. *Computers & Industrial Engineering* 2012;63(2):530-537.
[16] J. Park, H. Bae, J. Bae. Cross-evaluation-based weighted linear optimization for multi-criteria ABC inventory classification. *Computers & Industrial Engineering* 2014;76(0):40-48.
[17] S. Hatefi, S. Torabi. A common weight linear optimization approach for multicriteria ABC inventory classification. *Advances in Decision Sciences* 2015;2015:1-11.
[18] J. Rezaei, S. Dowlatshahi. A rule-based multi-criteria approach to inventory classification. *International Journal of Production Research* 2010;48(23):7107-7126.
[19] Y. Chen, W.L. Kevi, D. Marc Kilgour, K.W. Hipel. A case-based distance model for multiple criteria ABC analysis. *Computers & Operations Research* 2008;35(3):776-796.
[20] B. Soylu, B. Akyol. Multi-criteria inventory classification with reference items. *Computers & Industrial Engineering* 2014;69:12-20.
[21] M.-C. Yu. Multi-criteria ABC analysis using artificial-intelligence-based classification techniques. *Expert Systems with Applications* 2011;38(4):3416-3421.
[22] F. Partovi, M. Anandarajan. Classifying inventory using an artificial neural network approach. *Computers & Industrial Engineering* 2002;41(4):389-404.
[23] R. Ernst, M. Cohen. Operations related groups (ORGs): a clustering procedure for production/inventory systems. *Journal of Operations Management* 1990;9(4):574-598.
[24] T. Ladhari, M.Z. Babai, I. Lajili. Multi-criteria inventory classification: new consensual procedures. *IMA Journal of Management Mathematics* 2016;27(2):335-351.
[25] R. Zhang, W. Hopp, C. Supatgiat. Spreadsheet implementable inventory control for a distribution center. *Journal of Heuristics* 2001;7(2):185-203.
[26] R.H. Teunter, M.Z. Babai, A.A. Syntetos. ABC classification: service levels and inventory costs. *Production and Operations Management* 2010;19(3):343-352.
[27] Tsai, C.-Y., and S.-W. Yeh. 2008. "A multiple objective particle swarm optimization approach for inventory classification." International Journal of Production Economics 114 (2): 656-666.
[28] D. Mohammaditabar, S. Hassan Ghodsypour, C. O'Brien. Inventory control system design by integrating inventory classification and policy selection. *International Journal of Production Economics* 2012;140(2):655-659.
[29] S.-T. Wang, M.-H. Li. An analysis of the optimal multiobjective inventory clustering decision with small quantity and great variety inventory by applying a DPSO. *The Scientific World Journal* 2014;2014 Article ID 805879:15 pages.
[30] M. Millstein, L. Yang, H. Li. Optimizing ABC inventory grouping decisions. *International Journal of Production Economics* 2014;148:71-80.
[31] F. Lolli, R. Gamberini, A. Regattieri, E. Balugani, T. Gatos, S. Gucci. Single-hidden layer neural networks for forecasting intermittent demand. *International Journal of Production Economics* 2017;183:116-128.
[32] A.A. Syntetos, J.E. Boylan, J.D. Croston. On the categorization of demand patterns. *Journal of the Operational Research Society* 2005;56(5):495-503.
[33] J.D. Croston. Forecasting and stock control for intermittent demands. *Operational Research Quarterly* 1972;23(3):289-303.
[34] A.A. Syntetos, J.E. Boylan. The accuracy of intermittent demand estimates. *International Journal of Production Economics* 2005;21(2):303-314.
[35] A.A. Syntetos, M.Z. Babai, J. Davies, D. Stephenson. Forecasting and stock control: a study in a wholesaling context. *International Journal of Production Economics* 2010;127(1):103-111.
[36] R.H. Teunter, A.A. Syntetos, M.Z. Babai. Determining order-up-to levels under periodic review for compound binomial (intermittent) demand. European Journal of Operational Research 2010;203(3):619-624.