Sales Volume Prediction and Application to Materials Trading

Marc Souply, Marc Malmaison, François Rioult, Bertrand Cuissart

Sales Volume Prediction and Application to Materials Trading

Marc Souply*†, Marc Malmaison*, François Rioult†, Bertrand Cuissart†
*RMAN Sync, 60 rue Philippe Livry Level 14760 Bretteville-sur-Odon, FRANCE
{*}firstname.lastname@rman-sync.com
†Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC Campus 2 Côte de Nacre 14000 CAEN, FRANCE
{†}firstname.lastname@unicaen.fr

***Abstract*—The reliability of sales forecasting is critical for an industrial decision support system dedicated to raw material retailers. However, it turned difficult to train and maintain a custom model dedicated to each of the numerous references. For every reference, it would be needed to select the most accurate algorithm together with its relevant features, then, to exhaustively test every relevant combination of parameters. This was the reason why we explored an approach based on auto parametrization of well-known predictive models, while adding specific seasonal features. From our experiments, the Dynamic Harmonic Regression (DHR) based on ARMA stood out as being the most effective model for popular products: it reached a fair accuracy while requiring a reasonable cost to train. However, when it came to more volatile products, a simple prediction like the average sales per week over a year often performed the best. Thus, YearlyMean saved computational resources that could then be used to exhaustively train DHR or LSTM models on some company key products, leading to a potential improvement of their forecasts. Then, one details the implementation of a smart computing machine learning process based on predictive scenarios that seek for a trade-off between the consumed resources and the predictive performances.**

***Index Terms*—smart computing, data analysis, neural networks, arma model, random forest, sales forecast**

I. Introduction

Material dealers are little known to the public despite their key role in the construction industry. On the surface, the basic principle of these trades seems simple: to buy in large quantities from suppliers and wholesalers construction materials such as plasterboard, pipes, paint, and then to resell these products to craftsmen on construction sites. Sales volume prediction is a regression problem that consists of using the description of a period to predict its sales volume. While this problem is critical in the materials trading sector where sales margins are low and storage costs are high, none of the popular prediction methods are currently used by players in the field. Most often, the players rely on empirical methods linked to their business skill, close to chartist [2] analysis, older methods such as the Wilson model [12], [10] or the Chaikin [1] oscillator analysis.

If these methods and the business knowledge of domain specialists can guarantee a certain profitability, a decision support based on relevant sales prediction could assist the players and potentially increase their profit margins. It is therefore worth asking why traditional predictive models such as ARIMA, random forests or neural networks are not already integrated into ERP modules. This can be explained by a combination of factors. From a strictly logistical point of view, materials trading is not easy to automate: there is a desire to maintain expertise and human contact, prices are often negotiable, suppliers' end-of-year bonuses are difficult to evaluate. A product usually has several alternatives, transport rates are complex to predict. It is usual to build up a stock to satisfy the customer, even if the latter is at a loss, and the activity is also the result of complex arrangements between the players. All these characteristics make a strict automation of the field difficult.

A sales volume prediction system is then hard to build, it is however the cornerstone of an intelligent system for this domain. There are several reasons for this complexity. The first is the number of different items: with an average of 20,000 items per trader, proposing a predictive solution for all these products that can be trained and updated in a reasonable time is a computational challenge. For example, it is not appropriate to train a neural network for each of the products. Also, the selection of relevant descriptors is a difficult task. Some products are weather sensitive, others depend on one or more specific seasonality (monthly, quarterly), and it is impossible to test all possible descriptors to determine an optimal selection. Finally, the sales volume at the product level is too fine to predict the sales of an entire warehouse. In practice, it is necessary to take a strategic view and consider each product as an element of the business to which only a portion of the predictive resources can be allocated. To be able to make a prediction, a metalearning approach like ensemble modeling [4] necessitates to train every model on every product. Other approaches [6] are focused on forecasting the future losses made by the investigated models, then to specifically choose the proper model according to the conditions. Both methods require to train many models to predict sales volumes or to forecast the related errors, which causes a very heavy computational burden, often an unaffordable one.

