Long short-term memory neural networks applied in demand forecast in the retail market

First F. Fukai[1], Second D. Cavalieri[1], Third F. Zanetti[2]

[1]*Dept. of Automation Engineering, Federal Institute of Espírito Santo*
Av. dos Sabiás, 330 - Morada de Laranjeiras, 29166-630, ES/Serra, Brazil
fernandafukai@ifes.edu.br, daniel.cavalieri@ifes.edu.br, fidelis.zanetti@ifes.edu.br

Abstract.

Sales forecasting is vital for the retail industry, supporting strategic decisions and operational planning. Traditional statistical methods, while common in time series analysis, often fail to capture high-dimensional data patterns and non-linear relationships between variables. In this context, Long Short-Term Memory (LSTM) networks provide significant improvements due to their ability to retain information over long periods, making them suitable for dynamic scenarios common in retail. This study examines the effectiveness of three LSTM architectures—Vanilla LSTM, ConvLSTM, and CNN-LSTM Multiscale—in sales forecasting. Boruta algorithm was tested for feature selection and their effects on performance and training time was assessed. The models were evaluated through Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and Mean Absolute Scaled Error (MASE). A comparative analysis highlights the strengths of each LSTM model in time series forecasting. The results indicate that the ConvLSTM architecture outperforms the other models in most metrics in the study case. While feature selection improved training time, it affected performance, increasing error. This study concludes that LSTM-based models effectively handle the complexities of time series data, showcasing the value of LSTM networks in advanced retail analytics.

Keywords: Sales Forecasting, Time-Series, Deep Learning, Long Short-Term Memory

1 Introduction

Sales forecasting is an important aspect of business management, playing a major role in resource allocation, marketing and influences decisions to provide products and services [1]. The capability to predict future trends can optimize inventory and optimize customer experience [2]. However, inaccurate forecasts can lead to over-stock or stock-out [2], increasing inventory costs and decreasing profits. Running out of stock can lead to companies losing competitive advantages and customers' trust, missing out opportunities to maximize sales [3]. Supply chain operations are cost-oriented and retailers need to optimize their operations to carry less financial risks. Therefore, the usage of technological tools and predictive methods is becoming more popular and necessary for retailers.

Generally, time series forecasting techniques fall into the two main categories of statistical and computational intelligence methods [4]. Within statistical techniques, autoregressive integrated moving average (ARIMA), Holt-Winters moving average, and exponential smoothing are presented as classical linear models [4]. However, most real-world time series exhibit non-linear characteristics. To address these challenges, computational intelligence, particularly machine learning, has emerged as a powerful tool capable of manipulating large volumes of data and identifying complex patterns that traditional statistical approaches may overlook. Applications of machine learning techniques in the supply chain domain for time series forecasting have proven effective in many retail case studies [5–7].

The application of deep learning techniques, such as Recurrent Neural Networks (RNNs) , Convolutional Neural Networks (CNNs) and Long Short-Term Memories (LSTMs), has demonstrated promising results in improving the accuracy and performance of demand forecasting models. These techniques have been particularly effective in the context of high-dimensional sales data and projecting it into a high-dimensional space for processing, thus improving the ability to adjust sales curves closer to reality.

In this paper, we investigate three deep learning-based architectures (Vanilla LSTM, ConvLSTM, and LSTM-CNN Multiscale) for accurate prediction of sales forecasting of a pharmaceutical company. The rest of the paper is organized as follows: Section 2 presents the literature. The Research Methodology is explained in Section 3. The

results are discussed in Section 4. Finally, Sections 5 and 6 discuss the results of the research study and provide concluding remarks.

2 Related work

Recurrent Neural Networks (RNNs) are a class of artificial neural networks designed to recognize sequential patterns and predict subsequent events in a series. Unlike traditional Artificial Neural Networks (ANNs), RNNs incorporate feedback loops, enabling connections between nodes and allowing signals to travel in multiple directions [8].RNNs have been widely applied in various domains of the supply chain. For instance, Ard et al. [9] demonstrated the superiority of RNNs in forecasting spare parts, while Rahman et al. [10] highlighted their effectiveness in medium and long-term predictions of electricity consumption. Additionally, Hribar et al. [11] compared RNNs to linear regression and extreme machine learning algorithms for forecasting natural gas demand, incorporating variables such as past temperatures and time markers for holidays and other events. Their findings indicated that RNNs outperformed these alternative methods. However, RNNs are not universally effective in the supply chain due to their short memory and the vanishing gradient problem, which complicates the training process [8].

