DEVELOPMENT AND COMPARISON OF MACHINE LEARNING MODELS APPLIED TO THE PREDICTION OF IBOVESPA PERFORMANCE TRENDS

João Vitor de Paiva Marcoti[1]

ABSTRACT:

The models presented in this paper were compared across the periods of 2010, 2015, and 2020 until 2023. It was established that for a model to be considered satisfactory, its precision must exceed the percentage of days in which the Ibovespa appreciated during the test period. Four initial experiments were conducted, with multiple technical analysis indicators as input variables. Recurrent Neural Network (RNN) models showed the best average performances. The hybridization of the best-performing models in the initial experiments did not surpass their individual performances in the last experiment. Even with accuracy and precision close to 60%, the best models still performed near the base model, so they can't be considered good enough to be used as a consistent investment strategy. For future work exploring the prediction of Ibovespa trends over longer periods, such as months or years, rather than just a single day, may provide better results.

Keywords: Deep learning. Financial time series. Forecasting. Statistics. Stock market prediction.

Data da submissão: 07-10-2025
Data do aceite: 10-24-2025

INTRODUCTION

A predictive model aims to anticipate events based on past occurrences, assisting in decision-making and proving pivotal in solving complex problems (Alves, 2019).

The development of a predictive model involves analyzing a collection of historical data to define a method that enables the prediction of a future event of interest. Thus, the accuracy of a predictive model involves assessing the equivalence between its forecast and the actual occurrence (Alves, 2019).

The most critical aspect of a predictive model is identifying the best variables to define an event or characteristic (Alves, 2019).

Machine learning models employing Artificial Neural Networks (ANNs) are the most used technique in prediction algorithms, as they consistently demonstrate good performance in stock price prediction. Additionally, algorithms based on Decision Trees (DTs) are also widely employed in such applications, effectively describing the cause-and-effect relationship between information (Tsai; Wang, 2009).

Artificial Neural Networks are systems based on a collection of connected units or nodes called artificial neurons, inspired by neurons in an animal's biological brain. Like synapses in

[1] Engenheiro Elétrico (UEM) e Economista (Unicesumar). joaovmarcotti@hotmail.com

a biological brain, each connection can transmit a signal to other neurons. An artificial neuron receives signals, processes them, and then it can signal to the neurons connected to it. Typically, neurons are organized into layers, with different layers performing various transformations on their inputs. Signals travel from the first (input) layer to the last (output) layer, passing through intermediate (hidden) layers multiple times (Hardesty, 2017). The hidden layer can add a non-linear component to the neural network through a nonlinear function such as a sigmoid (Hyndman; Athanasopoulos, 2021).

The simplest neural network contains no hidden layers, just input and output layers, and is equivalent to a linear regression model. A layer is composed of coefficients, and each coefficient has a weight associated with it, and the output value is obtained by the combination of the input values (Hyndman; Athanasopoulos, 2021).

Initially an ANN has random values as weights, and these values change with the training iterations through the minimization of a cost or error function (such as MSE or RMSE) (Hyndman; Athanasopoulos, 2021).

The most basic model of an ANN is the Multilayer Perceptron (MLP). The aim of this model is to approximate a function F. For a classifier, Y = F(X) maps an input X to a category of Y, defining a mapping Y = F(X; θ) and learning the parameter values θ that result in the best approximation of the function Y. This model represents a feedforward neural network as information flows through neurons from the initial to the final layers, reaching the output without feedback connections. If such connections were present, it would be a Recurrent Neural Network (RNN) (Goodfellow; Bengio; Courville, 2016).

As mentioned earlier, RNN involves feedback connections between neuron layers. This property allows neural networks of this type to store information while processing new inputs, making them suitable for tasks that require considering previous inputs, making them suitable for applications involving a significant sequence of values, such as a time series. Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) are examples of RNNs (Goodfellow; Bengio; Courville, 2016).

Decision Trees is a technique that can be divided into two types, varying according to the type of value assumed by the target variable: Classification Trees, assuming discrete values, and Regression Trees, assuming continuous values (Wu, 2007). However, according to Silva and Sátiro (2022), the use of algorithms of this type has become less common due to the emergence of models that combine multiple Decision Trees, known as bagging and boosting models. These models yield better results than individual Decision Trees applications (Nabipour et al., 2020).

Bagging models, such as Random Forest, work by associating Decision Trees in parallel, with the final result given by the average of individual results. On the other hand, boosting models, such as Gradient Boosting, Light Gradient Boosting, and X Gradient Boosting, involve a sequential association, with each Decision Tree aiming to improve the result of the preceding tree (Silva; Sátiro, 2022).

