Design and Evaluation of Features for Algorithmic Trading Models

Newton Paulo Linchen[a], Renata Galante[a], Adriano C. M. Pereira[b]

[a]*Federal University of Rio Grande do Sul (UFRGS) - Department of Computer Science, Av. Bento Gonçalves, 9500, INF - Agronomia, Porto Alegre, 91509-900, Rio Grande do Sul, Brazil*
[b]*Federal University of Minas Gerais (UFMG), Department of Computer Science (DCC), Rua Reitor Píres Albuquerque, ICEx - Pampulha, Belo Horizonte, 31270-901, Minas Gerais, Brazil*

Abstract

Time series of financial trading assets are known to have stochastic properties which turn prediction into an almost futile endeavor. In Economics, the mainstream theory of the Efficient Market Hypothesis proposes that any attempt to predict the future prices of a tradable asset is in vain, and should not be pursued. In the last decades, the advent of machine learning algorithms gave the investing community interesting tools for advancing the prediction research. However, we understand that algorithms are not enough to make successful predictions: in order to build better models the researcher should employ feature development, especially with the knowledge and experience of a practitioner and specialist in the field. In this work, we evaluate the performance of a classification algorithm (QDA - Quadratic Discriminant Analysis) with the addition of features, comparing the results with a benchmark (buy-and-hold) and a baseline experiment. The investigation was carried out using the Bovespa Index Futures Contract (Ibovespa Futuro), by making short-term predictions in a simulated environment.

Keywords: Feature Engineering, Machine Learning, Algotrading, Classification, Prediction, Stock Markets, Time Series, Quadratic Discriminant Analysis.

1. Introduction

Financial time series are regarded as stochastic processes, with mean equal to zero and even distributions (without skewness), with the presence of fat tails, which differentiate them from gaussian distributions. This properties make almost impossible to predict future data (future prices), and that is the reason why in Economics this effort is regarded as futile, by the Efficient Market Hypothesis defenders.

The advent of machine learning algorithms gave new tools for the investment researchers, but the algorithms by themselves are not exactly capable to extract the value which resides in the financial time series, due to its high levels of noise. In fact, in order to allow algorithms to make sufficiently adequate predictions, the researcher should employ the use of features, especially features developed by specialists in the field.

By incorporating specialist-generated features into machine learning models, traders and investors can gain insights into market trends and make more informed decisions. However, it is important to note that the quality and relevance of these features can vary depending on the expertise of the specialist and the specific market conditions being analyzed.

In this work, we propose that feature development can improve algorithm results in financial markets by enhancing the capability of extracting value (signal) from the noisy dataset. The research was made using a classification algorithm with no hyperparameters, which allows to maintain the same algorithmic structure throughout different sets of experiments, each experiment containing both training of the algorithm in part of the dataset and testing of the model on the remainder of the dataset. For this matter, we advocate the use of both proprietary features (SALIB - Structural Analysis Library) as well as the mainstream features of the talib.py library (TALIB - Technical Analysis Library).

The experiments, when compared with the Brazilian market's benchmark and with a baseline experiment, displayed an improvement in the results, both for the SALIB features and TALIB features, as well as for the join use of both feature libraries (experiments with SALIB and TALIB features).

This article is organized as follows: Chapter 1 presents the issue of feature development for algotrading. Chapter 2 reviews the relevant work on algotrading by researchers of several countries, and analyses the use of features by type. Chapter 3 proposes a methodology for investigating the relevance of features in algotrading models. Chapter 4 reveals the results of a set of experiments with different types of features, and analyses this results comparing to a Benchmark and Baseline. Chapter 5 concludes this work and addresses future works.

2. Related Work

Specialist-generated features refer to specific data points or indicators that are created by domain experts in the field of financial markets and algorithmic trading. These features are used to inform machine learning algorithms and help them make more accurate predictions or decisions. They derive from direct trading experience and knowledge about trading theory and practice.

In the context of financial markets and algorithmic trading, specialist-generated features may include: technical indicators, which are mathematical calculations based on market data such as price and volume. Examples include moving averages, relative strength index (RSI), and Bollinger Bands, among others. Fundamental data, which refers to information about a company or asset that may impact its value, such as earnings reports and economic indicators. Sentiment analysis, which involves analyzing the tone and content of news articles, social media posts, and other sources of information to determine the overall sentiment towards a tradable asset or particular market. Order book data, that includes information about the buy and sell orders for a particular asset, such as the number of orders clustered at different price levels. For last, market microstructure data refers to data on the behavior of market participants, such as bid-ask spreads and order flow, transformed into features to enhance price predictions.

