Multivariate multi-output LSTM for time series forecasting with intermittent demand patterns

Gabellini M.*, Calabrese F.*, Regattieri A.*, Ferrari E*.

**Department of Industrial Engineering,*
University of Bologna, Viale del Risorgimento 4, 40136 – Bologna – Italy
(matteo.gabellini5@unibo.it, francesca.calabrese9@unibo.it, alberto.regattieri@unibo.it, emilio.ferrari@unibo.it)

Abstract: Compared to other data sources, demand time series are easily available in the industrial context. Providing accurate forecasts based on this kind of data for components with intermittent demand patterns is thus fundamental in many applications. Examples include optimizing inventory levels and the selection of the best tradeoff between holding and stockout costs in the spare parts management context. Recently, deep learning and machine learning models have been proposed to address this need. Compared to the more traditional ones, these methods better model nonlinear patterns in data. On the other hand, they require more effort in the parameter tuning phase, making it difficult to optimize at an item level in real life. In addition, relying only on single time series has some limitations. This study proposes a new approach based on a multivariate multi-output long short-term memory neural network to reduce time spent tuning and capturing interactions between different items' consumption data. The model is tested on a real spare parts dataset of a mechanical company. Croston's method and its variations, together with a multi-layer perceptron neural network, are used to compare the results.

Keywords: Intermittent demand, LSTM neural network, multivariate time series forecasting, spare parts.

I. INTRODUCTION

Spare parts management has increasingly acquired relevance during the last decades, moving from the idea of a "necessary evil" to a source of potential profits. Empirical evidence has shown that spare parts margins are significantly higher than product margins [4]. Furthermore, managerial literature stresses the role of the spare parts business as a long-term revenue, a profit source, a lever for differentiation, and a source of knowledge on products and customers [8],[25].

Spare parts management involves many related areas, such as inventory control and spare parts forecasting [21]. In particular, accurate demand forecasting is crucial to inventory control [16] and requires specific attention in this field due to some peculiar characteristics. The first element of attention is the presence of a large portion of zero values and a significant variability among non-zero values in an item's consumption history (usually referred to as intermittent and lumpy demand). In addition, the ever-increasing number of parts managed and the high responsiveness required due to downtime costs by customers make forecasting in this field a complex matter [1]. Some practical implications of the previous problems can be found in [14] and [23]. In the former, authors analyzing a case study in the aerospace sector have to deal with about 30'000 items and state that a complete forecast for the demands of all components needs about 330h. In the latter, it is estimated that a two-hour delay for an aircraft can cost an airline up to 150,000$.

As a result of the complexity and relevance of accurate forecasts in this field, researchers have tested different methodologies in time. Recently, two main directions have been delineated: methods relying on time series data and methods that leverage contextual data often referred to as installed base information, such as maintenance schedules, equipment age, or operating conditions. Time series forecasting methods, in turn, can be divided into parametric and non-parametric ones. While demand is assumed to follow a hypothesized probability distribution in the former, the latter derives the lead time demand distribution from the data [24]. Within this last category, an increasingly investigated field concerns the use of neural networks (NN). NN is considered a versatile tool that can capture nonlinear patterns in data, such as intermittence and lumpiness, better than most time series methods [19], [2].

This paper extends the research on intermittent and lumpy time series forecasting with neural network and propose a new approach based on a multivariate multioutput long short-term memory neural network (MMO_LSTM). The model has been benchmarked against different Croston variations and the NN proposed in [22]. This new approach aims to make the most of time-series data by looking for hidden patterns between different items. In addition, the time for tuning aims to be reduced as the model is able to deal with prediction on multiple components simultaneously.

The rest of the paper has been organized as follows. Section 2 reviews previous neural network works in spare parts forecasting based on time series data and summarize the main research gap. Section 3 gives an overview on the theoretical background of methods implemented in the study. Original Croston and its variations are presented as they are object of the comparison in 3.A. The focus is then posed on the main building block on which the proposed model is based in 3.B. In 3.C, the new proposed model is illustrated. Section 4 discuss the experimental setup and the case study. In the end, Section 5 and 6 present the results and conclusions, respectively.

