XLI CILAMCE
XLI Ibero-Latin American Congress on Computational Methods in Engineering
16-19 November, 2020 FOZ DO IGUAÇU PARANÁ - BRAZIL

LSTM Ensemble Approach for Demand Forecasting in Supply Chain Management

Paulo J. D. A. Barbosa, Daniel C. Cavalireri

PROPECAUT - Espírito Santo's Federal Institute
Serra - ES, Brazil
paulobarbosa.vix@gmail.com, daniel.cavalieri@ifes.edu.br

Abstract. Perishable product handling represents great challenge to supply-chain management. These items have shorter shelf-life and most of them have special needs for transportation, storage and display. Inappropriate treatment of these products, as well as the inability of the retailer to sell them within its shelf-life leads to waste of products that affects the entire supply chain, society and environment. The trouble to manage these products is to match the stock replenishment with its future demand and retail profits are significantly affected by perishable losses. This paper proposes an approach of an one-step ahead forecasting for single time series using LSTM Ensemble. The forecasts obtained by the proposed method yielded smaller forecasting error compared to the best single LSTM, best single ARIMA and to the 12-month average sales - method commonly used by retail.

Keywords: Long Short-Term Memory, Ensemble, Autoregressive Integrated Moving Average, Forecasting, Supply Chain Management

1 Introduction

Perishable product handling represents great challenge to supply-chain management. These items have shorter shelf-life and most of them have special needs for transportation, storage and display. Inappropriate treatment of these products, as well as the inability of the retailer to sell them within its shelf-life leads to waste of products that affects the entire supply chain, society and environment. According to the Brazilian Supermarket Association [1], in 2018, these losses summed up to 1.89% of gross revenue and $1.53 billion in absolute numbers. The most affected product categories are bread and bakery, sea food, meats, rotisserie, prepared foods, frozen foods and produce. Together, these categories represent about 26% of the loss cost over the entire Brazilian retail revenue.

The trouble to manage these products is to match the stock replenishment with its future demand. Overestimate the demand leads to financial resources stagnation; food wasting; unnecessary warehousing usage, energy consumption and human resources allocation. On the other hand, underestimate the demand can cause invaluable impact on the final consumer, growing dissatisfaction and leading to it to competitor store [2]. Small retailers can rely only on average sales and, at most, apply simple supply-chain management techniques, such as economical order quantity (EOQ) [3]. That policy implicates in higher safety stock and fixed warehouse cost. In opposition, large retail chains can afford supply-chain management personnel responsible for forecasting future demand based on the sales history for each product or group of product.

Inspired by that challenge, the main objective of this paper is to present a flexible, automatable and accessible method for demand forecasting. In order to achieve that, the authors developed a solution that allows fast replication of experiments from data preparation to the final forecast results. The proposed solution is based on open source programming language and cloud-based processing environment for performing an one-step ahead forecasting for single time series. Aiming at the reality of retail, all experiments were based on real sales data available on Kaggle from a large Ecuadorian-based grocery retailer. In the experiments to be presented, two different forecast predictors and its corresponding ensemble were compared to each other and to the 12-month moving average sales - method commonly used by retail. The forecasts obtained by the proposed ensemble method yielded smaller forecasting error compared to the best single sub-models predictors and to the 12-month average sales.

2 Theoretical background

Since forecasting represent such an important role in business, researchers try to keep improving their methods. Makridakis open competitions are one the most important events in this field of study and help to evaluate and compare state of the art with standard methods. The M4-Competition tested 61 forecast methods dividing them into statistical and machine learning (ML) benchmarks and standards for comparison [4]. This section describes the used time series forecast methods, the proposed ensemble method and the metrics used to evaluate models performance.

2.1 Autoregressive Integrated Moving Average

The Autoregressive (AR) Integrated (I) Moving Average (MA), also known as ARIMA, was proposed by Box and Jenkings [5] in their publication in 1970. A common way to denote an ARIMA model is by setting p, d and q values. Being p the number of lagged observations included in the model; q, the number of times that difference operation is applied to raw observations; and d, the size of the moving average window.

ARIMA is the most widely used method for time series forecasting due to its generality, simplicity and various succeeded application evidences. Capable to deal with any kind of time series data [6], it is also one of the two comparison standards from M4-Competition. When modeling a time series data set, the practical problem is to choose p, d and q values to minimize forecast errors [7].