Additionally, the Support Vector Machine (SVM) method can also be employed in predictive models. SVM is a highly versatile model capable of performing linear and non-linear classifications, regressions, and even outlier detection. This model aims to draw hyperplanes to separate input data in a way that the ideal hyperplane maximizes the distance between the two closest sample points (Géron, 2019).

Naive Bayes models have a relatively simple mathematical basis, founded on principles of statistical probability through the application of Bayes' theorem and the assumption of conditional independence among their input variables. Among the types of Naive Bayes algorithms, three stand out: Gaussian, Bernoulli, and Multinomial, named based on the data distribution they are derived from (Brownlee, 2020).

Furthermore, the combination of different machine learning techniques has shown better performances compared to employing a single technique (Tsai; Wang, 2009).

A machine learning model can have its performance evaluated through different methods, depending on the task proposed and the model applied. A classification model is usually evaluated based on its accuracy, that measures the proportion of cases in which the model produces the correct output (Goodfellow; Bengio; Courville, 2016).

Besides accuracy, it is possible to use other metrics to evaluate different levels of performance of classifiers: precision, recall (or sensitivity) and specificity (Boehmke; Greenwell, 2020):

- Precision measures how accurately does the classifier predicts the positive prediction (for the number of predictions made, how many were correct).
- Recall represents how accurately does the model classifies positive events (for the number of positive events, how many were correctly predicted).
- Specificity measures how accurately does the model classifies negative events (for the number of negative events, how many were correctly predicted).

According to Boehmke and Greenwell (2020), a good binary classifier has high precision and sensitivity, so the model performs well when it predicts that an event will happen and also when it won't happen and minimizes both false positives and false negatives. To capture it it's often applied the ROC curve (Figure 1), which is represented by plotting the false positive rate as the x axis and the true positive rate as the y axis.

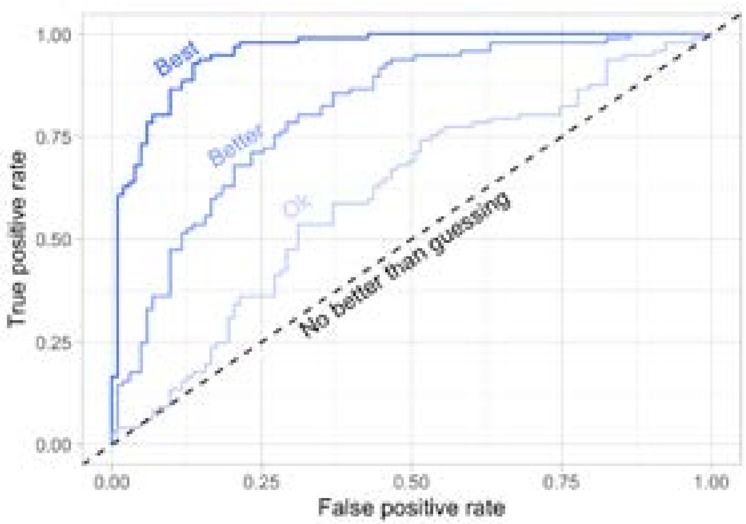

Figure 1. Representation of the ROC curve (Boehmke; Greenwell, 2020).

The area under the ROC curve is known as AUC. The closer the curve is to the upper right corner, the larger its Area Under the Curve (AUC) is, and the better the model's performance is (Boehmke; Greenwell, 2020).

Regarding regression models, it's necessary to use other methods to evaluate their performance. The most common metrics are the Mean Squared Error (MSE) and the Root Mean Squared Error (RMSE) (Boehmke; Greenwell, 2020):

- MSE is equal to the average of the squared error (being error difference between the actual and predicted values). The squared component results in larger errors having larger penalties.
- RMSE represents the square root of the squared error (square root of the MSE) and has the same units as the response variable.

Thus, companies and stakeholders in stock markets worldwide have sought techniques and tools that enable the prediction of asset performance over a specific period and aid in decision-making processes. Predicting the value of stocks traded on the exchange is a challenging and complex process due to the unpredictable nature of their long-term movements (Nabipour et al., 2020).

There are two methodologies for predicting stock values: fundamental analysis and technical analysis. Fundamental analysis focuses more on a company's accounting information than on its traded stock, relying on the history of its financial statements. On the other hand, technical analysis is based on the past movements of a company's stock, including the patterns formed by it. Therefore, it is possible to apply machine learning techniques and models to the technical analysis of a stock (Mohri, 2018).

