Machine Learning with Applications 5 (2021) 100060

Contents lists available at ScienceDirect

Machine Learning with Applications

journal homepage: www.elsevier.com/locate/mlwa

Feature selection and deep neural networks for stock price direction forecasting using technical analysis indicators

Yaohao Peng [a,b,*], Pedro Henrique Melo Albuquerque [a], Herbert Kimura [a], Cayan Atreio Portela Bárcena Saavedra [a]

[a] University of Brasilia, Campus Darcy Ribeiro, Brasilia, Distrito Federal, 70910–900, Brazil

[b] Brazilian Ministry of Economy, Esplanade of Ministries, Block P, Brasilia, Distrito Federal, 70048-900, Brazil

ARTICLE INFO

Keywords:
Deep learning
Technical analysis indicators
Time-series forecasting
Market efficiency
Trading profitability

ABSTRACT

This paper analyzes the factor zoo, which has theoretical and empirical implications for finance, from a machine learning perspective. More specifically, we discuss feature selection in the context of deep neural network models to predict the stock price direction. We investigated a set of 124 technical analysis indicators used as explanatory variables in the recent literature and specialized trading websites. We applied three feature selection methods to shrink the feature set aiming to eliminate redundant information from similar indicators. Using daily data from stocks of seven global market indexes between 2008 and 2019, we tested neural networks with different settings of hidden layers and dropout rates. We compared various classification metrics, taking into account profitability and transaction costs levels to analyze economic gains. The results show that the variables were not uniformly chosen by the feature selection algorithms and that the out-of-sample accuracy rate of the prediction converged to two values — besides the 50% accuracy value that would suggest market efficiency, a "strange attractor" of 65% accuracy also was achieved consistently. We also found that the profitability of the strategies did not manage to significantly outperform the Buy-and-Hold strategy, even showing fairly large negative values for some hyperparameter combinations.

1. Introduction

Financial variables are hard to predict. Over the decades, many scholars and market practitioners found various empirical evidence and stylized facts concerning financial variables' unpredictability, ranging from stock and commodity prices to exchange and interest rates. The efficient market hypothesis, which states that no economic agent can consistently obtain higher returns than the market, remains one of the most important theoretical results in finance. However, numerous studies had sought to analyze potential market inefficiencies and to predict the future trends of financial variables. In particular, the forecasting of the price and the directional movement of a stock price is still a largely debated and studied topic in finance.

Concerning this stream of research, a wide variety of models have been tested and retested, and a large number of variables have been listed as potential sources of useful information to make price predictions. However, while the number of reported significant variables increases, the models become increasingly complex and harder to interpret in an intuitive and economically consistent way: as discussed in Peng and Nagata (2020), an adequate model should lie on an optimal middle-ground between (i) describing well the data taken from the sample (a small in-sample error), and (ii) deriving patterns that are likely to generalize well for future and yet unseen data (a small overall complexity). Since the past data is filled not only with useful information but also an intrinsic component of noise, merely "memorizing" the past data tends to be not enough to make good predictions for out-of-sample data. In statistics, this trade-off is also known as the "bias–variance dilemma".

Specifically in finance, the advancements in artificial intelligence led to an "overflow" of potential informative independent variables, as studies from the scientific literature use increasingly more different variables to explain financial relationships or to predict a particular financial variable. In the scope of asset pricing, a large number of different factors were proposed by the recent literature, leading to a proliferation of candidate explanatory factors, dubbed as a "factor zoo" by Cochrane (2011); similarly, there are also various indicators from both fundamental and technical analysis listed in the stock price prediction literature. In addition, many other technical analysis indicators are used by investors and market practitioners but were not analyzed

* Corresponding author at: Brazilian Ministry of Economy, Esplanade of Ministries, Block P, Brasilia, Distrito Federal, 70048-900, Brazil.
E-mail addresses: peng.yaohao@economia.gov.br (Y. Peng), pedroa@unb.br (P.H.M. Albuquerque), herbertkimura@unb.br (H. Kimura), cayan@lamfo.unb.br (C.A.P.B. Saavedra).
URL: http://www.lamfo.unb.br (P.H.M. Albuquerque).

https://doi.org/10.1016/j.mlwa.2021.100060
Received 19 December 2020; Received in revised form 24 May 2021; Accepted 2 June 2021
Available online 8 June 2021

by the recent scientific literature in finance; therefore, the variety of technical analysis indicators used for prediction reflects a "factor zoo" of its own.

In this sense, this paper intends to identify which technical analysis indicators have been applied in recent scientific researches and financial market trading, as well as to test the predictive performance of those features using deep neural networks with different architectures and regularization parameters. We also analyze potential improvements of feature selection methods to identify the most informative indicators and to yield profitable strategies under the presence of transaction costs. Therefore, the main contributions of this paper can be summarized in the following points:

- **Contribution 1**: Provide a comprehensive compendium of technical analysis indicators considered in recent scientific articles on stock prices prediction, as well as other variables that are used by market professionals that can serve as additional sources of information for researches about stock price prediction;
- **Contribution 2**: Evaluate the relative importance of each technical analysis indicator used by the literature and the market using various feature selection methods;
- **Contribution 3**: Test the empirical performance of deep neural networks for seven markets by applying different settings of architecture and regularization, evaluating not only the classification metrics but also estimating the maximum bearable transaction cost from the perspective of an investor in order to obtain actual profitability from the yielded strategies

This paper is structured as follows: Section 2 discussed the relevance of feature selection in financial forecasting and presented the recent literature on machine learning techniques applied to stock price predictions and detailed the theoretical contributions of this paper with regard to the technical analysis indicators applied in this literature; Section 3 describes the methods applied in this paper, presenting the structure and regularization of deep neural networks, feature selection algorithms (namely SFFS, TS, and LASSO) and the details of our empirical experiments; Section 4 discussed the results in terms of predictive performance and profitability of the models for the seven analyzed markets and discussed the practical contributions of our findings; finally, Section 5 summarized the conclusions of the study and presented suggestions for further research.