The review of the relevant literature comprised works from 2001 to 2020, from authors and researchers around the globe: United States, Canada, Brazil, Greece, Holand, Turkey, Moroco, India, China, Singapore, Tailand, Philipines and Taiwan. In these works, we found that when features were used (at all), most of them were TALIB features (Technical Indicators), a smaller amount were derived from Market Data (OHLC price information, Return, Volume, Exogenous Time Series and Market Internals). This is expressed in Table 1.

Table 1: Features in the literature

Feature Type	Number of feature occurrences	Percentage of occurrence
Technical Indicators	67	62%
Market Data	39	36%
Statistical Properties	2	2%

Source: Compiled by the author

The literature indicates, also, that rarely the authors relied in only one type of feature. From the works reviewed, only 5 used exclusively “Technical Indicators”, meanwhile only 9 works used exclusively “Market Data”.

The following works presented the features derived from “Market Data”: Moody and Saffell (2001), Joseph et al. (2012), Weng (2009), Tay and Cao (2002), Martinez et al. (2009), Sezer et al. (2017), Li et al. (2018), You et al. (2019), Torralba (2019), Cho et al. (2019), Song and Lee (2019), Joosery and Deepa (2019), Louwerse and Rothkrantz (2014), Patil et al. (2020) and Upadhyay et al. (2016) presenting “OHLC prices data”, i.e., the following prices for the time windows: the opening price, the highest price, the lowest price and the closing price.

The works Li et al. (2018), Castro et al. (2023), Roca and M'ol (2015),Caramico (2010), Calainho (2015), Torralba (2019), Cho et al. (2019) and Wang (2020) presented the feature “Return”, which means the variation of the opening and closing prices of the analyzed time window.

The feature “Volume”, occurred in the works: Weng (2009), Li et al. (2018), Torralba (2019), Cho et al. (2019) and Louwerse and Rothkrantz (2014). Volume is a measure of liquidity of the trading asset, and its important in studies of market impact of buy and sell orders. It's peculiar to see this important feature not appearing in much more research works.

“Exogenous Time Series” as features occurred in the works: Joseph et al. (2012), Weng (2009), Tay and Cao (2002) and Li et al. (2018). This type of feature would be any time series aside from the trading asset time series itself. It allows to perform multivariate analysis.

Fundamental data were presented in the following works: Weng (2009), Li et al. (2018), Cao and Tay (2001) and Song and Lee (2019) in the form of Multiples. Features derived from Economic Data occurred in Martinez et al. (2009) and Song and Lee (2019). Again, important market measures (the fundamental multiples) appearing in such a small number of works. Fundamental data are regarded to be the primary source of valuation for public companies, and it's considered to be very important data.

Technical indicators appeared in the following works:Castro et al. (2023), Caramico (2010), Madge and Bhatt (2015), Calainho (2015), Cao and Tay (2003), Chalvatzis and Hristu-Varsakelis (2019), Leung et al. (2014), Zhang et al. (2018), Labiad et al. (2018), Song and Lee (2019), Wang (2020), Chen et al. (2017), Louwerse and Rothkrantz (2014) and Patil et al. (2020). This type of features are price and/or volume derived formulas, which are extensively used by the school of analysis called Technical Analysis, considered to be, alongside with the Fundamental Analysis, one of the major influences on market participants (brokers, investors, traders and asset managers).

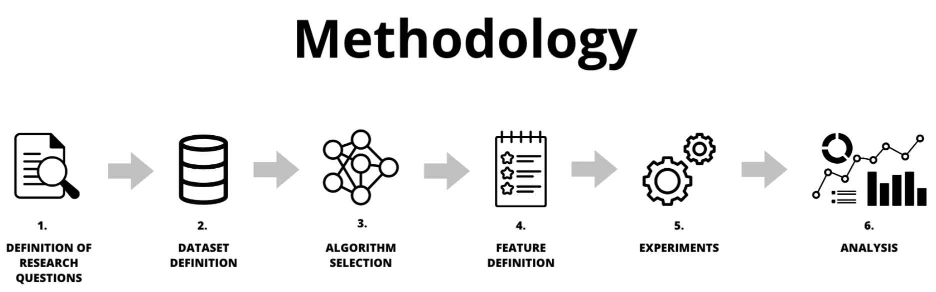
Statistical transformations occured in You et al. (2019) and Labiad et al. (2018), meanwhile features derived from "News" only appeared in Oncharoen and Vateekul (2018).