To address these limitations, Long Short-Term Memory (LSTM) networks were developed [12]. LSTMs are a type of RNN designed to capture the most relevant information from data by distinguishing between short-term and long-term dependencies [8]. This capability allows LSTMs to effectively analyze and forecast over extended periods [13]. LSTMs are particularly adept at producing long-term forecasts due to their reliance on historical data [12]. In a study conducted by El Filali et al.[14], an optimized LSTM network was proposed for demand forecasting in a pharmaceutical Moroccan company. Using the grid search method to automatically select the optimal forecasting model, the LSTM network outperformed traditional methods like ETS and ARIMA, showcasing its strength in capturing nonlinear features in time series data. Abbasimehr and Paki Abbasimehr and Paki[15] proposed a hybrid model combining LSTM and multi-head attention for time series forecasting. Their method demonstrated superior performance over standard time series forecasting techniques and other hybrid models, achieving the best average rank across 16 datasets. Also, Bandara et al.[6] evaluated a real-world e-commerce database from Walmart.com, performing sales forecasting with LSTM models and shown that this method have outperformed the state-of-the-art univariate forecasting techniques.

As mentioned before, machine learning and its advanced branch, deep learning, have revolutionized predictions by creating detailed, layered models that understand how different factors interrelate. In this context, the case studies dedicated to advanced methods in deep learning, and have increasingly highlighted the effectiveness of LSTMs for time series prediction, demonstrating their capability in this field. While LSTM applications have shown promising results, indicating improved forecasting accuracy, there remains potential for further enhancements to better suit retail environments.

3 Methodology

3.1 Dataset

Our experiments utilized the Kaggle public dataset, containing sales and store information for 1,115 chain stores of the pharmaceutical company Rossmann. The dataset spans from January 1, 2013, to July 31, 2015, and is organized into three .csv files: a training set, a test set, and supplementary data about the stores. To address the challenges and accelerate the training process, we extracted a sample dataset containing three semesters of the original dataset. The sample was also enriched with 15 new features derived from the features of the original dataset. Feature selection with the Boruta algorithm was implemented, reducing the original number of features to 13, as detailed in Table 1.

Table 1. Dataset, sampling and features

Dataset	**Start Date**	**End Date**	**Features**	**Data points**
Original Dataset	01-01-2013	31-07-2015	20	844,338
Pre-processed Sample	31-01-2014	31-07-2015	35	478,820
Boruta Sample	31-01-2014	31-07-2015	13	478,820

3.2 Feature selection with Boruta

Developed by Kursa and Rudnicki [16] for use in R, Boruta is an all-relevant feature selection algorithm based on feature importance from Random Forest. Boruta creates "shadow" features by duplicating and shuffling the values of real features to remove any association with the response. A Random Forest model is then trained on the expanded dataset, and feature importance is recorded. Boruta tests if each real feature's importance is significantly higher than the maximum importance of its shadow counterparts. Features consistently more important than their shadows are deemed relevant. This process is repeated to ensure robustness [16]. The aplication of the Boruta algorithm reduced the preprocessed dataset from 35 to 13 features. Results regarding performance and interpretability will be discussed in the next chapter.

3.3 LSTM

The LSTM network, introduced by Hochreiter and Schmidhuber in 1997 [12], is an enhancement of RNNs designed to address the vanishing and exploding gradient problems inherent in traditional RNNs [17]. Each LSTM block contains a memory cell and three gates: an input gate (i_t), a forget gate (f_t), and an output gate (o_t), which manage the flow of information to and from the cell state (c_t). Each gate performs a specific function [18]:

- **Forget gate (f_t)**: Determines which information from the cell state should be discarded, ensuring irrelevant information is removed.
- **Input gate (i_t)**: Decides which new information should be added to the cell state, crucial for updating it with relevant data.
- **Output gate (o_t)**: Regulates the information that is output from the LSTM cell, determining which part of the cell state will be used for predictions.

These gates enable LSTM networks to maintain and update long-term memory, effectively capturing temporal dependencies in sequential data, making these networks particularly powerful for tasks involving time series forecasting, speech recognition, and other applications where understanding the order and context of data points is essential [12, 18].