2. Theoretical background

2.1. Factor zoo and feature selection in finance

Feature selection is an NP-hard problem that involves finding an optimal subset from the original features that leads to the best predictive performance. Since that it is unfeasible to test the models using all possible feature subsets, there are many possible heuristic rules to perform this task. As indicated in the survey paper of Xue, Zhang, Browne, and Yao (2015), a core challenge is the scalability of the feature selection methods as the number of observations and features grow, not only in terms of computational cost but also regarding the interaction between the variables and its impact on real-world decision-making, reinforcing the need for more sophisticated algorithms. In this sense, Xue, Xue, and Zhang (2019) proposed a particle swarm optimization algorithm with a self-adaptive mechanism for feature selection on large-scale datasets, as an improvement over existing evolutionary computation methods that perform well for smaller-scale feature selection, but that may require a costly parameter tuning when dealing with datasets of high dimensionality and may get stuck in local optima given larger solution spaces.

In financial applications, a recent research topic related to feature selection and the challenge of scalability is the "factor zoo". The "factor zoo" refers to the big number of features that was reported to be significant to model the expected value of stock market returns in comparison to classic asset pricing models with linear structure and a only a small number of features, such as Fama and French (1993, 1996). The evolution of asset pricing models towards more complex models is discussed in Harvey, Liu, and Zhu (2016), which identified an extensive set composed of 316 variables with reported significance to predict the cross-section of expected financial returns. The study argues that most factors do not bring significant improvements to the models' performance, evidencing the presence of many non-informative and redundant variables that add more noise than explanatory power. Furthermore, testing for the factors' significance using t-tests, the authors concluded that the high number of non-significant factors might likely indicate that many research findings reported in financial economics papers actually do not hold, favoring instead classic and more parsimonious models. These results highlight a common challenge in predicting financial variables, which is the existence of a large number of potentially useful features, with many of them having high correlation, thus carrying a large degree of redundant information as the number of predictors grow.

Indeed, the recent financial literature indicates that the useful portion of the information contained in a large number of variables from the "factor zoo" can be summarized in a small number of principal components. For instance, Kozak, Nagel, and Santosh (2020) analyzed the effects of high dimensionality in financial models, testing the effects of introducing nonlinear interactions between 130 factors up to degree-3 polynomials, and subsequently applying dimensionality reduction techniques considering ℓ_1 and ℓ_2 regularizations to increase the model's sparsity. The results showed that a very small number of principal components are able to capture almost all of the out-of-sample explanatory power, while most principal components are non-informative. Moreover, the introduction of additional regularized principal components does not hinder the model's sparsity and does not improve predictive performance either.

Similar results were found in Feng, Giglio, and Xiu (2020), which proposed a Two-Pass Regression approach with Double Selection LASSO with Monte Carlo simulations, incorporating the existence of model-selection mistakes that lead to an omitted variable bias. Using 99 risk factors as inputs and testing for data between 1980 and 2016 of companies listed in NYSE, AMEX, and NASDAQ, the authors reported that most recent proposed factors are statistically "redundant" or "useless", while relatively few were shown to be statistically "useful". Furthermore, the significance of the variables selected by the proposed method was shown to be more stable than the standard LASSO model. The findings of those studies emphasize the importance of regularization techniques to reduce the models' level of complexity, filtering out the redundant information from the "factor zoo".

Comparing the relevant features from the "factor zoo" and classic models in asset pricing, Hwang and Rubesam (2019), tested linear models from a set of 83 factors from the asset pricing literature "factor zoo" using a Bayesian estimation for seemingly unrelated regression models. The authors tested the models for US stocks from 1980 to 2016 and found out that only 10 factors were selected by the proposed method. In addition, only 5 to 6 factors showed actual significance on explaining the assets returns, including some selected factors that, in certain periods, are not covered in traditional factor models like Fama and French (1992, 2015). The authors also pointed out that the only factor that was consistently selected as a relevant variable throughout the periods was the excess market return. The results is consistent with the findings of Laloux, Cizeau, Bouchaud, and Potters (1999), Nobi, Maeng, Ha, and Lee (2013) and Sensoy, Yuksel, and Erturk (2013), which identified that (i) the market systematic risk is responsible for the largest eigenvalue of financial covariance matrices in many different financial markets and time horizons; (ii) this highest eigenvalue is significantly larger than the remaining ones; and (iii) the vast majority of eigenvalues accounts for noisy information, falling into the theoretical bounds of a purely random Wishart covariance matrix. Nevertheless, as discussed in Peng, Albuquerque, do Nascimento, and

Table 1
Technical analysis indicators used in recent financial prediction studies that applied machine learning models.

