# Predicting Stock prices in Latin America using Associative Deep Neural Networks

Juan Fernando Gallego Rojas
jgalle47@eafit.edu.co

**Director**
Paula María Almonacid Hurtado
palmona1@eafit.edu.co

*Maestría en Ciencias de los Datos y Analítica,*
*Universidad EAFIT,*
*Medellín, Colombia.*

**Abstract**

The stock market is a critical sector of the global economy, and predicting stock prices is of great interest to investors and companies. However, the movements of the market are volatile, non-linear, and complicated. This topic has attracted the attention of researchers, who have proposed formal models that demonstrate accurate predictions can be made with appropriate variables and techniques. Deep learning algorithms are often used for this purpose due to their superior accuracy in time series-based and complex pattern analysis. This paper proposes to predict the opening, closing, highest, and lowest stock prices of select Latin American market indexes using associative deep neural networks that can simultaneously predict related values based on the Long Short-Term Memory (LSTM) technique, known for its high accuracy in this area. As well as using classic econometric methods for the analysis of time series such as ARIMA models. The proposed model achieved a good performance in terms of prediction, which in turn allows finding interesting trading opportunities for investors. The results of the models were measured using the average RMSE of the predicted prices metric and compared with those obtained using a naive model.

***Keywords***— Stock market, Latin American market indexes, Machine learning, Deep learning, Associated network, Deep recurrent neural network, Multi-output, Multi-input.

# 1 Project description

## 1.1 Problem Statement

The stock market is an important driver of the global economy [1], in which numerous investors and companies interact with a wide range of financing and investment products offered by this market, according to their needs concerning liquidity, return and risk expected.

Particularly, talking about investments in stocks. It has been of great interest to investors to have techniques that help them predict stock prices [1], [2] to support their financial decisions in order to increase their expected profits and reduce their risk exposure. However, stock price movements are volatile, non-linear, and complicated [3], [4] and therefore, challenging to predict.

In addition, this paper can contribute to the need to have this type of works that predict prices for the Latin American financial markets, since these have not been carried out or there are very few works on the subject for this market. Additionally, the inputs or predictions of this work can be used to solve problems related to risk and portfolio Management, as well as financial engineering.

Finally, the importance of simultaneously obtaining the opening, closing, high and low price predictions (what is our proposal) is that the trader gets a complete picture of how the market is expected to evolve. This allows him to make informed decisions and consider all aspects of price movement, additionally, he can identify potential trading opportunities. He can also use this information to determine strategic entry and exit points, as well as set stop-loss and take-profit levels. This in turn provides a solid foundation for risk management. The trader can assess the expected price ranges and adjust his strategy accordingly. For example, if the prediction indicates a narrow range between the high and low price, the trader can choose to reduce his position or look for lower risk strategies. So these predictions allow the trader to refine and optimize his trading strategies. He can as well tailor his approach to specific market movements, such as taking advantage of expected volatility or following certain price patterns. In addition, saving time by not having to perform separate analyzes for each price. This allows the trader to focus on executing and managing his trades instead of spending a lot of time analyzing prices.

## 1.2 Justification

Stock market performance prediction has always been an important and prominent research topic. [5], [6], due to its strong potential to generate financial profit and to help investors to make prudent financial decisions in the stock market minimizing their risk exposure.

Before the era of machine learning, there have been traditional statistical time series techniques, which have been very useful for modeling stock prices, among which stand out, the auto-regressive conditional heteroscedastic (ARCH) methods [7] and auto-regressive moving average (ARMA) [8] or auto-regressive integrated moving average (ARIMA). However, these models assume that price movements follow a behavior under a linear mathematical scheme, while the stock market evolves dynamically under a non-linear scheme, which reduces the ability of these models to predict stock prices [9].

In addition to the above, stock prices tend to have a lot of noise due to the wide range of factors that can affect them [10], [11], [12], [13],[14], [15], such as socio-ecological and sentimental, which makes their modeling very complex using traditional statistical methods [16]

Currently, taking advantage of the development of computational power, researchers have been able to explore, combine and develop advanced techniques to address this task. It can be seen in formal propositions exhibiting that with appropriate variables and techniques, accurate models can be designed, in which, it has been shown that machine learning techniques have obtained more accurate results than traditional techniques, highlighting the LSTM with the best performance among these [17]. Machine learning methods are suited to address this prediction problem because they are capable to capture complex and non-linear patterns in stock prices [9].

The research team is composed of Director Paula Maria Almonacid, an economist from Universidad EAFIT, who holds a Master of Science in Economics from the Montreal University and a Ph.D. in Statistics from Universidad Nacional. She has interests in financial econometrics, Bayesian statistics, machine learning, deep learning, and financial markets. The team also includes Student Juan Fernando Gallego, who has a degree in finance with an emphasis on corporate finance and financial risk management from Universidad EAFIT and is a Master's candidate in Data Science and Analytics, expected to graduate in July 2023 from Universidad EAFIT. Juan's interests lie in financial modeling, financial markets, and analytics. Director Almonacid, with her preparation, knowledge, and experience, has not only shown interest in the topic of this article but has also worked on related issues. This interest is shared by Juan, who will support the research with his knowledge in finance and the insights he has gained from his master's program.

# 2 Objectives

## 2.1 General objective

This paper proposes to predict the opening, closing, highest, and lowest stock prices of select Latin American market indexes for the following day using associative deep neural networks.

## 2.2 Specific objectives

- Apply and evaluate the performance of associative deep neural networks based on LSTM in order to predict some selected Latin American market indexes on the next day.
- Examine the descriptive characteristics or components of time series data on the prices and yields of stock market assets from underexplored markets, specifically in Latin America.
- Contrast the components of the time series of yields of the stock market under study with the stylized facts found in the literature.
- Find profit opportunities according to the patterns found in the series and determine the type of efficiency of the studied market.

The rest of this paper is organized as follows. Section 3 discusses the state of the art about stock prices, investments, and Latin American market indices. Section 4 presents the theoretical framework about the model to be implemented, while Section 5 exposes the data and the methodology to be used in this project. Section 6 details the process carried out in this paper, describing data, transformations used and process of fitting models. Section 7 presents the results obtained. Section 8 shares conclusions of this work. Section 9 outlines the expected outcomes of this paper. Section 10 exposes the data management plan and Section 11 explains the purpose of this work and who will benefit from it.

# 3 State of art

The stock exchange is a marketplace that enables transactions of financial instruments or securities, such as aforementioned stocks, between interested parties. In this environment, there are a series of stock markets, which represent securities that are traded in a specific region or country, where it is common to find market indexes, which follow the performance of a certain group or section of financial instruments of the stock market and is often used by investors to monitor market movements and as a basis for selection of financial assets in which to invest.

The stock price indicates the current market value of a company listed on the stock market. This price represents how much the stocks are trading as agreed by buyers and sellers who trade this financial asset. These transactions between buyers and sellers occur when they agree on a price for the exchange operation. And the fact that both parties agree is due to the expectations or needs that these actors have, on the basis of which they define their investment strategy. In order to give clarity on these transactions in the stock market, some usual examples are mentioned: A company seeking to finance itself lists on the stock exchange hoping to sell its stocks and thus get the money that needs for its business and on the other hand, investors will be interested in buying these stocks if they expect the company to grow and increase its value, which will increase the value of the invested capital bringing profits for investors. Another case is when investors either expect the market to go down; which means that the price of the stock of interest will fall and therefore, they define a current position of sell and in the future, they close this position with buy or they expect the market to go up; which indicates that the price of the stock of interest will rise and therefore, they establish a position in the present of buy and in the future they will close it with a sell. In both cases, investors earn on the price difference as long as their expectation is true. And also happens that in the face of an urgency or rumor an investor seeks to sell his stocks for a different price than one that is currently being negotiated and in another part there is an agent seeing an investment opportunity in this situation. Faced with an uncertain future and, as already mentioned, a wide range of factors that can affect stock prices, different expectations and needs results in interactions between agents in the stock market, leading to variations in stock prices.

In a working period of trading on listed stocks on the stock exchange, the following four prices occur: the opening price, which is the price of the first trade when the stock exchange opens. The highest price and the lowest price, which refer to

the top and bottom prices obtained in the operations that happens at that time. And finally, the closing price, which is the final price at which a given stock is traded during that session. The latter is usually different from the opening price of the following period, due to outside working hours operations that change its perceived valuation by the agents. It should be noted that the number of stocks traded in a given period of time is the trading volume. According to the description given about these prices, their relationship is appreciated in stock market transactions in a period that in summary, begins with the opening price, followed by the variations of this period from that entry opening value, resulting in the highest and lowest price, and ending these variations with the closing price. These four values are the ones that this work seeks to predict simultaneously, taking advantage of the relationship between them. We propose to work on the prediction of stock prices of the Latin American market indexes, which have not only adequate history in order to have good information available as input for the model to be implemented, but also because, to my knowledge, there are no works carried out like the one proposed in this article with data from Latin America and therefore could represent a contribution.

There is a lot of literature on stock price prediction, among the recent ones are works where authors emphasize the usual machine learning techniques applied, such as stock price prediction of India's major automotive company Tata Motors using four machine learning learning models, getting that the Prophet model with logistic regression as the best [18]. Predicting high variations in the year 2020 using LSTM with adam optimizer which was able to get adequate accuracy, better than traditional techniques [3]. Prediction of Open, High, Low and Closing stock prices in Indonesian exchange employing an LSTM, getting a 94.57% of accuracy [19]. LSTM model is again seen in Microsoft's stock prices prediction, obtaining better RSME than others machine learning algorithms [20]; and related to SET100 index stocks prices prediction, testing it through trading operations strategies achieving higher return than others strategies [13]. Opening and closing stock prices prediction of 42 firms listed in Istanbul Stock Exchange National 100 Index (ISE-100) proposing various machine learning methods and evaluating them by RMSE, MSR and R-squared, resulting that Multilayer Perceptrons and LSTM was better [4].

Also, some investigations highlight the data which they are working with using machine learning techniques over multi-source data or data transformations, like Hybrid deep neural network architecture with CNN and LSTM, using multi-source information-fusion to integrate stock-related information from six (6) heterogeneous sources from the Ghana Stock Exchange (GSE) and obtaining good prediction accuracy over this dataset compared with other single dataset with individual data sources [11]. Prediction over four stock market groups from Tehran stock exchange working with ten technical indicators, calculating these as continuous data and converting them to binary data, using nine machine learning models, resulting that RNN and LSTM outperform other prediction models with considerable difference for continuous data [15].

Aditionally, there are studies about combinations, transformations or hybrid machine learning models, including two models phases, first a Fast Recurrent Neural Networks (Fast RNNs) used for stock price predictions for the first time and second a hybrid deep learning model developed by utilising the best features of FastRNNs, Convolutional Neural Networks, and Bi-Directional Long Short Term Memory models, getting better results by RMSE and computation time for real time predictions than auto Regressive Integrated Moving Average, FBProphet, LSTM, and other proposed hybrid models [21]. Hybrid modelling technique based on several machine learning and deep learning models used to stock price prediction of the Reliance Industries Limited, getting LSTM-based univariate model as the most accurate [5]. LSTM network based on Pearson's correlation coefficient and a Bayesian-optimized LightGBM hybrid model against LSTM-BO-XGBoost hybrid model, the LSTM-LightGBM hybrid model, the LSTM-XGBoost hybrid model, the single LSTM network model and the RNN network model, which was able to obtain a better generalization ability to predict the fluctuations of stock prices [22]. Combining CNN architecture that stands for feature extraction of data, with a LSTM for predicting and using a Kalman Filter to improve the model accuracy evaluated for Apple Inc. and SP 500 index [23]. Stock prediction over 400 stocks in China employing a combination of Extreme learning machine (ELM)-and discrete wavelet transform (DWT)-based denoising to denoise stock data to eliminate the influences of short-term random events on the continuous trend, getting better performance than others 12 machine learning models [24]. Hybrid deep neural network model (HDNNM) consists of two stages, first classification stage uses CNN to structure a three-category classification model to predict if stock is up, flat or down, later this is used as input to regression models based on CNN-long short-term memory (CNN-LSTM), then evaluating it by four indicators showing better performance than others [25]. Associated deep recurrent neural network model based on LSTM, which returns multiple outputs with multiple inputs, used to predict at the same time the opening, lowest, and highest prices of a stock and then comparing it against LSTM and deep RNN, getting superior prediction accuracy than these, over 95% [2]. And another hybrid modeling approach implements eight regression models and four deep learning-based

regression models using LSTM, highlighting an LSTM-based univariate model as the most accurate.

From the previous investigation, it can be seen that machine learning techniques get precise results in stock price prediction, of which; LSTM networks have remarkable performance. In addition, deep machine learning techniques can capture valuable patterns and relationships in the data reflected as more accurate predictions especially using data sets rich in the relevant information. And finally, the structuring of these advanced techniques enhances their relationships and results, reaching accurate predictions. Since the objective of this paper is to predict four related stock prices at the time, it will be taken as the basis of the author's work [2] in which they developed an associated neural networks model based on LSTM deep recurrent networks capable to predict three values simultaneously: the opening, the lowest and the highest prices of a stock, with an accurate results. It also should be noted that in this research there are a lack of works that simultaneously make predictions of financial values, which may be a gap in the literature that our work tries to reduce.

# 4 Theoretical framework

The approach that is addressed in this study in order to obtain an accurate prediction of the prices of the selected Latin American indices, taking into account their own characteristics, consists of the implementation of deep learning methods, specifically LSTM networks. Next, the fundamental characteristics of the method and the particular architecture that is implemented in this study will be described

The base component is the Long Short-Term Memory (LSTM) networks, which are a type of recurrent neural networks, able to learn order dependence from sequential information.

The structure of a unit of the LSTM neural networks is composed of an input gate, forget gate and output gate, with which the unit operates by looping through the sequence of information to be processed and determining how the information flows through it and thus defining the state of the neuron (memory) in each time-step and how its state is updated. Their structure is shown in figure 1.

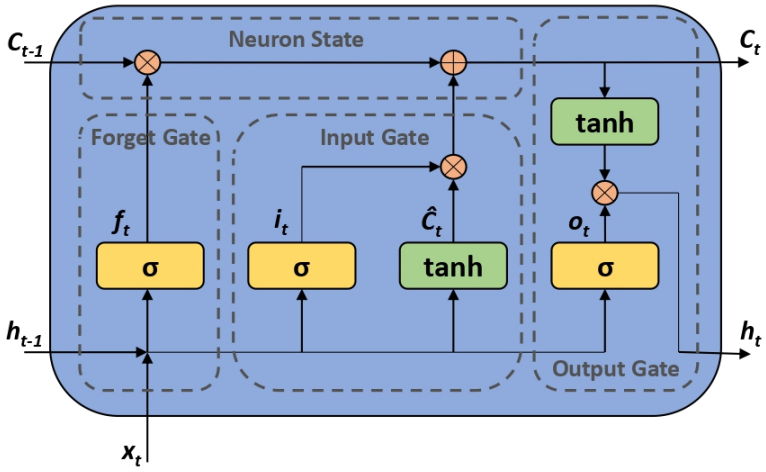


Figure 1: LSTM unit structure. Source: own elaboration.

The forget gate decides what information should be discarded from the current neuron state. To do this, this forget layer has a sigmoid activation function ($\sigma$) that, taking into account the output information of the previous neuron ($h_{t-1}$) and the input information of the current neuron ($x_t$), returns real values between interval 0 and 1 for each value of previous neuron state ($C_{t-1}$), directly proportional to its adjustment with this information or to the degree of conserving it.

The probability of forgetting is given by the following equation 1:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{1}$$

Where:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2}$$

The input gate decides how much new information is added to the neuron state. For this, based on the output information of the previous neuron ($h_{t-1}$) and the input information of the current neuron ($x_t$), this input layer first computes a sigmoid activation function ($\sigma$) to establish what information must be updated and second, another layer computes an activation function tanh that produces candidate vectors ($\hat{C}_t$) for the new neuron state. These calculations are presented in the following equations 3, 4:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{3}$$

$$\hat{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{4}$$

Where:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{5}$$

Based on the above operations, the current neuron state is computed as follows 6:

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t \tag{6}$$

Forgetting the parts of the previous neuron state and adding the number of the new candidates that have been decided to update in the current state.