II. RESEARCH BACKGROUND

A first paper using NN in the context of intermittent demand is [15], the authors proposed a Multi-Layer Perceptron architecture (MLP). One input layer (IL), one output layer (OL), and one hidden layer (HL) with three neurons (units) have been used. Twenty-four components have been studied with an average demand interval (ADI), computed as reported in [27], ranging from 3,38 to 5,44. NN has been trained based on 624 daily historical consumption observations for each product. In contrast, the following 343 observations have been used to evaluate its prediction performance. Historical data has been organized to provide the model, for each of the time step t, the value of the demand at the end of the immediately preceding period (D_{t-1}) and the number of periods separating the last two non zero demand transactions as of the end of the immediately preceding period (INZ_{t-1}). Based on these values, the back-propagation (BP) algorithm has been chosen as an optimization algorithm (OA) with a learning rate (LR) value of 0,1 to predict the value of demand transaction for the one-step ahead period (D_{t+1}). NN reports a mean absolute percentage error (MAPE) as computed in [12], ranging from 102% to 153%.

Another MLP has been proposed by [22]. In this study, the same number of items, historical observations' length, number of layers and neurons, learning rate, and momentum factor of the previous work have been used on a dataset with an ADI ranging from 2,63 to 3,28 to predict D_{t+1}. In contrast to the previous work, the partition between training and testing has been set to 80-20% and the number of consecutive periods with no demand transaction immediately preceding the target period ($ZCUM_{t-1}$) has been used in combination with D_{t-1} to forecast the value of D_{t+1}. Values of MAPE ranging from 95% to 145% have been founded.

In [18], two different MLP configurations have been studied on a monthly consumption dataset of 1000 items. Both NNs have been trained on 36 observations, while the following 100 have been used for validation. For each time step, non-zero demand value (NZD) and inter demand intervals (IDI) have been provided as input considering different lags ranging from one time step back (t-1) to 3 times step back (t-3), and 1 to 3 neurons have been tested in the hidden layer. In the first model, called NN-DUAL, the prediction concerned the value of IDI and NZD at t+1, while the second model, called NN-RATE, directly provided the demand rate (DRATE) at t+1. The Levenberg Marquardt (LM) optimization algorithm has been used for both models during 1000 epochs.

A different architecture based on Recurrent Neural Network (RNN) has been proposed by [3], 30 items have been considered, and 55 monthly historical observations have been used in the training phase. In contrast, 12 observations have been reserved for the validation phase. One input layer, one output layer, one context layer, and one hidden layer have been adopted, and a number of neurons ranging from 1 to 15 have been tested in the hidden layer to find the best architecture. In addition to D_{t-1}, INZ_{t-1}, and $ZCUM_{t-1}$, five more variables have been extracted from data and used as predictors for establishing the value of D_{t+1}: the number of consecutive periods with demand transactions immediately preceding the target period ($NZCUM_{t-1}$), the number of periods between the target period and first non-zero demand, immediately preceding the target period ($IFNZ_{t-1}$), the number of periods between the target period and first zero demand, immediately preceding target period (IFZ_{t-1}), the mean of demand for six periods immediately preceding the target period ($DMEAN_{t-6}$) and the maximum demand among six periods, preceding the target period ($DMAX_{t-6}$). BP with a learning rate of 0.01 has been used as an optimization algorithm, and values of MAPE ranging from 51% to 165% have been estimated across all items. Still, no indication about the ADI value of the dataset has been provided.