Variable	References
Simple moving average	Chang and Fan (2008), Chang, Liu, Lin, Fan, and Ng (2009), Huang and Tsai (2009) and Thawornwong, Enke, and Dagli (2003) Chang et al. (2012), Kara, Boyacioglu, and Baykan (2011), Vanstone and Finnie (2010) and Yu, Chen, Wang, and Lai (2009) Chen, Cheng and Tsai (2014), Creamer (2012), de Oliveira, Nobre, and Zárate (2013) and Patel, Shah, Thakkar, and Kotecha (2015a) Chiang, Enke, Wu, and Wang (2016), Gorenc Novak and Velušček (2016), Patel, Shah, Thakkar, and Kotecha (2015b) and Weng, Ahmed, and Megahed (2017) Alhashel, Almudhaf, and Hansz (2018) and Henrique, Sobreiro, and Kimura (2018)
Weighted moving average	Gorenc Novak and Velušček (2016), Kara et al. (2011) and Patel et al. (2015a, 2015b) Alhashel et al. (2018) and Henrique et al. (2018)
Exponential moving average	Ang and Quek (2006), Tay and Cao (2001), Vanstone and Finnie (2010) and Yu et al. (2009) Chen, Xiao, Sun, and Wu (2017), Creamer (2012), Gorenc Novak and Velušček (2016) and Ticknor (2013) Alhashel et al. (2018), Nakano, Takahashi, and Takahashi (2018) and Weng et al. (2017)
Momentum	Creamer (2012), Kara et al. (2011), Kim (2003), Kim and Han (2000) and Yu et al. (2009) Chen, Cheng et al. (2014), de Oliveira et al. (2013), Patel et al. (2015a) and Patel et al. (2015b) Chiang et al. (2016), Gorenc Novak and Velušček (2016) and Weng et al. (2017)
Stochastic K%	Chang et al. (2009), Kim (2003), Kim and Han (2000), Kwon and Moon (2007) and Thawornwong et al. (2003) Huang and Tsai (2009), Kara et al. (2011), Vanstone and Finnie (2010) and Yu et al. (2009) Chang et al. (2012), Chen, Kuo, Huang and Chen (2014), de Oliveira et al. (2013) and Ticknor (2013) Patel et al. (2015a, 2015b) Alhashel et al. (2018) and Nakano et al. (2018)
Stochastic D%	Chang and Fan (2008) and Kim (2003), Kim and Han (2000), Kwon and Moon (2007) Chang et al. (2009), Huang and Tsai (2009), Kara et al. (2011) and Yu et al. (2009) Chang et al. (2012), Chen, Kuo et al. (2014), de Oliveira et al. (2013) and Ticknor (2013) Nakano et al. (2018) and Patel et al. (2015a, 2015b)
Slow stochastic D%	Kim (2003), Kim and Han (2000) and Yu et al. (2009)
Relative strength index	Armano, Marchesi, and Murru (2005), Kim (2003), Kim and Han (2000) and Thawornwong et al. (2003) Chang and Fan (2008), Chang et al. (2009), Huang and Tsai (2009) and Kwon and Moon (2007) Kara et al. (2011), Vanstone and Finnie (2010) and Yu et al. (2009) Chang et al. (2012), Creamer (2012), de Oliveira et al. (2013) and Ticknor (2013) Chen, Cheng et al. (2014), Gorenc Novak and Velušček (2016) and Patel et al. (2015a, 2015b) Alhashel et al. (2018), Nakano et al. (2018) and Weng et al. (2017) Henrique et al. (2018)
Moving average convergence–divergence	Armano et al. (2005), Kwon and Moon (2007), Tay and Cao (2001) and Thawornwong et al. (2003) Chang and Fan (2008), Chang et al. (2009), Huang and Tsai (2009) and Yu et al. (2009) Chang et al. (2012), Creamer (2012), Kara et al. (2011) and Vanstone and Finnie (2010) Chen, Cheng et al. (2014), Chen, Kuo et al. (2014), de Oliveira et al. (2013) and Patel et al. (2015a) Alhashel et al. (2018), Chiang et al. (2016), Nakano et al. (2018) and Patel et al. (2015b)
William's R%	Chang et al. (2009), Huang and Tsai (2009) and Kim (2003), Kim and Han (2000) Chang et al. (2012), Kara et al. (2011), de Oliveira et al. (2013) and Ticknor (2013) Alhashel et al. (2018), Chen, Cheng et al. (2014) and Patel et al. (2015a, 2015b)
Accumulation/Distribution oscillator	Kara et al. (2011), Kim (2003), Kim and Han (2000) and Yu et al. (2009) Alhashel et al. (2018), Henrique et al. (2018) and Patel et al. (2015a, 2015b)
Commodity channel index	Kara et al. (2011), Kim (2003), Kim and Han (2000) and Yu et al. (2009) Alhashel et al. (2018), Gorenc Novak and Velušček (2016) and Patel et al. (2015a, 2015b)
Rate of change	Armano et al. (2005), Kim (2003), Kim and Han (2000) and Tay and Cao (2001) Chang et al. (2009), Creamer (2012), Gorenc Novak and Velušček (2016), Weng et al. (2017) and Yu et al. (2009) Alhashel et al. (2018)
Disparity	Kim (2003), Kim and Han (2000), Weng et al. (2017) and Yu et al. (2009)
Price oscillator	Kim (2003), Kim and Han (2000) and Yu et al. (2009)
Psychological line	Chang and Fan (2008), Chen, Cheng et al. (2014) and Huang and Tsai (2009)
Directional indicator up	Huang and Tsai (2009)
Directional indicator down	Huang and Tsai (2009)
Bias	Chang and Fan (2008), Chang et al. (2009, 2012), Chen, Cheng et al. (2014) and Huang and Tsai (2009)
Volume Ratio	Huang and Tsai (2009)
A Ratio	Huang and Tsai (2009)
B Ratio	Huang and Tsai (2009)
Average true range	Henrique et al. (2018) and Vanstone and Finnie (2010)
Bollinger band upper	Alhashel et al. (2018), Creamer (2012) and de Oliveira et al. (2013)
Bollinger band lower	Alhashel et al. (2018), Creamer (2012) and de Oliveira et al. (2013)
Directional movement indicator	Alhashel et al. (2018)
Keltner channel upper band	Alhashel et al. (2018)
Keltner channel lower band	Alhashel et al. (2018)
Triangular moving average	Alhashel et al. (2018)
Moving average envelope upper	Alhashel et al. (2018)

(*continued on next page*)

Machado (2019), the variance held by the top eigenvalue was smaller upon the application of nonlinear estimator for covariance matrices, suggesting that part of the information that would be considered as irrelevant for linear models have predictive value when combined with nonlinear methods.