The last output gate decides what to return based on the input information and the neuron memory. To get this, it computes a sigmoid activation function based on the output information of the previous neuron ($h_{t-1}$) and the input information of the current neuron ($x_t$) to decide which parts of the current neuron state and control neuron state to filter on the result. These operations are shown in the following equations 7, 8:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{7}$$

$$h_t = o_t * \tanh(C_t) \tag{8}$$

With this base component, the LSTM-based deep recurrent neural networks (DRNN) are built to increase the model's expected expressive power [2], which in essence repeats the LSTM unit loop operation process several times, resulting in many LSTM units that conform to a deep network. These deep recurrent neural networks can be seen in the following figure 2:

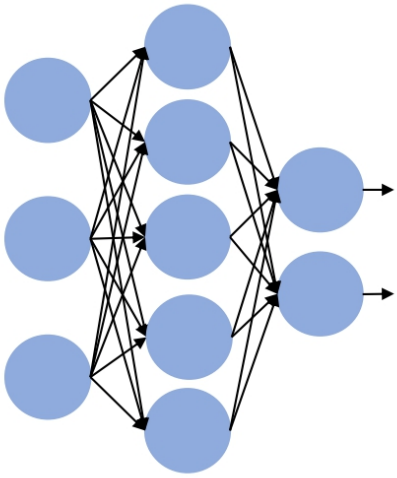

Figure 2: Deep recurrent neural network structure. Source: reproduced from [2].

And, in addition, the Dropout method is considered, which is a regularization method that temporarily randomly deactivates neurons in each iteration of the model training process, getting that the input values are being trained by variations of the deep neural networks with different compositions of neurons. So that, it generalizes better and therefore prevents neural networks from Overfitting [26]. The illustration of figure 3 of a deep recurrent neural networks with Dropout is shown below:

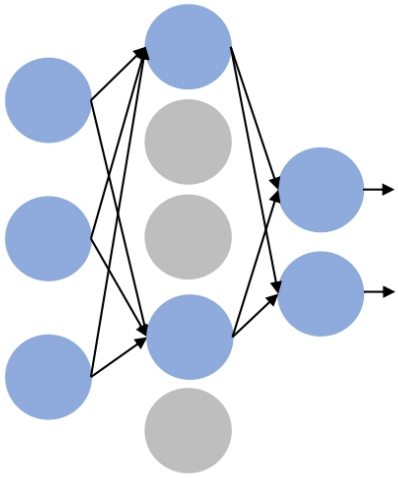

Figure 3: Deep recurrent neural networks with Dropout structure. Source: reproduced from [2].

Putting all together, the schema with proposed deep recurrent neural networks (DRNN) based on LSTM considering the mentioned regularization method is shown below in figure 4:

In which input values are processed in the training phase by a neural network composed of multiple LSTM units and during this process, the Dropout method is applied to avoid the Overfitting problem.

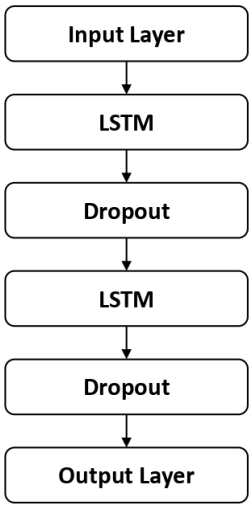


Figure 4: Deep recurrent neural networks (DRNN) schema based on LSTM. Source: reproduced from [2].

Finally, based on this DRNN, the structure of the associative neural networks model to be implemented, capable of predicting four multiple values, is built. This is illustrated in the following figure 5:

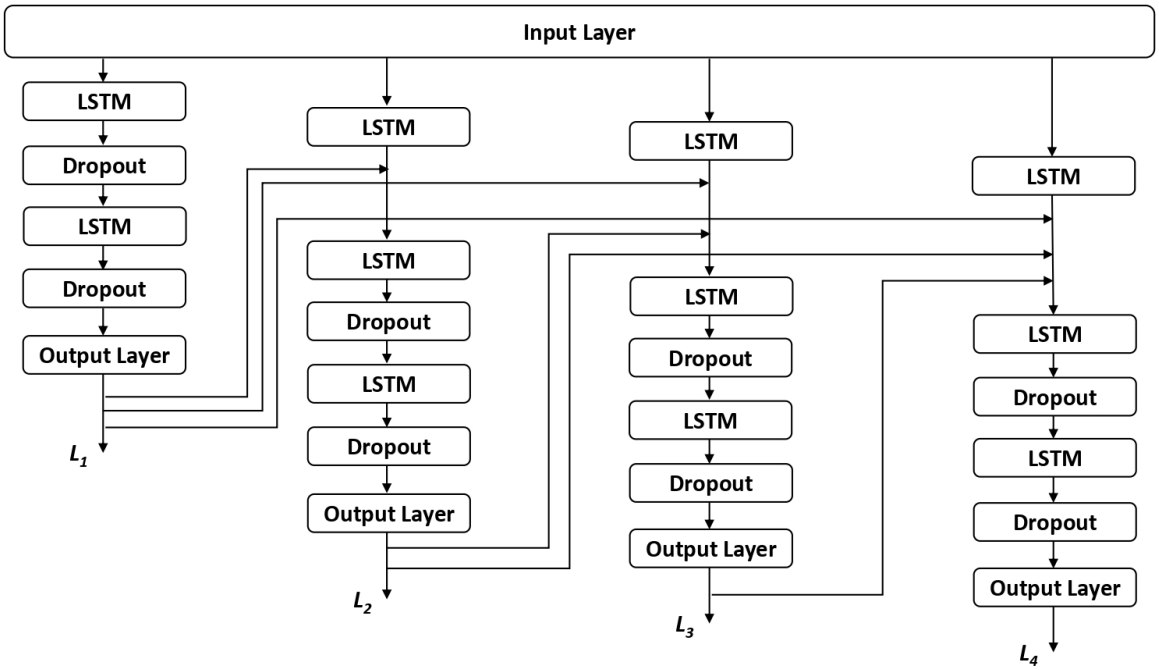


Figure 5: Associative model structure based on DRNN. Source: reproduced from [2].

In which, each DRNN schema predicts one value that will be used as additional input values to predict the following ones in the subsequent DRNN schemas interconnected in the associative neural networks model structure.

# 5 Methodology

This section shares steps carried out for development of this work, beginning with collection, analysis and pre-processing of the data. Then, the modeling process in which, based on descriptive statistics, the relationships of variables and possible transformations to be applied to the data were analyzed. These steps contribute to achievement of specific objectives related to description and contrast of components of the time series studied. In addition, the naive model was implemented as a reference point, then the ARIMA model was implemented with its due process to complement exploration and characterization of each of the series studied (it also contributes to objectives mentioned in previous steps) and finally we carried out the proposed associative deep neural networks model to predict the four proposed prices simultaneously, adjusting it with the hyperparameter tuning process on transformed data. Afterwards, the performance evaluation of the models used and the respective conclusions were made. These last steps contribute to the achievement of the rest of the specific objectives and the general objective related to the application, evaluation of the model and to examine possible opportunities to support investment strategies according to their performance. And finally, technical specifications of the software and hardware used are mentioned.

## 5.1 Data Collection

The first step in this study involved collecting data on select Latin American market indexes. We retrieved data from Bloomberg.

### 5.1.1 Dataset

There are 4 data sets of the indices that integrate the MILA: COLCAP, IPSA, MEXBOL, IGBVL. However, since May 2015 prices of the IGBVL index stopped varying, and for that time this index had a change in weighting by capitalization, which possibly affected information captured by Bloomberg, as can be seen in line graph 6. Taking this argument into account, we decided not to include the IGBVL index in the analysis (the lack of uniformity with the other indices), therefore we will work with the indices: COLCAP, IPSA and MEXBOL. For each one, information is available for the trading days from January 1, 2010 to August 29, 2022 of the opening, closing, highest, lowest prices and the volume in dollar units. And as additional variables, indicator of volatility market; VIX and 10-year and 30-year US Treasury bond rates, over the time range mentioned. These eight technical parameters will be evaluated as input variables to estimate the four proposed output values for the next day. Each data set is divided into a training set for the years 2010-21 and a test set for the year 2022. Technical parameter labels are shown in table 1.

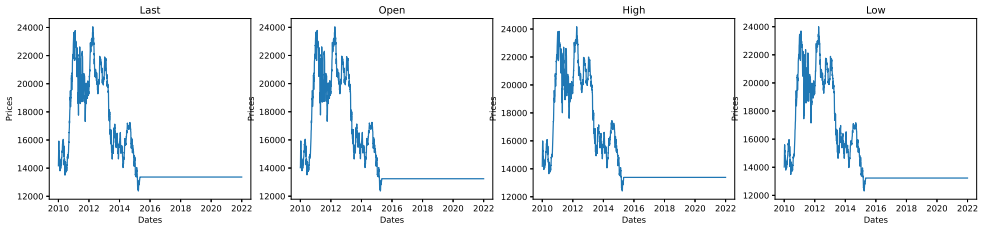


Figure 6: Line graph of variables to predict of IGBVL index. Source: own elaboration.

| Technical parameter | Label |
|---|---|
| Opening price | Open |
| Closing price | Last |
| Highest price | High |
| Lowest price | Low |
| Volume | Volume |
| Indicator of volatility market | VIX |
| 10-year US Treasury bond rate | H15T10Y |
| 30-year US Treasury bond rate | H15T30Y |

Table 1: Labels used for technical parameters. Source: own elaboration.

## 5.2 Data Preprocessing

After collecting the data, we explored and preprocessed it to ensure its quality and suitability for analysis. We examined its descriptive statistics to know the behavior of data. We removed any missing values, corrected any inconsistencies, and checked for outliers. We also normalized the data to ensure that all the variables are on the same scale, which is a necessary step when using deep learning algorithms.

## 5.3 Model Development

To predict the stock prices of the selected Latin American market indexes, we use persistence model as baseline then we implement an ARIMA model traditionally used for time series forecasts and finally we employed associative deep neural networks using the Long Short-Term Memory (LSTM) technique. The LSTM technique is known for its high accuracy in time series-based and complex pattern analysis. We trained our model using a portion of the data and tested its performance on the remaining data.

## 5.4 Model Evaluation

To evaluate the performance of our model, we use the total loss function $L_{total}$, which is the root mean squared error ($RMSE$) of each stock price to predict ($L_i$). The smaller the value of $L_{total}$, the better the accuracy of the prediction model. These formulas are shown in the following equations 9, 10:

$$L_{total} = \frac{1}{n}\sum_{i=1}^{n} L_i \tag{9}$$

Where the loss function $L_i$ is the root mean square error ($RMSE$) between observed values ($Y_i$) and predicted values ($\hat{Y}_i$) for each value to predict:

$$L = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2} \tag{10}$$

It is proposed as a validation technique to evaluate the candidate models to use walk-forward validation with a fixed time window for the training set. Which consists of; iteratively, train the model with the training set, then predict the first value from the validation set and store it for evaluation, then take it one step further, adding the current value of this validation observation to the training set and discarding the oldest keeping its time window, to retrain the model and predict the next value to evaluate and so on until the entire validation set is evaluated with the indicated metric (loss function). As detailed in the next illustration 7:

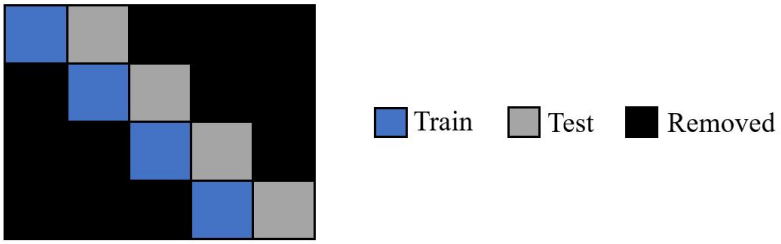


Figure 7: Validation technique. Source: own elaboration.

### 5.5 Software and Hardware

We used Python as our primary programming language and various libraries such as TensorFlow, Keras, and Pandas for data manipulation, model development, and evaluation. The hardware used for training the models was a computer with a GPU to accelerate the computations.

In summary, this study employed associative deep neural networks using the LSTM technique to predict the opening, closing, highest, and lowest stock prices of select Latin American market indexes. We collected, explored, preprocessed and normalized the data, developed our model using Python, TensorFlow, Keras, and Pandas, and evaluated its performance using $L_{total}$ metric, applying the validation technique said. This process is detailed in next section 6.

## 6 Analysis of Results

In this section, the exploration and preprocessing of data, estimation and selection of models and finally results and conclusions, will be developed for the three data sets. This process will be detailed for the COLCAP dataset, although it was also carried out for rest of indices analyzed and all tables and figures presented in this section will be included in the appendix for other indices and their analyzes are based on same principles. Ultimately summary results will be shared for all indices.

### 6.1 Data exploration and preprocessing

We will analyse COLCAP training set data as follows below:

#### 6.1.1 Univariate statistics

Some univariate statistics are calculated to get an idea of the structure of the data that we are working with. These are detailed in table 2:

| | Last | Open | High | Low | Volume | VIX | H15T10Y | H15T30Y |
|---|---|---|---|---|---|---|---|---|
| Count | 3131.00 | 3131.00 | 3131.00 | 3131.00 | 3131.00 | 3131.00 | 3131.00 | 3131.00 |
| Mean | 1496.75 | 1496.68 | 1504.42 | 1487.99 | 1.18E+08 | 18.11 | 2.20 | 2.96 |
| Min | 894.03 | 894.03 | 916.39 | 880.72 | 5.47E+05 | 9.14 | 0.52 | 0.99 |
| P25 | 1348.16 | 1348.29 | 1354.98 | 1341.72 | 1.88E+07 | 13.41 | 1.74 | 2.53 |
| P50 | 1499.16 | 1498.21 | 1507.89 | 1491.36 | 3.14E+07 | 16.29 | 2.21 | 2.96 |
| P75 | 1657.56 | 1657.98 | 1665.79 | 1648.42 | 8.83E+07 | 20.63 | 2.67 | 3.32 |
| Max | 1942.37 | 1942.37 | 1956.89 | 1940.38 | 3.36E+09 | 82.69 | 4.01 | 4.85 |
| Range | 1048.34 | 1048.34 | 1040.50 | 1059.66 | 3.36E+09 | 73.55 | 3.49 | 3.86 |
| IQR | 309.40 | 309.69 | 310.81 | 306.70 | 6.95E+07 | 7.23 | 0.93 | 0.79 |
| Stand_dev | 192.22 | 192.24 | 192.04 | 192.19 | 2.47E+08 | 7.18 | 0.70 | 0.78 |
| Coeff_var | 12.84% | 12.84% | 12.77% | 12.92% | 208.70% | 39.67% | 31.96% | 26.22% |
| Skewness | -0.15 | -0.15 | -0.14 | -0.17 | 5.04 | 2.59 | -0.07 | 0.13 |
| Kurtosis | -0.75 | -0.75 | -0.79 | -0.71 | 37.07 | 11.94 | -0.12 | -0.01 |

Table 2: COLCAP univariate statistics. Source: own elaboration.

From these univariate statistics we have the following observations:

- Very similar statistics can be seen for four values to be predicted: opening price, closing price, higher and lower.
- Average value of opening and closing price is around 1495, highest price is 1503 and lowest price is 1487.
- Minimum, Percentiles (indicator that shows the proportion of data that is below its value), IQR (range of fifty percent of central region) and Maximum of values to be predicted, show an important variation.
- Standard deviation (average spread of the data around the mean) of values to be predicted is around USD 189. Relatively large.
- Coefficient of variation (percentage of standard deviation with respect to mean) shows that the variable with greatest variability is Volume, with a coefficient greater than 200%, which indicates serious problems of high dispersion.
- Skewness indicates shape of data, where for four values to be predicted and for the 10-year treasury bond rates they have a leftward slope in their frequency distribution (mean $<$ median), for others it is toward right (mean $>$ median) and this latter to a greater extent for Volume and VIX variables.
- kurtosis is about concentration of values around the central zone of its frequency distribution, where Volume and VIX variables present their highest values considerably higher than 3; which indicates that its extreme values are distant from the center.

Considerable scatter observed in data is likely to make it difficult to get accurate predictions, if these are caused by random fluctuation (eg non-systematic).