While risks are limited when predicting all the sales on all the warehouses because a general average stabilizes the system, it is not the case when the granularity is the sales prediction of a particular warehouse, because they can fluctuate greatly according to parameters that are difficult to determine or anticipate. There are different physical granularities, from

the single reference to the entire network and the warehouse. From another point of view, the temporal granularity is also very important and shows a great diversity of problems: it is easier to predict monthly sales than weekly sales. However, weekly accuracy may be necessary, depending on the supply time and the turnover rate of the products. Our work thus consists in proposing a predictive system optimizing the total learning time for all products, by adapting the physical (warehouse) and temporal (weekly predictions) granularities.

This text provides practical lessons resulting from our work in the field. Specifically, we will illustrate the following points:

- the interest of generating seasonal descriptors according to known sales;
- the comparison of four existing predictive models when it turns impossible to determine an optimal parameterization specific to each product;
- the orientation of computational resources towards the most relevant products while minimizing the backlash on the general prediction of the other items.

II. Materials and methods

Since the goal of our work was to predict the sales volumes of different construction items, we evaluated models obtained from real sales data.

A. Data

The data used resulted from collecting and summing daily sales volumes of 1,000 products in a trading company, over three years. In this paper, we focus at first on the analysis of the results obtained from the ten products generating the most sales. The learning and testing periods are specified by Table I. Saturdays and Sundays did not generate sales and are therefore excluded from the study. Specifically, in the case of time series, a cross-validation scheme cannot be used as the observations of the learning set have to chronologically precede the observations of the test set [8, Chapter 5.10]. Moreover, there are few data available, thus the trade off between the sizes of the training set, the validation set and the test set is a complex one. Here, we chose to dedicate 93% of the data for training, 3% for validation and 3% for the test (see Table I). For confidentiality reasons, the data cannot be detailed, nor can the nature of the products processed. The prediction problem was a regression problem: each object represented a sales period to which was associated a quantity, the latter being the information targeted by the prediction. In the materials trading, a retailer has to be able to immediately supply the most frequently demanded references. Thus, one can consider that, for these references, demand is met: there is no bias in predicting sales instead of demand.

The descriptors used to represent a day could be separated into three types. First, three pieces of weather information were considered: wind strength, precipitation volume and temperature of the warehouse area. All these descriptors were continuous variables. Second, the date was decomposed into eight discrete variables: the number of the day in the current year, the number of the week, the number of the month, but also the number of the week in the month, *etc*. Finally, four calendar events were considered in the form of binary variables: holidays, confinement, COVID-19 period and the presence of extraordinary activity. In total, we had 15 descriptors.

TABLE I
The data used and their partition between training and test samples.

Dataset	Period	Size (in week)
learning	from 2017-12-18 to 2020-08-30	141
validation*	from 2020-08-31 to 2020-09-27	5
test	from 2020-09-28 to 2020-10-30	5

* : validation is used for LSTM and RF tuning, for the two other models, validation set is included in training set

TABLE II
Details on the studied products.

product	average	max volume	min volume	relative stdev $\frac{\sigma}{\mu}$
C2-2	1333	4452	24	0.60
C2-1	370	1176	0	0.67
C4-1	1112	4460	0	0.69
C1-3	121	508	2	0.70
C3-1	64	231	1	0.71
C1-4	273	1010	3	0.71
C1-1	445	1872	11	0.73
C1-2	133	546	1	0.74
C1-5	103	458	0	0.77
C1-6	177	1125	0	0.86

The target variable corresponded to the sales made on a given week, on a specific reference in a merchant's warehouse. The exact nature of the products was anonymized for confidentiality reasons but they were grouped by category (C1, C2, C3) and a variation in the category (-1, -2, *etc.*). Table II details the average, the maximum sales volume, the minimum, and the relative standard deviation for each of the products studied. It can be noticed that there is no obvious correlation between the average sales of a product and the associated standard deviation, especially for products C1-2 and C2-3. In view of the differences between each maximum value and the associated mean, we could expect the presence of extraordinary peaks of activity.