The literature shows that regarding the use of features for machine learning modeling of the financial markets (algotrading), the researchers focus on building and enhancing the algorithms rather than spend much time developing new proprietary features. As the talib.py library provides a comprehensive set of technical indicators, its relatively easy to instantiate Technical Analysis features as a proxy for new proprietary features, which would require from the authors extensive knowledge and experience with trading the markets or the particular asset.

3. Methodology

Our methodology in this work follows a six-step process: the definition of research questions, dataset definition, algorithm selection, feature definition, experiments and analysis, as displayed in the Figure 1.

Figure 1: Methodology



Source: compiled by the author

The research questions are related to the impact that the use of specialist-generated features may have in the results of a machine learning trained model: what happens when we increase the number of features, what's the impact of using different sets of features (SALIB or TALIB), how the results change when combining both types of features (SALIB and TALIB). Does these variations in the number and types of features affect significantly the model's results?

Next, we have the dataset definition. In order to analyze the variation in the model's results only by the impact of the different combinations of features, we decided to maintain the same dataset for all experiments. In this case, the chosen dataset would be the price history (time series) of the mini-contract of the Brazilian stock index futures, (Ibovespa Futuro), from february 13th, 2014, to september 22th, 2022. The time series is of the "intraday" type, representing the "15 minute chart" of the trading asset: each datapoint corresponds to an intraday "bar" or "candle" of the 15 minute time window, which contains the following OHLC structure: "Open" (opening price for the data point), "High" (highest price for the 15 minute interval), "Low" (lowest price for the 15 minute interval) and "Close" (the last price of the 15 minute interval).

The next step would be the selection of the algorithm used to perform the experiments. By the same token, we would like to isolate this variable, so that the only real change from experiment to experiment would be the different set of features used in the training of the model. Therefore, the chosen algorithm should have no hyperparameters, and should remain stable from one training to the other. We chose the classic machine learning classification algorithm, the QDA - Quadratic Discriminant Analysis, which would fit the desired specifications.

Here we have the step of feature definition, in which we would list all the features allowed in the experiments. In this work, we use two sets of specialist-generated features:

1. The proprietary SALIB features (Structural Analysis Library, developed by the author), which derives from price data, ascribing to it dimensional and spatial relationships. They are:
   (a) "Hour" - Sets the hourly timestamp for the datapoints.
   (b) "Gap" - The "opening gap" of a trading day: the difference between the opening price of the day and the closing price of the previous day. Formula: GAP = (Open - Close.1)/Close.1
   (c) "CTO - Close-to-open" - The daily return for a specific (determined) trading day. Can be instantiated for any daily data point of the time series. Formula: (Close - Open)/Open
   (d) "Monthly Quadrant" - A feature that compares the current price of the datapoint with the prices of the current month, and ascribes to it a spatial relationship. The month is divided in three "thirds" or price zones (beginning, middle and end of the month), and divided in two hemispheres: "positive", if the current price is above the opening price of the month, and "negative" if it's below. So, the current price, at any point would be in one of the six quadrants (beginning above, beginning below, middle above, middle below, end above, end below).
   (e) "IBS - Internal Bar Strenght" - Feature that measures the "strenght" of the closing price within the daily range. Formula: IBS = (Close-Low)/(High-Low).
   (f) "From High" - Measures the percentage distance of the current price to the hightest price of the time window. Formula: FHIGH = (H.n - C.0) / H.n
   (g) "From Low" - Measures the percentage distance of the current price to the lowest price of the time window. Formula: FLOW = (C.0 - L.n) / L.n
   (h) "DAILYPOSITION" - Sets a spatial relationship between two daily OHLC structures ("bars" or "candles"). The current day (or any instantiated day) could be "Above" (H¿H.1 and L¿L.1),

"Inside" (H¡H.1 and L¿L.1), "Below" (H¡H.1 and L¡L.1) or "Outside" (H¿H.1 and L¡L.1) the limits of the high and the low of the previous daily bar (or "candle").