The aforementioned results corroborate that the "factor zoo" of financial variables can be reduced to a smaller set of relevant features without inducing significant losses to the models' explanatory power. Moreover, the results suggest that the high dimensionality of the "factor zoo" can be explored not only by filtering out irrelevant features but

Table 1 (*continued*).

Variable	References
Moving average envelope lower	Alhashel et al. (2018)
Rex oscillator	Alhashel et al. (2018)
Negative volume index	Creamer (2012)
Positive volume index	Creamer (2012)
Volume adjusted moving average	Chavarnakul and Enke (2009)
Highest price ratio	Vanstone and Finnie (2010)
Lowest price ratio	Vanstone and Finnie (2010)
Opening price	de Oliveira et al. (2013)
Closing price	de Oliveira et al. (2013)
Minimum price	de Oliveira et al. (2013)
Minimum price	de Oliveira et al. (2013)
Volume	Chang et al. (2009), de Oliveira et al. (2013) and Gorenc Novak and Velušček (2016)
Volume momentum	Chiang et al. (2016)
Moving price level percentage	Chiang et al. (2016)
Percent price oscillator	Chiang et al. (2016)
Parabolic stop and reverse	Alhashel et al. (2018) and Gorenc Novak and Velušček (2016)
On balance volume	Creamer (2012), Nakano et al. (2018), de Oliveira et al. (2013) and Tay and Cao (2001)
Volatility	Tay and Cao (2001)
Money flow index	Thawornwong et al. (2003)
Variance ratio	Yu et al. (2009)
Linear regression slope	Yu et al. (2009)

Table 2
Technical analysis indicators not used in recent high-impact financial studies.

Variable	References
Acceleration band up	Trading Technologies (2019)
Acceleration band down	Trading Technologies (2019)
Accumulation/Distribution index	Fidelity Investments (2019) and Trading Technologies (2019)
Money flow multiplier	StockCharts (2019)
Accumulation distribution line	StockCharts (2019) and TradingView (2019)
Absolute price oscillator	Fidelity Investments (2019) and Trading Technologies (2019)
Aroon indicator positive	StockCharts (2019) and Trading Technologies (2019) Fidelity Investments (2019) and TradingView (2019)
Aroon indicator negative	StockCharts (2019) and Trading Technologies (2019) Fidelity Investments (2019) and TradingView (2019)
Aroon oscillator	StockCharts (2019) and Trading Technologies (2019)
Average directional movement index	StockCharts (2019) and Trading Technologies (2019) Fidelity Investments (2019)
Average true range Percent	Fidelity Investments (2019)
Average volume	Fidelity Investments (2019)
Bollinger band width	Fidelity Investments (2019) and StockCharts (2019) TradingView (2019)
Bollinger band %B	Fidelity Investments (2019) and StockCharts (2019) TradingView (2019)
Band width	Trading Technologies (2019)
Chaikin money flow	Fidelity Investments (2019) and StockCharts (2019) TradingView (2019)
Chaikin oscillator	StockCharts (2019) and TradingView (2019)
Chaikin volatility	Fidelity Investments (2019)
Chande momentum oscillator	Fidelity Investments (2019) and Trading Technologies (2019)
Chandelier exit long	StockCharts (2019)
Chandelier exit short	StockCharts (2019)
Choppiness index	TradingView (2019)
Coppock curve	StockCharts (2019)
Detrended price oscillator	Fidelity Investments (2019) and StockCharts (2019) TradingView (2019)
Donchian channel	Cavendish Astrophysics (2011), TradingView (2019)
Double exponential moving average	Trading Technologies (2019)
Double smoothed stochastic	Fidelity Investments (2019)
Ease of movement	StockCharts (2019) and TradingView (2019)
Force index	StockCharts (2019) and TradingView (2019)
Hull moving average	Fidelity Investments (2019)
Kaufman's adaptive moving average	StockCharts (2019)
Linear regression intercept	StockCharts (2019) and Trading Technologies (2019) Fidelity Investments (2019)
Linear regression trend line	Fidelity Investments (2019) and Trading Technologies (2019)
MACD histogram	StockCharts (2019)
Mass index	StockCharts (2019)
Raw money flow	Fidelity Investments (2019)
Midpoint	Trading Technologies (2019)
Midprice	Trading Technologies (2019)
Normalized average true range	Trading Technologies (2019)
Standard support 1	Fidelity Investments (2019)
Standard support 2	Fidelity Investments (2019)

(*continued on next page*)

Table 2 (*continued*).