2.2 Long Short-Term Memory

State of the art of Machine Learning forecast methods, Long Short-Term Memory (LSTM) networks are a special case of a Recurrent Neural Network (RNN). LSTM was first proposed by Hochreiter and Schmidhuber [8] to solve the vanishing gradient problem caused by the gradient descent algorithm during the network weight updating for long series. Because it is a neural network, LSTM can learn both linear and nonlinear relations from data [9]. In addition, since the group of cells are able to maintain memory, the context of input values over time is learned. The basic LSTM networks architecture - Vanilla LSTM - consists of a single LSTM layer and one fully connected layer for regression. A LSTM layer, in turn, is based on cells units composed by tree elements defined as gates: input gate (i_t), output gate (o_t) and forget gate (f_t). Each cell is fed with input value x_t, previous cell's output $h_{t\text{-}1}$ and previous cell's state $c_{t\text{-}1}$. From each cell derives the output h_t and the cell's state c_t. Sigmoid and hyperbolic tangent activation functions; summation, concatenation and convolution operations are executed inside the cell. For mathematical formulation, see [10].

Recent papers has shown that it is possible to improve basic LSTM forecasts by applying feature engineering to input data [11], implementing more complex architecture such stacked or multivariate LSTM [12], and mostly by applying ensemble learning [13–15].

2.3 Ensemble learning

Ensemble learning aims to minimize forecast error by combining multiple predictors under a rule or a meta-learner such as Linear Regression, Random Forests or Artificial Neural Networks [16, 17]. Ensemble strategies can be classified in the following groups: bagging, boosting and Stacking. Bagging, also known as Bootstrap Aggregation, consists of retraining a model by resampling the training data. It is Not very used in time series applications due to the existing dependency on the sequence input data [18]. Boosting improves poorly predicted values by increasing their input importance on the next predictor model's training. Finally, Stacking combines multiple forecasters as input for an upper layer model.

Multivariate Linear Regression (LR) for Ensemble Learning have been widely used for a long time [19, 20]. It consists of calculating the weights for each input variable that minimizes the overall error to corresponding line (single input) plane (two dimensional) or hyper plane (high dimensional). Its usage for ensemble forecasters, specially, LSTM models lacks of publications.

2.4 Evaluation

The choice of the right metric for forecasting evaluation is affected by a number of factors: presence of zero, negative values and outliers; comparing different data sets and benchmark an experiment. The best approach is usually to analyze the results based on multiple metrics.

Scale-based methods can be used to compare different forecasters on the same data. For methods such as Rooted Mean Squared Error ($RMSE$) and Mean Absolute Error (MAE), the resulting metric is in the same scale as the data easing the interpretation. Despite ($RMSE$) is widely used, (MAE) is less sensible to outliers in the forecasts results [21].

Scale-independent methods are important to evaluate same forecaster on different data sets. Methods such as Mean Absolute Percentage Error ($MAPE$) and Mean Absolute Scaled Error ($MASE$), make possible to benchmark the experimented forecast model [22].

3 Proposed Solution

In this section all the steps will be presented to enable the implementation of the proposed model and run the experiments to be described. The solution were developed using Python 3.7 on Google Colab as well as TensorFlow, Keras and Scikit-Learn libraries for machine learning techniques.

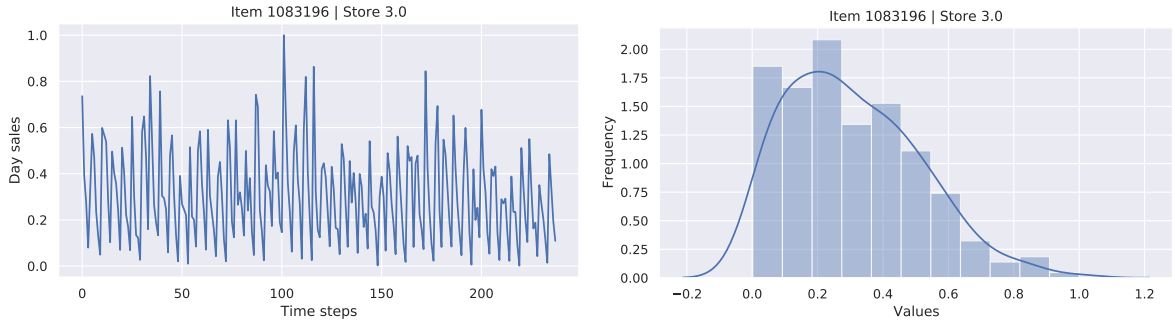
3.1 Data Preparation

The experiments were based on real sales data available on Kaggle from a large Ecuadorian-based grocery retailer: Corporación Favorita [23]. The data set contains point of sales of 4100 numbered items divided into item families and item classes. All items are labeled as perishable or not. The sales are derived from 54 different stores, from January 2013 to August 2017. The data set also contains complementary information such as: "on promotion", holiday calendar, oil price time series and clustering groups for the stores.