   (i) "WEEKLY POSITION" - Same price relationships described by DAILYPOSITION, but instantiated with weekly price data.

2. The TALIB features, which is an open source library that contains several Technical Indicators based on Technical Analysis.

   (a) RSI - Relative Strenght Index

   $$RSI = 100 - \frac{100}{(1 + FR)}$$

   (b) "Stochastic Oscillator"

   $$\%K = \frac{100 * (\text{Preço} - LN)}{(HN - LN)}$$

   $$\%D = MMA(\%K, X)$$

   (c) "MACD - Moving Average Convergence Divergence"

   $$\text{MACD}(x, y, z) = \text{MME1}(x) - \text{MME2}(y)$$

   (d) "DI+/DI- Index (ADX)"

   $$+DM = H_0 - H_1$$

   $$-DM = L_1 - L_0$$

   $$TR = \text{máx}\{H_0 - L_0, H_0 - C_1, L_0 - C_1\}$$

   (e) "CCI - Commodity Channel Index"

   $$CCI = \frac{1}{0{,}015} \times \frac{PT - MM(PT)}{\sigma(PT)}$$

   (f) "MFI - Market Facilitation Index"

   $$MFI = \frac{(HIGH - LOW)}{VOLUME}.$$

(g) "Williams % R (WILLR)"

$$\%R = \frac{-(\text{MAX (MAX}(i-n)) - \text{CLOSE}(i))}{(\text{MAX (MAX }(i-n)) - \text{MIN (MIN }(i-n)))} * 100$$

(h) "TRIX - Triple Exponential Average"

i. EMA1 = EMA(Close)
ii. EMA2 = EMA(EMA1)
iii. EMA3 = EMA(EMA2)
iv. TRIX = (EMA3 [today] - EMA3 [yesterday]) / EMA3 [yesterday]

The remainder of the methodology deals with the experiments and analysis, which we will address in the next Section. The experiments start with a Benchmark, then a Baseline experiment, followed by the experiments with different feature combinations.

4. Experiments: analysis and discussion

The key characteristic of the experiments was to provide a constant framework for training and testing the models, without variations in the algorithm or the dataset. The idea was that only the addition of features should impact the results. Our goal was to isolate the features impact from any other variable, so the expected variation in the results could be referred to the features used.

Therefore, the algorithm deployed in the experiments was the QDA classificator, which had no hyperparameters and remained constant from experiment to experiment. Also, the same dataset was used in all the experiments, i.e., the intraday time series (15 minute interval) of the Brazilian stock market index futures contract.

Before the experimentation with the features, we proposed a Benchmark evaluation and a Baseline experiment. This would provide a basis for comparing results with the features experiments.

The Benchmark corresponded to a "buy-and-hold" simulated investment, with a simulated "buy order" executed in the first datapoint (15 min OHLC bar or "candle"), and a closing simulated "sell order" in the last datapoint. This procedure would allow the Benchmark to capture the price variation for the entire duration of the dataset. The Benchmark resulted negative. For this holding period, the result was a -14,95% (minus almost fifteen percent).

For the Baseline experiment, we simulated a long-only approach (allowing only simulated buy orders, in order to profit from the upside of price movement). As the dataset is the 15 minute interval time series for the asset, we ascribed the "holding period" of the "trading" (simulated negotiation) for two "candles" or OHLC bars, which means a holding period of 30 minutes. This categorizes the approach as day trading, in the jargon of the financial community. The Baseline results were also negative, with a -19,65% return for 15.475 simulated trades, with 50.33% success rate (accuracy).

It's important to acknowledge that this results (or any other results in this work) were not computed with trading costs, for the following reason: the purpose of this work was not to develop a complete trading strategy, which would require an entire new set of concerns and procedures. To develop a tradable strategy, it's required not only a working set of parameters and trained models but, also, the simulation of price costs, slippage occurrences (the difference of the intended price with the actual traded price), volume capacity (how large is your trading capital for the available liquidity), and many other concerns, such as risk parameters, risk constraints, time constraints, etc. This effort would only be justified if we were to develop an actual trading strategy, for actual exploitation of a trading "edge". The complete disclosure of a trading strategy would regard it unexploitable, for the investing community could make it worthless as a trading edge expires when used by sufficiently large volume of capital. So the purpose of this work is to acknowledge the differences in results when using features, but as an scientific endeavor, not a commercial or financial pursuit.