Variable	References
Standard resistance 1	Fidelity Investments (2019)
Standard resistance 2	Fidelity Investments (2019)
Fibonacci support 1	Fidelity Investments (2019)
Fibonacci support 2	Fidelity Investments (2019)
Fibonacci resistance 1	Fidelity Investments (2019)
Fibonacci resistance 2	Fidelity Investments (2019)
Demark pivot point	Fidelity Investments (2019)
Demark support	Fidelity Investments (2019)
Demark resistance	Fidelity Investments (2019)
Price channel upper	StockCharts (2019) and Trading Technologies (2019)
Price channel lower	StockCharts (2019) and Trading Technologies (2019)
PPO histogram	StockCharts (2019)
Percentage volume oscillator	StockCharts (2019)
PVO histogram	StockCharts (2019)
Price volume trend	Trading Technologies (2019) and TradingView (2019)
Pring's know sure thing oscillator	StockCharts (2019) and TradingView (2019)
Pring's special K	StockCharts (2019)
Relative vigor index	Fidelity Investments (2019)
Standard error	Fidelity Investments (2019)
Stochastic RSI	Fidelity Investments (2019) and TradingView (2019)
Triple exponential moving average	Trading Technologies (2019)
Triple exponential moving average oscillator	StockCharts (2019) and Trading Technologies (2019) TradingView (2019)
True strength index	StockCharts (2019)
Typical price	Fidelity Investments (2019)
Ulcer index	StockCharts (2019)
Ultimate oscillator	StockCharts (2019) and Trading Technologies (2019) Fidelity Investments (2019) and TradingView (2019)
Volume oscillator	Fidelity Investments (2019)
Volume price trend	Cavendish Astrophysics (2011)
Volume weighted average price	StockCharts (2019) and Trading Technologies (2019) TradingView (2019)
Vortex indicator positive	StockCharts (2019)
Vortex indicator negative	StockCharts (2019)
Welles wilder's smoothing average	Trading Technologies (2019)

Table 3
Distribution of the number of times that each technical analysis indicator (Literature + Market) was chosen by the feature selection methods throughout the seven markets.

Technical analysis indicators	Number of times chosen
DPO	21
HULL, MFM	16
ADO, APO, BIAS, DEMA, VOLAT	15
MOM, NVI, RVI, VOLR	14
ADX, BB_BW, BWW, DMI, DSS, STRSI, FORCE, MQO_BETA	13
ADL, ATRP, DIU, MASS, MQO_STD, NATR, ULTOSC	12
AVOL, BRATIO, CCI, CHOSC, CVOL, EMV, HPR, PSK, PSY, REX, RMF, ROC, STOCH_D, VAMA, VMOM	11
AR_NEG, AR_POS, CHOPPINESS, CMO, MQO_ALPHA, MQO_PRED, PVOH, RSI, VARR, VOLUME, VOOSC	10
ARATIO, CLOSE, COPP, DID, KAMA, KST, LPR, MACDH, MIDPOINT, MPP, OBV, OSCP, PVI, PVOI, STOCH_K	9
AD, ATR, CMF, DS1, NVOI, PERC_B, PVT, TP, TSI	8
AR_OSC, DISP, EMA, FR2, KC_L, MAE_UP, PPOH, SS1, STOCH_D_SLOW, VPT	7
AB_DOWN, BB_LOW, DONCHIAN, FS1, FS2, KC_U, MAE_LOW, MFI, PC_DOWN, SAR, TRIX, WILL_R	6
BB_UP, CHAND_SHORT, DR1, FR1, MIDPRICE, OPEN, PVO, SR1, SR2, SS2, TRIMA, VWAP	5
AB_UP, CHAND_LONG, MACD, PD1, PPO, ULCER	4
MAXX, MINN, TEMA	3
PC_UP, SMA, WWS	2
WMA	1

also by adding a nonlinear component to allow the identification of additional patterns to the data. Inspired by these findings, we took into consideration both issues in this paper, performing similar analyses over the set of technical analysis indicators to verify whether the relevant information is also concentrated in a small number of features and to which extent the elimination of non-informative features can enhance the predictive performance and the profitability of the respective trading strategies.

The literature findings on the "factor zoo" for asset pricing tasks can also be extended to the context of stock price prediction using technical analysis indicators. Since there is also a large number of features being used as predictors, the use of feature selection methods becomes important, notably when the variables are highly correlated among themselves. For example, Creamer (2012) described a trading algorithm for high-frequency data of EURO STOXX 50 and DAX index Futures, applying machine learning techniques, such as boosting and bagging. The paper incorporated many technical indicators, trading rules, and liquidity indicators, and the empirical analysis showed that the whole set of variables, although containing indicators with a high degree of redundancy, yielded models with lower overall error rates.

As discussed in Xue, Tang, Xu, Liang, and Neri (2021), in addition to the number of features chosen as relevant and their classification accuracy, the feature selection problem should also take into account the issue of data unreliability, which is likely to occur when the corresponding dataset contains a large proportion of missing values. The authors applied five multi-objective optimization algorithms on six incomplete data sets, using mean imputation and the KNN classifier to evaluate the feature subsets, amongst which the non-dominated sorting