As in a real supply chain forecasting situation, there is a huge amount of data. Therefore, the data preparation began by daily aggregating item sales per item, per store, and then filtering the desired data applying the following sequence:

1. Keep only perishable items.
2. Keep topmost selling item class from each family.
3. Keep only time series, at least, 2 years long and missing no more then 2.5% values.
4. Select topmost selling items with 100% holiday matching for missing values.
5. Normalize data by scaling it from 0 to 1.
6. Treat outlier samples by replacing them with its three-sigma limit.

From the filtering presented above, the selected data set were Item 1083152 from Store 47 with $total\ sales = 564.74$, $mean = 0.3361$ and $standard\ deviation = 0.2190$. The time series and histogram plots from the selected time series are shown in Figure 1.



(a) Selected data set time series plot.

(b) Selected data set histogram.

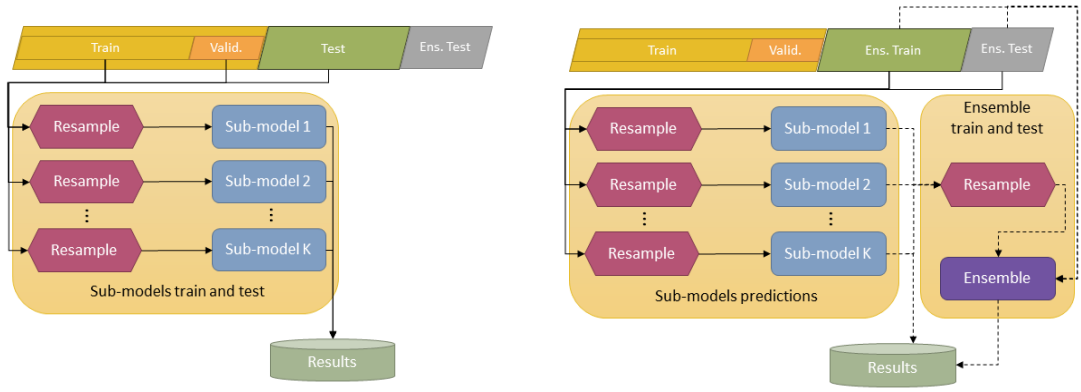
Figure 1. Selected data set time series plot and histogram.

3.2 Train, validation and test splitting

First, the time series were trimmed so the first day in the series were a Monday and the last day were a Sunday. Following that, all the splitting were rounded to the next value that made the resulting data sets sizes divisible by 7 (a whole week). Respecting that condition and the hold-out split (70-30% ratio), the data set was divided in four

groups at the end of the data set (most recent): Train set (105 days), Validation set (42 days), Test set (63 days), Ensemble Test set (28 days).

As shown in the simplified diagram from Figure 2a, during the sub-models training, the Test Set is passed for sub-models performance evaluation. This data set is also used for the ensemble train, therefore, during ensemble, the Test Set will be referred as Ensemble Train Set. Then, the Ensemble Test Set is used to evaluate ensemble performance as shown in Figure 2b.



(a) Data preparation and sub-models training and testing .

(b) Ensemble training and testing.

Figure 2. Proposed method for sub-models and ensemble training and testing.

3.3 Sub-model

Two kinds of sub-models were trained: ARIMA and LSTM, 21 sub-models each.

The ARIMA models were generated by sweeping p, q and d from 0 to 5, 0 to 2 and 0 to 1, respectively. Except from p,q,d = (0,0,0) and other impossible combinations.

The LSTM sub-models were modeled by changing the following parameters within intervals: $input\ sequence = [1, 2, 3, 4, 5]$ and $number\ of\ LSTM\ cells = [8, 16, 32, 64, 128]$. The other parameters were set as fixed: $dropout = 0.3$, $batch\ size = 7$, $learning\ rate = 0.05$, *Adam* optimizer, $Training\ epochs = 100$ and *Mean Squared Error* as loss. Despite the learning rate and epochs fixed settings, both *early stopping* and *reduce learning rate on plateau* were utilized as fitting callbacks.

