Instituto de Tecnologia e Liderança
Public Report
October 9, 2025

ASSET RETURN PREDICTION USING DEEP LEARNING ALGORITHMS

Cristiane Coutinho
Inteli
Author
cristiane.coutinho@sou.inteli.edu.br

Adriana Vieira
Universidade Beira do Interior
Co Supervisor
adriana.coelhovieira@gmail.com

Geraldo Vasconcelos
Inteli
Supervisor
geraldo.vasconcelos@prof.inteli.edu.br

ABSTRACT

This study aimed to analyze the performance of different deep learning architectures applied to the forecasting of financial asset returns using historical time series from the Brazilian Stock Exchange (B3). Four models LSTM, CNN, Transformer, and MLP were evaluated to assess their predictive accuracy and generalization capacity in a context marked by high volatility, noise, and non-stationarity.

The dataset covered the period from 2021 to 2024 and included opening, closing, high, low, and trading volume prices. The target variable, return, was computed as the difference between closing and opening prices divided by the opening price. Data were normalized and divided into training (January–August) and testing (September–December) subsets.

The results highlighted the inherent complexity of financial time series forecasting. The CNN achieved low numerical errors (RMSE = 0.0175, MAE = 0.0079) but exhibited a negative R^2 and extremely high MAPE, revealing poor generalization. The Transformer model produced low absolute errors (RMSE = 0.0086, MAE = 0.0048) but generated nearly constant predictions, suggesting limited learning capability. The MLP achieved a high R^2 (0.8688) but suffered from extremely high MAPE, indicating overfitting. Meanwhile, the LSTM yielded unsatisfactory results (R^2 = -1145.32), failing to capture temporal dependencies and showing severe prediction errors.

Regarding the proposed hypotheses, H1 (superiority of LSTM architectures) was not confirmed; H2 (effectiveness of Transformer models) was partially validated; H3 (lower overfitting in deep learning models compared to traditional ones) was partially refuted; H4 (inefficiency of MLP in modeling temporal dependencies) was confirmed; and H5 (increasing model complexity raises overfitting risk) was corroborated.

Overall, while deep learning demonstrates potential for financial forecasting, none of the tested models achieved robust generalization. The chaotic, volatile, and non-stationary nature of financial markets poses substantial challenges for predictive modeling. Future work should focus on hybrid approaches combining statistical and neural methods, feature engineering, and integration of exogenous variables such as macroeconomic indicators and sentiment analysis. This study reinforces the need for systematic experimentation and deeper understanding of neural architectures' limitations when applied to financial time series prediction.

Keywords: Deep Learning, Time Series, Neural Networks, LSTM, Financial Forecasting.

1. INTRODUCTION

When lending resources to institutions, a trading environment is created with the objective of obtaining financial returns based on investments. The returns generated from these transactions are referred to as “securities.” When acquiring a debt instrument, the investor also assumes the risk associated with that return, i.e., the possibility of depreciation of the investment.

To maximize returns, a technological framework was developed to support and establish the relationship between time and causality in asset gains. According to Guedes (2022), the development of algorithms in the financial sector was gradually introduced into trading markets and evolved as more technical analyses of mathematics, probability, and statistics were incorporated into so-called “investment robots.”

With this technological advancement, the development of algorithms using Machine Learning and Deep Learning became increasingly feasible and attractive, being applied for asset selection and portfolio optimization. Shimabukuro (2024) attributes these advances to the growing computational processing speed and the reduction in physical memory size, making the implementation of optimization and prediction algorithms more cost-effective.

Among deep learning techniques, recurrent neural networks such as LSTMs have proven particularly efficient for modeling financial time series, as they incorporate past information and memory update mechanisms, allowing current data to be related to remote states in time while avoiding gradient vanishing or exploding issues.

In addition to LSTMs, vision transformer-based architectures such as ViT, DeiT, Swin, and ConvMixer have also been applied to financial asset prediction. In these approaches, time series are converted into 2D images enriched with technical indicators, enabling classification of market movements as Buy, Sell, or Hold.

2. LITERATURE REVIEW

Mesquita (2020) applies LSTM neural networks to predict financial series from the Bovespa, using ten time series as the analysis base. The main objective was to evaluate the relationship between prediction results and the validity of the Random Walk Hypothesis (RWH), tested via the variance ratio. LSTM was chosen for its ability to capture long-term dependencies in sequential data, overcoming limitations of traditional recurrent networks such as gradient vanishing and exploding. The study demonstrates that series rejecting the RWH achieve better predictive performance, reinforcing that selecting statistically appropriate data influences model effectiveness. Additionally, results were compared with benchmarks like Buy and Hold and random classifiers, showing potential operational advantages for trading strategies based on deep learning.

Gezici (2024) explores the use of transformer-based architectures for predicting asset prices and directional movements. The study evaluates different variants of vision transformers (ViT, DeiT, and Swin) and a convolutional patch-based approach (ConvMixer), in addition to comparisons with traditional methods like CNN. To apply these architectures, the authors transformed one-dimensional financial time series into two-dimensional images, enriched with over 50 technical indicators (e.g., RSI, EMA, MACD), used as model inputs. Results indicate that ConvMixer achieved the highest test accuracy, followed by ViT, while Swin outperformed DeiT. Although traditional CNNs still performed well, transformers showed greater potential, suggesting a promising trend for their use in finance.

Shimabukuro (2024) applied both classical econometric models and LSTM recurrent neural networks to predict financial time series. Econometric models considered linear relationships between values and error terms, such as MA, AR, ARMA, ARIMA, and SARIMA, capturing seasonal patterns and temporal dependencies. In contrast, LSTMs do not assume linearity, enabling learning of complex relationships between current and past data while incorporating mechanisms that prevent gradient vanishing or exploding in long sequences. The network was trained iteratively using log-returns and fractional differentiation, with weight adjustment via backpropagation and gradient descent, batch processing, and early stopping to prevent overfitting. The study highlights LSTMs' ability to learn non-linear relationships in financial series, offering a predictive alternative superior to traditional methods.

TITLE	ARCHITECTURE	METRICS
Use of an LSTM Neural Network and Variance Ratio Test for Predictions in Bovespa Asset Series	LSTM	Comparison with Buy and Hold, Random Classifier, Variance Ratio
Deep Transformer-Based Asset Price and Direction Prediction	ViT, DeiT, Swin, ConvMixer, CNN	Test accuracy, architecture comparison, label distribution (Buy, Hold, Sell)
Deep Learning Applied to Ibovespa Return Prediction: An Analysis of LSTM Performance Using Log-Returns and Fractional Differentiation	LSTM with Log-Returns and Fractional Differentiation	RMSE, MAE, MAPE, MASE, R^2, comparison with baseline

3. JUSTIFICATION

Soares (2023) highlights that machine learning algorithms are widely explored in regression and classification problems for predicting asset returns and classifying specific assets. Conversely, Shimabukuro (2024) points out limiting factors related to applying machine learning models to financial assets, including large data volume requirements, low sensitivity to market noise, and risk of overfitting.