The internal mechanisms of an LSTM block allow it to propagate error gradients over longer sequences, addressing the limitations of traditional RNNs. The gating mechanisms of the LSTM architecture provide a robust framework for learning and maintaining long-term dependencies, making it an indispensable tool in the field of sequential data modeling.

3.4 Vanilla LSTM

The Vanilla LSTM comprises a single LSTM layer followed by a Dense output layer. This variant is the most commonly used model in the literature due to its straightforward structure [19]. The LSTM layer processes the input sequence time step by time step, creating an internal state representation that serves as the learned context for making predictions. This design leverages the inherent sequence support of LSTM networks, allowing them to effectively model temporal dependencies and provide accurate predictions [19]. The Dense output layer translates the learned context into the final prediction, making it suitable for time series forecasting modelling. The Vanilla LSTM Model used comprises a single LSTM layer with 50 units, followed by two Dense layers with 100 units each.

3.5 CNN-LSTM Multiscale

The combination of Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) models, known as CNN-LSTM, has been widely explored and applied in various domains for time series forecasting and prediction tasks [20]. The multi-scale CNN-LSTM model, time-series features were constructed in the form of continuous feature maps as input. CNN is used to cascade the shallowerand deeper feature in different scales. The feature vectors of different scales are fused as the input of LSTMnetwork and short-term load forecasting based on LSTM network. Multiscale Model used in this paper includes two Conv1D layers with filters of 60 and kernel sizes of 7 and 30, respectively, followed by AvgPool1D and LSTM layers. Dense and Dropout layers are added to enhance model performance

3.6 ConvLSTM

The architecture of ConvLSTM involves extending the fully connected LSTM to incorporate convolutional structures in both the input-to-state and state-to-state transitions [21]. In our case, one-dimensional convolutional layers (Conv1D) apply filters to input data, extracting relevant features through convolutions along the temporal dimension, incorporating into LSTM network. Gates receive the generated resources as new appetizer. They are a reduced representation that captures only the most relevant so that the efficiency of the cell state update mechanism is improved. This approach aims to improve the accuracy of predicting long-term issues based on more efficient processing of sequences [21]. The ConvLSTM Model assessed is composed by a Conv1D layer with 64 filters and a kernel size of 3, followed by an LSTM layer with 50 units, a Flatten layer, and Dense and Dropout layers for better prediction accuracy.

3.7 Train-Test Split

The data was be divided into a 70% training set, 20% test and 10% validation. The division ensures chronological integrity. This method avoids anticipation bias, where future information can inadvertently influence the model training, and the models can be trained on previous data, tested on more recent data, and finally validated on the most recent data, respecting the natural order of the time series.

3.8 Parameter Settings

Table 4 describes the parameters used in the model. Additional functions include EarlyStopping (patience=10), which monitors the loss on the validation dataset and stops training if the metric does not improve for a predefined number of epochs. Lookback, was set to 7, implying that the models utilized data from the previous 7 time points to make a forecast.

Table 2. Parameters

Parameter	**Value**
Optimizer	adam
Loss Function	Mean Square Error (MSE)
Learning Rate	0.001
Batch Size	128
Training Epochs	55

3.9 Evaluation metrics

To evaluate the prediction accuracy, the MAE (Mean Absolute Error), RMSE (Root Mean Square Error) and MASE (Mean Absolute Scaled rror) are computed against the testing data, which are calculated as eq. 1, eq. 2 and eq, 3, respectively:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (1) \qquad \text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \quad (2) \qquad \text{MASE} = \frac{\text{MAE}}{\frac{1}{n-1}\sum_{i=2}^{n}|y_i - y_{i-1}|} \quad (3)$$

where y_i are the observed values, $\hat{y}_i$ are the predicted values, and n is the number of observations.

3.10 Results

Table 3 presents a comparison of the performance metrics—Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Scaled Error (MASE) — for the three analysed models, with and without

feature selection using the Boruta method. The models evaluated include a baseline average-based model, Vanilla LSTM, ConvLSTM, and Multiscale CNN-LSTM.

Table 3. Performance Metrics Comparison for Different Models With and Without Feature Selection Using the Boruta Method.