B. Indicators of prediction quality

Different indicators of the prediction quality for each models were used: MAE, AIC and MAPE [8, Chapter 3.4 and 5.5], detailed below. These indicators could sometimes be used for model development (see Section II-C) in a framework of optimization (MAE for neural network and random forest) or choice of parameterization (AIC for the DHR model), and sometimes they were useful to estimate the quality of a model in terms of prediction, as an indicator of the risk incurred.

Based on week granularity of the sales volumes to be predicted, we note:

- y the actual value of sales;

- $\widehat{y}$ the sales forecast;
- $\overline{y}$ the average sales.

a) Mean Absolute Error (MAE): The MAE was used when computing models (e.g., neural network and random forest, see Section II-C) as the objective function to minimize: $MAE = \frac{\sum_{i=1}^{n} |\hat{y}_i - y_i|}{n}$.

b) Akaike Information Criteria (AIC): AIC provides an estimate of the information loss when using a model and parameterization to represent the process that generates the data. The AIC allowed to choose, between several parameterizations of a model (for example ARMA or DHR, see section II-C), the one that best represented the training data. However, it did not give any information on the intrinsic quality of the model relative to others and could not be a criterion for model selection. $AIC = 2k - 2\ln(L)$ where k is the number of parameters to be estimated and L is the maximum likelihood.

c) Mean Absolute Percent Error (MAPE): :

We evaluated the relevance of our predictions in terms of risk using a normalized version of the Mean Absolute Error (MAE): the Mean Absolute Percent Error (MAPE). MAPE is the empirical mean of the prediction deviations normalized by the true value: $\mathcal{E} = \frac{1}{n} \sum \frac{|\hat{y} - y|}{y}$.

According to the granularity, we considered for a week w that y_w was the sales of this week, as the sum of the sales of the corresponding days. The associated prediction error was: $\mathcal{E}_{week} = \frac{1}{n_{week}} \sum_w \frac{|\hat{y}_w - y_w|}{y_w}$.

C. Predictive models

In the experiments reported in Section III, we compared four predictive models.

a) Yearly Mean (YM): YM was a basic prediction model, where the $\widehat{y_i}$ value of the product was considered to be determined by:

With 52 the number of weeks in a year:

$$\hat{y}_w = \frac{1}{52} \sum_{j=1}^{52} y_{w-i}$$

The principle behind this naive model was to propose a prediction based on the average sales of a product during the last 52 weeks. It is widely used in this industry at the moment.

b) Recurrent Neural Network with Long Short Term Memory (RNN LSTM): : The neural network is a model dating back to the 1950s [11]. Initially neglected due to a lack of computing power, this model has been popular since the 1990s. The recurrent neural network is a variant allowing to take into account the sequentiality of the input data, by linking the output of a cell to its input. Specifically, the short- and long-term recurrent neural network is a recurrent neural network in which each cell handles three additional signals: input, output and forget. This configuration allows to modify at each iteration a state related to the hidden layers of the model, influencing the next signals passing through the neuron. The optimization of this model is done in our case by minimizing the MAE.

c) Random Forests (RF): [5] Random forests, although mainly used to solve classification problems, can also be used in the regression framework as it is the case here. Each tree is generated by drawing N observations from the training set, and selecting a sample of the total number of descriptors related to these observations. Once these N draws and this selection of B descriptors have been made, a decision tree is trained for each draw, whose growth is limited by the setting of a maximum depth. During the prediction, each tree will provide its prediction and the final value will be an average of these. The optimization of this model is done by minimizing the MAE.

d) Dynamic Harmonic Regression based on an ARIMAX model (DHR): [8, Chapter 9.5] The ARMA models [3, Chapitre 3] from which ARIMAX is derived are based on two main processes: autoregression, meaning that a variable $X(t)$ can be explained by its past values, and a Moving Average over forecast error process, meaning that the variable can be explained by the accumulation of white noise (the errors) from previous predictions. An ARMA model then considers that X can be explained based on the previous values of X, and the previous errors. The optimization of this model is done by minimizing the AIC.