### 6.1.2 Line graph

The line graph for each technical parameter is illustrated in figure 8:

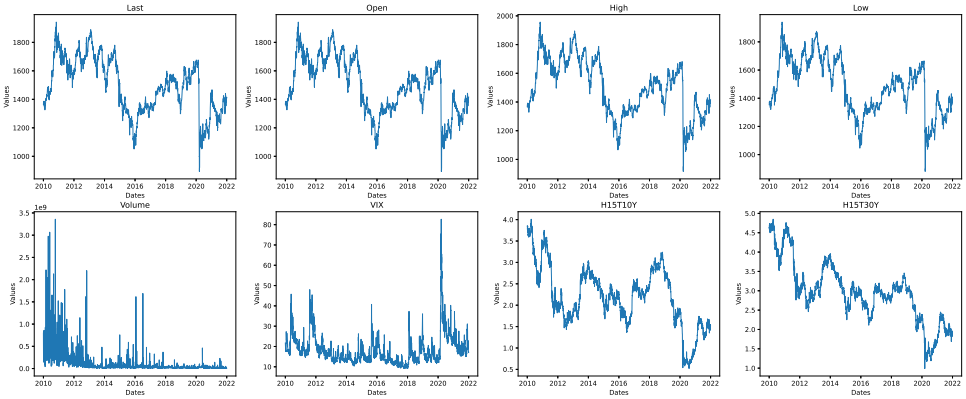


Figure 8: Line graph for all variables. Source: own elaboration.

And we have observations too:

- Trends of growth and decrease over time are appreciated.
- Considerable fluctuations are observed over the years, probably the data does not have a stationary behavior.
- There is great variability in year 2020 for values to be predicted and for VIX and in the first years and in 2016 for Volume. And a considerable drop for the four prices is observed in 2016 and 2020.
- These high fluctuations in mentioned years are possibly related to the mortgage crisis of 2008; still recovering in the first years of analysis range, debt crisis in Europe of 2016 and nd beginnings of the COVID-19 pandemic of 2020.

As observed it may be beneficial to perform transformations on the data for modeling.

### 6.1.3 Histogram and density graph

The histogram and density graph for each technical parameter is shown in figure 9:

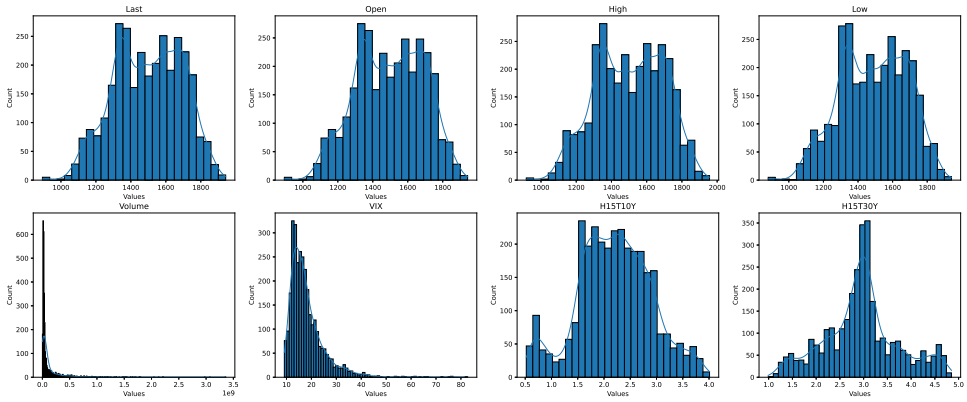


Figure 9: Histogram and density graph for all variables. Source: own elaboration.

From which respective inclinations of distributions in data can be seen, reported in the Skewness and Kurtosis statistic. It seems that the four prices have a bimodal behavior.

### 6.1.4 Box and whisker plot

Initially, the exploration of the distributions of all the variables for the entire time of analysis was carried out using the box and whiskers 10. Afterward, these distributions were examined for each study year to observe possible trends and seasonalities 11.

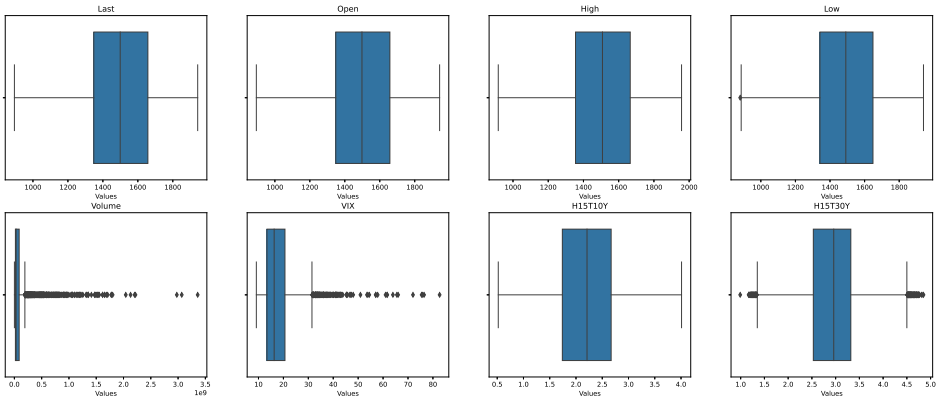


Figure 10: Box and whisker plot for all variables. Source: own elaboration.

This figure shows to a greater extent possible outliers in variables Volume and VIX, as warned by Kurtosis. In addition, Volume variable has a higher concentration of its values in central region, as expected due to having the highest Kurtosis value.

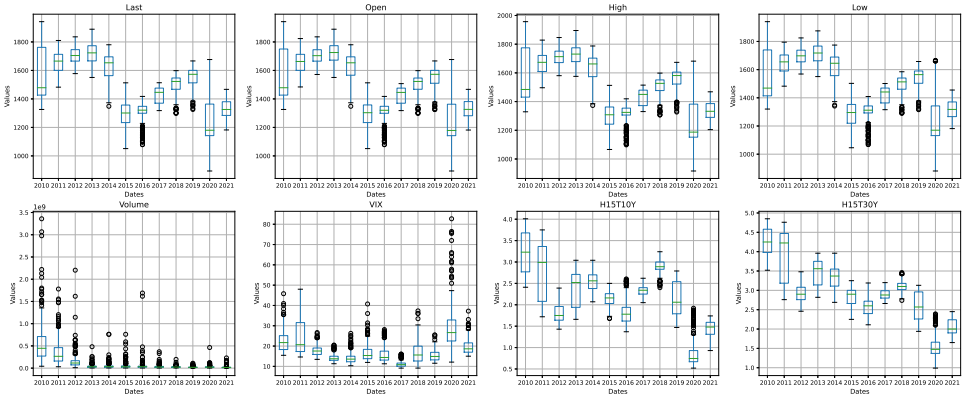


Figure 11: Box and whisker plot for all variables each year. Source: own elaboration.

These diagrams provide more information about data:

- Median values (green line) for data show a trend such as appears to be non-linear.
- Spread of central region (blue boxes) of the data differs, perhaps not consistently over time.
- For values to be predicted, possible outliers are seen in years 2016, 2018 y 2019. In addition, greater dispersion is observed in years 2010 y 2020.
- Volume and VIX variables present more years with atypical values
- For variables of US treasury bond rates, although they have similar behaviors, for 30-year variable it can be seen that it has fewer years with atypical values.

Annual fluctuations observed seem not to be systematic and therefore complex to model.

### 6.1.5 Multivariate statistics

The Granger causality test was carried to evaluate relationship between variables, to understand if a variable has explanatory power with respect to any other variable and so for all. The maximum number of lags for which the test is computed is defined with a rule of thumb: (T/3)/n, where T=sample size and n=number of variables. Resulting in 130 lags.

To do this, variables must have a stationary behavior, therefore its stationarity is evaluated with the Dickey-Fuller test. Additionally, its rolling mean and standard deviation are examined to define an adequate transformation.

The results of the Dickey-Fuller test for original variables are shown in the table 3:

| | **Last** | **Open** | **High** | **Low** | **Volume** | **VIX** | **H15T10Y** | **H15T30Y** |
|---|---|---|---|---|---|---|---|---|
| Test Statistic | -2.47 | -2.71 | -2.40 | -2.56 | -2.53 | -5.50 | -2.51 | -1.98 |
| p-value | 0.12 | 0.07 | 0.14 | 0.10 | 0.11 | 0.00 | 0.11 | 0.30 |
| Lags Used | 8.00 | 3.00 | 12.00 | 9.00 | 29.00 | 9.00 | 0.00 | 11.00 |
| Observations Used | 3122.00 | 3127.00 | 3118.00 | 3121.00 | 3101.00 | 3121.00 | 3130.00 | 3119.00 |
| Critical Value (1%) | -3.43 | -3.43 | -3.43 | -3.43 | -3.43 | -3.43 | -3.43 | -3.43 |
| Critical Value (5%) | -2.86 | -2.86 | -2.86 | -2.86 | -2.86 | -2.86 | -2.86 | -2.86 |
| Critical Value (10%) | -2.57 | -2.57 | -2.57 | -2.57 | -2.57 | -2.57 | -2.57 | -2.57 |

Table 3: Dickey-Fuller test with original COLCAP variables. Source: own elaboration.

From the test, it can be seen that for almost all variables they have a p-value greater than 0.05; significance level regularly used, which indicates that the null hypothesis of being a non-stationary process cannot be rejected, except for the VIX variable. Therefore, to have stationarity for all variables, an appropriate transformation must be applied. To define it, statistics over time are examined in Fig. 12.

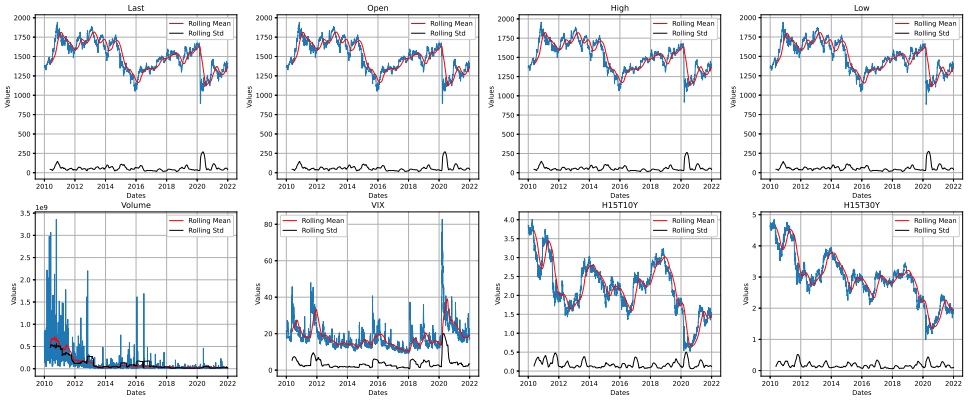


Figure 12: Rolling mean and standard deviation for all variables. Source: own elaboration.

It is observed that both mean and standard deviation vary, therefore Logarithm-Differentiation transformation will be used, to remove changes in both statistics over time. This variation also happens to VIX variable, therefore, although its null hypothesis was not rejected, this transformation is applied to dissipate these changes in trend for all variables.

Finally, we evaluate that all transformed variables present a stationary behavior 4 and then we performed the Granger causality test.

| | Last | Open | High | Low | Volume | VIX | H15T10Y | H15T30Y |
|---|---|---|---|---|---|---|---|---|
| Test Statistic | -14.32 | -14.20 | -11.53 | -16.09 | -17.33 | -23.08 | -11.09 | -18.82 |
| p-value | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Lags Used | 17.00 | 17.00 | 24.00 | 12.00 | 29.00 | 7.00 | 19.00 | 9.00 |
| Observations Used | 3112.00 | 3112.00 | 3105.00 | 3117.00 | 3100.00 | 3122.00 | 3110.00 | 3120.00 |
| Critical Value (1%) | -3.43 | -3.43 | -3.43 | -3.43 | -3.43 | -3.43 | -3.43 | -3.43 |
| Critical Value (5%) | -2.86 | -2.86 | -2.86 | -2.86 | -2.86 | -2.86 | -2.86 | -2.86 |
| Critical Value (10%) | -2.57 | -2.57 | -2.57 | -2.57 | -2.57 | -2.57 | -2.57 | -2.57 |

Table 4: Dickey-Fuller test with transformed COLCAP variables. Source: own elaboration.

According to the results presented in Table 4, the null hypothesis of non-stationarity of Test DF is rejected for all the variables analyzed, therefore they have a stationary behavior. We proceed with the Granger causality test presented in the next table 5.

| | Last_x | Open_x | High_x | Low_x | Volume_x | VIX_x | H15T10Y_x | H15T30Y_x |
|---|---|---|---|---|---|---|---|---|
| Last_y | 1.0000 | 0.0000 | 0.0000 | 0.0000 | 0.1202 | 0.0002 | 0.0000 | 0.0000 |
| Open_y | 0.0000 | 1.0000 | 0.0000 | 0.0000 | 0.0112 | 0.0000 | 0.0000 | 0.0000 |
| High_y | 0.0000 | 0.0000 | 1.0000 | 0.0000 | 0.0109 | 0.0000 | 0.0000 | 0.0000 |
| Low_y | 0.0000 | 0.0000 | 0.0000 | 1.0000 | 0.0469 | 0.0000 | 0.0000 | 0.0000 |
| Volume_y | 0.0789 | 0.0767 | 0.1041 | 0.0915 | 1.0000 | 0.0056 | 0.1720 | 0.1783 |
| VIX_y | 0.2466 | 0.1031 | 0.3333 | 0.3867 | 0.0001 | 1.0000 | 0.0692 | 0.0012 |
| H15T10Y_y | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0895 | 0.0001 | 1.0000 | 0.0000 |
| H15T30Y_y | 0.0000 | 0.0001 | 0.0000 | 0.0000 | 0.2098 | 0.0000 | 0.0000 | 1.0000 |

Table 5: p-value matrix of Granger causality test. Source: own elaboration.

The row are the response variable (Y) and the columns are the predictor variable (X). This matrix returns p-values testing whether X forecasts Y. If p-value is less than a significance level (usually 0.05), then the null hypothesis of non-Granger causality, which indicates that variable X and their lagging values do not explain variation of Y is rejected and vice versa.

According to the results, four values to predict, VIX indicator and US Treasury bond rates are relevant to explain variables to be predicted: Last, Open, High, Low. Volume, although with p-values higher than the previous ones, also seems to be useful to explain almost all of these, except for Last.

## 6.2 Persistence model

It is proposed to take as the baseline of the estimation and performance method, the observation of previous period as an estimate of observation of following period. Candidate models will be compared on this basis, also known as naive forecast. The $L_{total}$ of this model is 19.6345. Forecast results by variable to predict are shown in the following figure below 13:

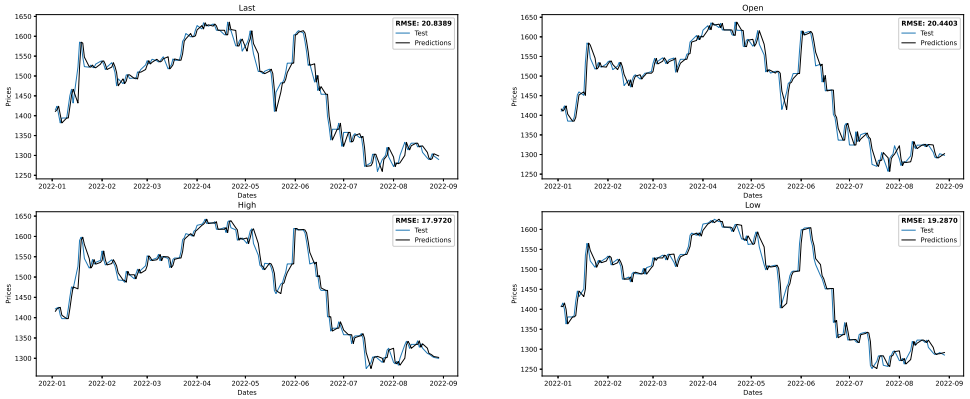


Figure 13: Persistence model forecast results. Source: own elaboration.

## 6.3 The ARIMA model

ARIMA models were implemented to address the prediction problem for each of the four indicated values using only information from their lags. We worked with values with stationary behavior that we already have (the transformed variables), as they should be for ARIMA model. However, to evaluate the predictions with respect to the test set by variable, an inverse transformation of the values of the predictions was applied, where for the first value to be predicted from test set, entire training set will be used to fit the model and then do this iteratively for following values to be predicted, the window of training data set will be moved by adding next observation and removing oldest one; keeping its size, according to the validation technique to be used.

### 6.3.1 The ACF and PACF graphs

To define the possible order values of ARIMA model, the ACF and PACF are shown in the figure 14.