In recent years, many researchers have used machine learning algorithms to create tools and analyze historical financial data to assist in investment decision-making (Obthong et al, 2020).

Financial market-related news and data collected from social networks were used as inputs by Jeong, Kim e Yoon (2018) to train a model to assist on decision-making related to stocks.

Accurate results directly depend on the correct choice of the employed machine learning model and the type of collected information (Obthong et al, 2020). Table 1 resumes the use of different quantitative models in stock performance prediction applications according to Obthong et al. (2020).

Table 1. Use of quantitative models in stock performance prediction (Obthong et al, 2020).

Model	Used
Clustering	
K-Means	X
SOM - Self Organizing Maps	X
Hierarchical clustering	X
Predicting	
RF - Random Forest	X
SVM - Support Vector Machine	X
MLP - Multilayer Perceptron	X
LSTM - Long Short-Term Memory	X
RNN - Recurrent neural networks	X
GAs - Genetic Algorithms	X
KNN - K Nearest Neighbor	X
SVR - Support Vector Regression	X
MCS - Monte Carlo Simulation	X

Model	Used
ANNs - Artificial Neural Networks	X
CART - Classification and Regression Trees	X
GP - Gaussian Processes	X
BSM - Black Scholes Model	X
GRNN - Generalized Regression Neural Network	X
RBF - Radial Basis Function Neural Networks	
BPNN - Back propagation neural network	X
LR - Logistic Regression	X
HMM - Hidden Markov Model	X
Classifying	
SVM - Support Vector Machine	X
KNN - K Nearest Neighbors	X
LR - Logistic Regression	
ANNs - Artificial Neural Networks	

On the other hand, Table 2 presents the percentage of each machine learning model used for predicting stock values present in 30 papers selected by Kumar, Sarangi e Verma (2021).

Table 2. Percentage of each model used in the 30 papers studied (Kumar; Sarangi; Verma, 2021).

Model	Usage (%)
SVM - Support Vector Machine	21
NN - Neural Network	33
ANN - Artificial neural network	24
CNN - Convolutional neural network	6
RNN - Recurrent neural network	6
SVR - Support vector regression	3
GAN - Generative adversarial network	3
NB - Naïve Bayes	3
Hybrid approach	9

Time series data consists of continuous information collected over time (annually, monthly, weekly, daily, or even hourly, minutely, or secondly). The closing price of a stock is an example of time series data (Obthong et al, 2020).

A time series consists of four components: trend, cycle, seasonality, and remainder. The trend represents the data's direction over a long period, which can be stable, increasing, or decreasing, and it does not have to be linear. The cycle refers to movement patterns lasting over a year or longer, typically influenced by economic or business cycles and without a fixed frequency. Seasonality reflects data movements influenced by specific periods, influenced by natural, business, social, or cultural causes, always with a fixed and known frequency. The remainder component is equal to what lefts after trend, seasonality and cycle components have been subtracted from the data (Hyndman; Athanasopoulos, 2021), and represents short-term irregular movements in the time series, caused by events like disasters, wars, or diseases, often impacting companies for short durations (Obthong et al, 2020).

Despite good performance during training and testing phases, historical data-based prediction models do not guarantee accuracy due to uncertainties influenced by political and economic conditions, exemplified by events such as the Covid-19 pandemic or conflicts like the Ukraine-Russia war (Obthong et al, 2020).

Certain machine learning models have shown good performance in predicting asset values with a low error rate, including ANN, RNN, LSTM, SLSTM, and BLSTM (Obthong et al, 2020). However, in Bluvol (2022) ARIMA models outperformed artificial neural networks models.

1 MATERIALS AND METHODS

The methodology of this study involved developing and comparing multiple machine learning models to predict the Ibovespa next-day trend, classifying it as appreciation (1) or depreciation (0) across five experiments.

In the first experiment, explanatory variables included the closing ratio (closing price divided by the rolling average closing price over the period) and trend (number of days the index increased in the previous period) over the last 3, 5, and 10 business days for indices such as Ibovespa, S&P500, Gold, Euronext 100, SSE Composite - Shanghai, 5-year US Treasury bonds, and the exchange rates of the US dollar and euro relative to the Brazilian real

In the second experiment, a linear relationship was calculated between each explanatory variable from the first experiment and the Ibovespa value of the following day. The models were then fed with the five highest positive correlations and the five lowest negative correlations.