Given these challenges, investigating strategies with advanced Deep Learning algorithms, such as LSTMs and transformers, becomes relevant, as it allows exploration of more robust predictive methods capable of capturing complex and non-linear relationships in financial time series, contributing to more informed and accurate investment decisions.

4. OBJETIVOS

4.1. GENERAL OBJECTIVE

Predictive analysis of financial returns using different neural network architectures with deep learning techniques on a time series.

4.2. SPECIFIC OBJECTIVES

- Conduct exploratory analysis of financial time series, identifying patterns, trends, seasonality, outliers, and correlations among relevant variables.
- Implement different neural network architectures: MLP, LSTM, CNN1D, and Transformers for asset return prediction.
- Compare predictive performance of the architectures using quantitative metrics (RMSE, MAE, MAPE, R^2).
- Assess the models' generalization ability by testing on different financial assets and time windows.
- Propose recommendations for the use of the most effective architectures, discussing their advantages and limitations in real financial market applications.
- Validate models using precision, recall, F1-score, and accuracy.

5. RESEARCH HYPOTHESES

- H1: Recurrent deep learning architectures (LSTM) have higher predictive accuracy in financial time series than traditional feedforward architectures.
- H2: Attention-based or transformer models (Transformers) can better capture market noise and generalize results across different financial assets.
- H3: Deep learning models exhibit lower overfitting compared to traditional machine learning models (e.g., linear regression or random forest) when applied to large volumes of financial data.

- H4: MLP networks may be less efficient at capturing temporal dependencies in financial series.
- H5: Increasing model complexity (number of layers and units) improves predictive performance only up to a point, after which overfitting risk becomes significant even with early stopping.

6. RESEARCH PROBLEM

Which deep learning architectures demonstrate the greatest generalization capability in predicting financial returns, and which stand out in terms of evaluation metrics?

7. RESEARCH METHODOLOGY

7.1. DATA DESCRIPTION

This study used publicly available real financial time series data from the Brazilian stock exchange B3 (Brasil, Bolsa, Balcão). Historical data were collected from B3's official COTAHIST files, provided annually in .ZIP format containing structured text files (.txt) according to B3's layout. Data from the last four years (2021–2024) were extracted, containing variables on quotation date, ticker, maximum price, minimum price, and closing price. The data were organized into CSV files per year with the following variables:

VARIABLE NAME	TYPE	DESCRIPTION
data	Data	Trading date of the asset
ticker	Nominal	Asset identification name
preco_abertura	Numeric	First traded price of the asset
preco_max	Numeric	Highest price during the day
preco_min	Numeric	Lowest price during the day
preco_fechamento	Numeric	Closing price of the asset
volume	Numeric	Total trades multiplied by respective prices

In addition, the number of rows contained in each dataset and the quantity of unique tickers were also observed.

DATASET	ROWS	UNIQUE TICKERS
Dataset 2021	1831862	101995
Dataset 2022	2117440	121201
Dataset 2023	2257424	133292
Dataset 2024	2635561	167131

7.2. NEURAL NETWORK ARCHITECTURE DEFINITIONS

The choice of LSTM, CNN1D, Transformers, and MLP architectures is based on each model's characteristics and applicability to asset return prediction.

- **LSTM**: Recurrent networks capable of processing sequential data, overcoming gradient vanishing or exploding issues through gated memory units. Used for capturing long-term dependencies.
- **CNN 1D**: Originally applied in computer vision, CNNs are effective in financial price prediction by detecting local patterns in time series. Used to extract short-term features and repetitive patterns in asset returns.
- **Transformers**: Based on attention and self-attention mechanisms, allowing efficient relation of distant events in a series. Employed for capturing long-range dependencies and outperforming recurrent and convolutional networks in complex tasks.
- **MLP**: Multilayer perceptrons process inputs densely and non-sequentially. Used as a baseline for comparing performance gains of specialized models (LSTM, CNN, Transformers).

7.3. DATA CLEANING AND EXPLORATORY ANALYSIS

- **VERIFICATION OF NULL, EMPTY, OR ZERO VALUES**

Initially, an analysis was conducted to identify rows containing at least one column with a null, empty, or zero value. Rows that met these criteria were removed from the dataset, ensuring greater consistency and reliability of the data used in the models.

- **CREATION OF THE RETURN COLUMN**

Considering that the objective of the study is to predict financial returns, a specific column called return was created. This column corresponds to the result of the difference between the closing price and the opening price, divided by the opening price, representing the daily percentage variation of the asset.

- **REMOVAL OF DUPLICATES**

It was verified that the dataset contains no duplicate records, ensuring that each row corresponds to a unique trading event.

- **OUTLIER TREATMENT**

Outliers in the dataset were not treated, given that the distribution of the variables of interest exhibits right skewness. Removing or modifying these extreme values could distort the natural variability of the data and negatively impact the predictive capacity of the models, since outliers reflect significant movements in financial returns that the model needs to learn. Thus, all values were retained, preserving the integrity of the original distribution and allowing the neural network architectures to learn real patterns, including extreme events, which are relevant for more accurate predictions in financial time series.

- **NORMALIZATION / STANDARDIZATION**

The normalization and standardization step of the numerical variables involved applying MinMax normalization to all numerical features in the dataset, with the goal of scaling the values between 0 and 1. This normalization was performed to ensure that the high value of the volume, compared to other variables, would not cause overfitting.

- **ENCODING OF CATEGORICAL VARIABLES**

Categorical variables will not be used in the model.

- **ELIMINATION OF EXAMPLES WITH FEWER THAN 250 INSTANCES**

For the development of the neural network model, it is necessary that the input related to each ticker contains at least 250 examples, which corresponds to the approximate number of trading days in a year. In this way, having continuous data for temporal analysis contributes to achieving better results.

- **CORRELATION MATRIX**

Due to the small number of variables, a correlation matrix will not be included in this study.

- **NUMBER OF EXAMPLES FOR EACH VARIABLE**

For the exploratory analysis, the datasets were divided by year: 2021, 2022, 2023, and 2024. For each year, the number of examples (rows) was verified.