The I in the acronym ARIMAX corresponds to a need to differentiate the time series to make it stationary, and the X adds to the model the possibility of using the descriptors presented previously as terms in the regression. In the context of a prediction based on multiple seasonalities, some of which are long, (52 for a year of working weeks for example), it is possible to move from a SARIMA(X) model to an ARIMAX model where the different seasonalities are extracted and considered as descriptors of the model. We later denominate this method as *DHR*.

D. Parametric optimization of the predictive models

A specific parameterization was necessary for each predictive model. When the number of references was small, we look for an optimal parameterization for each model and each product. However, for our application, it was impossible to dedicate a specific model to each of the 20,000 products because the computation time would be prohibitive. For example, LSTM need $36mn$ to test the 162 combinations of values, and would lead to $500days$ of computing time for the 20,000 products.

In addition to the selection of model parameters, the selection of relevant parameters could also be preprocessed. Most parameters selection algorithms are very expensive since they result from optimizations based on testing all possible combinations. In this work, we tried to use random forest feature subset selection [9] to improve our prediction on every model involved in this study. DHR was the only one improved by this feature selection, and was therefore the only model on which this filter was used.

E. Creation of seasonal descriptors

In order for a predictive model to incorporate seasonal variations associated with the sales of a product, it was necessary to

introduce the causes of these variations in the form of descriptors. Once determined, these seasonalities would become new descriptors. The calculation of the seasonality began by using an auto-correlation function (ACF) [3, Chapter 2] to determine the time lags with which the target variable appears to be correlated. For example, a peak on the ACF repeated with a 4-weeks lag (see Figure 1) suggested monthly seasonality, a 12- or 13-weeks lag for trimestrial seasonality, and so on. Once the predominant seasonality was determined, a Seasonality-Trend Loess (STL) [7] decomposition of the target variable was performed. STL separated the time series into three additive components: a trend, a seasonality, and a residual that had the properties of noise. Figure 2 shows on its upper part the series to be decomposed and on the other three graphs the components whose addition gives the original series. The seasonality component gave rise to a new series which was a new input descriptor to feed the model. This operation of decomposition according to the predominant seasonality was repeated until all significant peaks of auto-correlation (exceeding the gray area on Figure 1) were considered.

For some models, RF or DHR, a simplification of the seasonal descriptors improved the results. This simplification consisted in truncating the Fourier transformation of the seasonality signal to filter out the high harmonics, which induced a generalization of the seasonality. Figure 3 shows on its upper part the 20-weeks seasonality and the lower part the simplification obtained.

F. Implementation tools

To experiment on our data, we used a Python implementation with some of it's popular library : pandas, to manipulate dataset of products, sklearn for random forest forecast, importance filtering and data scaling, pmdarima for DHR forecast, tensorflow for LSTM forecast, numpy and statsmodels were used to isolate and transform seasonality descriptors.

III. Results and discussion

This section presents the results of our experiments in sales volume prediction on the top ten selling products. We begin by discussing the value of adding seasonal decomposition descriptors and then compare the different models. We conclude by proposing predictive scenarios combining the best models.

A. Interest of the seasonal decomposition

The value of adding seasonal descriptors can be seen in the results of Table III. This table shows the weekly error rates of the three methods, with and without the seasonal decomposition. The seasonal decomposition always improved the results. For the rest of the analysis, seasonal descriptor will be included in models, as they always improved prediction and are cost effective to compute.

It is also relevant to note that the DHR process had better performance (29.41% MAPE) than the traditional seasonal model SARIMAX (32.65%). In addition, the training was faster ($15s$ for DHR, $4.6mn$ for SARIMAX) as SARIMAX needs to tune 4 more parameters.