For the third experiment, explanatory variables included multiple technical analysis indicators: simple and weighted moving average of the closing price, momentum, stochastic K, stochastic D, Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD), Signal line, Larry Williams ratio and Commodity Channel Index (CCI).

These indicators were calculated relative to the Ibovespa closing prices for 3, 5, and 10 previous business days based on the equations presented in Figure 2. The values obtained were continuously arranged over time.

Name of indicators	Formulas
Simple 10-day moving average	$\frac{C_t + C_{t-1} + \cdots + C_{t-10}}{10}$
Weighted 10-day moving average	$\frac{((n) \times C_t + (n-1) \times C_{t-1} + \cdots + C_{10})}{(n + (n-1) + \cdots + 1)}$
Momentum	$C_t - C_{t-n}$
Stochastic K%	$\frac{C_t - LL_{t-n}}{HH_{t-n} - LL_{t-n}} \times 100$
Stochastic D%	$\frac{\sum_{i=0}^{n-1} K_{t-i}\%}{n}$
RSI (Relative Strength Index)	$100 - \frac{100}{1 + \left(\sum_{i=0}^{n-1} Up_{t-i}/n\right) / \left(\sum_{i=0}^{n-1} Dw_{t-i}/n\right)}$
MACD (moving average convergence divergence)	$MACD(n)_{t-1} + 2/n + 1 \times (DIFF_t - MACD(n)_{t-1})$
Larry William's R%	$\frac{H_n - C_t}{H_n - L_n} \times 100$
A/D (Accumulation/Distribution) Oscillator	$\frac{H_t - C_{t-1}}{H_t - L_t}$
CCI (Commodity Channel Index)	$\frac{M_t - SM_t}{0.015 D_t}$

C_t is the closing price, L_t the low price, H_t the high price at time t, $DIFF$: $EMA(12)_t - EMA(26)_t$, EMA exponential moving average, $EMA(k)_t$: $EMA(k)_{t-1} + \alpha \times (C_t - EMA(k)_{t-1})$, α smoothing factor: $2/1 + k$, k is time period of k day exponential moving average, LL_t and HH_t mean lowest low and highest high in the last t days, respectively, $M_t : H_t + L_t + C_t/3$; $SM_t : \left(\sum_{i=1}^{n} M_{t-i+1}\right)/n$, $D_t : \left(\sum_{i=1}^{n} \mid M_{t-i+1} - SM_t \mid\right)/n$, Up_t means the upward price change, Dw_t means the downward price change at time t.

Figure 2. Equations of technical analysis indicators (Kara; Boyacioglu; Baykan, 2011).

Simple moving average measures the average value of the closing price in the last N periods and weighted moving average gives higher weights to recent values, making it more sensitive to recent price changes (Kara; Boyacioglu; Baykan, 2011).

Momentum measures the rate of change in the price over time. Positive values indicate an uptrend while negative values indicate a downtrend (Kara; Boyacioglu; Baykan, 2011).

Stochastic K compares the current closing price to the lowest low (LL) and highest high (HH) over the last N periods (Kara; Boyacioglu; Baykan, 2011). Stochastic D measures the simple moving average of stochastic K over the last N periods (Kara; Boyacioglu; Baykan, 2011).

Relative Strength Index (RSI) compares the magnitude of recent uptrends (Up) and downtrends (Dw) of price changes (Kara; Boyacioglu; Baykan, 2011).

MACD (Moving Average Convergence Divergence) compares the exponential moving averages of the last N + 9 and N + 23 periods (being N = 3, 5 and 10 business days in this paper) (Kara; Boyacioglu; Baykan, 2011).

Larry Williams ratio compares the current closing price and lowest price to the highest price of the last N periods on a 0 to 100 scale (Kara; Boyacioglu; Baykan, 2011).

Commodity Channel Index (CCI) is the most complex indicator in this paper. It measures the difference between the index's typical price (average value between the highest price, the lowest price and the closing price) and its moving average of the period N (Kara; Boyacioglu; Baykan, 2011).

In the fourth experiment, the technical analysis indicators applied to the third experiment were transformed into a deterministic form, representing an upward or downward trend in the Ibovespa as 1 and 0, respectively.