A different contribution has been offered by [19]. Here, three different architectures, respectively, a feed-forward (FF), a time delay (TD), and a recurrent neural network (RNN), have been compared on 24 items in the automotive sector. In the study, 61 to 76 weekly historical observations have been provided to forecast the future demand consumption for 1, 3 and 5 step ahead ($D_{t+1;t+3;t+5}$) in a dataset with ADI ranging from 1,18 to 3,78. Recognizing the complexity of optimizing networks at an item level, due to the effort required for parameter tuning, the authors suggest using an Extreme Learning (EL) mechanism to simplify the procedure as the only parameters that need to be adjusted are the number of hidden neurons. A linear relationship between ADI and MAPE has been found, with a latter value ranging from about 50% to about 120%. A summary is presented in Appendix A.

Overall, the scientific literature on neural networks for intermittent and lumpy time series forecasting suggests some uncovered opportunities for further investigation. Except for the work [19], little attention has been posed to neural network implementations able to reduce the great amount of time that they require in the tuning phase. Clearly, a relevant problem if looking at the ever-increasing number of spare parts that need to be managed. In addition, previous models based their prediction only on a single item's consumption history. However, forecasting based only on single time series has some limitations [6],[5],[10]. Considering the interactions between items and that they usually share some functional and technical similarities, some hidden patterns could be found by looking at multiple consumption time series simultaneously. On the one hand, the main novelty of this contribution is thus to propose a neural network model able to reduce time spent in tuning. On the other hand, the proposed data modelling phase aims to make the model investigate hidden patterns from the analysis of the interaction of multiple items' consumption time series.

III. METHODS

A. *Croston*

Croston method [9] is one of the standard benchmarks for spare parts demand forecasting. He proposed an approach that made a separate forecast for the demand-interval (P_t) and the demand size (Z_t). The forecast for demand per period (D_t) is then calculated as the ratio of the forecast for demand size and demand interval. The equations of Croston's method are the following:

$$Z_{t+1} = \alpha z_t + (1 - \alpha) Z_{t-1} \quad (1)$$

$$P_{t+1} = \beta p_t + (1 - \beta) P_{t-1} \quad (2)$$

$$D_{t+1} = Z_t / P_t \quad (3)$$

Where p_t is the actual value of time between consecutive non-zero transactions at the instant t and z_t is the real value of the last non-zero demand at the end of the review period t. At the same time, α and β are smoothing parameters, and they are project choices. Over the years, many scholars have proposed modifications of Croston, such as those of [27] (SBA) and of [26] (SBJ). These modification results in correcting the value of D_{t+1} computed in (3) with different factors as follow:

$$D_{t+1} = Z_t / P_t * (1 - \beta/2) \quad (SBA)$$

$$D_{t+1} = Z_t / P_t * (1 - (\beta/(2 - \beta)) \quad (SBJ)$$

B. *MLP and LSTM neural network*

An extensive NN review can be found in [28]. As the authors state, one of the most widely used types of NN is Multi-layer perceptrons (MLP). MLP architecture can be seen as the sum of three different macro-layers, each with a specific number of so-called "neurons". The network architecture is also characterized by interconnections of layers that determine its behaviour. MLP and NN generally require a former training phase to learn hidden patterns inside data. In this phase, an historical training dataset is provided. As this dataset contains both the historical values of variables and the historical observations generated in the face of those variables, the model can learn the nonlinear relationship between input and output data. Technically speaking, the training phase consists in finding the right value of the variables W and b in the following formula:

$$Y = g((X * W) + b)) \quad (4)$$

Where X is the input tensor, W is the weight tensor, and b is the bias tensor. A tensor product is performed between X and W, and the resulting tensor is then added with tensor b. In the end, the so-called transfer function (g) is applied. Once the proper value of W and b has been determined in the training phase, operation (4) can be propagated from layer to layer to convert input data (X) into the final prediction. Once the training has been completed, the model's prediction capability is evaluated on new input dataset, usually referred to as the validation dataset.