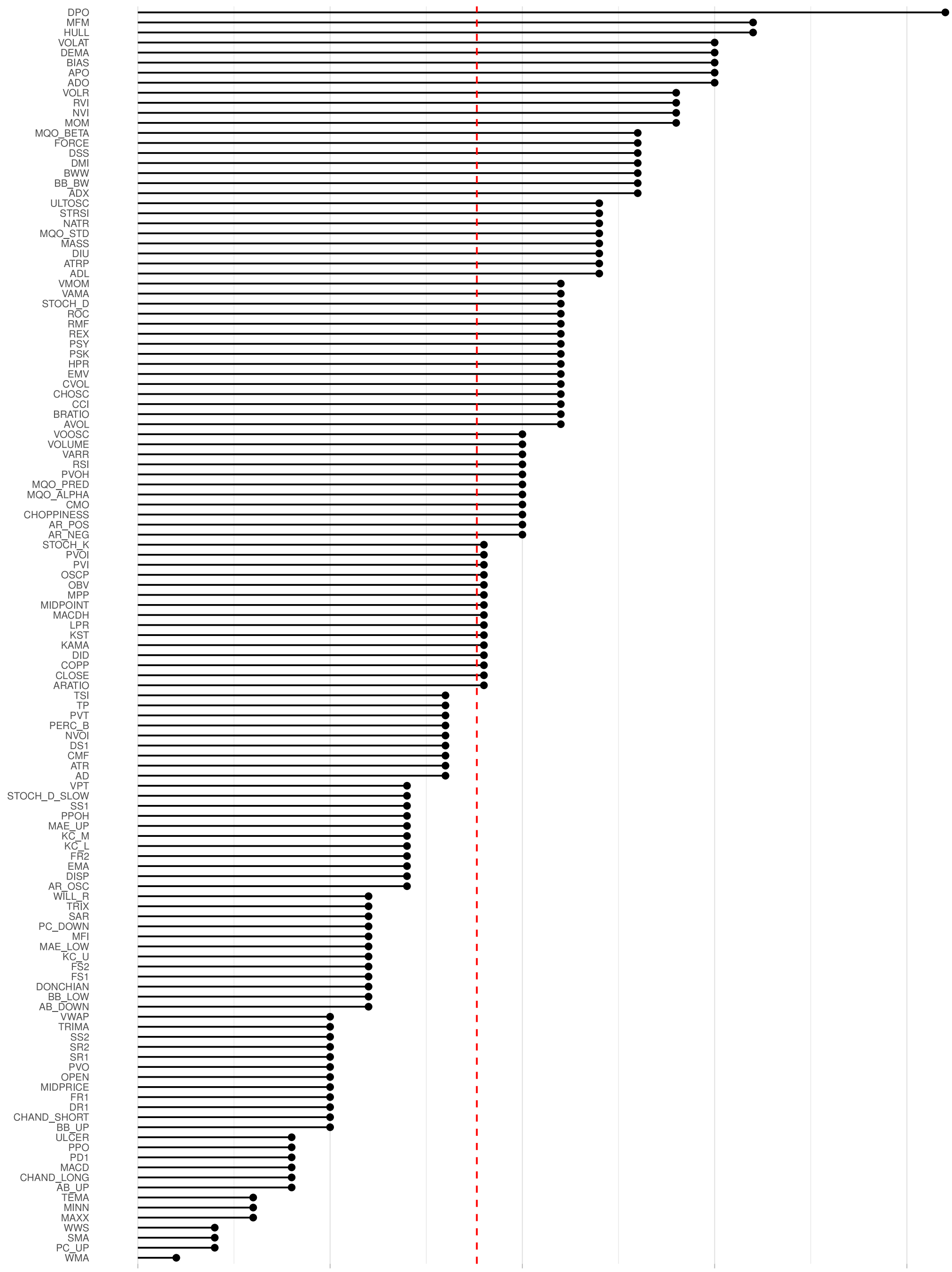

Fig. 1. Histogram of the number of times that each technical analysis indicator (Literature + Market) was chosen by the feature selection methods throughout the seven markets. The dashed vertical line indicates the average value considering all indicators.

genetic algorithm-III (NSGA-III) presented the best results in terms of inverted generational distance and hyper-volume. In financial contexts, missing values are also common in datasets like annual accounting reports and commodity prices, and depending on the method applied to perform data imputation, the forecasting performance or the conclusions of a causal inference can differ significantly, as seen in works like Choudhury and Pal (2019) and Kofman and Sharpe (2003).

In turn, de Oliveira et al. (2013) considered 46 variables distributed into macroeconomic variables, firm fundamental indexes, historical prices, and technical analysis indicators to predict the stock price direction of a Brazilian firm. The authors applied a filter feature selection method with correlation criterion, reducing the variable set to 18 features and applying them to a shallow artificial neural network (henceforth ANN) with one hidden layer, which yielded an out-of-sample accuracy rate of 87.5%. Finally, Jadhav, He, and Jenkins (2018) applied a feature selection procedure based on a genetic algorithm using three machine learning models as a baseline for credit scoring prediction, finding out that the classification performance can be boosted using an appropriate feature selection method, which can also provide a measure of the importance of the analyzed variables for decision-making.

The findings presented in this subsection reveals two challenges for forecasting financial variables: on the one hand, the application of nonlinear models can help to extract complex patterns that increase prediction performance over linear models; on the other hand, the existence of a big number of potentially useful features adds a high

Table 4
Distribution of the number of times that each technical analysis indicator (Only Literature) was chosen by the feature selection methods throughout the seven markets.

Technical analysis indicators	Number of times chosen
ADO	16
BIAS, DIU, MACD, VOLR, VOLUME	14
CCI, DID, LPR, NVI, STOCH_K, WILL_R	13
REX, VARR	12
ARATIO, DISP, DMI, STOCH_D	11
HPR, MPP, PPO, PSY, ROC, WMA	10
MOM, MQO_BETA, OSCP, SAR, VOLAT	9
BRATIO, MFI, OBV, PVI	8
MAE_LOW	7
RSI, STOCH_D_SLOW, VAMA, VMOM	6
ATR, BB_UP, CLOSE, SMA	5
BB_LOW, KC_L, MAE_UP, MAXX, MINN	4
EMA, OPEN	3
KC_U, TRIMA	2

level of noise on the models, thus demanding the application of feature selection methods as a preprocessing step in order to keep only the most informative variables for the models' training. Hence, in this paper, we tackle both challenges — by applying various settings of deep learning models jointly with regularization and filtered feature sets obtained from different feature selection techniques. Thus, the next subsections focus on each of these issues, with Section 2.2 providing a review of the literature on the application of nonlinear models in financial contexts, while Section 2.3 details the size of the "factor zoo" for technical analysis indicators, listing as well indicators used by market professionals but not explored by scientific researches.