The transformation process from continuous to deterministic values followed these guidelines, based on Patel, Shah, Thakkar, and Kotecha (2015):

- Simple Moving Average = 1 if the closing value is greater than the simple moving average of the past N days. Otherwise, it's set to 0.
- Weighted Moving Average = 1 if the closing value is greater than the weighted moving average of the past N days; otherwise, it's set to 0.
- Momentum = 1 if the momentum is positive; otherwise, it's set to 0.
- Stochastic K = 1 if the current day's K stochastic value is greater than the previous day's; otherwise, it's set to 0.
- Stochastic D = 1 if the current day's D stochastic value is greater than the previous day's; otherwise, it's set to 0.
- Relative Strength Index (RSI) = 1 if the RSI is above 70 or lower than the previous day's RSI; otherwise, it's set to 0.
- Moving Average Convergence Divergence (MACD) = 1 if the current day's MACD value is greater than the previous day's; otherwise, it's set to 0.
- Larry Williams' Ratio = 1 if the current day's Larry Williams' ratio is greater than the previous day's; otherwise, it's set to 0.
- Commodity Channel Index (CCI) = 1 if the CCI is above 200 or lower than the previous day's CCI; otherwise, it's set to 0.

In the fifth experiment, hybrid LSTM-GRU and GRU-LSTM models were developed, using the same input variables from the third experiment to evaluate if these models outperform the previous ones.

Furthermore, to examine the models' performance across different time periods, the 5 experiments were conducted using three time frames: 2010-2023, 2015-2023, and 2020-2023, which were chosen to evaluate the models' performance across different market conditions and over short, medium, and long-term periods.

Thus, machine learning models of the following types were developed and evaluated: Gaussian Naive-Bayes (GNB), Bernoulli Naive-Bayes (BNB), Random Forest (RF), Support Vector Machine (SVM), Gradient Boosting (GB), X Gradient Boosting (XGB), Light Gradient Boosting (LGB), Multilayer Perceptron (MLP), Long Short Term Memory (LSTM), and Gated Recurrent Unit (GRU), as well as the hybrid LSTM-GRU and GRU-LSTM models.

The parameters of all models were determined through grid search parameterization, except for LSTM and GRU, whose parameters were selected through experimentation.

With the aim of profiting just from long positions in the index, without taking short positions, the precision metric (Equation 2) was used alongside accuracy (Equation 1) to evaluate the models. In this context, precision measures the probability that a model correctly predicts the Ibovespa's appreciation on the following day.

For a model to be considered satisfactory, its precision must exceed the percentage of days in which the Ibovespa appreciated during the test period, thus outperforming a base model that predicts index appreciation every day.

The market data was collected through the YFinance Python library, which provides companies' historical prices and financial statements directly from Yahoo Finance. The dataset has daily frequency and includes Ibovespa's daily high, low, and closing prices from January 2010 to September 2023.

The algorithms were developed using the Python programming language in the Google Colab environment. The Pandas library was utilized for data manipulation, while Seaborn and Matplotlib were used to visualize the historical index series and the heatmap of linear correlations between variables. Numpy was applied for mathematical operations, and Scikit-learn, Keras, and TensorFlow were utilized to develop machine learning models.

Therefore, this paper represents an exploratory and quantitative research endeavor, seeking to quantify, through statistical models, the behavior of a sample, in this case, the Brazilian stock market index, the Ibovespa.

2 RESULTS AND DISCUSSION

Four initial experiments were conducted to compare the performance of the developed machine learning models when fed with different variables and over different time periods (2010-2023, 2015-2023, and 2020-2023).

Table 3. Models' accuracy in the first experiment.

Model	2010	2015	2020	Mean
Bernoulli Naive Bayes	0.4778	0.4754	0.5059	0.4864
Gated Recurrent Unit	0.5109	0.5205	0.5378	0.5231
Gaussian Naive Bayes	0.4886	0.5042	0.5059	0.4996
Gradient Boosting	0.5157	0.4975	0.4743	0.4958
Light Gradient Boosting	0.5244	0.5059	0.4585	0.4963
Logistic Regression	0.4962	0.5178	0.4743	0.4961
Long-Short Term Memory	0.5228	0.5256	0.5458	0.5314
Multilayer Perceptron	0.5027	0.5178	0.4901	0.5035
Random Forest	0.4973	0.5144	0.4704	0.4940
Support Vector Machine	0.5244	0.5178	0.4901	0.5108
X Gradient Boosting	0.4930	0.4805	0.4901	0.4879

Table 4. Models' precision in the first experiment.