	Sample			Sample Boruta		
Model	MAE	RMSE	MASE	MAE	RMSE	MASE
Average Model	0,650	0,88	0,82	0,650	0,88	0,82
Vanilla LSTM	0,385	0,564	0,491	0,484	0,705	0,618
ConvLSTM	**0,351**	**0,516**	**0,447**	**0,388**	**0,560**	**0,495**
CNN-LSTM Multiescale	0,450	0,591	0,573	0,485	0,640	0,619

The baseline average-based model remains consistent with MAE, RMSE, and MASE values of 0.650, 0.88, and 0.82, showing that feature selection did not impact its performance. Without feature selection, the Vanilla LSTM model achieves MAE, RMSE, and MASE values of 0.385, 0.564, and 0.491, respectively. However, with feature selection (Boruta), the performance slightly decreases to 0.484 (MAE), 0.705 (RMSE), and 0.618 (MASE). This suggests that feature selection might have removed some relevant features for this model. The ConvLSTM model performs the best without feature selection with MAE, RMSE, and MASE values of 0.351, 0.516, and 0.447, respectively. With feature selection, the performance slightly decreases to 0.388 (MAE), 0.560 (RMSE), and 0.495 (MASE). Despite this, the ConvLSTM model still outperforms the Vanilla LSTM and CNN-LSTM models with and without feature selection. Without feature selection, the CNN-LSTM Multiscale model shows MAE, RMSE, and MASE values of 0.450, 0.591, and 0.573, respectively. With feature selection, these metrics change to 0.485 (MAE), 0.640 (RMSE), and 0.619 (MASE). This indicates a decrease in performance similar to the Vanilla LSTM model when feature selection is applied.

Table 4 shows the training times for different neural network models using a dataset processed with the Boruta method for dimensionality reduction. It compares three models: Vanilla LSTM, ConvLSTM, and Multiscale CNN-LSTM, with training times reported for both Boruta-processed and unprocessed samples. The table also indicates the training epoch at which each model stopped, either due to early stopping or meeting a performance criterion.

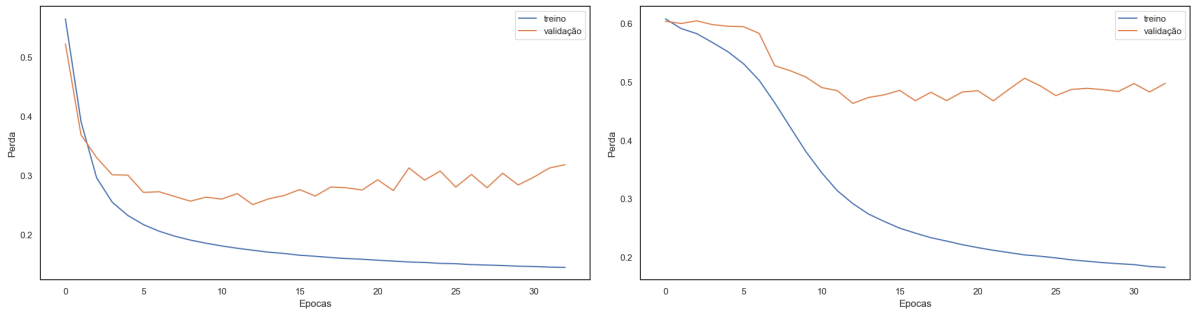
Table 4. Training time for the model on the dataset with dimensionality reduction, indicating the stopped epoch

Model	**Boruta Training Time**	**Training Time**	**Epoch**
Vanilla LSTM	8m 6.3s	32m 34.5s	42
ConvLSTM	16m 20.7s	18m 22.6s	50
CNN-LSTM Multiscale	10m 34.0s	117m 38.2s	50

The training time for the Vanilla LSTM model with Boruta feature selection is significantly shorter (8m 6.3s) compared to the full dataset (32m 34.5s). The model stops at 42 epochs, indicating a more efficient training process with feature selection. The ConvLSTM model's training time with Boruta feature selection is 16m 20.7s, compared to 18m 22.6s without feature selection. The model stops at 50 epochs in both cases, suggesting that while feature selection reduces training time, it does not affect the number of epochs needed for convergence. The training time for the CNN-LSTM Multiscale model with Boruta feature selection is 10m 34.0s, significantly shorter than the 117m 38.2s without feature selection. The model stops at 50 epochs in both cases, similar to the ConvLSTM model.