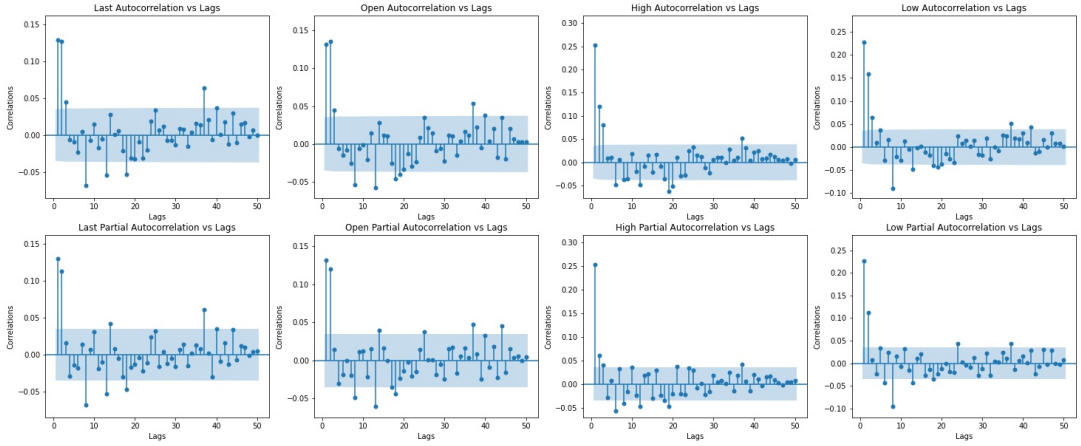


Figure 14: ACF and PACF graphs for each transformed variable. Source: own elaboration.

It can be seen that in general possible values for p are 1 and 2 and for q are 1, 2 and 3. The possible combinations of these values are evaluated in ARIMA model using Grid Search method and the one with the smallest $L$ metric is selected.

### 6.3.2 Forecast results

The ARIMA model with the best combination of parameters p and q for each variable is presented in the following graph 15.

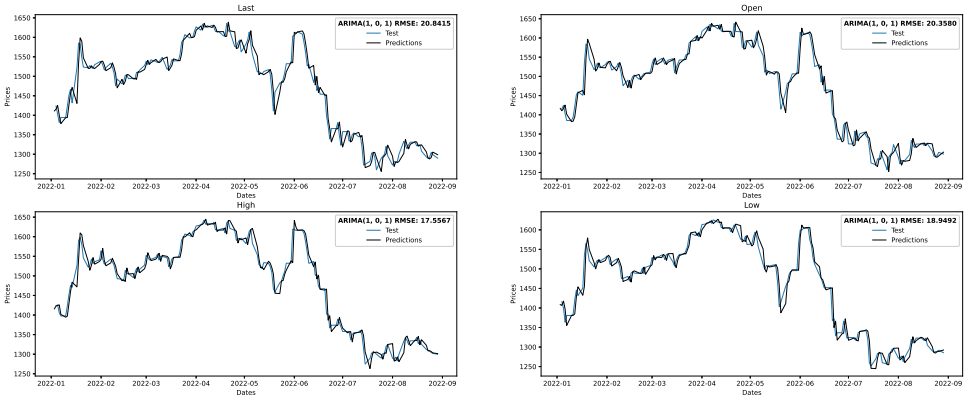


Figure 15: ARIMA model forecast results. Source: own elaboration.

The $L_{total}$ of this model is 19.4264, slightly better than persistence model that was 19.6345.

### 6.3.3 Residual analysis

To complete the process of ARIMA model, its residuals are evaluated.

The histogram and the density graph of residuals 16 are shared to analyze if a like Gaussian distribution is presented, which is directly related to validity of ARIMA model.

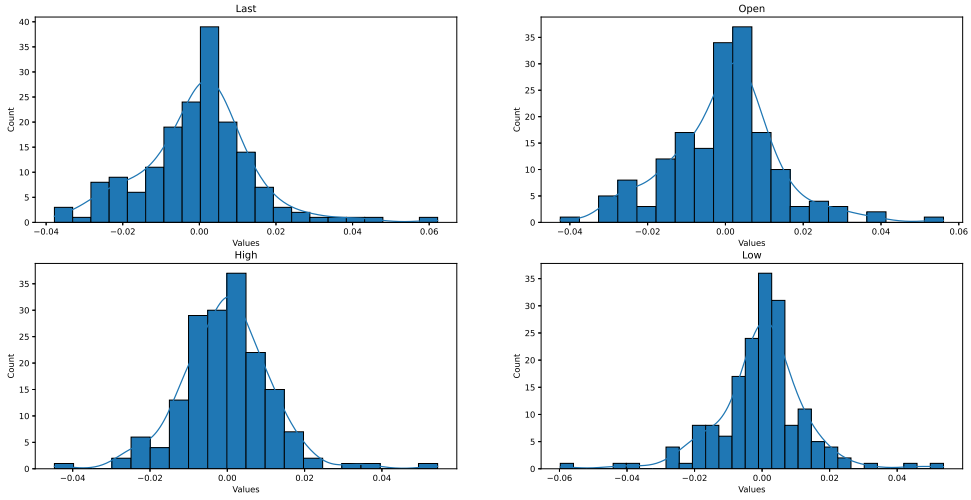


Figure 16: Histogram and density residuals graph by ARMA model per transformed variable. Source: own elaboration.

A distribution similar to the Gaussian is appreciated, with a slight inclination.

It is also evaluated if residuals show correlation. If so, it would indicate that the model has more opportunities to model the temporal structure in the data. For this, the ACF (Autocorrelatioin Function) and PACF (Partial Autocorrelation

Function) graphs for the residuals are analyzed 17:

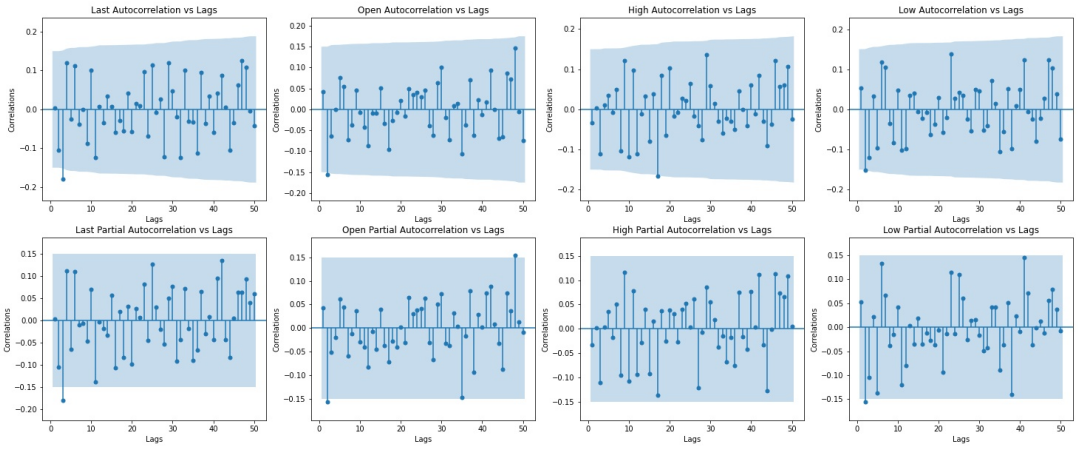


Figure 17: ACF and PACF graphs for each transformed variable after implementing the ARIMA models. Source: own elaboration.

The presence of autocorrelation in residuals is not appreciated.

## 6.4 LSTM associative networks model

In this section, we present how we implemented the model to predict the four mentioned values simultaneously.

### 6.4.1 Model structure details

According to the proposed structure in figure 5, the input data consist of eight variables of the data set with time steps, having three dimensions: (records, lags, variables), which then follow the sequence of the presented network scheme. in figure 4, starting at an LSTM neural network layer, then going through a LSTM with dropout layer, followed by another LSTM layer, then another LSTM with dropout layer, and finally a dense one-dimensional output layer. All layers except dense layer return sequences; that is, its output states of each time steps, preserving these dimensions, but since the output of last dense layer does not return sequences but rather has two dimensions (records, units). To concatenate this return with initial input data to predict next value and so on for the other values, these input data are first passed through an LSTM layer that does not return sequences so that it has two dimensions (records, units) and thus can be concatenated, increasing the number of units and then reshaping it by adding 1 lag so that it has the three dimensions needed to pass it through the proposed sequence of network scheme. Finally four predicted values are contained and thus obtain the appropriate output.

### 6.4.2 Model setting

The transformed variables (Logarithm-Differentiation and also the Min-Max Scaled Variables) were taken as input in the model, in order to improve the performance of the model in terms of its predictive power. This taking into account that one of the steps suggested in the analysis of time series is to stationarize the series, which can be achieved through simple transformations. And then to evaluate predictions values in original test set, the respective inverse transformation was applied to them.

To fit and validate this model, both training set and test set must have a data set to train and another to predict. And with respect to the aforementioned validation technique, the time window to be used and maintained for historical data to be trained will be given by timestep hyperparameter. This process is carried out with transformed data and finally to evaluate the metric $L_{total}$ on test set, predictions are taken to their original representation.

For each data transformation, it is defined as fixed values a hyperbolic tangent activation function for LSTM layers and for training model, batch size of 64, loss function of mean squared error and Adam optimizer are established. And for adjustment of the model, the possible combinations of variation of following hyperparameters will be evaluated: timestep values of 10 and 30, units values of 50 and 100, epochs values of 30 and 50 and dropout value that varies from zero; no dropout, increasing by 0.1 up to 0.8. Resulting in 144 combinations to be evaluated and the one that obtains the best performance of the selected metric in test set will be kept.

### 6.4.3 Prediction results

The histogram and density graphs of results of the metric $L_{total}$ in original test set for each transformation of the data is presented below 18 to compare them:

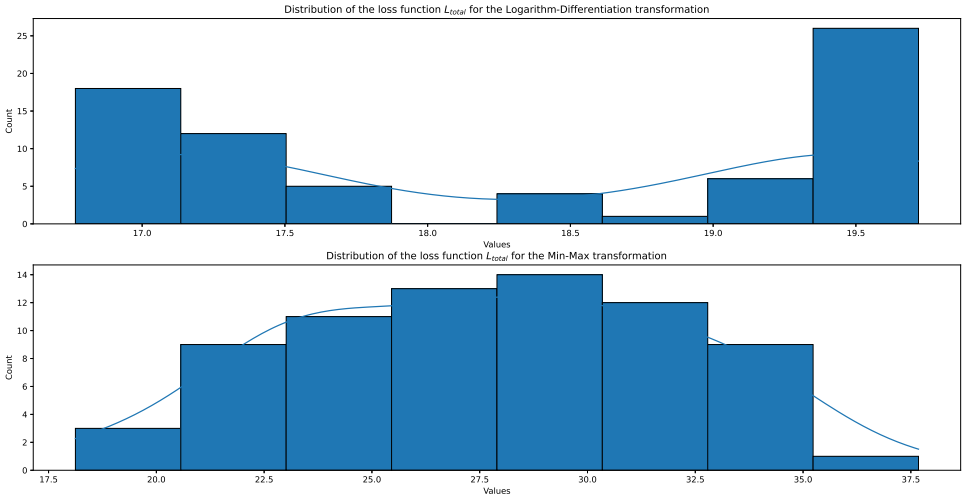


Figure 18: Histogram and density $L_{total}$ graph by both transformations used. Source: own elaboration.

When examining results of both transformations, it can be observed that the Logarithm-Differentiation transformation achieves lower and less sparse error rates than the Min-Max transformation, its best result was 16.7662 compared to 18.1225 using the second transformation. This shows that it is worth evaluating other transformations when working with these neural networks and that they achieve good performance; in our best case, when operating with seasonal transformations. And it also shows importance of carrying out a hyperparameter adjustment process to find a combination of hyperparameters with a remarkably better result than other combinations evaluated.

This best value of the metric $L_{total}$ that was obtained with the Logarithm-Differentiation transformation, was computed under the following hyperparameters: hyperbolic tangent activation function, batch size of 64, loss function of mean squared error, Adam optimizer, timestep of 10, units of 50, epochs of 50 and dropout value of 0. This best combination of the proposed model will be detailed in remainder of this section.

Next, a graph 19 is shared with the evolution of the root mean squared error and the loss by epochs in training stage of the model with transformed data sets, to analyze competition of the best model:

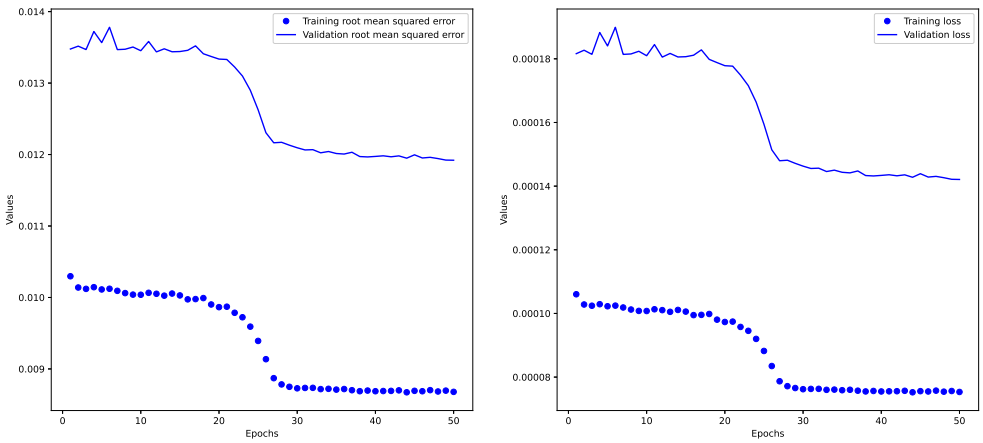


Figure 19: Loss functions in transformed data sets in the model training phase for the Logarithm-Differentiation transformation. Source: own elaboration.

Where it can be seen how error rate of the model is reduced and its result is stabilized, which is a good sign of model fit.

And with respect to the regularization method, figure 20 presents results of the metric $L_{total}$ in the original test set, for each dropout value evaluated on best combination of other hyperparameters; keeping them constant, to analyze their behavior:

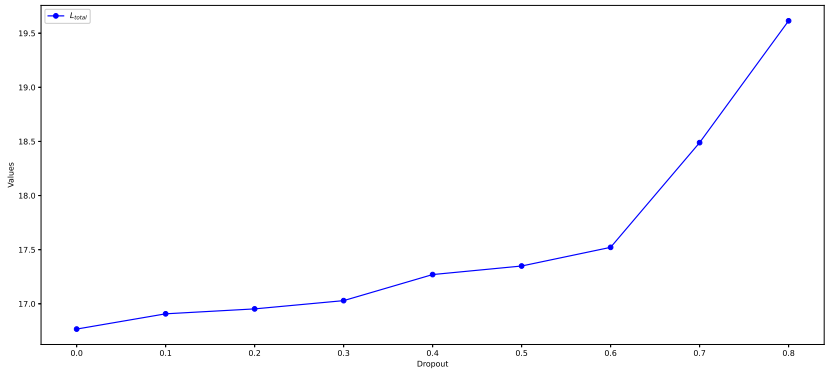


Figure 20: $L_{total}$ for each dropout value evaluated on the best combination of other hyperparameters for the Logarithm-Differentiation transformation. Source: own elaboration.

It is curious how, for data set worked on, best result with mentioned combination of hyperparameters is achieved without dropout and it tends to deteriorate as it increases. It was possible to capture the pattern of the data without regularization and perhaps when it was smoothed its bias increased.

It is worth examining performance of best combination of hyperparameters, varying the dropout for the min-max transformation, which can be seen in figure 21:

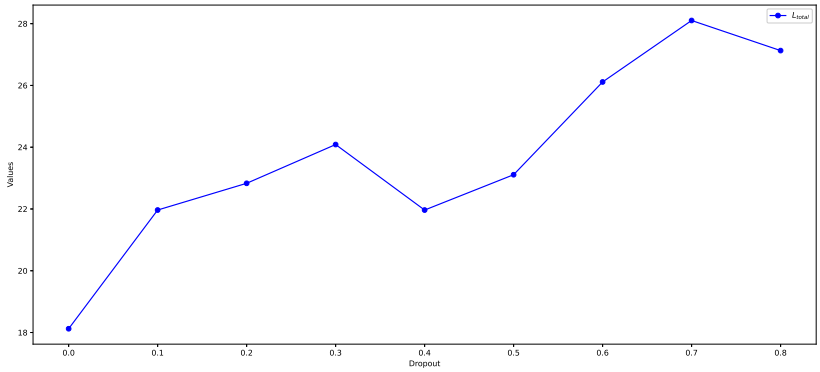


Figure 21: $L_{total}$ for each dropout value evaluated on the best combination of other hyperparameters for the Min-Max transformation. Source: own elaboration.