After the Benchmark and Baseline evaluation, we started the experiments with features. For these experiments, we maintained the same structural framework: the same algorithm was used, the same dataset, and the same holding period of 30 minutes "per trade" or simulated negotiation. All the parameters remained the same, so the only actual change in results would come from the different feature combinations.

Three sets of experiments were proposed: the first one computing exclusively SALIB features, had a total of 2879 different experiments (training of the algorithm an testing of the trained model), each one with a different combination of the features. The second set of experiments computed only TALIB features with 255 combinations.The third set had comprised both SALIB and TALIB features, for a total of 4080 experiments. Despite the high number of variations, the features were combined according to proprietary-developed restrictions, necessary to optimize combinations of relevance for short-term trading (from a trader's point of view), and to avoid possible combinatory explosion.

The results for the first set of experiments are displayed in Table 2, which contains the most significant results for the combinations of SALIB features. The classification criteria was the percentage result in the Testing Period (Testing Results %). We observe a mean of 72% return for the set of 10 best feature combinations.

Table 2: Best SALIB features combinations.

FEATURE COMBINATION	CONSISTENCY MEASURE	Nº OF TRADES (TRAINING PERIOD)	Nº OF TRADES (TESTING PERIOD)	TRAINING RESULTS (%)	TESTING RESULTS (%)
quadrant daily_position0 daily_position1 weekly_position0 weekly_position1 weekly_position2	1,01	23878	6108	350	90
daily_position0 daily_position1 weekly_position0 weekly_position1 weekly_position2	1,01	23878	6108	350	90
quadrant fhigh0 flow0 weekly_position0 weekly_position1	1,07	21132	4917	135	72
fhigh0 flow0 weekly_position0 weekly_position1	1,07	21132	4917	135	72
gap cto1 quadrant ibs1 fhigh0 flow0 daily_position0 weekly_position0 weekly_position1 weekly_position2	1,05	16988	3481	262	66
gap cto1 ibs1 fhigh0 flow0 daily_position0 weekly_position0 weekly_position1 weekly_position2	1,05	16988	3481	262	66
gap cto1 cto2 quadrant ibs1 fhigh0 flow0 daily_position0 weekly_position0 weekly_position1 weekly_position2	1,03	17046	3237	351	66
gap cto1 cto2 ibs1 fhigh0 flow0 daily_position0 weekly_position0 weekly_position1 weekly_position2	1,03	17046	3237	351	66
quadrant weekly_position0 weekly_position1 weekly_position2	1,03	23739	6075	171	64
weekly_position0 weekly_position1 weekly_position2	1,03	23739	6075	171	64

Source: Compiled by the author

The results for the second set of experiments are displayed in Table 3, which contains the most significant results for the combinations of TALIB features. The classification criteria was the percentage result in the Testing Period (Testing Results %). We observe a mean of 37,76% return for the set of 10 best feature combinations.

Table 3: Best TALIB features combinations.

FEATURE COMBINATION	CONSISTENCY MEASURE	Nº OF TRADES (TRAINING PERIOD)	Nº OF TRADES (TESTING PERIOD)	TRAINING RESULTS (%)	TESTING RESULTS (%)
RSI MACD ADX MFI WILLR TRIX	0,95	8187	3603	229	45,97
RSI STOCHASTIC MACD ADX MFI TRIX	0,95	8136	3574	231	44,39
RSI MACD ADX CCI MFI TRIX	0,94	8153	3577	231	41,89
RSI MACD ADX MFI TRIX	0,92	7851	3608	259	40,33
RSI STOCHASTIC ADX MFI WILLR TRIX	0,93	8423	3588	227	39,11
MACD ADX CCI MFI TRIX	0,93	8220	3642	216	34,85
STOCHASTIC MACD ADX MFI WILLR TRIX	0,96	8296	3573	187	34,4
RSI STOCHASTIC MACD MFI TRIX	0,94	7202	3314	193	33,07
RSI MACD ADX WILLR TRIX	0,89	8132	3685	267	31,85
MACD ADX MFI TRIX	0,96	8339	3646	166	31,74

Source: Compiled by the author

The third set of experiments best results are displayed in Table 4, which contains the most significant results for the combinations of both SALIB and TALIB features. The classification criteria was the percentage result in the Testing Period (Testing Results %). We observe a mean of 66,53% return for the set of 10 best feature combinations.