Model	2010	2015	2020	Mean
Bernoulli Naive Bayes	0.5023	0.4937	0.4959	0.4973
Gated Recurrent Unit	0.5533	0.5337	0.5274	0.5381
Gaussian Naive Bayes	0.5186	0.5355	0.4815	0.5119
Gradient Boosting	0.5391	0.5098	0.4651	0.5047
Light Gradient Boosting	0.5484	0.5180	0.4620	0.5095
Logistic Regression	0.5195	0.5296	0.4685	0.5059
Long-Short Term Memory	0.5460	0.5293	0.5357	0.5370
Multilayer Perceptron	0.5239	0.5178	0.4901	0.5106
Random Forest	0.5209	0.5241	0.4742	0.5064
Support Vector Machine	0.5244	0.5178	0.4901	0.5108
X Gradient Boosting	0.5177	0.4987	0.4793	0.4986

Based on the results obtained in the first experiment, it's observed that the accuracy of 7 models decreased with the reduction of the time period, while it increased in only 4 models, among them the 2 models with the highest average accuracy: GRU and LSTM, with 0.5231 and 0.5314 respectively.

Regarding precision, all models showed a performance loss with the reduction of the time period. The models with the best average performance were again GRU and LSTM with 0.5381 and 0.5370 respectively, along with Gaussian Naive Bayes, which also surpassed the average base model of 0.5108.

Table 5. Models' accuracy in the second experiment.

Model	2010	2015	2020	Mean
Bernoulli Naive Bayes	0.4826	0.4813	0.5099	0.4913
Gated Recurrent Unit	0.5321	0.5350	0.5480	0.5384
Gaussian Naive Bayes	0.4794	0.4898	0.5336	0.5009
Gradient Boosting	0.4913	0.5204	0.4625	0.4914
Light Gradient Boosting	0.4935	0.5272	0.5138	0.5115
Logistic Regression	0.5054	0.5051	0.4743	0.4949
Long-Short Term Memory	0.5288	0.5214	0.5200	0.5234
Multilayer Perceptron	0.4967	0.5051	0.5059	0.5026
Random Forest	0.4913	0.5153	0.4822	0.4963
Support Vector Machine	0.4946	0.5136	0.4901	0.4994
X Gradient Boosting	0.4816	0.4949	0.5099	0.4955

Table 6. Models' precision in the second experiment.

Model	2010	2015	2020	Mean
Bernoulli Naive Bayes	0.5064	0.4956	0.5000	0.5007
Gated Recurrent Unit	0.5494	0.5592	0.5542	0.5543
Gaussian Naive Bayes	0.5029	0.5026	0.5455	0.5170
Gradient Boosting	0.5128	0.5212	0.4483	0.4941
Light Gradient Boosting	0.5165	0.5343	0.5039	0.5182
Logistic Regression	0.5288	0.5138	0.4767	0.5064
Long-Short Term Memory	0.5617	0.5312	0.5167	0.5365
Multilayer Perceptron	0.5200	0.5139	0.4972	0.5104
Random Forest	0.5161	0.5228	0.4795	0.5061
Support Vector Machine	0.5198	0.5136	0.4901	0.5078
X Gradient Boosting	0.5056	0.5079	0.5000	0.5045

As in the first experiment, the LSTM and GRU models excelled above the rest, with average accuracies of 0.5234 and 0.5384 respectively, followed by LGB with 0.5115.

Regarding average precision, five models outperformed the base model of 0.5092: GRU, LSTM, LGB, GNB, and MLP, with the first one notably achieving an average precision of 0.5543, an improvement compared to the first experiment.

Table 7. Models' accuracy in the third experiment.

Model	2010	2015	2020	Mean
Bernoulli Naive Bayes	0.5015	0.4923	0.4893	0.4944
Gated Recurrent Unit	0.4892	0.5317	0.5487	0.5232
Gaussian Naive Bayes	0.5083	0.4938	0.5036	0.5019
Gradient Boosting	0.5034	0.4923	0.4464	0.4807
Light Gradient Boosting	0.5152	0.4938	0.4429	0.4840
Logistic Regression	0.5083	0.5031	0.4679	0.4931
Long-Short Term Memory	0.4921	0.5085	0.5596	0.5201
Multilayer Perceptron	0.5113	0.5046	0.5393	0.5184
Random Forest	0.5328	0.5138	0.5357	0.5274
Support Vector Machine	0.5113	0.5046	0.4786	0.4982
X Gradient Boosting	0.5348	0.4862	0.4750	0.4987

Table 8. Models' precision in the third experiment.