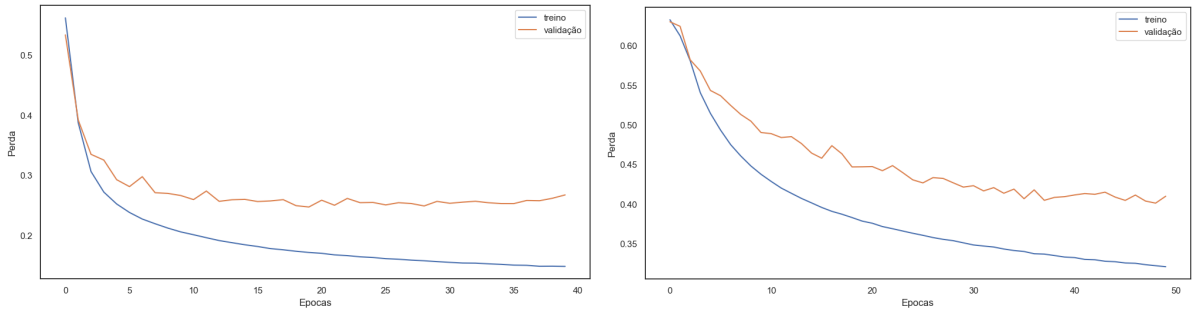
The following figures illustrate the behavior of the loss functions over the training epochs for LSTM models. These models were trained with both standard sampling and sampling with dimensionality reduction using the Boruta method. The Vanilla LSTM models, as observed in Figures 1a and 1b, show a rapid and stable decline in training loss with the increase in epochs, indicating good learning. The validation loss, although higher than the training loss, also shows a decreasing trend, suggesting that the models are generalizing adequately.

For the Multiscale CNN-LSTM models depicted in Figures 2a and 2b, the loss curves indicate effective learning, with the validation loss stabilizing after the initial epochs, which is indicative of no significant overfitting.



(a) Vanilla LSTM with Standard Sampling (b) Vanilla LSTM with Boruta Sampling

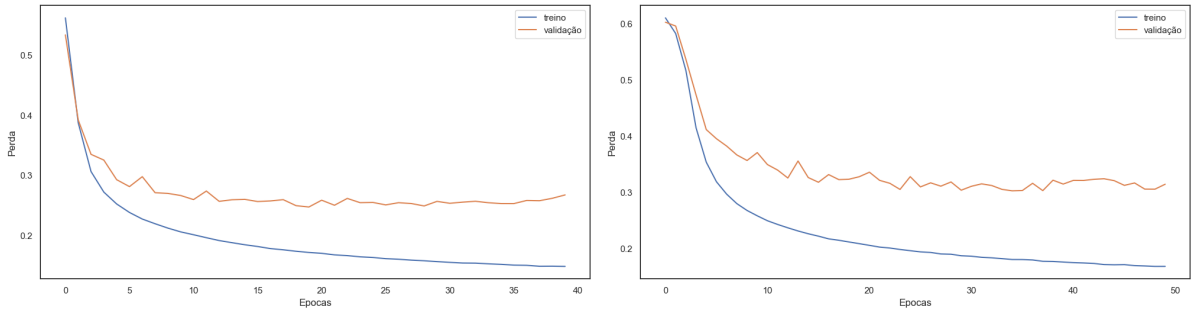
Figure 1. Loss Curves for Vanilla LSTM Models



(a) Multiscale CNN-LSTM with Standard Sampling (b) Multiscale CNN-LSTM with Boruta Sampling

Figure 2. Loss Curves for Multiscale CNN-LSTM Models

Similarly, Figures 3a and 3b show the loss graphs for the ConvLSTM models, which behave similarly to the Vanilla LSTM models, with the loss curves demonstrating consistent learning throughout the training and validation epochs.



(a) ConvLSTM with Standard Sampling (b) ConvLSTM with Boruta Sampling

Figure 3. Loss Curves for ConvLSTM Models

4 Conclusions

In this study, three Long-Short Term Memory (LSTM) network models were implemented to predict overall sales of a company. Variations of LSTM networks were tested and their performance was assessed using metrics on a real-world retail dataset. Among the models, the ConvLSTM outperformed the others in all evalution metrics. We concluded that LSTM networks are effective for accurate prediction. However, ConvLSTM produced the best results on our dataset, while the Vanilla LSTM model demonstrated efficiency in terms of computational complexity during training. Additionally, using the Boruta feature selection algorithm reduced training time and achieved acceptable results. For future work, we plan to optimize LSTM hyperparameters using GridSearch, introduce other machine learning benchmarks and evaluate performance on the complete dataset.