Table 4: Best TALIB features combinations.

FEATURE COMBINATION	CONSISTENCY MEASURE	Nº OF TRADES (TRAINING PERIOD)	Nº OF TRADES (TESTING PERIOD)	TRAINING RESULTS (%)	TESTING RESULTS (%)
quadrant weekly_position0 weekly_position1 weekly_position2 TRIX	1,05	22296	5488	178,82	69,33
weekly_position0 weekly_position1 weekly_position2 TRIX	1,05	22296	5488	178,82	69,33
gap quadrant fhigh0 flow0 weekly_position0 weekly_position1 weekly_position2 ADX MFI TRIX	1,07	15167	2951	239,64	66,44
gap fhigh0 flow0 weekly_position0 weekly_position1 weekly_position2 ADX MFI TRIX	1,07	15167	2951	239,64	66,44
quadrant weekly_position0 weekly_position1 weekly_position2 ADX MFI WILLR	0,98	24120	6186	326,71	65,98
weekly_position0 weekly_position1 weekly_position2 ADX MFI WILLR	0,98	24120	6186	326,71	65,98
gap quadrant fhigh0 flow0 weekly_position0 weekly_position1 weekly_position2 ADX MFI	1,06	16580	3603	184,87	65,92
gap fhigh0 flow0 weekly_position0 weekly_position1 weekly_position2 ADX MFI	1,06	16580	3603	184,87	65,92
quadrant weekly_position0 weekly_position1 weekly_position2 ADX	1,04	23870	6160	159,63	64,97
weekly_position0 weekly_position1 weekly_position2 ADX	1,04	23870	6160	159,63	64,97

Source: Compiled by the author

Table 5 combines the information from all sets of experiments. And they are compared to the Benchmark and Baseline results:

Table 5: Overview of the results

Experiment	**Results**
Benchmark	-14.95%
Baseline	-19.65%
Average of 10 best SALIB	72.00%
Average of 10 best TALIB	37.76%
Average of 10 best SALIB + TALIB	66.53%

We found that any type of specialist-generated features, whether SALIB or TALIB brings positive results when compared with the Benchmark and Baseline experiments. A case can be made for the data mining properties of feature development, and as the results are from the Testing Period, the capacity to enable the trained model to generalize it's learning in future data.

The repository with the actual code and dataset used in the experiments it's available here: https://github.com/NewtonLinchen/FeatureExperiments

5. Conclusion

The analysis of the related work and the experiments conducted by the author allowed us to conclude that there is enormous potential for enhanced trading results if the development of features receives substantial part of the research effort. However, we also conclude that rarely its the case of authors innovating in feature discovery and development. The use of standard libraries such as talib.py allows any researcher to deploy features without much (or any) knowledge about the financial markets, especially about trading.

The experiments demonstrated the positive impact on results for price prediction, when features were added to the modeling process. Both the author's proprietary set of features (SALIB) as the more common Technical Analysis features (TALIB) showed improvement on results, when compared with the Benchmark and Baseline.

Future work could provide further development of proprietary features, as well as testing this concept with different asset classes and global markets.

References

Calainho, F.D., 2015. Previsão da direção de índices da bovespa por intermédio de máquina de suporte vetorial .

Cao, L., Tay, F.E., 2001. Financial forecasting using support vector machines. Neural Computing & Applications 10, 184–192.

Cao, L.J., Tay, F.E.H., 2003. Support vector machine with adaptive parameters in financial time series forecasting. IEEE Transactions on neural networks 14, 1506–1518.

Caramico, A., 2010. ProposiÇÃo de um modelo preditivo do ibovespa por meio da utilizaÇÃo de redes neurais artificiais doi:10.13140/RG.2.2.21376.48645.

Castro, C., De, A., Braga, P., Andrade, A., 2023. Aplica ao de um modelo ensemble de redes neurais artificiais para previsão de séries temporais não estacionárias .

Chalvatzis, C., Hristu-Varsakelis, D., 2019. High-performance stock index trading: making effective use of a deep lstm neural network. arXiv preprint arXiv:1902.03125 .

Chen, G., Chen, Y., Fushimi, T., 2017. Application of deep learning to algorithmic trading. Tech. Rep, Tech. Rep. .