Model	2010	2015	2020	Mean
Bernoulli Naive Bayes	0.5134	0.4974	0.4740	0.4949
Gated Recurrent Unit	0.5556	0.6154	0.5833	0.5848
Gaussian Naive Bayes	0.5099	0.4983	0.4675	0.4919
Gradient Boosting	0.5333	0.4979	0.4266	0.4859
Light Gradient Boosting	0.5248	0.4991	0.4247	0.4829
Logistic Regression	0.5130	0.5047	0.4590	0.4922
Long-Short Term Memory	0.6190	0.5228	0.6571	0.5996
Multilayer Perceptron	0.5113	0.5046	0.5373	0.5177
Random Forest	0.5315	0.5166	0.5130	0.5204
Support Vector Machine	0.5113	0.5046	0.4786	0.4982
X Gradient Boosting	0.5477	0.4927	0.4532	0.4979

In the third experiment, the RF model showed an improvement in accuracy, not only outperforming the baseline model for the first time but also achieving higher accuracy than the other models.

Regarding precision, the LSTM and GRU models outperformed the other models, achieving average precisions of 0.5996 and 0.5848, respectively.

Table 9. Models' accuracy in the fourth experiment.

Model	2010	2015	2020	Mean
Bernoulli Naive Bayes	0.5059	0.4946	0.4875	0.4960
Gated Recurrent Unit	0.4912	0.5116	0.5254	0.5094
Gaussian Naive Bayes	0.5069	0.4915	0.4910	0.4965

Model	2010	2015	2020	Mean
Gradient Boosting	0.5088	0.4961	0.4659	0.4903
Light Gradient Boosting	0.5216	0.4961	0.5448	0.5208
Logistic Regression	0.5157	0.5039	0.4982	0.5059
Long-Short Term Memory	0.4951	0.5209	0.4964	0.5041
Multilayer Perceptron	0.4980	0.5023	0.5125	0.5043
Random Forest	0.5176	0.4761	0.5054	0.4997
Support Vector Machine	0.5265	0.4900	0.4624	0.4930
X Gradient Boosting	0.5029	0.4961	0.5090	0.5027

Table 10. Models' precision in the fourth experiment.

Model	2010	2015	2020	Mean
Bernoulli Naive Bayes	0.5179	0.5000	0.4694	0.4958
Gated Recurrent Unit	0.5345	0.5135	0.5070	0.5183
Gaussian Naive Bayes	0.5190	0.4971	0.4726	0.4962
Gradient Boosting	0.5169	0.5010	0.4615	0.4931
Light Gradient Boosting	0.5294	0.5010	0.5185	0.5163
Logistic Regression	0.5245	0.5071	0.4833	0.5050
Long-Short Term Memory	0.5870	0.5188	0.4624	0.5227
Multilayer Perceptron	0.5092	0.5062	0.4948	0.5034
Random Forest	0.5229	0.4830	0.4891	0.4983
Support Vector Machine	0.5334	0.4961	0.4474	0.4923
X Gradient Boosting	0.5120	0.5013	0.4915	0.5016

In this experiment, it's first observed a reduction in runtime for parameterization, training, and testing the models compared to the third experiment.

Regarding the models' accuracy, there is a noticeable drop in performance compared to the previous experiment, especially for the RF, GRU, and LSTM models. The same trend is observed for precision.

From the analysis of the initial four experiments, the third experiment highlighted the significantly different performance of the LSTM and GRU models regarding both accuracy and precision compared to the other models and even to each other in the other three experiments.

Therefore, it was decided to create hybrid models, LSTM-GRU and GRU-LSTM, apply them to the data from the third experiment, and evaluate their accuracy (Table 11) and precision (Table 12).

Table 11. Models' accuracy in the fifth experiment.

Model	2010	2015	2020	Mean
LSTM-GRU	0.5231	0.5077	0.5271	0.5193
GRU-LSTM	0.4956	0.5123	0.5271	0.5117
LSTM	0.4921	0.5085	0.5596	0.5201
GRU	0.4892	0.5317	0.5487	0.5232

Table 12. Models' precision in the fifth experiment.

Model	2010	2015	2020	Mean
LSTM-GRU	0.5779	0.5144	0.6250	0.5724
GRU-LSTM	0.5352	0.6000	0.5333	0.5562
LSTM	0.6190	0.5228	0.6571	0.5996
GRU	0.5556	0.6154	0.5833	0.5848

Based on the results obtained, unfortunately, the hybridization of LSTM and GRU models did not show improvement, presenting lower performance compared to the models applied individually in both accuracy and precision. Furthermore, it was observed that the LSTM-GRU hybridization outperformed the GRU-LSTM model.

3 FINAL CONSIDERATIONS

This study achieved its initial objective: the development and comparison of multiple machine learning models to predict the Ibovespa performance trend.