MLP has shown remarkable performance in different fields; however, its learning mechanism gives it no memory. Different other NN variations have been proposed to process a sequence or a temporal series of data points and overcome this problem. Examples are recurrent neural networks [11]. One of the latest proposed architectures is the long short-term memory neural network (LSTM). The major innovation of LSTM (see Figure 1) is the presence of a memory cell c_t able to accumulate state information. Different gates control operations like accessing, writing and clearing this cell. Every time a new input comes, its information will be accumulated in the cell if the input gate i_t is activated. Also, the past cell status c_{t-1} could be "forgotten" in this process if the forget gate f_t is on. Whether the latest cell output c_t will be propagated to the final state h_t is further controlled by the output gate o_t. One advantage of using the memory cell and gates to control information flow is that the gradient will be trapped in the cell and be prevented from vanishing too quickly, which is a critical problem for the previous models [17].

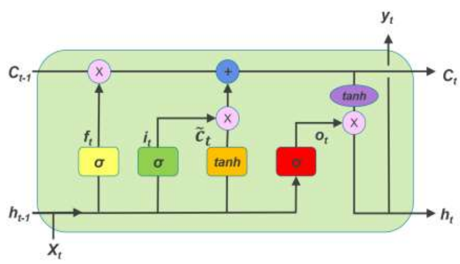

Fig. 1. LSTM model [17]

C. Proposed model: Multivariate Multi output LSTM

As [13] stated, spare parts demand may depend on many factors, and relying only on past consumption might not be accurate. In addition, the high number of spare parts makes the tuning phase almost impossible at an item level [19]. Previous neural network works have emphasized architecture that minimizes the accuracy error or optimizes the tradeoff between holding and stockout cost. However, little attention has been posed to the aforementioned practical problems. The main novelty of this paper is instead that of proposing a model able to both reach good accuracy metrics and deal with the previous practical implementation issues.

Keras library [7] has been used to implement the model. Keras is a high-level framework used to develop deep learning models. The adjective "deep" in the term deep learning means that the final model is obtained by stacking multiple models with different functions (usually referred to as layers). The proposed multivariate multi-output long short-term memory neural network (MMO_LSTM) has been obtained by stacking an LSTM layer over an MLP layer. This last layer has a number of output neurons equal to the number of considered items. This configuration is thus able to exploit the LSTM capability of best analyzing time series data. In addition, the MLP layer allows obtaining multi-output propriety. In addition, previous models in literature deal with the prediction of a single item at a time and thus require a proper tuning phase for each item. The proposed model instead, provide an unique output prediction for all considered items. As a consequence the time spent in tuning phase and the number of different tuned models that need to be used could be reduced.

Particular attention has been posed in data modelling phase with the objective of overcoming the limit of using single time series data. Previous literature have only relied on single items consumption history. However, components have interactions inside a final product and they usually share technical or functional similarities. As a consequence, prediction of an item's future consumption should be based not only on its single history but also looking at hidden pattern with the history of other items. In this paper the multivariate propriety of the proposed model has been designed to manage this issue. To do this, 3-dimensional input tensors have been projected. As figure 2 shows, items that share technical and functional similarities have been selected. Afterward, their historical time series consumption data were extracted from the ERP database. Subsequently, the historical time series has been split into two parts. In order to transform raw data into a format that the model can deal with, sliding windows have been used. The historical training predictors (X_train) and prediction (Y_train) tensors have been built by applying a sliding window to the first part of the previous split time series. Instead, the validation predictors (X_val) and prediction (Y_val) have been built by applying sliding windows to the second split portion. To give an example of the sliding window mechanism, consider starting at timestep 0 and considering, for example, a sliding window of 2 timesteps. The first row of X_train contains historical observations ranging from time step 0 to time step 2. In Y_train, the value of the 3rd historical observation is stored. The second row of X_train contains two historical observations starting from time step 1, while in Y_train is stored the demand value at timestep four and so on until all rows are stored. The same thing happens for the X_val and Y_val, but as stated above, the second split portion has been used to build them. Repeating this procedure for each item, the X_train, Y_train, and the X_val, Y_val matrix has become 3D tensors able to capture interactions between different items' s consumption history.