3.4 Ensemble

The proposed ensemble method is a homogeneous combination via Stacking. Therefore, predictions from a certain number sub-models of the same type will be combined accordingly to a choosing sequence.

The first parameter to be set is the number of sub-models K. In the following experiments, K were set from 2 up to 21 sub-models, yielding a total of 20 ensembles for each sub-model type. The proposed meta learner for the ensemble is Linear Regression (LR). The Ensemble must be fed with the K sub-models forecasts as a sequence, $\widehat{y}$, for Ensemble Training. The LR model also requires the real values, y, in order to calculate the weigh for each element in the input sequence that minimizes the residual sum of squares between the forecasts and the actual values, as stated by Equation 1. The pseudo-code for the LR ensemble method is described in Algorithm 1.

$$\underset{W}{argmin} \sum_{k=2}^{K} \|y - \hat{y}_k * W\|^2 \tag{1}$$

As stated before, the sub-models selection for ensemble were not at random. First, the sub-model with lower Ensemble Train loss is picked. Then, other the sub-models were chosen based on a heuristic that search for minimum loss; maximum Ensemble Train predictions and real values correlations; and minimum Ensemble Train predictions correlations with previous selected sub-models.

Algorithm 1 Linear Regression Ensemble

procedure $W_k(\widehat{y}_1, \widehat{y}_2, ...\widehat{y}_K; y)$ ▷ Ensemble Train Set
  Given: $\widehat{y}_k^t$, y^t, $t = (1...T)$ and $k = (1...K)$
  $\widehat{y} =$ **concatenate**$(\widehat{y}_1, \widehat{y}_2, ..., \widehat{y}_K)$
  $W =$ **initialize**$(W,$ size $= K \times 1)$
  $W =$ **Equation 1** $(\widehat{y}, W, y)$
procedure $\widehat{y}_e(\widehat{y}_1, \widehat{y}_2, ...\widehat{y}_K; W)$ ▷ Ensemble Test Set
  Given: $\widehat{y}_k^t$, $t = (1...T)$ and $k = (1...K)$
  $\widehat{y} =$ **concatenate**$(\widehat{y}_1, \widehat{y}_2, ..., \widehat{y}_K)$
  for $t = 1 : T$ **do**
    $\widehat{y}_e^t = \widehat{y}^t * W$

3.5 Experiments evaluation

The preliminary comparison of sub-models will be based on the loss derived from training. As stated in Section 3.3, the loss function set was Mean Squared Error (MSE) which is obtained by Equation 2. For deeper analysis on the results, normalized forecasts of all sub-models, $\widehat{y}$, and ensemble models, $\widehat{y}_e$, will be evaluated by its Rooted Mean Squared Error ($RMSE$), given by Equation 3, and Mean Absolute Error (MAE), given by Equation 4, compared to the 12-Month moving Average.

Those metrics will provide information on the accuracy of the forecasts and the presence of outliers in the results. Furthermore, since the metrics will be calculated based on normalized data, all metrics can be considered scale-independent enabling comparison between different data sets and also benchmark the proposed method.

$$MSE = \frac{1}{T} * \sum_{t=1}^{T} (y_t - \widehat{y}_t)^2 \tag{2}$$

$$RMSE = \sqrt{MSE} \tag{3}$$

$$MAE = \frac{1}{T} * \sum_{t=1}^{T} |y_t - \hat{y}_t| \tag{4}$$

4 Discussion on Results

Table 1 displays the MSE resulting from ARIMA and LSTM sub-models predictions over Ensemble Train Set. As expected, LSTM out-performs the ARIMA sub-models. The performance of the best sub-models and ensemble for each sub-models type are compared to the 12-Month moving Average by its $RMSE$ and MAE over Ensemble Test Set in Table 2.

Analyzing these results, it is clear that a direct application of ARIMA to forecast the experimented data set is not a good strategy. The best ARIMA sub-model performed worst then the 12-Month moving average. The proposed method for ensemble were successful on improving the base sub-models predictions $RMSE$ in both ARIMA (19.29%) and LSTM (2.77%).