The final results indicated that the models based on Recurrent Neural Networks (RNNs), specifically LSTM and GRU, had the best performance, with similar results through the experiments. Although the hybridization of these two models was proposed, its accuracy and precision were inferior to those of the models applied individually.

Even with accuracy and precision close to 60%, the best models developed and presented in this paper still performed similar to the base model, which predicts index appreciation every day, and therefore cannot be considered good enough to be used as a consistent investment strategy.

For future research, there's potential to test additional input variables, different time periods, more advanced machine learning models such as Convolutional Neural Networks (CNNs), or other hybrid models. Furthermore, exploring the prediction of Ibovespa trends over longer periods, such as months or years, rather than just a single day, may provide better results.

REFERENCES

ALVES, J. V. S. **Um modelo preditivo de cotação de ações de empresas estatais brasileiras utilizando redes neurais artificiais no ambiente MATLAB**. 2019. M.S. thesis, Universidade Federal de Uberlândia, 2019.

BLUVOL, L. M. **Análise de algoritmos de machine learning e redes neurais para previsão de preços de ações do Ibovespa**. 2022. M.S. Thesis. Fundação Getúlio Vargas, 2022.

BOEHMKE, B.; GREENWELL, B. **Hands-On Machine Learning with R**. Taylor & Francis Group, 2020. Available at: https://bradleyboehmke.github.io/HOML/. Accessed on: Sep. 16, 2025.

BROWNLEE, J. (2020). **How to Develop a Naive Bayes Classifier from Scratch in Python**. Machine Learning Mastery, Jan 10, 2020. Available: https://machinelearningmastery.com/

classification-as-conditional-probability-and-the-naive-bayes-algorithm/. Accessed on: Sep 27, 2023.

GÉRON, A. **Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow**: concepts, tools, and techniques to build intelligent systems. 2nd ed. O'reilly Media, 20219.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. The MIT Press, 2016. Cambridge.

HARDESTY, L. **Explained**: Neural networks. MIT. Apr 14, 2017. Available at: https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414. Accessed on: Nov. 20, 2022.

HYNDMAN, R. J.; ATHANASOPOULOS, G. **Forecasting**: principles and practice. 3. ed. Melbourne: OTexts, 2021. Disponível em: https://otexts.com/fpp3/. Accessed on: Aug. 21, 2025.

JEONG, Y.; KIM, S.; YOON, B. An Algorithm for Supporting Decision Making in Stock Investment through Opinion Mining and Machine Learning. **Proceedings of the Portland International Conference On Management Of Engineering And Technology (PICMET)**. Honolulu, United States of America. Aug 19-23, 2018.

KARA, Y.; BOYACIOGLU, M. A.; BAYKAN, Ö. K. Predicting direction of stock price index movement using artificial neural networks and support vector machines: the sample of the Istanbul stock exchange. **Expert Systems with Applications**, 38, 5311-5319. May 2011.

KUMAR, D.; SARANGI, P. K.; VERMA, R. A systematic review of stock market prediction using machine learning and statistical techniques. **Materials Today Proceedings**, 49, 3187-3191. Jan 2021.

MOHRI, M. **Foundations of Machine Learning**. 2nd ed. The Mit Press, 2018. Cambridge.

NABIPOUR, M.; NAYYERI, P.; JABANI, H.; MOSAVI, A.; SALWANA, E.; SHAHAB, S. Deep learning for stock market prediction. **Entropy**, Basel, v. 22, n. 8, p. 840. Jul 2020.

OBTHONG, M.; TANTISANTIWONG, N.; JEAMWATTHANACHAI, W.; WILLS, G. A Survey on Machine Learning for Stock Price Prediction: algorithms and techniques. **Proceedings Of The 2nd International Conference On Finance, Economics, Management And Business**. Lisboa, Portugal. Feb 2020.

PATEL, J.; SHAH, S.; THAKKAR, P.; KOTECHA, K. Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques. **Expert Systems with Applications**, 42, 259-268. Jan 2015.

SILVA, J. E.; SÁTIRO, R. M. O poder preditivo dos modelos boosting de machine learning no mercado brasileiro de ações. **Brazilian Journal of Quantitative Methods Applied to Accounting**, 11, 52-68. Dec 2022.

TSAI, C. F.; WANG, S. P. Stock Price Forecasting by Hybrid Machine Learning Techniques. **Proceedings of the International Multiconference of Engineers and Computer Scientists**. Hong Kong. 18-20. Mar 2009.

WU, X. Top 10 algorithms in data mining. **Knowledge and Information Systems**, 14, 1-37. Dec 2007.