YEAR	**NUMBER OF EXAMPLES**
2021	1628332
2022	1887323
2023	1990322
2024	2265471

	preco_abertura	preco_max	preco_min	preco_fechamento	volume	retorno
count	1.628332e+06	1.628332e+06	1.628332e+06	1.628332e+06	1.628332e+06	1.628332e+06
mean	5.742379e+01	5.912532e+01	5.582567e+01	5.744157e+01	4.850943e+08	1.299595e-02
std	5.633530e+02	5.727851e+02	5.558302e+02	5.645718e+02	7.160942e+09	3.808237e-01
min	1.000000e-02	1.000000e-02	1.000000e-02	1.000000e-02	1.000000e+00	-9.956522e-01
25%	3.600000e-01	4.100000e-01	3.200000e-01	3.600000e-01	1.500000e+05	-2.803738e-02
50%	1.600000e+00	1.740000e+00	1.500000e+00	1.600000e+00	1.360900e+06	0.000000e+00
75%	1.265000e+01	1.287000e+01	1.243000e+01	1.264000e+01	1.144000e+07	1.486520e-02
max	1.295160e+05	1.295160e+05	1.295160e+05	1.295160e+05	2.471088e+12	1.940000e+02

2021

	preco_abertura	preco_max	preco_min	preco_fechamento	volume	retorno
count	1.887323e+06	1.887323e+06	1.887323e+06	1.887323e+06	1.887323e+06	1.887323e+06
mean	5.919021e+01	6.130324e+01	5.712483e+01	5.921150e+01	3.743698e+08	1.562686e-02
std	5.541037e+02	5.668977e+02	5.423721e+02	5.551114e+02	6.582166e+09	5.902354e-01
min	1.000000e-02	1.000000e-02	1.000000e-02	1.000000e-02	1.000000e+00	-9.950495e-01
25%	3.000000e-01	3.500000e-01	2.700000e-01	3.000000e-01	1.078075e+05	-2.846488e-02
50%	1.320000e+00	1.430000e+00	1.230000e+00	1.330000e+00	9.600000e+05	0.000000e+00
75%	7.800000e+00	7.990000e+00	7.620000e+00	7.800000e+00	8.042438e+06	1.758191e-02
max	1.168860e+05	1.168860e+05	1.168860e+05	1.168860e+05	2.487180e+12	6.490000e+02

2022

	preco_abertura	preco_max	preco_min	preco_fechamento	volume	retorno
count	1.990322e+06	1.990322e+06	1.990322e+06	1.990322e+06	1.990322e+06	1.990322e+06
mean	4.338949e+01	4.477701e+01	4.207376e+01	4.341316e+01	2.934385e+08	1.686440e-02
std	5.231317e+02	5.309452e+02	5.163104e+02	5.238585e+02	4.997054e+09	4.856598e-01
min	1.000000e-02	1.000000e-02	1.000000e-02	1.000000e-02	1.000000e+00	-9.950000e-01
25%	2.500000e-01	2.900000e-01	2.200000e-01	2.500000e-01	6.760000e+04	-2.206736e-02
50%	1.100000e+00	1.180000e+00	1.010000e+00	1.100000e+00	5.599440e+05	0.000000e+00
75%	6.440000e+00	6.600000e+00	6.290000e+00	6.450000e+00	4.990200e+06	1.438849e-02
max	1.280060e+05	1.280060e+05	1.280060e+05	1.280060e+05	1.774876e+12	3.290000e+02

2023

	preco_abertura	preco_max	preco_min	preco_fechamento	volume	retorno
count	1.628332e+06	1.628332e+06	1.628332e+06	1.628332e+06	1.628332e+06	1.628332e+06
mean	5.742379e+01	5.912532e+01	5.582567e+01	5.744157e+01	4.850943e+08	1.299595e-02
std	5.633530e+02	5.727851e+02	5.558302e+02	5.645718e+02	7.160942e+09	3.808237e-01
min	1.000000e-02	1.000000e-02	1.000000e-02	1.000000e-02	1.000000e+00	-9.956522e-01
25%	3.600000e-01	4.100000e-01	3.200000e-01	3.600000e-01	1.500000e+05	-2.803738e-02
50%	1.600000e+00	1.740000e+00	1.500000e+00	1.600000e+00	1.360900e+06	0.000000e+00
75%	1.265000e+01	1.287000e+01	1.243000e+01	1.264000e+01	1.144000e+07	1.486520e-02
max	1.295160e+05	1.295160e+05	1.295160e+05	1.295160e+05	2.471088e+12	1.940000e+02

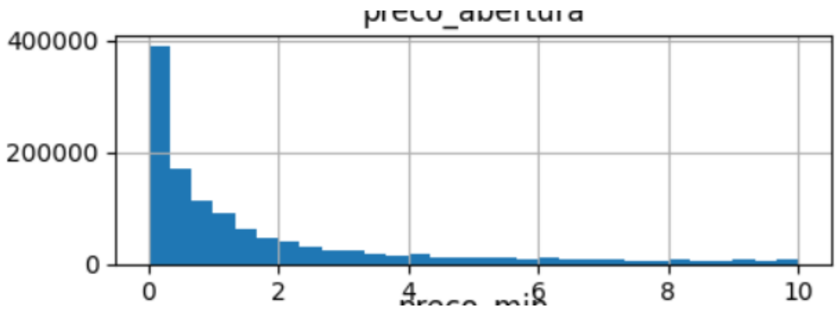
2024

Price Distribution: The average opening and closing prices range approximately from 41 to 57 units, with high standard deviation, reflecting the presence of extreme values (outliers). Minimum values are often close to zero, while maximum values exceed 100,000 units.

Trading Volume: The average daily trading volume ranges from approximately 2.46 × 10^8 to 4.85 × 10^8, with maximum values reaching 2.48 × 10^12, indicating days of exceptionally intense trading activity.

Financial Return: The average daily return is small, around 1.3% to 1.6%, while the standard deviation is relatively high due to occasional extreme fluctuations. The median return is zero in all years, indicating that most days have small variations.

- **VARIABLE DISTRIBUTION**



Distribution of opening price

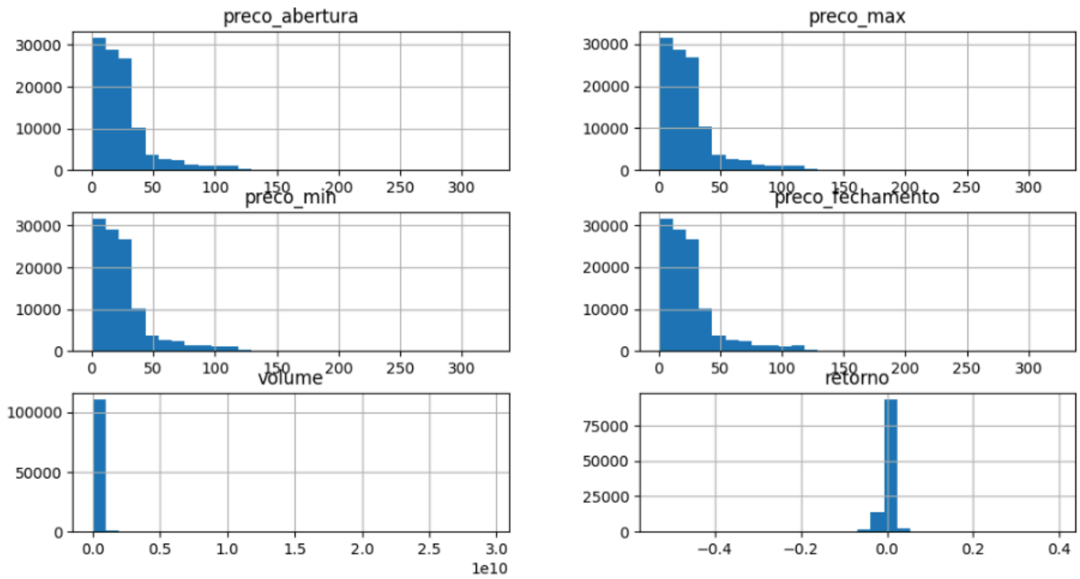
The following distribution describes the behavior of a variable in the dataset within the range of 0 to 10. The original dataset showed a large concentration of examples between 0 and 1. Even with this concentration, the dataset still presented “noise” related to very high values.