2.2. Nonlinearity and machine learning in financial forecasting

Machine learning methods became a widely studied topic over recent years due to their overall flexibility to the observed data and the absence of restrictive assumptions of distribution and functional forms. Instead of hypothesizing over cause–effect relationships, the basic premise of machine learning methods is to "learn from the data" and identify potential non-intuitive patterns that contribute to better forecasting (Chicco, 2017). One of the main features of machine learning is the ensemble of linear and nonlinear techniques, allowing the modeling of high complexity relationships with a reduced number of functions and hyperparameters. For instance, a neural network is essentially a linear regression with "chunks" of nonlinearity, and a classification support vector machine is a linear separator in a feature space with an arbitrarily high dimension, dependent on the applied Kernel function, which can reflect complex relationships (Duriez, Brunton, & Noack, 2017).

While the presence of nonlinear interaction variables can reveal additional patterns and joint significance between the original variables, it also augments the "zoo size" – *i.e.:* the number of features involved in the analysis. The presence of a non-informative feature can be problematic due to the possible spreading of its noisy effect to other useful factors: for instance, taking a set of n variables, the number of cross-interactions of degree 2 would yield $n + \binom{n}{2}$ variables, and of degree 3, $n + n(n-1) + \binom{n}{3}$ more variables. While the number of potentially useful regressors already leads to a high level of noise and high proneness of overfitting, taking the analysis to nonlinear relationships may hinder the model's overall predictive power even further. Analogously, Mullainathan and Spiess (2017) also points out that the inclusion of irrelevant variables as inputs for machine learning models can potentially harm the model's predicting performance.

In traditional econometrics, using highly correlated variables leads to the well-known problem of multicollinearity, which may imply larger standard errors on the estimates and interfere with the model's adequacy, making it less robust. Additionally, multicollinearity increases the chances of inaccuracies in the numerical optimization of the computer algorithms. In machine learning models, the effects are similar. The model tends to overfit the observed data and make inaccurate predictions in new samples. For instance, this can be seen in Guresen, Kayakutlu, and Daim (2011), in which ANNs and ensemble models were tested to predict the daily NASDAQ Index using a sample of 182 days between 2008 and 2009. The authors tested a hybrid neural network that introduced new input variables constructed from GARCH and EGARCH models, and the results of these methods were shown to be worse than the conventional multilayer perceptron, indicating that the added features introduced more noise than actual explanatory power.

Conversely, as discussed in Guyon and Elisseeff (2003), even highly correlated variables (positively or negatively) do not necessarily imply the absence of relevance since there may be informative nonlinear dependencies. In this sense, since machine learning methods rely heavily on nonlinear interactions, an apparently redundant variable subset, although potentially making the model more prone to overfitting and inaccurate predictions, in some cases can yield forecasts that are superior to results obtained by applying linear models.

When dealing with a high number of variables, a common approach to extract the most relevant information out of the feature set is to apply Principal Component Analysis (PCA), which aims at getting linear combinations that account for the largest proportions of explained variance. However, while PCA is very pragmatic, the principal components do not have an immediate implication in the real world, unlike feature selection methods, which take variables "as a whole", maintaining their respective economic and financial interpretations. Furthermore, PCA forces the principal components to be orthogonal — that is, linearly non-correlated. Thus, when nonlinear relationships of the candidate variables are being accounted for, the original interpretability suffers an additional loss since it can be unnatural to imagine the intuition of a polynomial or exponential version of a variable with clear economic meaning. Therefore, in this paper, we opted to take subsets of individual variables instead of combinations of features, given that the principal components themselves are already hard to interpret, and using PCA results as inputs to a nonlinear predictive model like ANN would further hinder the model's interpretability.

Regarding the use of machine learning techniques in stock price prediction, there are already many studies that explore this topic. For instance, Moghaddam, Moghaddam, and Esfandyari (2016) applied ANN to forecast the daily NASDAQ stock exchange return with data from January 2015 to June 2015 using past prices and the day of the week as input variables. Nayak, Pai, and Pai (2016) applied Boosted Decision Tree, Logistic Regression, and SVM to predict the Indian stock market trend using historical prices and market sentiment measured by posts on Twitter. Ramezanian, Peymanfar, and Ebrahimi (2019) applied an integrated model with genetic network programming, neural networks, and ARMA time-series models to predict the Iranian market's daily stock return between 2013 and 2017. Huang and Yen (2019) applied six machine learning models to predict financial distress using