TABLE III
Contribution of the seasonal decomposition on the MAPE error (in %).

model	without seasonality	with seasonality
DHR	31.45	**29.41**
LSTM	42.78	**34.00**
RF	35.81	**35.70**

TABLE IV
Weekly MAPE results of the different models on 10 top products.

product	YM	DHR	LSTM	RF
C2-2	36.23	**26.91**	32.06	31.41
C2-1	30.10	19.43	18.94	**17.26**
C4-1	29.65	25.86	**20.83**	22.63
C1-3	31.83	**25.84**	28.67	30.66
C3-1	26.73	26.27	34.45	**23,44**
C1-4	46.63	**39.65**	50.79	53.64
C1-1	35.40	**17.71**	32.96	26.77
C1-2	56.59	**44.56**	59.14	63.47
C1-5	32.91	38.83	**25.18**	37.07
C1-6	32.74	**29.07**	36.96	38.18
mean	35.88	**29.41**	34.00	34.45
training time	7.3 s	15 s	36 mn	3.4 mn

B. Model comparison

We trained DHR, LSTM, RF and YF for the ten best-selling products. For each model, Table IV gives the results by providing the mean absolute percentage error (MAPE) on these products, as well as the related learning error. It was easy to designate the model best suited to the constraints imposed: the DHR. This model was the most suitable because of its global predictive results and its low learning time. The results of the LSTM did not justify its training cost, and the RF, although faster, offered the least good results of the three models. The naive YM, while being the fastest to train had the worst performance.Figure 4 gives the same results, and allows to see that while error value were fluctuating, relative ranking remained the same for each model (peaks and troughs).

The good performance of the DHR was not surprising, as the algorithm includes an exhaustive search for the best parameterization which had little impact on the learning time. Resulting in a quickly trained model, whose parameterization is specific to each product.

It is important to keep in mind that, despite the inclusion of lockdowns and COVID19 in the descriptors, the October 2020 period was part of a year that was globally difficult to predict due to its exceptional character. Moreover, additional field parameters, not taken into account in the descriptors, could remove uncertainty: reliability of vendors to immediately record their sales, discount on products, price hike due to COVID. At the end, weekly forecasts prove acceptable for a good number of products, in particular seven products out of the ten were found to be below the 30 % error threshold.

C. Predictive scenarios

This study allowed us to determine the most relevant model to propose a global prediction of the ten most sold products.

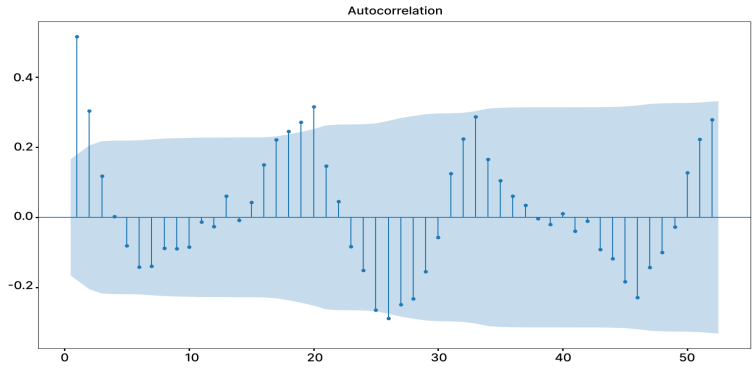

Fig. 1. ACF of the C2-2 with periodicity of 52.

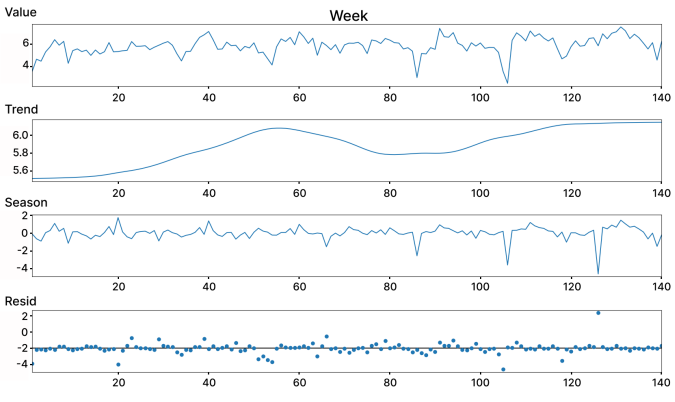

Fig. 2. STL decomposition of the C2-2 product with a seasonal periodicity of 20.