Cho, C.H., Lee, G.Y., Tsai, Y.L., Lan, K.C., 2019. Toward stock price prediction using deep learning, in: Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing Companion, pp. 133–135.

Joosery, B., Deepa, G., 2019. Comparative analysis of time-series forecasting algorithms for stock price prediction, in: Proceedings of the International Conference on Advanced Information Science and System, pp. 1–6.

Joseph, C.E., Nobakht, B., Loni, B., 2012. Stock market analysis and prediction using hidden markov models. Journal of Business and Economics Research (JBER) 10, 397–410.

Labiad, B., Berrado, A., Benabbou, L., 2018. Short term prediction framework for moroccan stock market using artificial neural networks, in: Proceedings of the 12th International Conference on Intelligent Systems: Theories and Applications, pp. 1–6.

Leung, C.K.S., MacKinnon, R.K., Wang, Y., 2014. A machine learning approach for stock price prediction, in: Proceedings of the 18th International Database Engineering & Applications Symposium, pp. 274–277.

Li, J., Rao, R., Shi, J., 2018. Learning to trade with deep actor critic methods, in: 2018 11th International Symposium on Computational Intelligence and Design (ISCID), pp. 66–71. doi:10.1109/ISCID.2018.10116.

Louwerse, V., Rothkrantz, L., 2014. Intraday stock forecasting, in: Proceedings of the 15th International Conference on Computer Systems and Technologies, pp. 202–209.

Madge, S., Bhatt, S., 2015. Predicting stock price direction using support vector machines. Independent work report spring 45.

Martinez, L., Hora, D., Palotti, J., Meira Jr, W., Pappa, G., 2009. From an artificial neural network to a stock market day-trading system: A case study on the bm&f bovespa, pp. 2006–2013. doi:10.1109/IJCNN.2009.5179050.

Moody, J., Saffell, M., 2001. Learning to trade via direct reinforcement. IEEE Transactions on Neural Networks 12, 875–889. doi:10.1109/72.935097.

Oncharoen, P., Vateekul, P., 2018. Deep learning using risk-reward function for stock market prediction, in: Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence, pp. 556–561.

Patil, P., Wu, C.S.M., Potika, K., Orang, M., 2020. Stock market prediction using ensemble of graph theory, machine learning and deep learning models, in: Proceedings of the 3rd International Conference on Software Engineering and Information Management, pp. 85–92.

Roca, R.L.R.d.S., M'ol, A.L.R., 2015. Prediction of the bovespa index by means of artificial neural networks .

Sezer, O.B., Ozbayoglu, A.M., Dogdu, E., 2017. An artificial neural network-based stock trading system using technical analysis and big data framework, in: proceedings of the southeast conference, pp. 223–226.

Song, Y., Lee, J., 2019. Design of stock price prediction model with various configuration of input features, in: Proceedings of the International Conference on Artificial Intelligence, Information Processing and Cloud Computing, pp. 1–5.

Tay, F.E., Cao, L., 2002. Modified support vector machines in financial time series forecasting. Neurocomputing 48, 847–861.

Torralba, E.M., 2019. Development of a deep learning-lstm trend prediction model of stock prices, in: Proceedings of the 2019 International Conference on Management Science and Industrial Engineering, pp. 126–133.

Upadhyay, V.P., Panwar, S., Merugu, R., Panchariya, R., 2016. Forecasting stock market movements using various kernel functions in support vector machine, in: Proceedings of the International Conference on Advances in Information Communication Technology & Computing, pp. 1–5.

Wang, S., 2020. The prediction of stock index movements based on machine learning, in: Proceedings of the 2020 12th International Conference on Computer and Automation Engineering, pp. 1–6.

Weng, C.K.C.V.J., 2009. A svm approach to stock trading .

You, D., Yao, X., Geng, X., Fang, X., Qu, S., 2019. Stock index prediction method based on dynamic weighted ensemble learning, in: Proceedings of the 2019 International Conference on Robotics Systems and Vehicle Technology, pp. 41–46.

Zhang, C., Ji, Z., Zhang, J., Wang, Y., Zhao, X., Yang, Y., 2018. Predicting chinese stock market price trend using machine learning approach, in: Proceedings of the 2nd International Conference on Computer Science and Application Engineering, pp. 1–5.