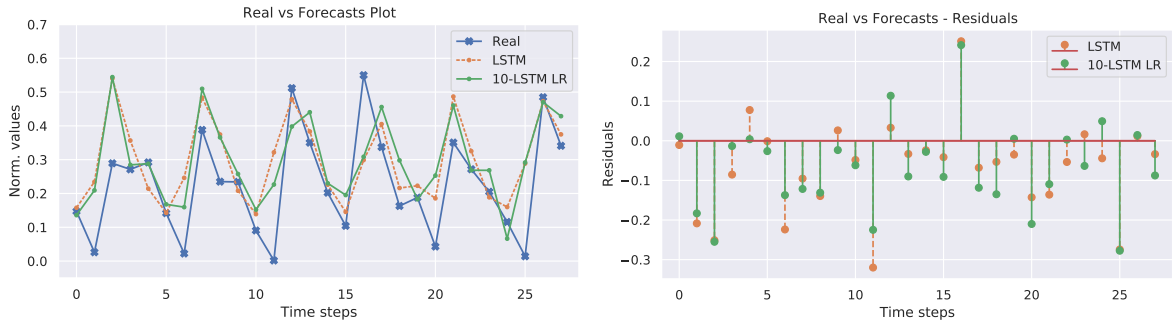
Table 1. Sub-models pre-ensemble MSE statistic describing.

	Count	Mean	Std	Min	25%	50%	75%	Max
ARIMA	21	0.042107	0.018361	0.025362	0.032569	0.036610	0.042107	0.107552
LSTM	21	0.029286	0.008151	0.017876	0.024210	0.027944	0.033120	0.048814

The overall best results derived from the 10 LSTM sub-models ensemble: $RMSE = 0.130002$. Although the corresponding MAE for the ensemble did not improve over base sub-model (-1.66%), the combination of these results indicates that the magnitude of the forecasting errors were smaller but more frequent, as indicated by Figure 3.

Table 2. Predictors ensemble Test set performance compared to 12-Month moving average.

	RMSE	Improv. over 12-Month Avg [%]	MAE	Improv. over 12-Month Avg [%]
12-Month Avg	0.171317	-	0.377965	-
Best ARIMA	0.180468	-5.34	0.391155	-3.49
7-ARIMA-LR	0.14565	14.98	0.330836	12.47
Best LSTM	0.133699	21.96	0.312730	17.26
10-LSTM-LR	0.130002	24.12	0.317909	15.89



(a) Actual values versus best LSTM and ensemble forecasts.

(b) Actual values versus best LSTM and ensemble residuals.

Figure 3. Actual values versus best LSTM and ensemble.

Apart from data preparation described in Section 3.1, the elapsed time from loading the selected data set, training and testing the 21 sub-models were 310 seconds for ARIMA and 238 seconds for LSTM. The average time required for training and testing all 20 ensemble combinations of sub-models predictions, from 2 up to 21 sub-models, were 0.35 seconds for ARIMA and 2 seconds for LSTM.

5 Conclusion

Forecasting is a key activity in supply-chain management (SCM). Farms, industries and retail must be able to predict future demand and resource availability to assist the decision making process and respond quickly in order to succeed in their operations. In the end, the decision about the whether to forecast the demand or just apply the 12-Month moving average should be a cost-benefit analysis on the availability and delivery time of the product; the cost for keeping and purchasing; and the level of perishability of the product.

The proposed method not only improves the accuracy of both benchmark forecasting methods, but it is also quite fast and simple to replicate for as many stock-keeping units (SKU) as the SCM personnel judges to be necessary. Although the LSTM sub-models training required some expertise for tuning its hyper-parameters, once it is defined by a specialist, the process can be repeated just as simple as it was for the ARIMA sub-models. Also, as expected for an ensemble model, the proposed method showed better potential for improving weak forecasters. Therefore, even for poorly tuned LSTM, the method may yield significant improvement.

Despite the success of the ensemble method, it showed potential for improvement hence the field of study is not widely explored. As a continuity of the research, it is possible to test more LSTM settings by including more hidden layers, trying other optimization methods and setting other hyper-parameters. Besides, a Multivariate LSTM implementation combining the target input series with the corresponding item price time series may yield even better results. About the ensemble method, it is possible to test other meta learners, especially machine learning methods, such as Random Forest and Multilayer Perceptron.

Being a decision making tool for assisting SCM, it would be important to test the extrapolation capability of the ensemble method by experimenting the procedure and hyper-parameters on a different store or even a different item time series. Yet, although it is very fast and automatable, if the items and stores could be clustered and

still present satisfactory results, the reduction of the computational infrastructure required would ease the retailers decision to implement the forecasting method for all SKU's.