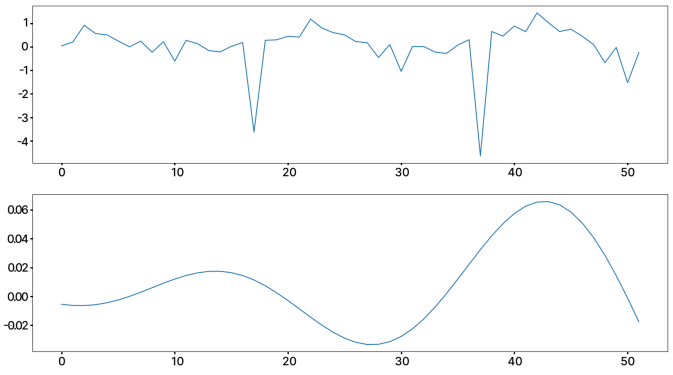

Fig. 3. Simplification by STL decomposition and Fourier order reduction.

However, we tried to generalize this process for the top 100 products, and the results can be viewed on Table V. Even if some products pulled upwards the week error (one product reached a 1000% error for example), the YM prediction model became the best model when the sales decreased. It implies that the choice of the best forecast model for a product could be related to the number of sales, or other parameters. In order

TABLE V
PREDICTIVE SCENARIO AGAINST EVERY PREDICTION.

model	week error	n. of first	n. of sec	n. of third	n. of fourth
YM	61,91 %	312	346	311	31
DHR	69,83 %	115	122	358	405
LSTM	65,81 %	127	178	171	524
scenario	56,90 %	492	318	152	38

to check this theory, we tried to build two decision tree. As the YM is the fastest and efficient model, we began to build a tree (YM-TREE) able to determine if the YM could be a reliable model for a specific product. Then we built another tree (LSTMDHR-TREE) to determine whether we should use an LSTM or a DHR, or if both processes could work.

To build these two trees, we used the prediction on 100 products of YM, DHR and LSTM in the following way: a model was considered viable to predict a specific product if its error was within a range of 5% with the best prediction error for this product. For example, our product C1-3 scored 31.83% (YM), 25.84 (DHR)%, 28.67% (LSTM) meaning the acceptable models for this product were DHR and LSTM. We then built the YM-TREE by targeting the presence of YM in the acceptable models, creating a binary target to predict the relevant use of YM.

We used few and easy-to-compute descriptors: relative standard deviation, total quantity of sales, number of sales, results of DHR and YM on the evaluation set. We did the same process for the LSTMDHR-TREE except that the classification task contained 3 classes : LSTM viable, DHR viable, LSTM AND DHR viable.

Once those two tree were built, we checked for each product if the YM was a good model. If so, YM was used to predict the sales. If it wasn't, LSTMDHR-TREE was used to guess which model would be suitable between LSTM or DHR (if both were acceptable, DHR would be preferred as it is faster). In order to evaluate this process, 10 products were randomly picked, removed from the prediction pool and predict by using the 90 others. This process was iterated with the 10 next products from the prediction pool and predicted it with the

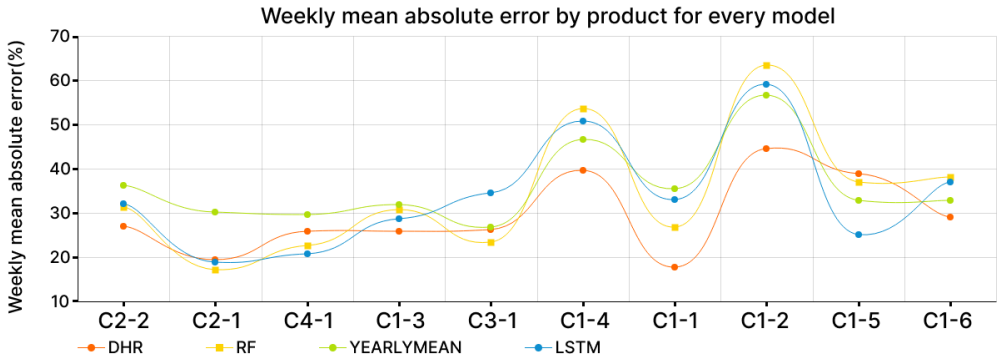

Fig. 4. Weekly absolute error.