- **DISTRIBUTION AFTER SEPARATION OF BUSINESS DAYS**

After removing tickers that contained fewer than 230 examples, a new evaluation of the distribution was carried out, which showed it to be less concentrated and with more balanced values:

- 2021

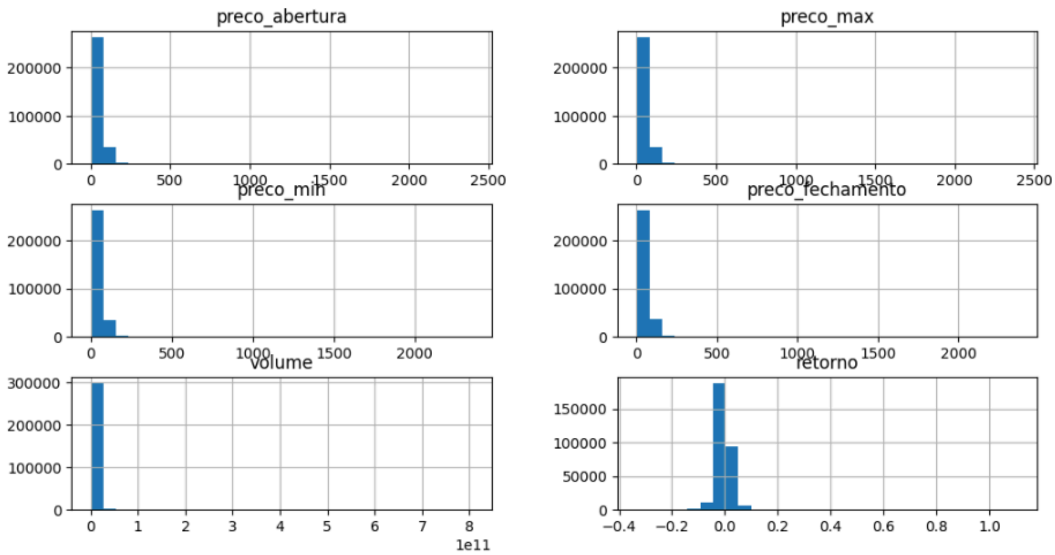
```python
df2021.hist(bins=30, figsize=(12,6))
plt.show()
```



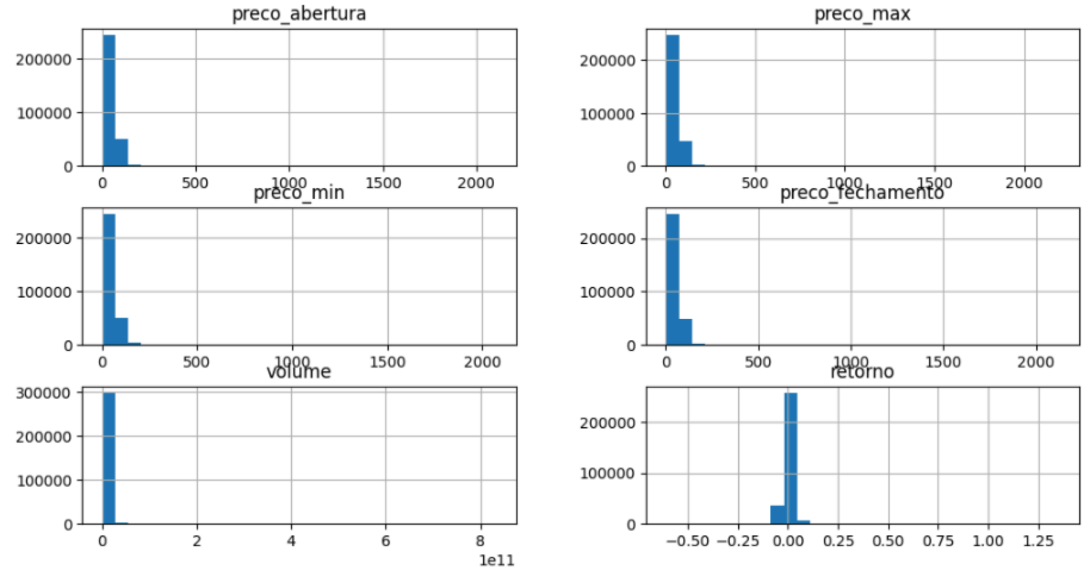
- 2022

Data related to the year 2022 still shows noise, regarding a few data points with high values.

```python
df2022.hist(bins=30, figsize=(12,6))
plt.show()
```