The same behavior can be seen for other transformation (Min-Max) in performance of the model with respect to the regularization method.

And finally figure 22 with best forecast results on original test set is shown:

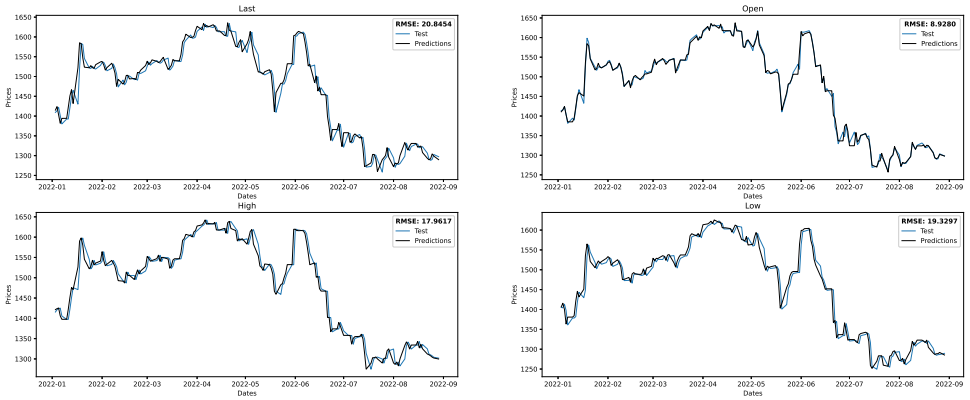


Figure 22: LSTM associative networks model forecast results. Source: own elaboration.

The $L_{total}$ of this model is 16.7662, noticeably better than ARIMA model that was 19.4264 and than persistence model that was 19.6345.

# 7 Summary results

This section share a summary results of best $L_{total}$ for each model for all evaluated indices. Below is a table 6 with the summarized best results of $L_{total}$:

| Ltotal | | | |
|---|---|---|---|
| **Indices** | **Persistence model** | **ARIMA model** | **LSTM associative networks model** |
| COLCAP | 19.6345 | 19.4264 | 16.7662 |
| IPSA | 53.2026 | 52.9827 | 41.2169 |
| MEXBOL | 505.9933 | 502.3751 | 420.7650 |

Table 6: Best $L_{total}$ summary results. Source: own elaboration.

In all cases, the ARIMA model achieves slightly better results than the base model, persistence model, however the proposed LSTM model obtains notoriously better results than others, evidencing its superior performance.

An, detail of adjusted hyperameters for the models for each index can be seen in the table 7:

| Adjuted hyperparameters | | | |
|---|---|---|---|
| **Indices** | **Persistence model** | **ARIMA model** | **LSTM associative networks model** |
| COLCAP | N/A | Last (p, d, q): (1, 0, 1)<br>Open (p, d, q): (1, 0, 1)<br>High (p, d, q): (1, 0, 1)<br>Low (p, d, q): (1, 0, 1) | Data transformation: Logarithm-Differentiation<br>Activation function: Hyperbolic Tangent<br>Loss funtion: Mean Squared Error<br>Optimizer: Adam<br>Batch size: 64<br>Timestep: 10<br>Untis: 50<br>Epoch: 50<br>Dropout: 0.0 |
| IPSA | N/A | Last (p, d, q): (1, 0, 1)<br>Open (p, d, q): (1, 0, 1)<br>High (p, d, q): (2, 0, 1)<br>Low (p, d, q): (2, 0, 3) | Data transformation: Logarithm-Differentiation<br>Activation function: Hyperbolic Tangent<br>Loss funtion: Mean Squared Error<br>Optimizer: Adam<br>Batch size: 64<br>Timestep: 30<br>Untis: 100<br>Epoch: 50<br>Dropout: 0.1 |
| MEXBOL | N/A | Last (p, d, q): (2, 0, 2)<br>Open (p, d, q): (1, 0, 2)<br>High (p, d, q): (1, 0, 1)<br>Low (p, d, q): (2, 0, 3) | Data transformation: Logarithm-Differentiation<br>Activation function: Hyperbolic Tangent<br>Loss funtion: Mean Squared Error<br>Optimizer: Adam<br>Batch size: 64<br>Timestep: 30<br>Untis: 50<br>Epoch: 50<br>Dropout: 0.0 |

Table 7: Best adjusted hyperparameters summary results. Source: own elaboration.

To close, it is worth noting that the persistence model does not require adjusting any hyperparameter, but its performance is inferior in all indices analyzed. From the ARIMA model it is worth mentioning that its best hyperparameters depend on variable to be predicted and will not necessarily be same for each variable of same index in a given period of time, although its performance is barely better than base model. And from the LSTM associative networks model it is also observed that best combination of hyperparameters depends on data to work with, and it is also remarkable that a model for each variable to be predicted is not required as in previous models, but rather just one structured model that simultaneously predicts them with the best results.

# 8 Conclusions

- Examining the data and its statistics is essential to choose the models to use and how to proceed in the modeling process. In our case, the transformation with best performance was chosen because statistics showed a changing behavior in mean and in standard deviation. The multivariate statistics also indicated that we had relevant variables for the prediction. In addition, possible values (p, d, q) to be evaluated in the ARIMA model were given with analysis of the ACF and PACF graphs, and additionally the graphs of error rate with respect to epochs in the training phase of the associative deep neural networks model showed a proper fit.
- It is worth trying models other than traditional ones, which can achieve better results. As seen in the results obtained in this article, the use of deep neural network models obtained more accurate results than a baseline model and a traditional ARIMA model, showing greater ability to capture the behavior of time series data with complex patterns.
- The model fitting process can be crucial to achieve a better performance in the estimation of a phenomenon in question. A clear example of this is presented in the histogram and density graph of distribution of the loss function $L_{total}$, which shows considerable dispersion that the error rates had for all combinations of hyperparameters and transformations evaluated, and that in certain cases the results were even worse than the baseline model which had the lowest performing, which reaffirms importance of this process, so that the model can achieve a better fit to the data to work with.
- This paper can represent a contribution in the prediction of stock prices, working with data from the Latin American market using the proposed model, which as far as I know, there are no works carried out like the one proposed in this work with data from Latin America.

# 9 Expected products

- Model for associated stock prices prediction developed through the application of concepts, elements and methods of analytics.
- Publishable thesis article.

# 10 Data management plan

This paper will work with the most complete historical daily data from the index MILA (Mercado Integrado Latioamericano), and also from the most representative indexes of the countries that pertain to the MILA. The information is of a public nature. This information may vary during the development of this work in order to improve the accuracy of the model.

# 11 Ethical aspects

The data will be used by the members of the work team: the director and the student, with the expectation of benefiting investors with a model that can support them in making investment decisions and also expect to represent a contribution due to the focus of this work on Latin American market indices.

# References


[1] D. Singh and B.K. Gupta. Closing price prediction of nifty stock using lstm with dense network. *Lecture Notes in Networks and Systems*, 302:382–392, 2022. cited By 0.

[2] G. Ding and L. Qin. Study on the prediction of stock price based on the associated network model of lstm. *International Journal of Machine Learning and Cybernetics*, 11(6):1307–1317, 2020. cited By 34.

[3] G. Bathla, R. Rani, and H. Aggarwal. Stocks of year 2020: prediction of high variations in stock prices using lstm. *Multimedia Tools and Applications*, 2022. cited By 0.

[4] U. Demirel, H. Cam, and R. Unlu. Predicting stock prices using machine learning methods and deep learning algorithms: The sample of the istanbul stock exchange. *Gazi University Journal of Science*, 34(1):63–82, 2021. cited By 1.

[5] M. Hirey, J. Unagar, K. Prabhu, and R. Desai. Analysis of stock price prediction using machine learning algorithms. 2022. cited By 0.

[6] S. Mehtab, J. Sen, and A. Dutta. Stock price prediction using machine learning and lstm-based deep learning models. *Communications in Computer and Information Science*, 1366:88–106, 2021. cited By 8.

[7] R. E. Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica*, 50(4):987–1007, 1982.

[8] G.E.P. Box, G.M. Jenkins, G.C. Reinsel, and G. L. Ljung. *Time series analysis: Forecasting and control: Fourth edition.* 2015.

[9] S. Banerjee and D. Mukherjee. Short term stock price prediction in indian market: A neural network perspective. *Studies in Microeconomics*, 10(1):23–49, 2022. cited By 0.

[10] R.S. Priya and C. Sruthi. Stock price prediction based on deep learning using long short-term memory. *Lecture Notes in Electrical Engineering*, 792:67–76, 2022. cited By 0.

[11] I.K. Nti, A.F. Adekoya, and B.A. Weyori. A novel multi-source information-fusion predictive framework based on deep neural networks for accuracy enhancement in stock market prediction. *Journal of Big Data*, 8(1), 2021. cited By 13.

[12] J. Kavinnilaa, E. Hemalatha, M.S. Jacob, and R. Dhanalakshmi. Stock price prediction based on lstm deep learning model. 2021. cited By 1.

[13] P. Piravechsakul, T. Kasetkasem, S. Marukatat, and I. Kumazawa. Combining technical indicators and deep learning by using lstm stock price predictor. pages 1155–1158, 2021. cited By 0.

[14] H.T. Nguyen, T.B. Tran, and P.H.D. Bui. An effective way for taiwanese stock price prediction: Boosting the performance with machine learning techniques. *Concurrency Computation*, 2021. cited By 0.

[15] M. Nabipour, P. Nayyeri, H. Jabani, S. Shahab, and A. Mosavi. Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis. *IEEE Access*, 8:150199–150212, 2020. cited By 54.

[16] P. Patil, C.-S.M. Wu, K. Potika, and M. Orang. Stock market prediction using ensemble of graph theory, machine learning and deep learning models. pages 85–92, 2020. cited By 7.

[17] A. Prasad and A. Seetharaman. Importance of machine learning in making investment decision in stock market. *Vikalpa*, 46(4):209–222, 2021. cited By 0.

[18] V.S. Charan, A. Rasool, and A. Dubey. Stock closing price forecasting using machine learning models. 2022. cited By 0.

[19] W. Budiharto. Data science approach to stock prices forecasting in indonesia during covid-19 using long short-term memory (lstm). *Journal of Big Data*, 8(1), 2021. cited By 11.

[20] Y. Liu. Analysis and forecast of stock price based on lstm algorithm. pages 76–79, 2021. cited By 0.

[21] K. Yadav, M. Yadav, and S. Saini. Stock values predictions using deep learning based hybrid models. *CAAI Transactions on Intelligence Technology*, 7(1):107–116, 2022. cited By 4.

[22] L. Tian, L. Feng, L. Yang, and Y. Guo. Stock price prediction based on lstm and lightgbm hybrid model. *Journal of Supercomputing*, 2022. cited By 0.

[23] A. Bhooshan and V.S. Hari. Recurrent neural network estimator for stock price. 2021. cited By 0.

[24] D. Wu, X. Wang, and S. Wu. A hybrid method based on extreme learning machine and wavelet transform denoising for stock prediction. *Entropy*, 23(4), 2021. cited By 5.

[25] W. Li, W. Huang, and A.-M. Zou. A hybrid deep neural network model for stock prediction. pages 483–489, 2021. cited By 0.

[26] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

# A Tables and figures for other indices analyzed

| | Last | Open | High | Low | Volume | VIX | H15T10Y | H15T30Y |
|---|---|---|---|---|---|---|---|---|
| Count | 3131.00 | 3131.00 | 3131.00 | 3131.00 | 3131.00 | 3131.00 | 3131.00 | 3131.00 |
| Mean | 4382.20 | 4382.50 | 4407.37 | 4356.75 | 7.32E+08 | 18.11 | 2.20 | 2.96 |
| Min | 2876.03 | 2876.03 | 3100.76 | 2850.78 | 6.07E+07 | 9.14 | 0.52 | 0.99 |
| P25 | 3918.58 | 3919.09 | 3939.07 | 3896.94 | 3.60E+08 | 13.41 | 1.74 | 2.53 |
| P50 | 4255.47 | 4256.11 | 4275.60 | 4229.70 | 5.49E+08 | 16.29 | 2.21 | 2.96 |
| P75 | 4809.74 | 4815.64 | 4837.88 | 4784.00 | 8.88E+08 | 20.63 | 2.67 | 3.32 |
| Max | 5880.47 | 5880.47 | 5894.93 | 5853.90 | 8.94E+09 | 82.69 | 4.01 | 4.85 |
| Range | 3004.44 | 3004.44 | 2794.17 | 3003.12 | 8.88E+09 | 73.55 | 3.49 | 3.86 |
| IQR | 891.16 | 896.55 | 898.81 | 887.06 | 5.28E+08 | 7.23 | 0.93 | 0.79 |
| Stand_dev | 562.94 | 563.03 | 562.36 | 561.90 | 6.15E+08 | 7.18 | 0.70 | 0.78 |
| Coeff_var | 12.85% | 12.85% | 12.76% | 12.90% | 83.97% | 39.67% | 31.96% | 26.22% |
| Skewness | 0.57 | 0.57 | 0.58 | 0.56 | 3.35 | 2.59 | -0.07 | 0.13 |
| Kurtosis | -0.57 | -0.57 | -0.60 | -0.55 | 20.42 | 11.94 | -0.12 | -0.01 |

Table 8: IPSA univariate statistics. Source: own elaboration.

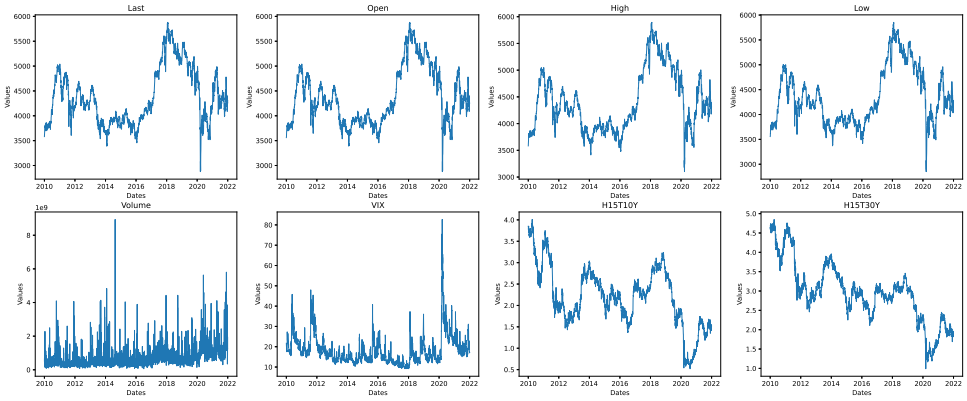


Figure 23: Line graph for the IPSA index values. Source: own elaboration.

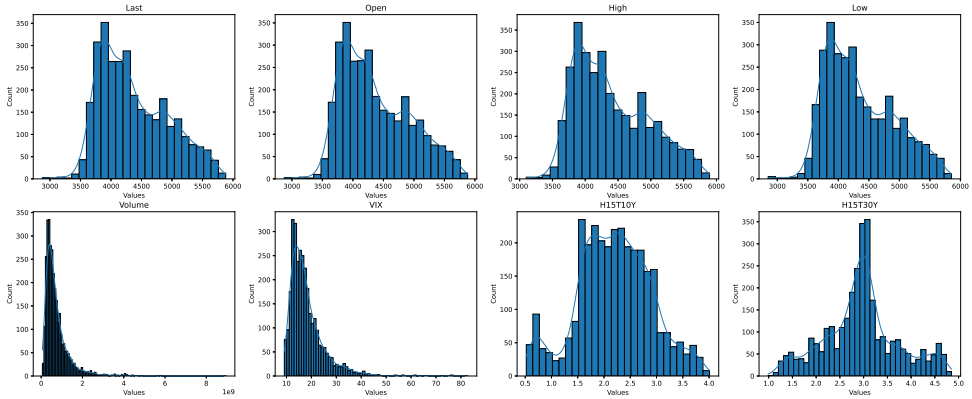


Figure 24: Histogram and density graph for the IPSA index values. Source: own elaboration.