90 others, until the prediction pool was empty. This process, named *random sub-sampling validation* [13], was done 100 times. To simplify, the YMTREE+LSTMDHR-TREE model would be referred as *scenario model* in the following.

The results of the scenario model are given on Table V and we added the number of time a model was the best to predict a product. Using the scenario model resulted in a significant improvement in predictions, at least 5% against any model, while showing that nearly half of the time the scenario managed to propose the best average prediction on the 10 random products chose at every iteration.

During 10 000 predictions (prediction on 100 products, 100 times), we observed that in average 70% of YM are computed, 20% of DHR and 10% of LSTM. The overall computation time on 10 products was then estimated to $39.4mn$ for the scenario, which was clearly endurable when compared to $73s$ for YM, $150s$ for DHR and $360mn$ for LSTM.

IV. Conclusion

This study was a practical comparison of four classical forecast models, in the specific framework of material sales prediction. Beyond the prediction performances, we paid particular attention to the learning times as well as to the selection of descriptors. We concluded that taking seasonality into account was efficient: although it slightly increased the computation time, it took into account the particularities of each product. With an adapted scenario model automatically choosing between YM, DHR and LSTM, we were able to improve the average prediction for a fully automatized prediction system on numerous products. While the paper reports a study computed on 100 products, a forecasting problem has to practically deal with up to $20,000$ items per trader. Consequently, there are potentially strong correlations between sales volumes of different items which are not yet exploited in our current model.

References

[1] Achelis, S.B.: Technical Analysis from A to Z. McGraw-Hill Professional (2014)
[2] Boutouria, N., Hamad, S.B., Medhioub, I.: Investor behaviour heterogeneity in the options market: Chartists vs. fundamentalists in the french market. Journal of Economics and Business **3** (2020)
[3] Box, G.E.P., Jenkins, G.M.: Time Series Analysis: Forecasting and Control. Holden-Day (1976)
[4] Breiman, L.: Stacked regressions. Machine learning **24**(1), 49–64 (1996)
[5] Breiman, L.: Random forests. Machine Learning **45**, 5–32 (2001)
[6] Cerqueira, V., Torgo, L., Pinto, F., Soares, C.: Arbitrated ensemble for time series forecasting. In: Joint European conference on machine learning and knowledge discovery in databases. pp. 478–494. Springer (2017)
[7] Cleveland, R.B., Cleveland, W.S., McRae, J.E., Terpenning, I.: STL: A seasonal-trend decomposition procedure based on loess. Journal of Official Statistics **6**, 3–73 (1990)
[8] Hyndman, R.J., Athanasopoulos, G.: Forecasting: Principles and Practice. OTexts (2013)
[9] Jaiswal, J.K., Samikannu, R.: Application of random forest algorithm on feature subset selection and classification and regression. In: 2017 World Congress on Computing and Communication Technologies (WCCCT). pp. 65–68. IEEE (2017)
[10] Jørgensen, P.L.: An analysis of the Solvency II regulatory framework's Smith-Wilson model for the term structure of risk-free interest rates. Journal of Banking and Finance **97**, 219–237 (2018)
[11] Rosenblatt, F.: The perceptron: a probabilistic model for information storage and organization in the brain. Psychological review **65**(6), 386 (1958)
[12] Smith, A., Wilson, T.: Fitting yield curves with long term constraints. Tech. rep., Bacon and Woodrow (2000)
[13] Xu, Q.S., Liang, Y.Z.: Monte carlo cross validation. Chemometrics and Intelligent Laboratory Systems **56**(1), 1–11 (2001)