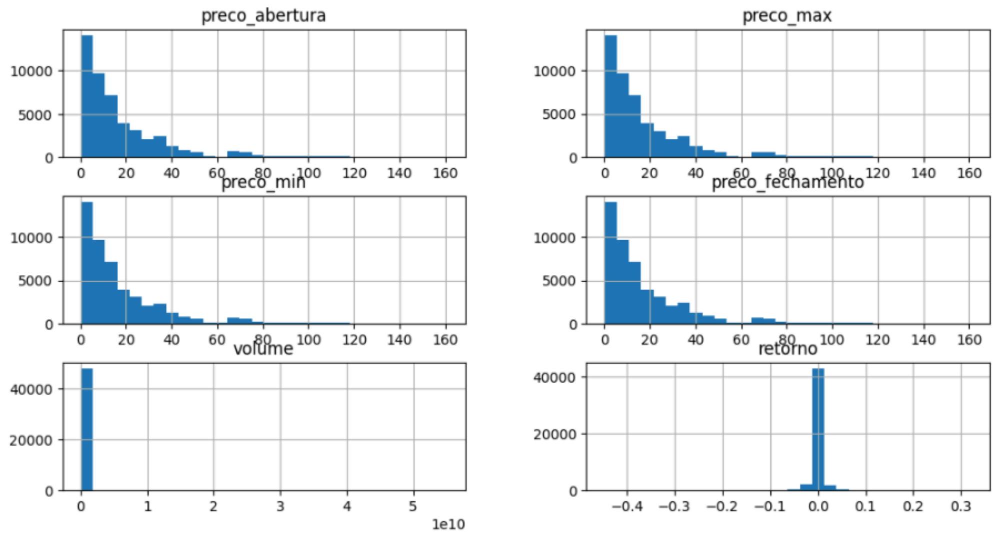


- 2023



- ## 2024

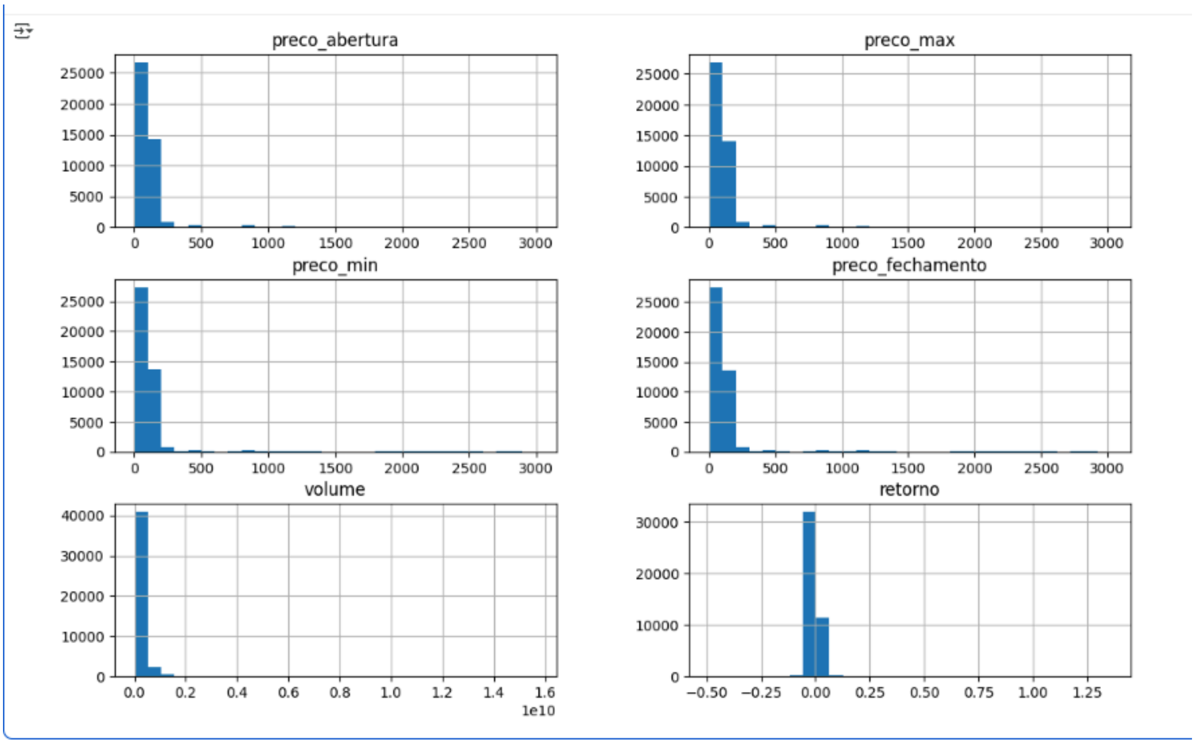
```
df2024.hist(bins=30, figsize=(12,6))
plt.show()
```



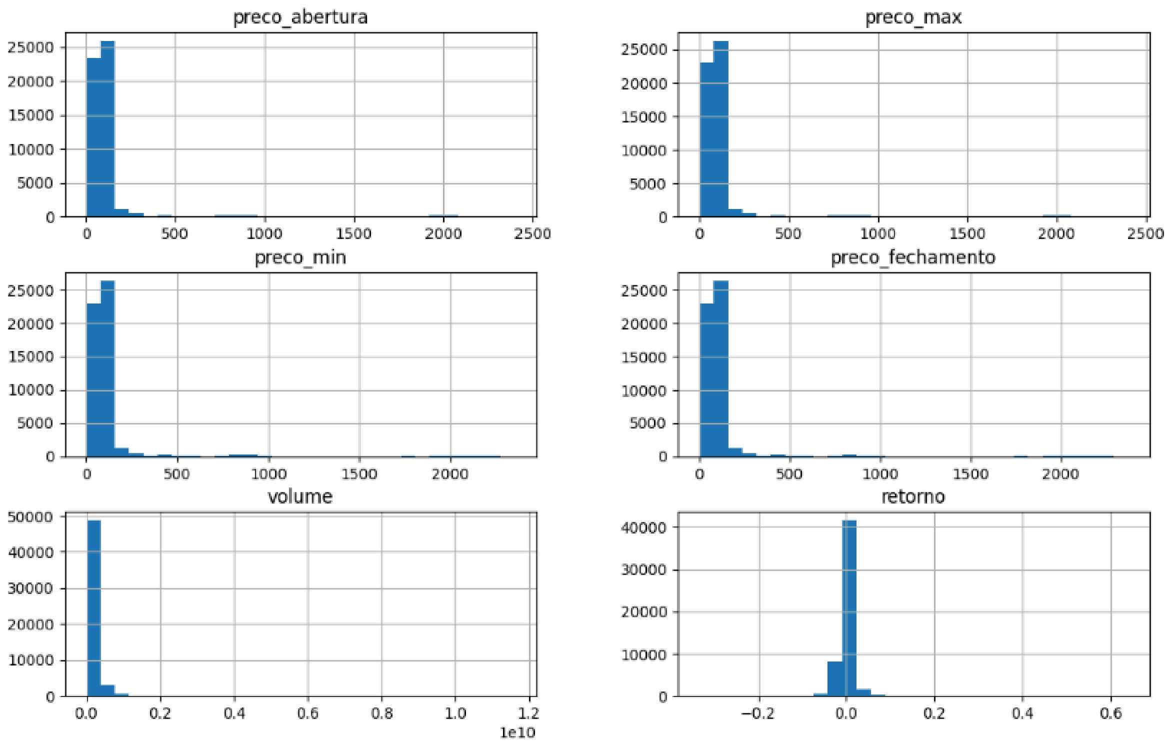
- ## DISTRIBUTION AFTER SEGREGATION BY MARKET TYPE AND BDI CODE

For preprocessing the data in neural networks, the data was segregated to analyze tickers related to the spot market and real estate investment funds. In this way, we obtained a distribution closer to normal.