References

[1] ABRAS, 2019. 19ª avaliação de perdas no varejo brasileiro de super mercados.
[2] Huber, J., Gossmann, A., & Stuckenschmidt, H., 2017. Cluster-based hierarchical demand forecasting for perishable goods. *Expert systems with applications*, vol. 76, pp. 140–151.
[3] Harris, F. W., 1913. How many parts to make at once. *The Magazine of Management*, vol. 10, n. 2, pp. 135–136.
[4] Makridakis, S., Spiliotis, E., & Assimakopoulos, V., 2020. The m4 competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, vol. 36, n. 1, pp. 54–74.
[5] Box, G. E. & Jenkins, G. M., 1970. Time series analysis forecasting and control. Technical report, WISCONSIN UNIV MADISON DEPT OF STATISTICS.
[6] CAMPOS, P. A. C., Clemente, A., & DE CORDEIRO, A. A. L., 2006. Aplicação do modelo arima para previsão do preço do frango inteiro resfriado no grande atacado do estado de são paulo. In *Anais do Congresso Brasileiro de Custos-ABC*.
[7] Da Veiga, C. P., Da Veiga, C. R. P., Catapan, A., Tortato, U., & Da Silva, W. V., 2014. Demand forecasting in food retail: A comparison between the holt-winters and arima models. *WSEAS transactions on business and economics*, vol. 11, n. 1, pp. 608–614.
[8] Hochreiter, S. & Schmidhuber, J., 1997. Long short-term memory. *Neural computation*, vol. 9, n. 8, pp. 1735–1780.
[9] Bousqaoui, H., Achchab, S., & Tikito, K., 2017. Machine learning applications in supply chains: long short-term memory for demand forecasting. In *International Conference of Cloud Computing Technologies and Applications*, pp. 301–317. Springer.
[10] Chen, G., 2016. A gentle tutorial of recurrent neural network with error backpropagation. *arXiv preprint arXiv:1610.02583*, vol. -.
[11] Cheng, Y., Xu, C., Mashima, D., Thing, V. L., & Wu, Y., 2017. Powerlstm: power demand forecasting using long short-term memory neural network. In *International Conference on Advanced Data Mining and Applications*, pp. 727–740. Springer.
[12] Krause, B., Lu, L., Murray, I., & Renals, S., 2016. Multiplicative lstm for sequence modelling. *arXiv preprint arXiv:1609.07959*, vol. -.
[13] Štěpnička, M. & Burda, M., 2017. On the results and observations of the time series forecasting competition cif 2016. In *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1–6. IEEE.
[14] Sun, S., Wei, Y., & Wang, S., 2018. Adaboost-lstm ensemble learning for financial time series forecasting. In *International Conference on Computational Science*, pp. 590–597. Springer.
[15] Choi, J. Y. & Lee, B., 2018. Combining lstm network ensemble via adaptive weighting for improved time series forecasting. *Mathematical Problems in Engineering*, vol. 2018.
[16] Armstrong, J. S., 2001. Combining forecasts. In *Principles of forecasting*, pp. 417–439. Springer.
[17] Krstanovic, S. & Paulheim, H., 2017. Ensembles of recurrent neural networks for robust time series forecasting. In *International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pp. 34–46. Springer.
[18] Dantas, T. M. & Oliveira, F. L. C., 2018. Improving time series forecasting: An approach combining bootstrap aggregation, clusters and exponential smoothing. *International Journal of Forecasting*, vol. 34, n. 4, pp. 748–761.
[19] Rosen, B. E., 1996. Ensemble learning using decorrelated neural networks. *Connection science*, vol. 8, n. 3-4, pp. 373–384.
[20] Feng, J., Lee, D.-K., Fu, C., Tang, J., Sato, Y., Kato, H., Mcgregor, J. L., & Mabuchi, K., 2011. Comparison of four ensemble methods combining regional climate simulations over asia. *Meteorology and Atmospheric Physics*, vol. 111, n. 1-2, pp. 41–53.
[21] Hyndman, R. J. & Koehler, A. B., 2006. Another look at measures of forecast accuracy. *International journal of forecasting*, vol. 22, n. 4, pp. 679–688.
[22] Bandara, K., Bergmeir, C., & Smyl, S., 2017. Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach. *arXiv preprint arXiv:1710.03222*, vol. -.
[23] Favorita, C., 2017. Corporación Favorita Grocery Sales Forecasting. Can you accurately predict sales for a large grocery chain?