Acknowledgements. The authors would like to thank the editor and reviewers. This research has been supported by the Federal Institure of Espírito Santo (IFES).

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

[1] R. J. Hyndman and G. Athanasopoulos. *Forecasting: principles and practice*. OTexts, Melbourne, Australia. Accessed on 10 July 2024, 2018.
[2] S. Levy and M. Sobolev. Improving sales forecast accuracy with machine learning. *Journal of Business Analytics*, vol. 7, n. 2, pp. 123–135, 2019.
[3] A. Beutel and A. Pasricha. Inventory optimization: Reducing costs and maximizing roi. *Journal of Supply Chain Management*, vol. 48, n. 1, pp. 45–58, 2012.
[4] M. Khashei and M. Bijari. A novel hybridization of artificial neural networks and arima models for time series forecasting. *Applied Soft Computing*, vol. 11, n. 2, pp. 2664–2675, 2011.
[5] K. Chandriah and R. Naraganahalli. Rnn / lstm with modified adam optimizer in deep learning approach for automobile spare parts demand forecasting. *Multimedia Tools and Applications*, vol. 80, 2021.
[6] K. Bandara, P. Shi, C. Bergmeir, H. Hewamalage, Q. Tran, and B. Seaman. *Sales Demand Forecast in E-commerce Using a Long Short-Term Memory Neural Network Methodology*, pp. 462–474, 2019.
[7] A. Giri and J. B. Smith. Improving sales forecasting with deep learning techniques. *Expert Systems with Applications*, vol. 189, pp. 113452, 2022.
[8] J. Dabounou. Deep learning: Les réseaux de neurones récurrents (rnn). *Data Value Consulting*. Accessed Sep. 20, 2021, 2021.
[9] A. K. Ard, A. Bekrar, A. A. E. Cadi, and Y. Sallez. Artificial intelligence for forecasting in supply chain management: a case study of white sugar consumption rate in thailand. *IFAC-PapersOnLine*, vol. 52, n. 13, pp. 725–730, 2019.
[10] A. Rahman, V. Srikumar, and A. D. Smith. Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks. *Applied Energy*, vol. 212, pp. 372–385, 2018.
[11] R. Hribar, P. Potočnik, J. Šilc, and G. Papa. A comparison of models for forecasting the residential natural gas demand of an urban area. *Energy*, vol. 167, pp. 511–522, 2019.
[12] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, vol. 9, n. 8, pp. 1735–1780, 1997.
[13] H. Abbasimehr, M. Shabani, and M. Yousefi. An optimized model using lstm network for demand forecasting. *Computers and Industrial Engineering*, vol. 143, 2020.
[14] A. El Filali, E. H. Ben Lahmer, S. El Filali, M. Kasbouya, M. A. Ajouary, and S. Akantous. A deep learning model using an optimized lstm network for demand forecasting. *Journal of Information Technology and Modelling Laboratory*, 2021.
[15] H. Abbasimehr and R. Paki. Improving time series forecasting using lstm and attention models. *Springer Nature*, 2021.
[16] M. B. Kursa and W. R. Rudnicki. Feature selection with the boruta package. *Journal of Statistical Software*, vol. 36, n. 11, pp. 1–13, 2010.
[17] C. Olah. Understanding lstm networks. *Blog*. Available at https://colah.github.io/posts/2015-08-Understanding-LSTMs/, 2015.
[18] A. Graves. *Speech Recognition with Deep Recurrent Neural Networks*. IEEE, 2013.
[19] Y. Wu, M. Yuan, S. Dong, L. Lin, and Y. Liu. Remaining useful life estimation of engineered systems using vanilla lstm neural networks. *Neurocomputing*, vol. 275, pp. 167–179, 2018.
[20] X. Guo, Q. Zhao, D. Zheng, Y. Ning, and Y. Gao. A short-term load forecasting model of multi-scale cnn-lstm hybrid neural network considering the real-time electricity price. *Energy Reports*, vol. 6, pp. 1046–1053, 2020.
[21] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems*, volume 2015-January, pp. 802–810, 2015.