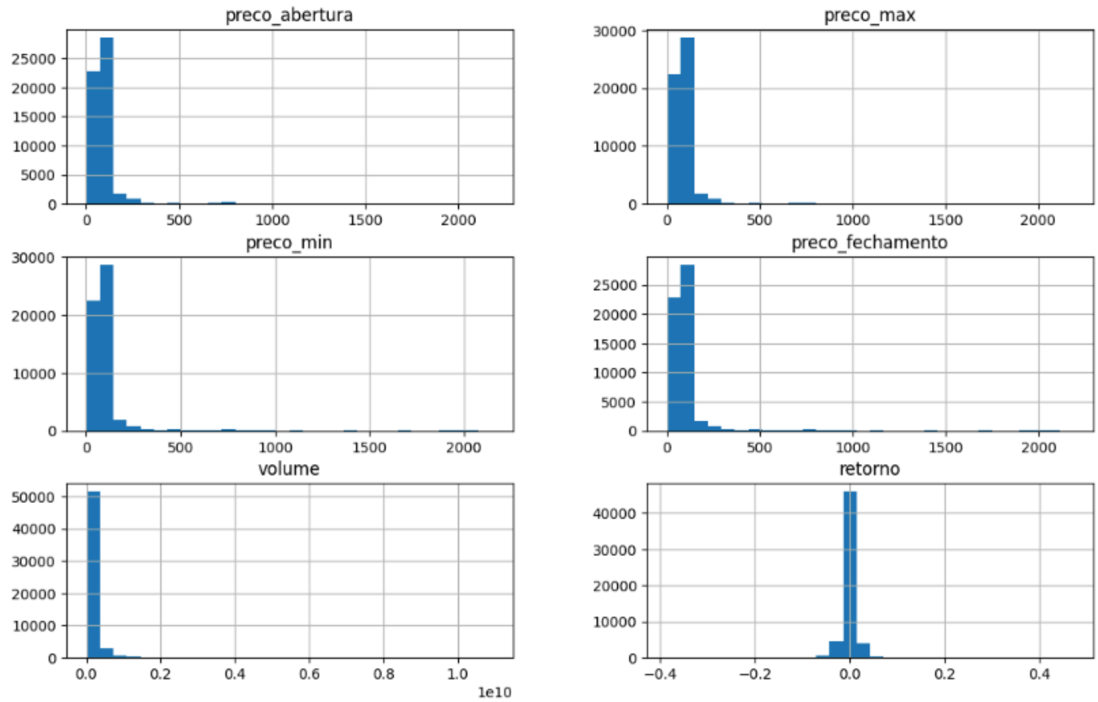
- **2021**


preco_abertura
preco_max
preco_min
preco_fechamento
volume
retorno
1e10

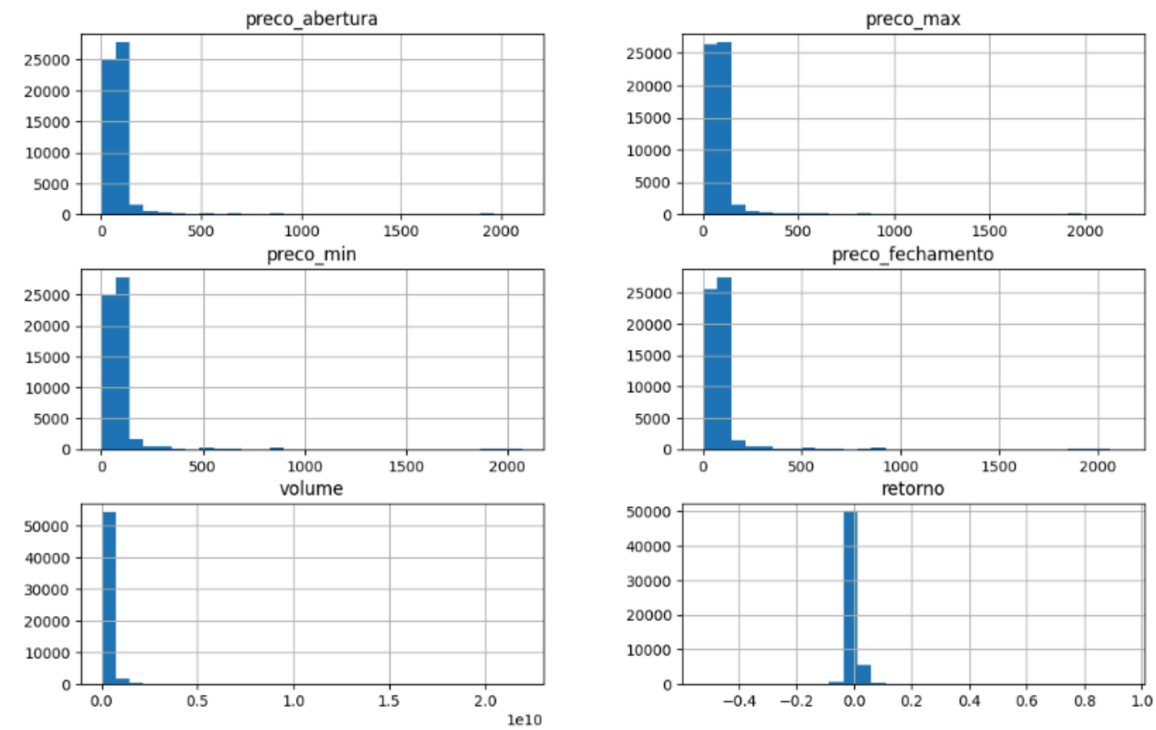
- **2022**


preco_abertura
preco_max
preco_min
preco_fechamento
volume
retorno
1e10

- **2023**



- **2024**



7.4. DATA PREPROCESSING

For the use of data in the tested neural networks, some preprocessing steps were necessary to ensure the networks could fully learn from the dataset. The first step was to organize the dataset temporally, that is, placing the oldest data first, following a chronological order. This organization was done together with the grouping of tickers, so that the same ticker remained grouped in chronological order. This is particularly important for LSTM networks.

Next, the dataset was filtered for a specific ticker of real estate investment fund assets over a 4-year period (from 2021 to 2024). After this organization, the dataset was split into training and testing sets. This step is essential for models focused on time series, as it is important that the dataset is correctly divided so the model can learn patterns chronologically. Thus, the dataset was divided into training data covering the months from January to August, allowing the model to also learn from seasonality, while the testing data comprised the following months.

7.5. ARCHITECTURES USED

MLP: This architecture is a simple multilayer perceptron designed for regression. The model starts with a dense layer of 64 neurons with ReLU activation, followed by an intermediate dense layer of 32 neurons, also with ReLU, allowing the network to capture nonlinear patterns in the data. The output layer consists of a single linear neuron to predict continuous values. Training uses the Adam optimizer, MSE (mean squared error) loss function, and MAE (mean absolute error) metric, with a batch size of 32 and a configurable number of epochs.

Transformers: The model was structured with two Transformer Encoder layers, each containing Multi-Head Attention with 4 heads, layer normalization, and an internal feed-forward network of 64 neurons. After stacking the encoders, the vectors were flattened, passed through a dense layer of 64 neurons with ReLU activation and Dropout, and finalized with a dense output layer with a single neuron for regression. The optimizer used was Adam with MSE loss.

LSTM: This architecture was based on the model presented by [4], with a difference in normalization. In the referenced work, the author normalizes the data within a specific layer of the network, allowing batch normalization. However, in this study, Z-Score normalization was replaced by MinMax normalization applied outside the neural network. The architecture was built to capture temporal dependencies in the series, starting with an LSTM layer with 125 units and a second LSTM layer with 75 units, using L2 regularization to prevent overfitting. The output consists of a single neuron with sigmoid activation and L1 regularization. The model was initially trained with the Adagrad optimizer and a binary loss function, then adjusted for continuous regression using MSE.