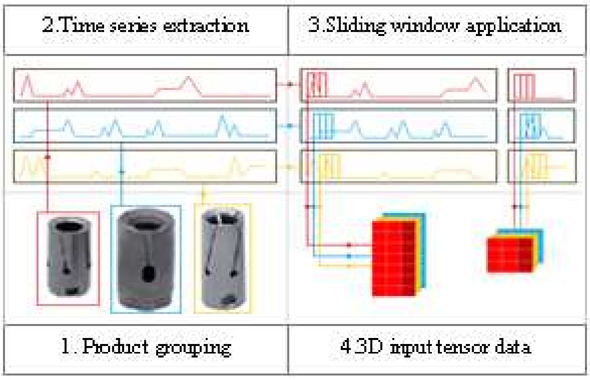

Fig. 2. Input data modeling process

IV. CASE STUDY

The proposed model has been applied to a real spare parts dataset of a mechanical company involved in the sector of machine tools. 138 mechanical collets with different internal and external diameters and different threads have been the object of the study. Their description of monthly ADI and CV^2 [27] is provided in Table 2.

TABLE 2
ADI AND CV^2 DATASET CLASSIFICATION

	Min	**Mean**	**Dev.std**	**Max**
ADI	*1,01*	*5,57*	*5,48*	*24*
CV^2	*0,33*	*6,34*	*6,58*	*32*

For each item, 96 monthly historical time series data points are used to train and validate the model. In particular, the first 72 observations are used as the training set, while the 24 remaining observations are used in the validation set. In the pre-processing data phase, a normalization procedure has been applied by subtracting to each data point the mean over the training set and dividing the result by the standard deviation.

Except for the mentioned parameters, other parameters have been set to default values provided by the Keras library for LSTM and MLP layers. Different sliding window sizes (1-3-4-6-12) have been tested to build the input tensor, and according to the results, a sliding window of three has been chosen as reported the best results. The batch size has been set to 72, and the number of epochs has been set equal to 150'000 with the option of early stopping training if for 300 consecutive epochs no improvements were found. RMS prop optimization algorithm has been used instead of the stochastic gradient descent algorithm. Different values for the learning rate have been tested (0.1-0.01-0.001-0.0001), and a value of 0.001 has been chosen as has reported the best result. The mean absolute error has been used as a loss function.

Original Croston, SBA, and SBJ variation and NN proposed in [22] have been compared on the validation dataset.
The python package of [20] has been used to implement the Croston method and its variation. The parameters for the NN_MUK are set as reported in [22]. The NN_MUK has been chosen among other NN implementations as in [19] is the one that reports the best results.

V. RESULTS

To compare the performance of different models, mean absolute percentage error (MAPE) as proposed by [12] and the total time to train and evaluate the model has been computed for 138 items. Results are summarized in Table 3.

TABLE 3
COMPARISON BETWEEN DIFFERENT MODELS

	MMO LSTM 4	**NN MUK**	**CRO**	**SBA**	**SBJ**
% Best	*43,1%*	*15,3%*	*30,7%*	*6,6%*	*4,4%*
Mean MAPE	*116,1%*	*134,4%*	*167,7%*	*152,1%*	*148,3%*
Min MAPE	*38,8%*	*37,6%*	*0,0%*	*33,4%*	*34,4%*
Dev.st MAPE	*50,2%*	*70,2%*	*152,9%*	*117,8%*	*106,1%*
Max MAPE	*415,3%*	*399,6%*	*959,8%*	*959,8%*	*959,8%*
Time (min)	*11,2*	*18,34*	*0,007*	*0,007*	*0,007*

The first row shows the percentage of times a model outperforms others in terms of MAPE. The second to fifth rows show respectively the mean, the minimum, the standard deviation, and the maximum value of MAPE across all items for each model. The third row shows the time models require to train and evaluate their performance for all items. Results show that MMO_LSTM outperforms other models 43,1% of the time, reduces the mean MAPE by 18,3 % with respect to NN_MUK, and is 63,8 % faster than NN_MUK. The high MAPE's upper bound reported for all models is probably due to the huge extension of the ADI value of the studied dataset, as the same value is limited to 5,44 in previous mentioned literature.