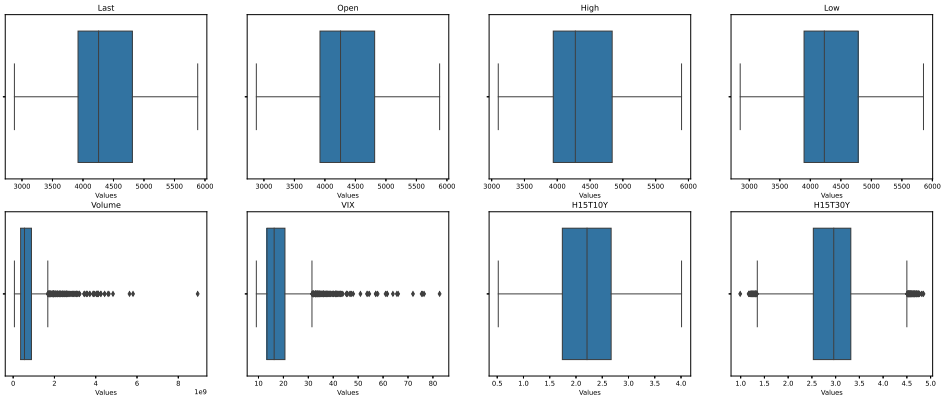


Figure 25: Box and whisker plot for the IPSA index values. Source: own elaboration.

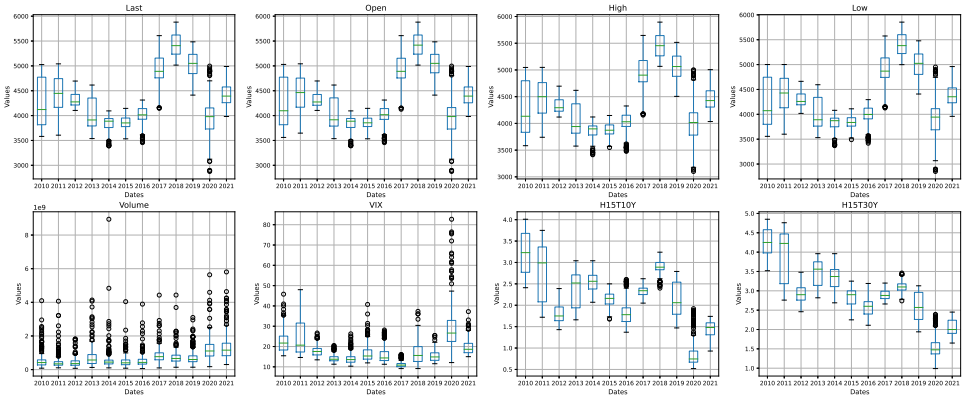


Figure 26: Box and whisker plot for the IPSA index values each year. Source: own elaboration.

| | Last | Open | High | Low | Volume | VIX | H15T10Y | H15T30Y |
|---|---|---|---|---|---|---|---|---|
| Test Statistic | -2.56 | -2.56 | -2.34 | -2.65 | -5.38 | -5.50 | -2.51 | -1.98 |
| p-value | 0.10 | 0.10 | 0.16 | 0.08 | 0.00 | 0.00 | 0.11 | 0.30 |
| Lags Used | 15.00 | 15.00 | 22.00 | 13.00 | 19.00 | 9.00 | 0.00 | 11.00 |
| Observations Used | 3115.00 | 3115.00 | 3108.00 | 3117.00 | 3111.00 | 3121.00 | 3130.00 | 3119.00 |
| Critical Value (1%) | -3.43 | -3.43 | -3.43 | -3.43 | -3.43 | -3.43 | -3.43 | -3.43 |
| Critical Value (5%) | -2.86 | -2.86 | -2.86 | -2.86 | -2.86 | -2.86 | -2.86 | -2.86 |
| Critical Value (10%) | -2.57 | -2.57 | -2.57 | -2.57 | -2.57 | -2.57 | -2.57 | -2.57 |

Table 9: Dickey-Fuller test with original IPSA variables. Source: own elaboration.

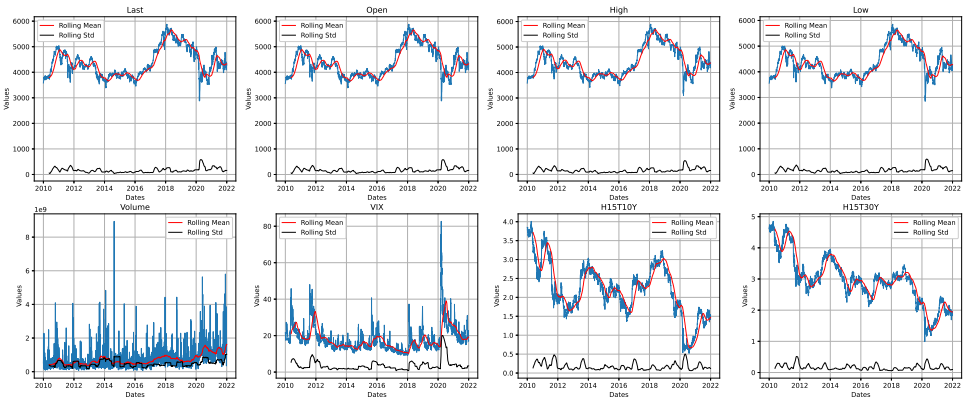


Figure 27: Rolling mean and standard deviation for each of the variables of the IPSA index. Source: own elaboration.

| | Last | Open | High | Low | Volume | VIX | H15T10Y | H15T30Y |
|---|---|---|---|---|---|---|---|---|
| Test Statistic | -14.17 | -14.80 | -12.60 | -14.02 | -16.29 | -23.08 | -11.09 | -18.82 |
| p-value | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Lags Used | 17.00 | 15.00 | 21.00 | 17.00 | 28.00 | 7.00 | 19.00 | 9.00 |
| Observations Used | 3112.00 | 3114.00 | 3108.00 | 3112.00 | 3101.00 | 3122.00 | 3110.00 | 3120.00 |
| Critical Value (1%) | -3.43 | -3.43 | -3.43 | -3.43 | -3.43 | -3.43 | -3.43 | -3.43 |
| Critical Value (5%) | -2.86 | -2.86 | -2.86 | -2.86 | -2.86 | -2.86 | -2.86 | -2.86 |
| Critical Value (10%) | -2.57 | -2.57 | -2.57 | -2.57 | -2.57 | -2.57 | -2.57 | -2.57 |

Table 10: Dickey-Fuller test with transformed IPSA variables. Source: own elaboration.

| | Last_x | Open_x | High_x | Low_x | Volume_x | VIX_x | H15T10Y_x | H15T30Y_x |
|---|---|---|---|---|---|---|---|---|
| Last_y | 1.0000 | 0.0000 | 0.0000 | 0.0000 | 0.1329 | 0.0000 | 0.0000 | 0.0000 |
| Open_y | 0.0000 | 1.0000 | 0.0000 | 0.0000 | 0.0097 | 0.0000 | 0.0000 | 0.0000 |
| High_y | 0.0000 | 0.0000 | 1.0000 | 0.0000 | 0.1936 | 0.0000 | 0.0000 | 0.0000 |
| Low_y | 0.0000 | 0.0000 | 0.0000 | 1.0000 | 0.0007 | 0.0000 | 0.0000 | 0.0000 |
| Volume_y | 0.0094 | 0.0079 | 0.0290 | 0.0008 | 1.0000 | 0.0321 | 0.0343 | 0.0139 |
| VIX_y | 0.0424 | 0.0082 | 0.0009 | 0.0167 | 0.0371 | 1.0000 | 0.0692 | 0.0012 |
| H15T10Y_y | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.7337 | 0.0001 | 1.0000 | 0.0000 |
| H15T30Y_y | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.5740 | 0.0000 | 0.0000 | 1.0000 |

Table 11: p-value matrix of Granger causality test for IPSA index. Source: own elaboration.

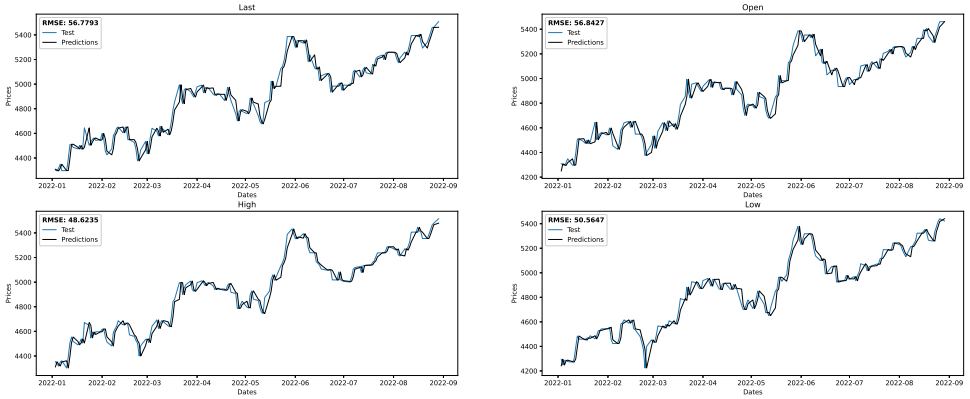


Figure 28: Persistence model forecast results for IPSA index. Source: own elaboration.

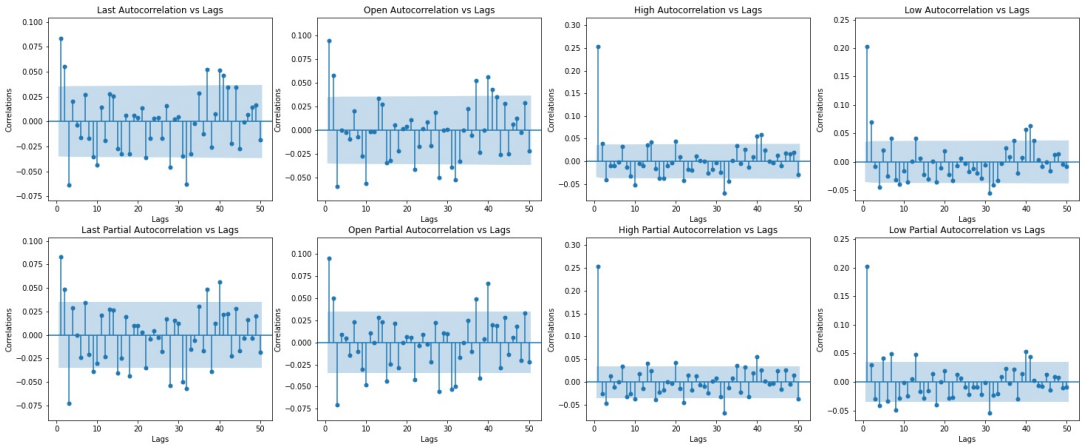


Figure 29: ACF and PACF graphs for each transformed variable for IPSA index. Source: own elaboration.

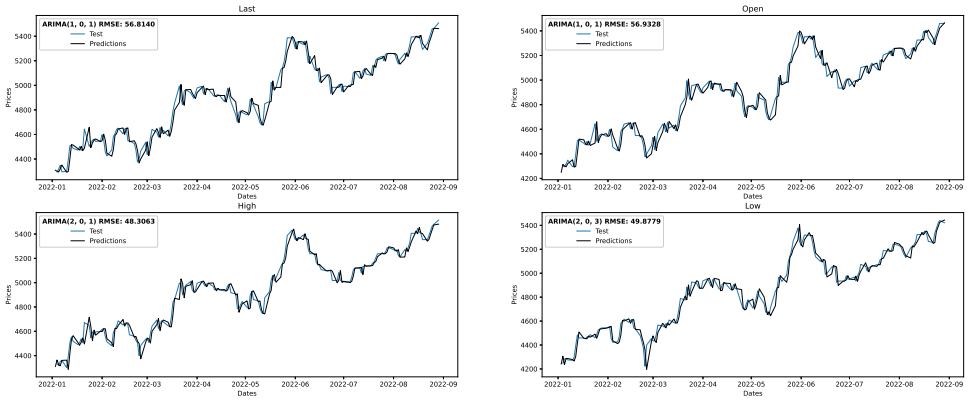


Figure 30: ARIMA model forecast results for IPSA index. Source: own elaboration.

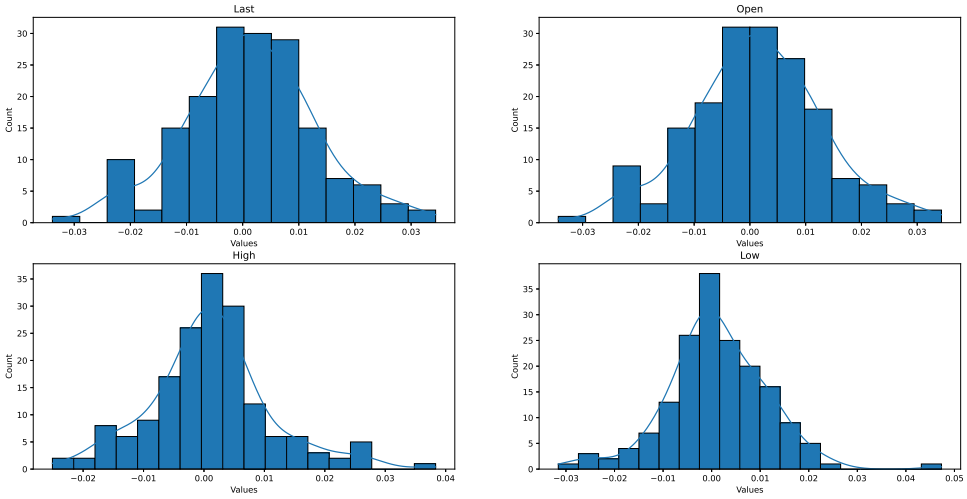


Figure 31: Histogram and density residuals graph by ARMA model per transformed variable for IPSA index. Source: own elaboration.

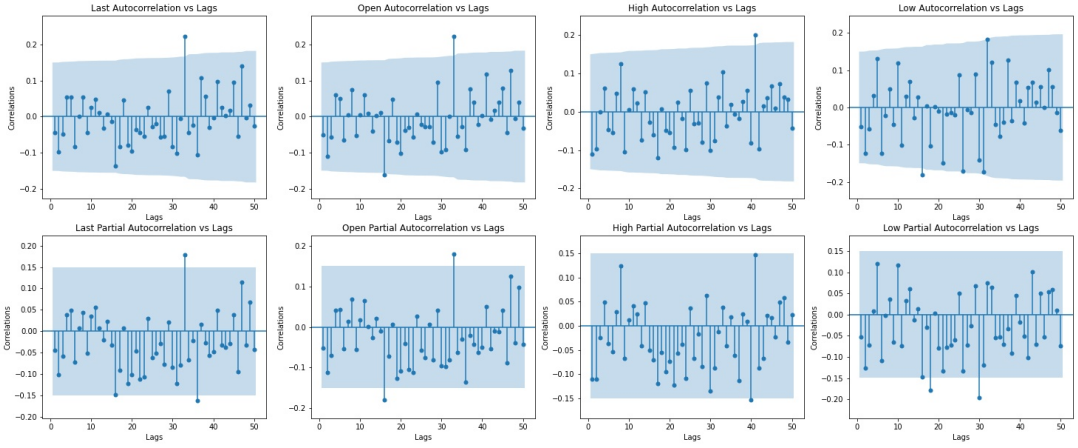


Figure 32: ACF and PACF graphs for each transformed variable after implementing the ARIMA models for IPSA index. Source: own elaboration.

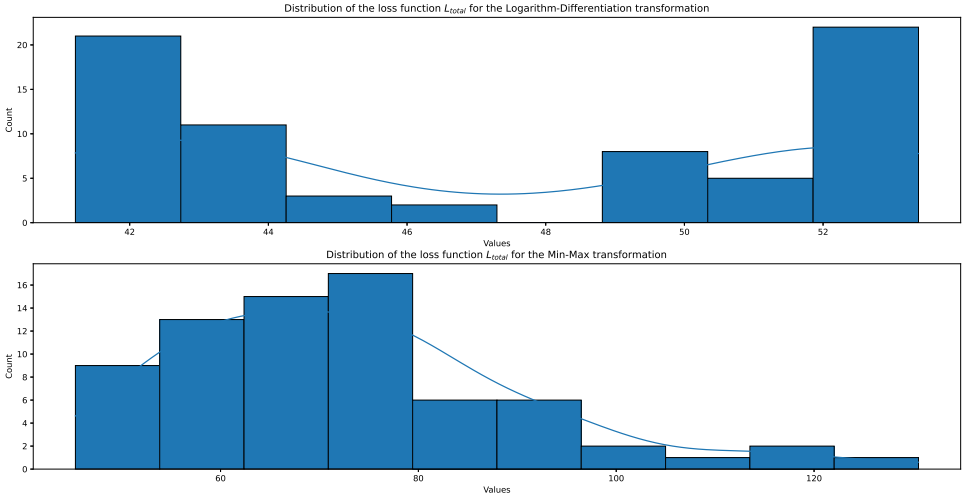


Figure 33: Histogram and density $L_{total}$ graph by both transformations used for IPSA index. Source: own elaboration.