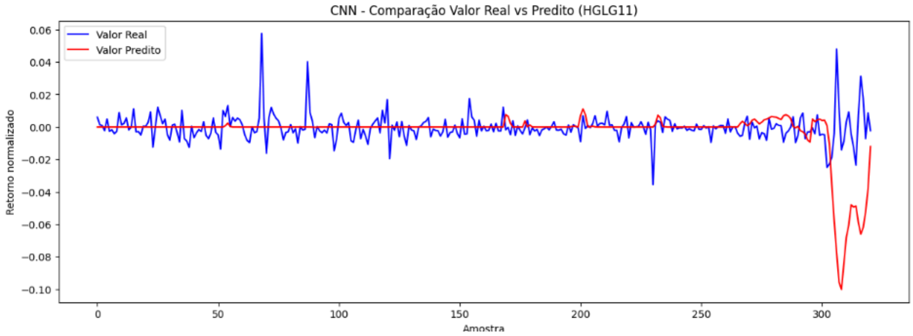
CNN: The convolutional network consists of two Conv1D layers to extract local patterns from the series (64 filters in the first block and 32 in the second, both with a kernel size of 3 and ReLU activation), followed by MaxPooling1D layers to reduce dimensionality and Dropout for regularization. After flattening, there is a dense layer of 64 ReLU neurons and a dense output layer with a single neuron for predicting continuous values.

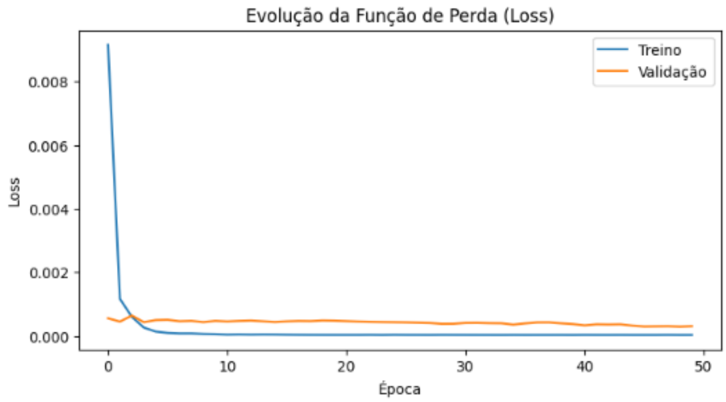
8. RESULTS

- **CNN**

```
RMSE: 0.017541
MAE:  0.007930
MAPE: 7158870591.917319
```

```
R²:    -3.694797
```

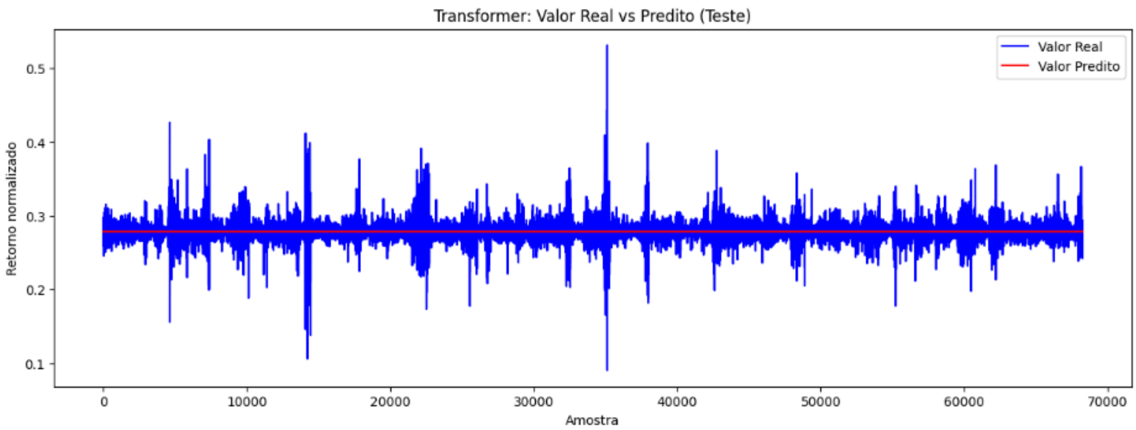


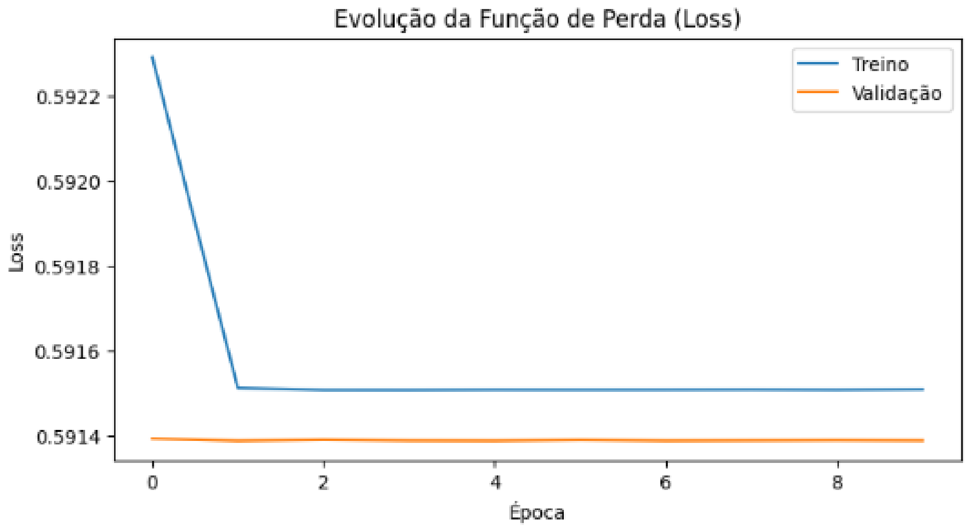


Although the RMSE and MAE values indicate low absolute errors in numerical terms, the extremely high MAPE and negative R^2 reveal that the model failed to adequately generalize the patterns from the training set. A negative coefficient of determination implies that the model's predictions are worse than simply using the mean of the actual values, suggesting that the CNN did not capture the underlying temporal and nonlinear relationships in the returns.

- **TRANSFORMERS**

```
RMSE: 0.008619
MAE:  0.004793
MAPE: 0.017349
R²:   -0.000720
```

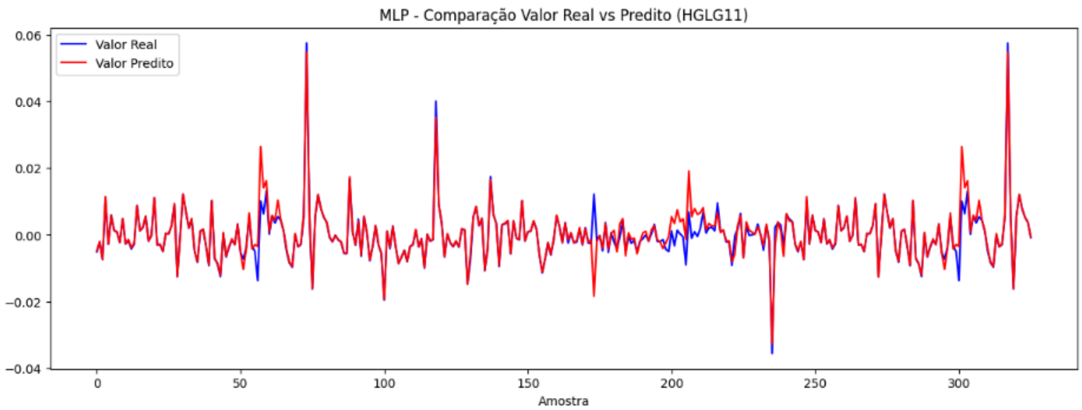




Despite the seemingly low absolute error values (RMSE and MAE), the negative R^2 indicates that the model could not explain the variance in the data. This behavior is consistent with the Transformer predicting nearly constant values close to zero, indicating that the network did not learn meaningful patterns during training.