In addition, Figure 3 shows the correlation between the reported MAPE of Croston variations and that of the MMO_LSTM method and some characteristics of the dataset like the ADI and the CV^2 value and the number of timestep with zero demand (NZD).

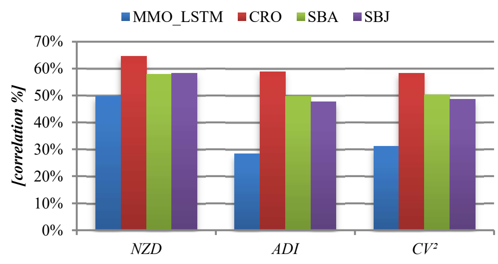

Fig. 3. Dataset feature correlation with MAPE

In the chart, the performance of the traditional method seems to be more conditioned by the variables mentioned above. Instead, the proposed model shows less dependence on them, remarking the possibility of achieving better results in context with high intermittences and lumpiness.

VI. CONCLUSIONS

Spare parts management has increasingly acquired relevance during the last decades, moving from the idea of a "necessary evil" to a source of potential profits. In addition, accurate forecasting plays an ever-increasing fundamental role in good inventory management.

Forecasting based on time series is widely used due to easy data availability, and NN has been increasingly tested in this field. However, forecasting based only on single time series has limitations. Components have interactions inside a final product, and they usually share technical or functional similarities. Consequently, the prediction of an item's future consumption should be based not only on its single history but also on hidden pattern with the history of other items. In addition, NN requires time to be tuned, and optimizing this task at an item level is almost impossible with the ever-increasing number of items that need to be managed.

In this paper a multivariate multioutput LSTM has been proposed and compared with different Croston variations and the NN proposed in [22]. The proposed model aims to overcome the problem mentioned above by looking at possible hidden patterns between the consumption history of different components and reducing the time for tuning as this task needs to be done for groups of items instead of single ones.

The mean absolute percentage error and total time to train and evaluate the model have been computed for 138 items with a mean ADI of 5,57 months. Results show that MMO_LSTM outperforms other models 43,1% of the time and is able to reduce the time for training and evaluating the model with respect to NN proposed in [22].

In conclusion, even if the model can reach improvements in accuracy and computing time, values of MAPE higher than 100% have been founded both in literature and in the study, remarking how challenging it is to make a forecast in the spare parts field also with modern technique. To address this problem and by looking at the limits of this paper, two future directions can be drawn. First, items have been grouped by looking at technical similarities, while providing a mathematical optimization procedure to the cluster creation could lead to performance improvements. In addition, this study considers only time-series data. Another option could be to consider different data sources, like the number of sold final products that adopts the studied spare parts, investigating thus information coming from the installed based.

REFERENCES

[1] Aktepe, A., Yanık, E. and Ersöz, S., (2021). Demand forecasting application with regression and artificial intelligence methods in a construction machinery company. *Journal of Intelligent Manufacturing*, 32(6), pp.1587-1604.

[2] Alsina, E.F., Cabri, G. and Regattieri, A., (2016). A neural network approach to find the cumulative failure distribution: Modeling and experimental evidence. *Quality and Reliability Engineering International*, 32(2), pp.567-579.

[3] Amin-Naseri, M. and Tabar, B., (2008). Neural network approach to lumpy demand forecasting for spare parts in process industries. *2008 International Conference on Computer and Communication Engineering*,.

[4] Bacchetti, A. and Saccani, N., (2012). Spare parts classification and demand forecasting for stock control: Investigating the gap between research and practice. *Omega*, 40(6), pp.722-737.

[5] Calabrese F., Regattieri A, Bortolini M, Galizia F.G, (2021). Fault Diagnosis in Industries: How to Improve the Health Assessment of Rotating Machinery. Smart Innovation, Systems and Technologies8th International *Conference on Sustainable Design and Manufacturing, KES-SDM 2021*. Volume 262(257–266)

[6] Calabrese, F., Regattieri, A., Bortolini, M., Galizia, F.G. and Visentini, L., (2021). Feature-based multi-class classification and novelty detection for fault diagnosis of industrial machinery. *Applied Sciences*, 11(20), p.9580.