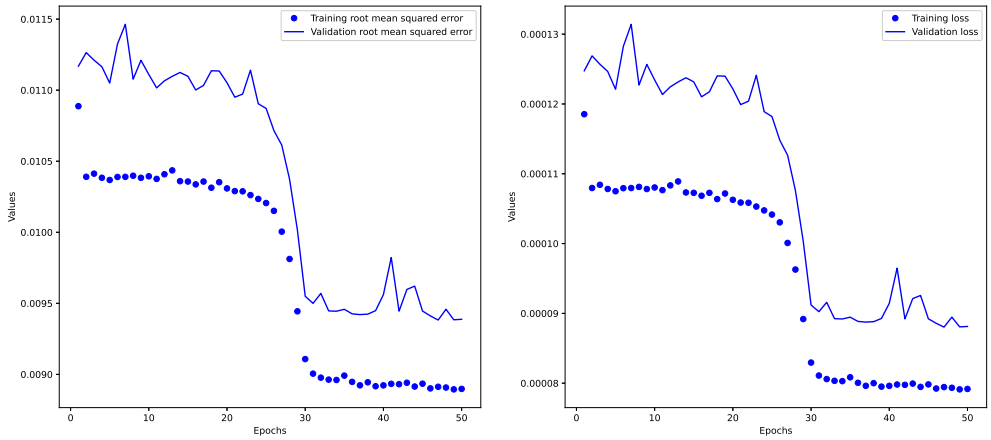


Figure 34: Loss functions in transformed data sets in the model training phase for the Logarithm-Differentiation transformation for IPSA index. Source: own elaboration.

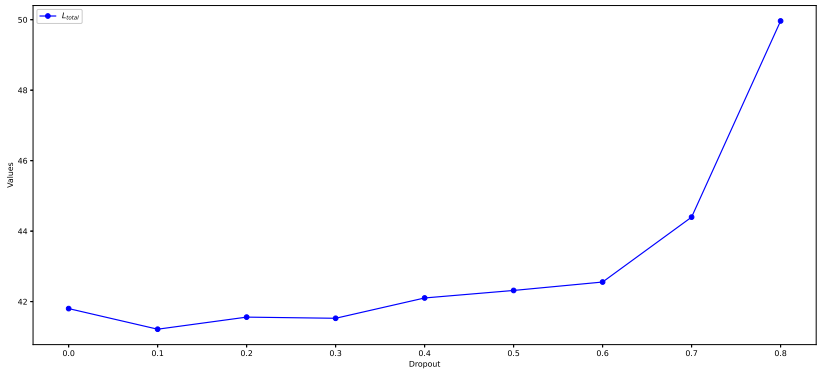


Figure 35: $L_{total}$ for each dropout value evaluated on the best combination of other hyperparameters for the Logarithm-Differentiation transformation for IPSA index. Source: own elaboration.

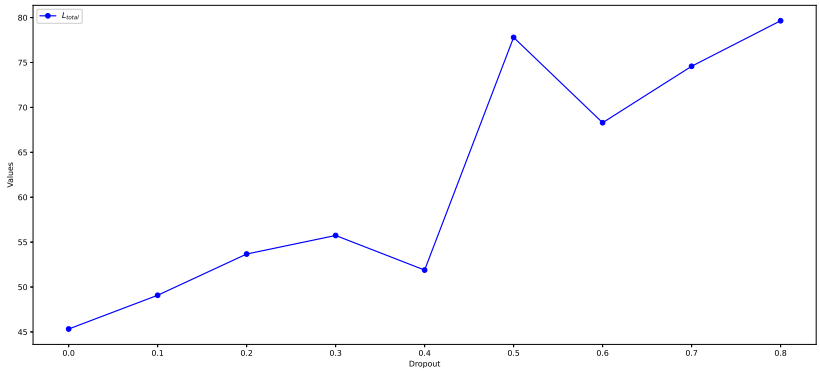


Figure 36: $L_{total}$ for each dropout value evaluated on the best combination of other hyperparameters for the Min-Max transformation for IPSA index. Source: own elaboration.

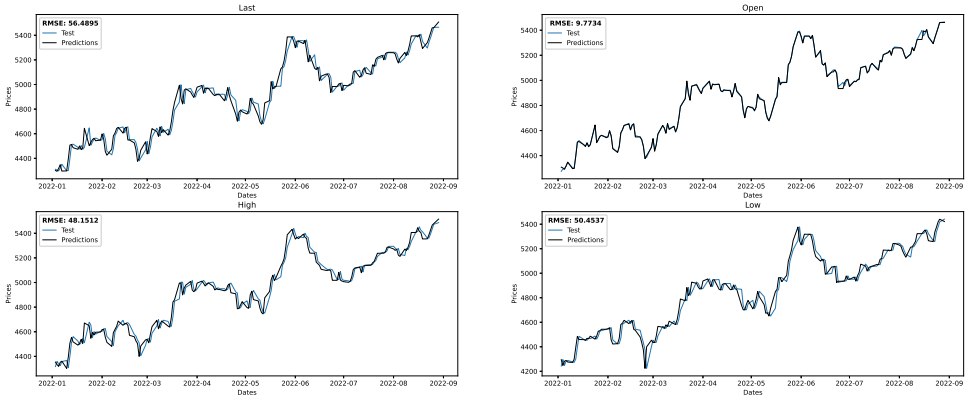


Figure 37: LSTM associative networks model forecast results for IPSA index. Source: own elaboration.

| | Last | Open | High | Low | Volume | VIX | H15T10Y | H15T30Y |
|---|---|---|---|---|---|---|---|---|
| Count | 3131.00 | 3131.00 | 3131.00 | 3131.00 | 3131.00 | 3131.00 | 3131.00 | 3131.00 |
| Mean | 42563.56 | 42562.49 | 42816.82 | 42293.37 | 1.92E+08 | 18.11 | 2.20 | 2.96 |
| Min | 30368.08 | 30376.64 | 30630.73 | 29926.06 | 5.24E+06 | 9.14 | 0.52 | 0.99 |
| P25 | 38683.60 | 38691.25 | 38980.09 | 38431.12 | 1.41E+08 | 13.41 | 1.74 | 2.53 |
| P50 | 43134.51 | 43122.19 | 43373.84 | 42877.55 | 1.76E+08 | 16.29 | 2.21 | 2.96 |
| P75 | 45955.41 | 45957.09 | 46251.22 | 45740.46 | 2.22E+08 | 20.63 | 2.67 | 3.32 |
| Max | 53304.74 | 53400.27 | 53630.53 | 53106.40 | 1.76E+09 | 82.69 | 4.01 | 4.85 |
| Range | 22936.66 | 23023.63 | 22999.80 | 23180.34 | 1.75E+09 | 73.55 | 3.49 | 3.86 |
| IQR | 7271.80 | 7265.84 | 7271.13 | 7309.34 | 8.15E+07 | 7.23 | 0.93 | 0.79 |
| Stand_dev | 5180.55 | 5181.12 | 5181.67 | 5177.63 | 9.84E+07 | 7.18 | 0.70 | 0.78 |
| Coeff_var | 12.17% | 12.17% | 12.10% | 12.24% | 51.17% | 39.67% | 31.96% | 26.22% |
| Skewness | -0.26 | -0.26 | -0.26 | -0.26 | 3.73 | 2.59 | -0.07 | 0.13 |
| Kurtosis | -0.62 | -0.63 | -0.62 | -0.62 | 31.96 | 11.94 | -0.12 | -0.01 |

Table 12: MEXBOL univariate statistics. Source: own elaboration.

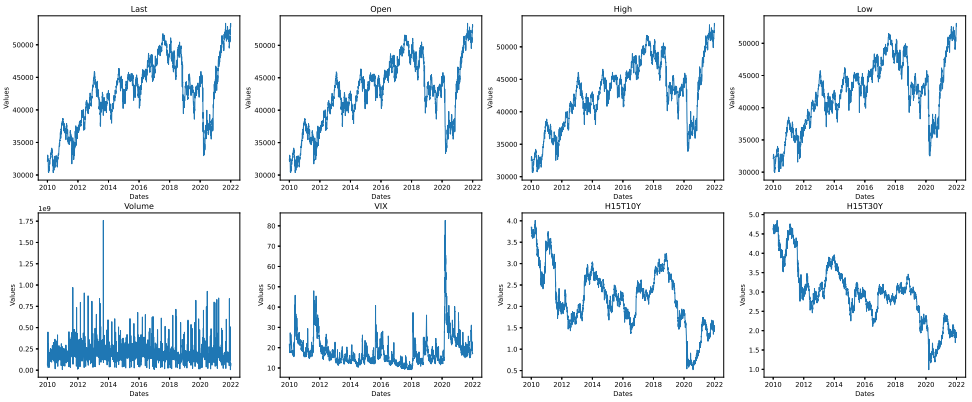


Figure 38: Line graph for the MEXBOL index values. Source: own elaboration.

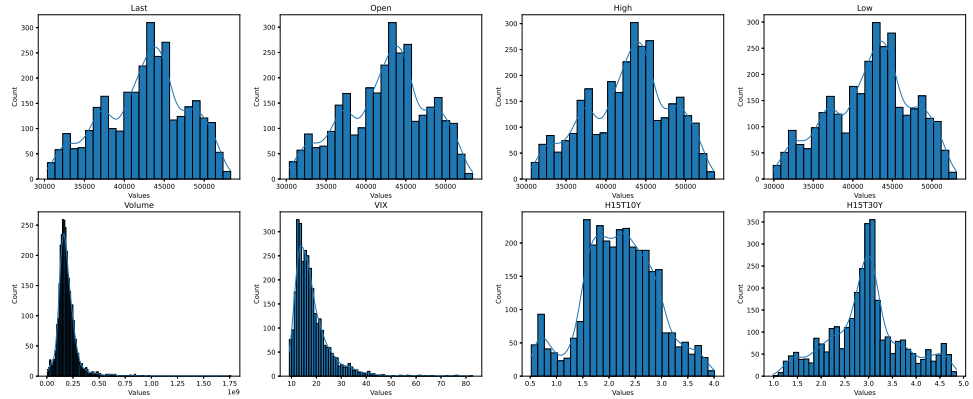


Figure 39: Histogram and density graph for the MEXBOL index values. Source: own elaboration.

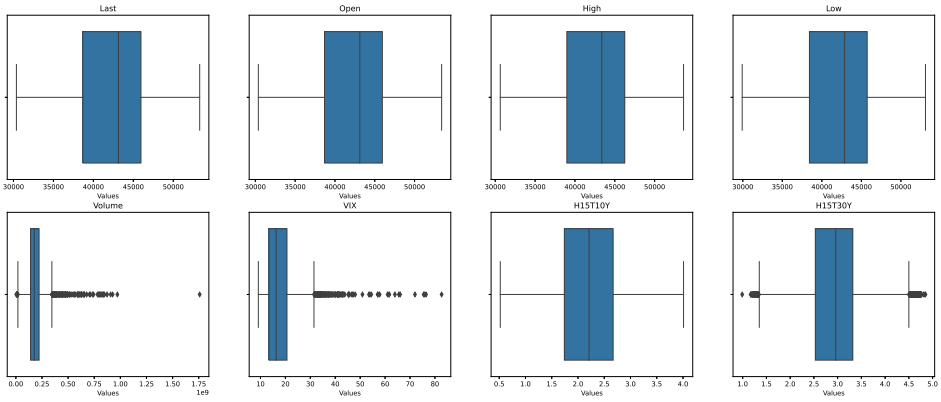


Figure 40: Box and whisker plot for the MEXBOL index values. Source: own elaboration.

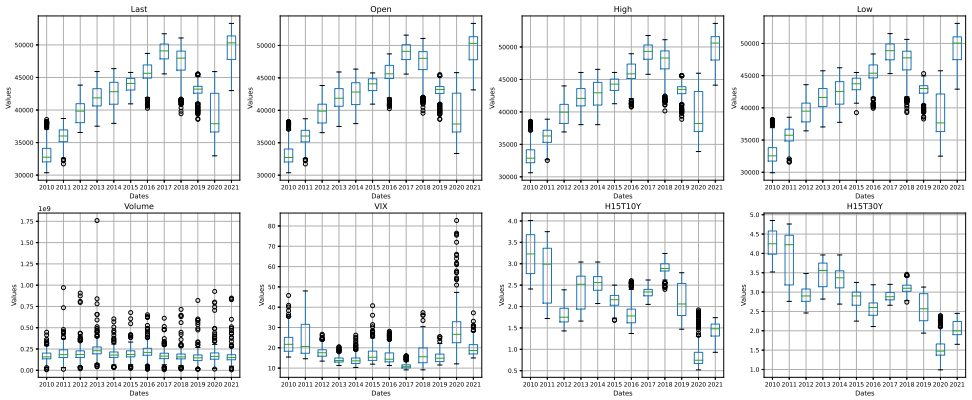


Figure 41: Box and whisker plot for the MEXBOL index values each year. Source: own elaboration.

| | **Last** | **Open** | **High** | **Low** | **Volume** | **VIX** | **H15T10Y** | **H15T30Y** |
|---|---|---|---|---|---|---|---|---|
| Test Statistic | -1.80 | -1.99 | -1.79 | -1.89 | -8.18 | -5.50 | -2.51 | -1.98 |
| p-value | 0.38 | 0.29 | 0.39 | 0.34 | 0.00 | 0.00 | 0.11 | 0.30 |
| Lags Used | 6.00 | 3.00 | 6.00 | 5.00 | 17.00 | 9.00 | 0.00 | 11.00 |
| Observations Used | 3124.00 | 3127.00 | 3124.00 | 3125.00 | 3113.00 | 3121.00 | 3130.00 | 3119.00 |
| Critical Value (1%) | -3.43 | -3.43 | -3.43 | -3.43 | -3.43 | -3.43 | -3.43 | -3.43 |
| Critical Value (5%) | -2.86 | -2.86 | -2.86 | -2.86 | -2.86 | -2.86 | -2.86 | -2.86 |
| Critical Value (10%) | -2.57 | -2.57 | -2.57 | -2.57 | -2.57 | -2.57 | -2.57 | -2.57 |

Table 13: Dickey-Fuller test with original MEXBOL variables. Source: own elaboration.

| | **Last** | **Open** | **High** | **Low** | **Volume** | **VIX** | **H15T10Y** | **H15T30Y** |
|---|---|---|---|---|---|---|---|---|
| Test Statistic | -22.46 | -33.57 | -24.36 | -26.88 | -16.68 | -23.08 | -11.09 | -18.82 |
| p-value | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Lags Used | 6.00 | 2.00 | 5.00 | 4.00 | 28.00 | 7.00 | 19.00 | 9.00 |
| Observations Used | 3123.00 | 3127.00 | 3124.00 | 3125.00 | 3101.00 | 3122.00 | 3110.00 | 3120.00 |
| Critical Value (1%) | -3.43 | -3.43 | -3.43 | -3.43 | -3.43 | -3.43 | -3.43 | -3.43 |
| Critical Value (5%) | -2.86 | -2.86 | -2.86 | -2.86 | -2.86 | -2.86 | -2.86 | -2.86 |
| Critical Value (10%) | -2.57 | -2.57 | -2.57 | -2.57 | -2.57 | -2.57 | -2.57 | -2.57 |

Table 14: Dickey-Fuller test with transformed MEXBOL variables. Source: own elaboration.