- **MLP**

```
RMSE: 0.002852
MAE:  0.000991
MAPE: 22684243453.417149
R²:   0.868868
```

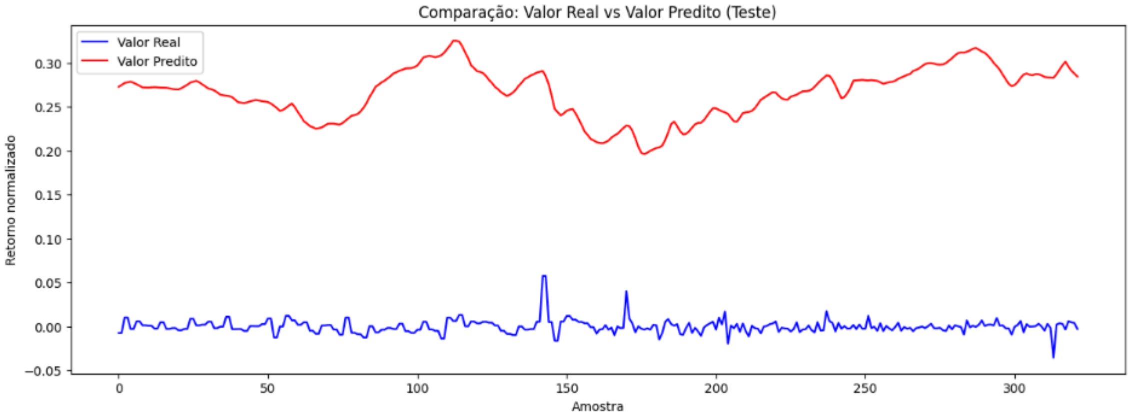


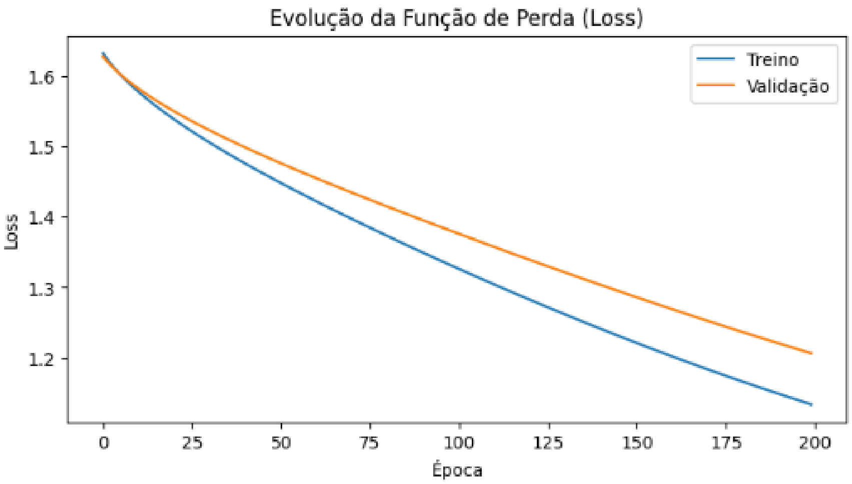
The MLP (Multilayer Perceptron) model showed inconsistent results, suggesting overfitting. While the R^2 indicates a good fit to the training data, the extremely high MAPE demonstrates that the model failed to generalize to the test data, resulting in distorted and unrealistic predictions.

- **LSTM**

```
RMSE: 0.267844
```

```
MAE:  0.266264
MAPE: 22616927431685.050781
R²:   -1145.322708
```





The LSTM (Long Short-Term Memory) model performed unsatisfactorily, showing difficulties in capturing temporal patterns in the data. The performance metrics were significantly high, with extremely elevated RMSE and MAPE, and an R^2 indicating that the model performed much worse than the mean of the samples. The highly negative R^2 reinforces the hypothesis that the model failed to generalize, producing predictions drastically different from actual values.

9. CONCLUSION

This study aimed to evaluate the performance of different deep learning architectures applied to financial time series forecasting, using asset returns as the target variable. Overall, the results highlighted the intrinsic complexity of the financial market, where high volatility, noise, and non-stationary data make it difficult to construct robust predictive models.

Regarding the hypotheses:

H1 — predicting the superiority of recurrent architectures (LSTM) over feedforward networks — was not fully confirmed. The LSTM showed signs of overfitting and poor generalization, suggesting that although it captures temporal dependencies, its ability to consistently predict future returns in this context is limited.

H2, which proposed that attention-based models (Transformers) would be more effective in capturing noise and generalizing patterns, was partially validated. The Transformer showed good performance in terms of positive R^2 and low average error, but in some runs it tended to produce static predictions (values near zero), indicating its efficiency depends strongly on data quality and quantity.

H3, regarding lower overfitting in deep learning models compared to traditional techniques, was partially refuted. Even with large datasets, neural networks can overfit when regularization and complexity control are inadequate. The MLP's behavior reinforces this observation: despite its simplicity, it overfit the training set and had low generalization ability.

H4 was confirmed, as the MLP, lacking memory or convolution mechanisms, was less effective in capturing temporal dependencies and sequential price patterns.

H5 was also corroborated, showing that increasing model complexity does not necessarily improve performance. Adding layers and units beyond a certain point increased the risk of overfitting, even with techniques like early stopping or dropout.

In general, financial time series forecasting remains a challenging problem, where advanced deep learning models can offer some gains but are far from providing a definitive solution.

10. REFERENCES

[1] GUEDES, Anderson Cerqueira et al. **Computação evolutiva para otimização de carteiras de estratégias de negociação no mercado financeiro.** 2022.

[2] SOARES, Taís Rigor. **Previsão de Retornos de Ativos no Mercado Financeiro Através de Métodos de Otimização.** 2023. Dissertação de Mestrado.

[3] SHIMABUKURO, Camilo Ilzo. **Deep Learning aplicado à predição de retornos do Ibovespa: uma análise do desempenho da rede neural LSTM utilizando log-retornos e diferenciação fracionária**. 2024. Tese de Doutorado. Universidade de São Paulo.

[4] MESQUITA, Caio Mário; OLIVEIRA, Renato; PEREIRA, Adriano César Machado. **Utilização de uma rede neural LSTM e teste da razão de variância para previsões em séries de ativos da Bovespa**, 2020.

[5] GEZICI, Abdul Haluk Batur; SEFER, Emre. **Deep transformer-based asset price and direction prediction. IEEE Access**, v. 12, p. 24164-24178, 2024.