[7] Chollet, F. et al., (2015). Keras. Available at: https://github.com/fchollet/keras

[8] Cohen, M. A., Agrawal, N., & Agrawal, V. (2006). Winning in the aftermarket. *Harvard business review*, 84(5), 129.

[9] Croston, J., (1972). Forecasting and Stock Control for Intermittent Demands. *Operational Research Quarterly (1970-1977),* 23(3), p.289.

[10] Del Buono, F.D., Calabrese, F., Baraldi, A., Paganelli, M. and Regattieri, A., (2021). Data-driven predictive maintenance in evolving environments: A comparison between machine learning and deep learning for novelty detection. In *Proceedings of the International Conference on Sustainable Design and Manufacturing (pp. 109-119).* Springer, Singapore.

[11] Elman, J. and Zipser, D., (1988). Learning the hidden structure of speech. *The Journal of the Acoustical Society of America*, 83(4), pp.1615-1626..

[12] Gilliland, M., (2002). Is forecasting a waste of time?. *SUPPLY CHAIN MANAGEMENT REVIEW*, V. 6, NO. 4 (JULY/AUG. 2002), P. 16-23: ILL.

[13] Gu, J., Zhang, G. and Li, K., (2015). Efficient aircraft spare parts inventory management under demand uncertainty. *Journal of Air Transport Management*, 42, pp.101-109.

[14] Guo, F., Diao, J., Zhao, Q., Wang, D. and Sun, Q., (2017). A double-level combination approach for demand forecasting of repairable airplane spare parts based on turnover data. *Computers & Industrial Engineering*, 110, pp.92-108.

[15] Gutierrez, R., Solis, A. and Mukhopadhyay, S., (2008). Lumpy demand forecasting using neural networks. *International Journal of Production Economics,* 111(2), pp.409-420.

[16] Hasni, M., Aguir, M., Babai, M. and Jemai, Z., (2018). Spare parts demand forecasting: a review on bootstrapping methods. *International Journal of Production Research,* 57(15-16), pp.4791-4804.

[17] Hochreiter, S. and Schmidhuber, J., (1997). Long Short-Term Memory. Neural Computation, 9(8), pp.1735-1780.

[18] Kourentzes, N., 2013. Intermittent demand forecasts with neural networks. *International Journal of Production Economics,* 143(1), pp.198-206.

[19] Lolli, F., Gamberini, R., Regattieri, A., Balugani, E., Gatos, T. and Gucci, S., (2017). Single-hidden layer neural networks for forecasting intermittent demand.*International Journal of Production Economics*, 183, pp.116-128

[20] Mohammadi, H., (2020). Croston, https://pypi.org/project/croston.

[21] Molenaers, A., Baets, H., Pintelon, L. and Waeyenbergh, G., (2012). Criticality classification of spare parts: A case study. *International Journal of Production Economics*, 140(2), pp.570-578.

[22] Mukhopadhyay, S., Solis, A. and Gutierrez, R., (2011). The Accuracy of Non-traditional versus Traditional Methods of Forecasting Lumpy Demand. *Journal of Forecasting*, 31(8), pp.721-735.

[23] pdfslide.us. 2022. *AIR TRANSPORT WORLD* - January (2013). (online) Available at: <https://pdfslide.us/documents/air-transport-world-january-2013.html> (Accessed 7 March 2022).

[24] Pinçe, Ç., Turrini, L. and Meissner, J., (2021). Intermittent demand forecasting for spare parts: A Critical review. *Omega*, 105, p.102513.

[25] Regattieri, A., Giazzi, A., Gamberi, M. and Gamberini, R., (2015). An innovative method to optimize the maintenance policies in an aircraft: General framework and case study. *Journal of air transport management*, 44, pp.8-20.