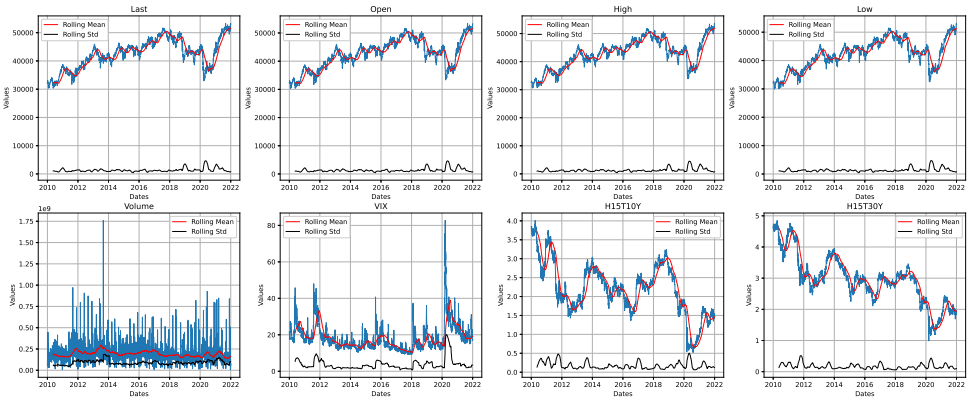


Figure 42: Rolling mean and standard deviation for each of the variables of the MEXBOL index. Source: own elaboration.

| | **Last_x** | **Open_x** | **High_x** | **Low_x** | **Volume_x** | **VIX_x** | **H15T10Y_x** | **H15T30Y_x** |
|---|---|---|---|---|---|---|---|---|
| Last_y | 1.0000 | 0.0000 | 0.0000 | 0.0019 | 0.2154 | 0.0004 | 0.0150 | 0.0000 |
| Open_y | 0.0000 | 1.0000 | 0.0000 | 0.0000 | 0.1782 | 0.0000 | 0.0000 | 0.0000 |
| High_y | 0.0000 | 0.0000 | 1.0000 | 0.0000 | 0.0004 | 0.0000 | 0.0000 | 0.0000 |
| Low_y | 0.0000 | 0.0000 | 0.0000 | 1.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Volume_y | 0.0002 | 0.0001 | 0.0000 | 0.0000 | 1.0000 | 0.0000 | 0.0196 | 0.0232 |
| VIX_y | 0.4025 | 0.1069 | 0.4053 | 0.2460 | 0.0003 | 1.0000 | 0.0692 | 0.0012 |
| H15T10Y_y | 0.0000 | 0.0004 | 0.0003 | 0.0000 | 0.0489 | 0.0001 | 1.0000 | 0.0000 |
| H15T30Y_y | 0.0001 | 0.0001 | 0.0005 | 0.0000 | 0.1540 | 0.0000 | 0.0000 | 1.0000 |

Table 15: p-value matrix of Granger causality test for MEXBOL index. Source: own elaboration.

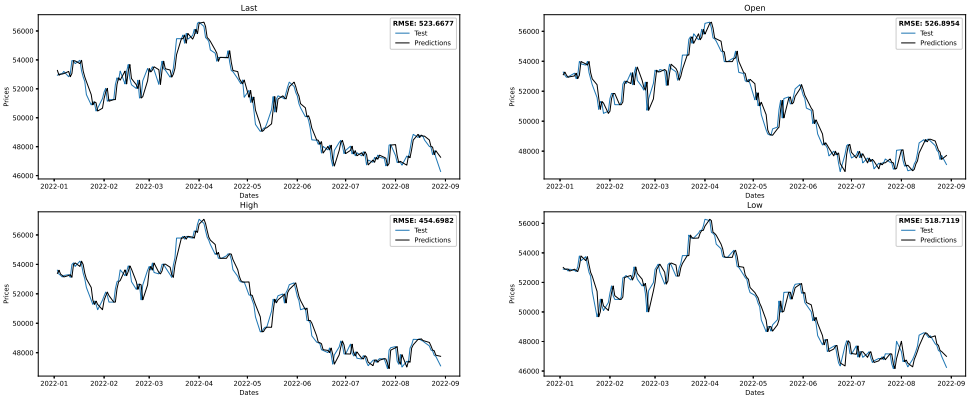


Figure 43: Persistence model forecast results for MEXBOL index. Source: own elaboration.

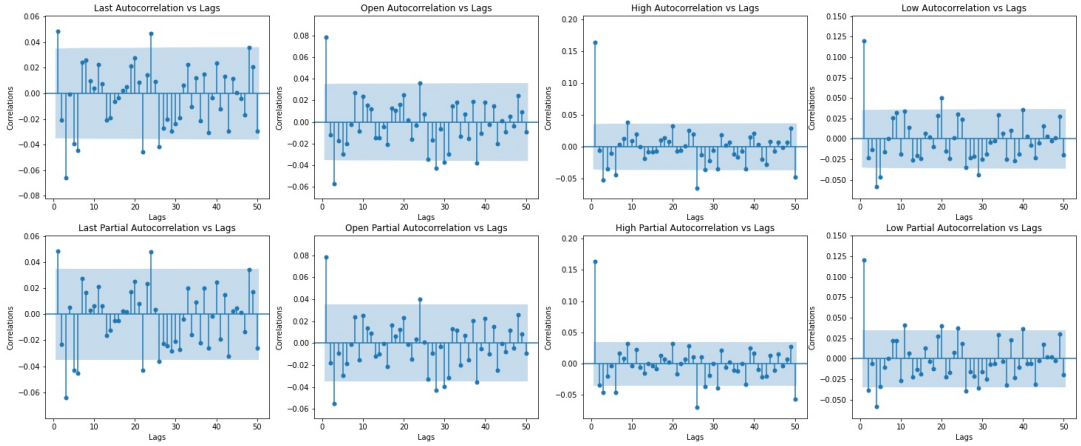


Figure 44: ACF and PACF graphs for each transformed variable for MEXBOL index. Source: own elaboration.

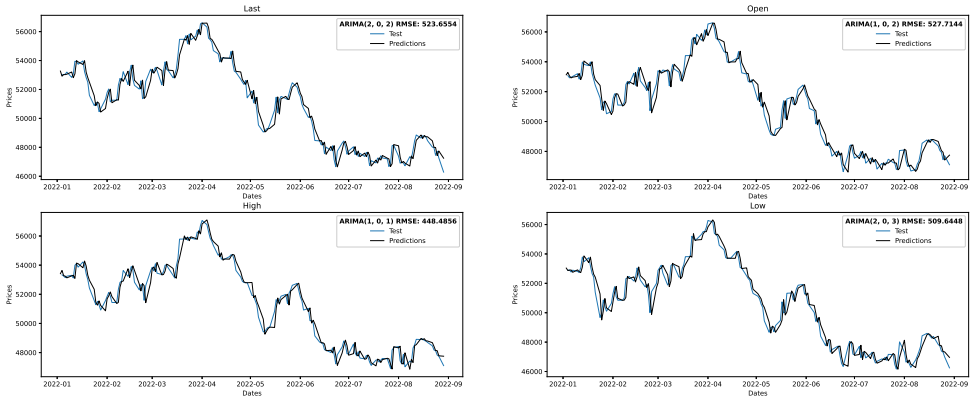


Figure 45: ARIMA model forecast results for MEXBOL index. Source: own elaboration.

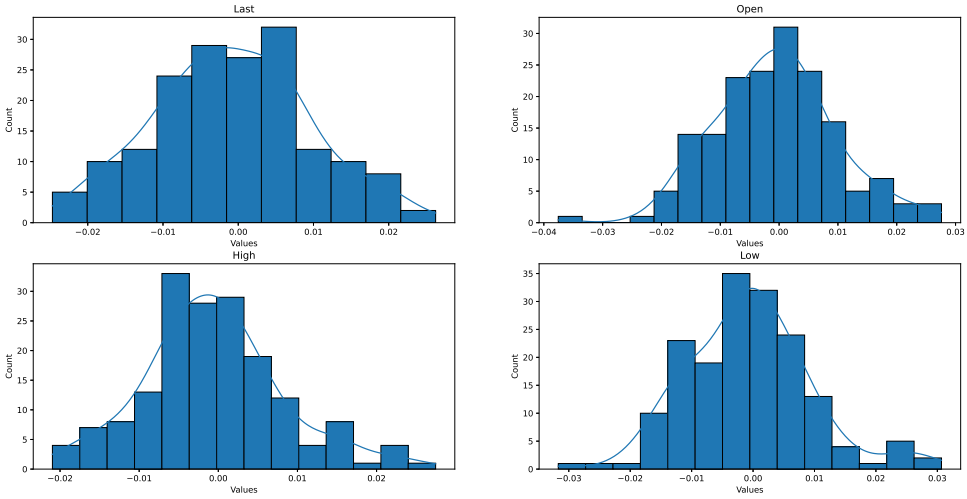


Figure 46: Histogram and density residuals graph by ARMA model per transformed variable for MEXBOL index. Source: own elaboration.

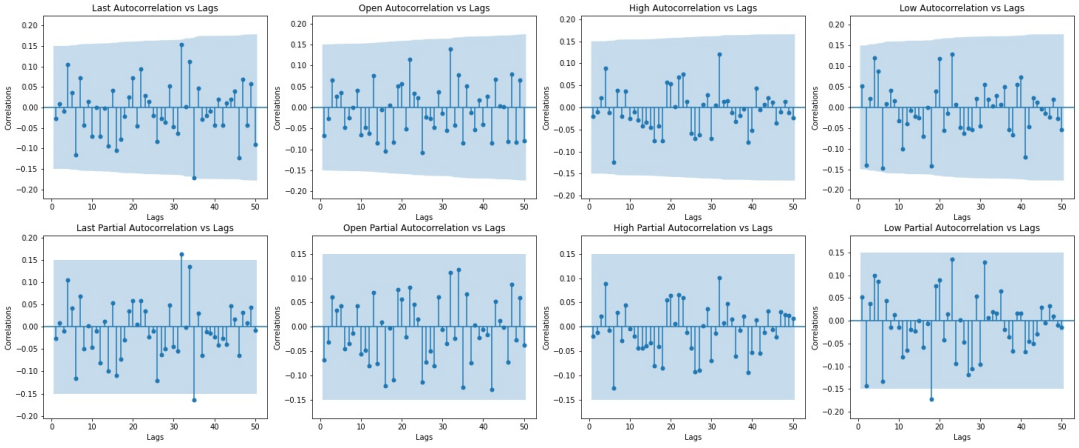


Figure 47: ACF and PACF graphs for each transformed variable after implementing the ARIMA models for MEXBOL index. Source: own elaboration.

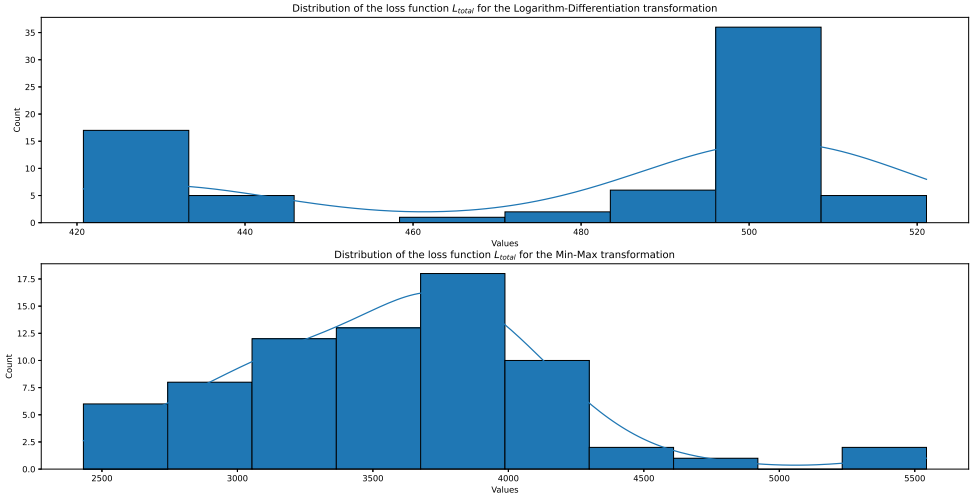


Figure 48: Histogram and density $L_{total}$ graph by both transformations used for MEXBOL index. Source: own elaboration.

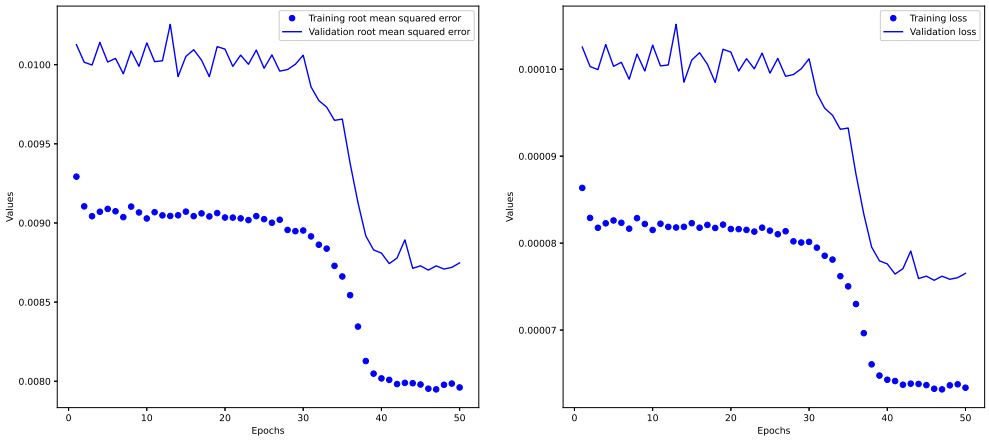


Figure 49: Loss functions in transformed data sets in the model training phase for the Logarithm-Differentiation transformation for MEXBOL index. Source: own elaboration.

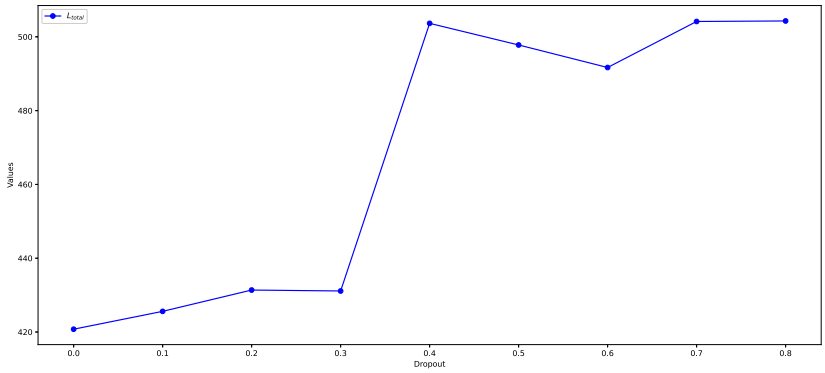


Figure 50: $L_{total}$ for each dropout value evaluated on the best combination of other hyperparameters for the Logarithm-Differentiation transformation for MEXBOL index. Source: own elaboration.

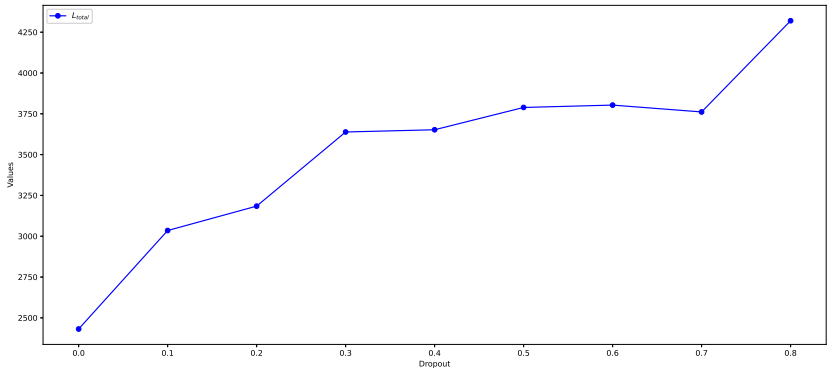


Figure 51: $L_{total}$ for each dropout value evaluated on the best combination of other hyperparameters for the Min-Max transformation for MEXBOL index. Source: own elaboration.

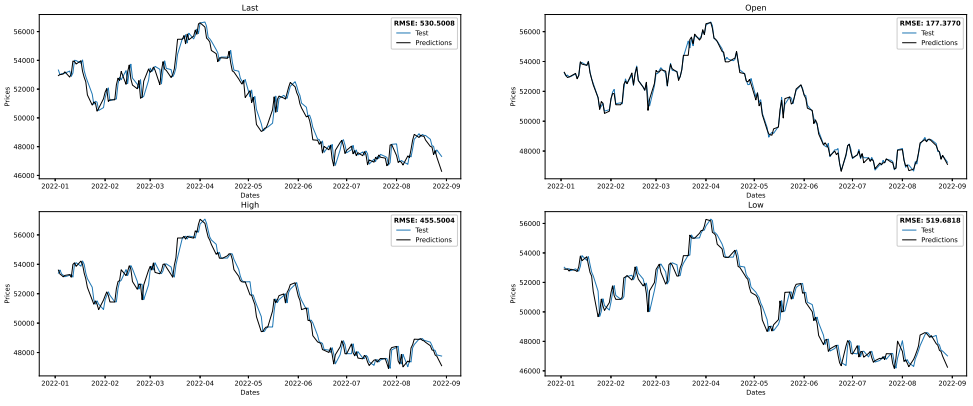


Figure 52: LSTM associative networks model forecast results for MEXBOL index. Source: own elaboration.