[26] Shale, E., Boylan, J. and Johnston, F., (2006). Forecasting for intermittent demand: the estimation of an unbiased average. *Journal of the Operational Research Society*, 57(5), pp.588-592..

[27] Syntetos, A. and Boylan, J., (2005). The accuracy of intermittent demand estimates. International *Journal of Forecasting*, 21(2), pp.303-314.

[28] Zhang, G., Eddy Patuwo, B. and Y. Hu, M., (1998). Forecasting with artificial neural networks:. *International Journal of Forecasting*, 14(1), pp.35-62.

Appendix A. LITERATURE SUMMARY

TABLE 1

NN MODELS FOR INTERMITTENT AND LUMPY DEMAND TIME SERIES FORECASTING

Reference	NN data		NN configuration	NN parameter				Dataset				Accuracy
Paper	Input variables	Output Variables	Type/ IL/HL/OL	Epochs	LR	OA	ADI	Time Aggr.	Item number	Series length	Training/ Test	MAPE
[15]	D_{t-1}; INZ_{t-1}	D_{t+1}	*MLP.* *1 layer,2 units* *1 layer,3 units* *1 layer,1 units*	*15’ 000*	*0.1*	*BP*	*3,38-5,44*	*day*	*24*	*967*	*65% / 35%*	*102%-153%*
[22]	D_{t-1}; $ZCUM_{t-1}$;	D_{t+1}	*MLP.* *1 layer,2 units* *1 layer,3 units* *1 layer,1 units*	*15’ 000*	*0.1*	*BP*	*2,63-3,28*	*day*	*24*	*967*	*80% / 20%*	*95%-145%*
[18]	$NZD_{t-1;t-2;t-3}$; $IDI_{t-1;t-2;t-3}$	NZD_{t+1}; IDI_{t+1}	*MLP.* *1 layer,2 units* *1 layer,1-3 units* *1 layer,2 units*	*1'000*	-	*LM*	-	*month*	*1’ 000*	*136*	*26 % / 74%*	-
[18]	$NZD_{t-1;t-2;t-3}$; $IDI_{t-1;\,t-2;\,t-3}$	$DRATE_{t+1}$	*MLP.* *1 layer,2 units* *1 layer,1-3 units* *1 layer,1 units*	*1'000*	-	*LM*	-	*month*	*1’ 000*	*136*	*26 % / 74%*	-
[3]	D_{t-1}; INZ_{t-1}; $ZCUM_{t-1}$; $NZCUM_{t-1}$; $IFNZ_{t-1}$; IFZ_{t-1}; $DMEAN_{t-6}$; $DMAX_{t-6}$	D_{t+1}	*RNN.* *1 layer,8 units* *1 layer,1-15 units* *1 layer,1 units*	-	*0.01*	*BP*	-	*month*	*30*	*67*	*82% / 18%*	*51%-165%*
[19]	D_{t-1}; I_{t-1}; $ZCUM_{t-1}$	$D_{t+1;t+3;t+5}$	*FF.* *1 layer,3 units* *1 layer,3 units* *1 layer,1 units*	*10'000*	*0.1*	*BP,EL*	*1,18-3,78*	*week*	*24*	*61-76*	*65% / 35%*	-
[19]	D_{t-1}; I_{t-1}; $ZCUM_{t-1}$	$D_{t+1;t+3;t+5}$	*TD.* *1 layer,3 units* *1 layer,3 units* *1 layer,1 units*	-	*0.1*	*BP,EL*	*1,18-3,78*	*week*	*24*	*61-76*	*65% / 35%*	-
[19]	D_{t-1}; I_{t-1}; $ZCUM_{t-1}$	$D_{t+1;t+3;t+5}$	*Type: R.N.N.* *1 layer,3 units* *1 layer,3 units* *1 layer,1 units*	*10'000*	*0.1*	*BP,EL*	*1,18-3,78*	*week*	*24*	*61-76*	*65% / 35%*	*50%-120%*