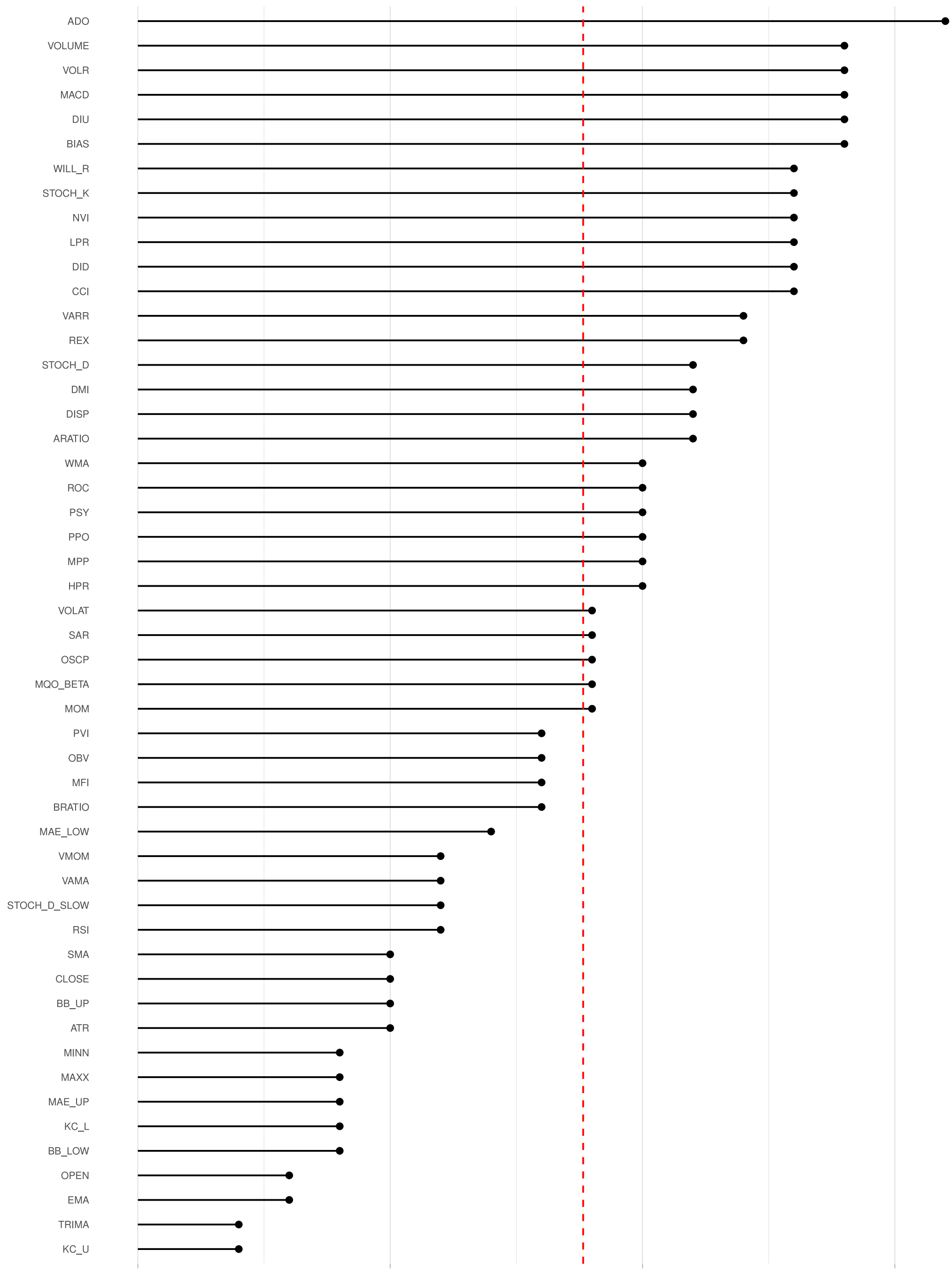

Fig. 2. Histogram of the number of times that each technical analysis indicator (Only Literature) was chosen by the feature selection methods throughout the seven markets. The dashed vertical line indicates the average value considering all indicators.

fundamentalist variables from financial statements of Taiwanese firms from 2010 to 2016. Amongst the tested models, gradient boosting had the best performance, while an ensemble model combining Deep Belief Network and Support Vector Machine performed better than the two models applied individually.

The improvements of nonlinearity in financial applications are also discussed in Gu, Kelly, and Xiu (2020), who applied various machine learning methods — namely principal components regression, partial least squares, generalized linear models, boosted regression trees, random forests, and artificial neural networks — and compared them to simple and penalized (ridge regression, LASSO, and elastic net) linear models. The study measures the risk premium of financial assets using data of nearly 30000 financial assets from NYSE and NASDAQ between 1957 and 2016. The results presented empirical evidence favoring machine learning models in terms of providing a more accurate description of the price oscillation patterns of the analyzed assets in comparison to traditional statistical methods. The authors credited the predictive performance gain to the introduction of nonlinear predictor interactions from those methods, which are not considered by commonly used econometric approaches. Moreover, the authors reported that all models converged to a similar set of "essential predictors" composed mainly by variations in the assets' liquidity and volatility.

Concerning artificial neural networks, Qiu, Song, and Akagi (2016) emphasized the theoretical ability of this class of models to approximate any nonlinear continuous function, reiterating that the nonlinear behavior of stock market returns is one of the main challenges of

Table 5
Out-of-sample prediction results for assets of S&P 100 Index.

Technical analysis indicators	Feature selection method	Hidden layers	Dropout	Accuracy	Precision	Recall	F-Score
Literature + market (Tables 1 and 2)	None	3	0	0.6468	0.6582	0.6785	0.6682
			0.3	0.6528	0.6725	0.6580	0.6651
		5	0	0.6447	0.6511	0.6937	0.6717
			0.3	0.6537	0.6653	0.6824	0.6738
		7	0	0.6474	0.6600	0.6749	0.6673
			0.3	0.6528	0.6545	0.7149	0.6834
	SFFS	3	0	0.6472	0.6683	0.6488	0.6584
			0.3	0.6496	0.6716	0.6484	0.6598
		5	0	0.6515	0.6653	0.6742	0.6697
			0.3	0.6492	0.6724	0.6448	0.6583
		7	0	0.6486	0.6684	0.6540	0.6611
			0.3	0.6473	0.6655	0.6572	0.6613
	TS	3	0	0.6511	0.6721	0.6526	0.6622
			0.3	0.6512	0.6611	0.6861	0.6733
		5	0	0.6521	0.6643	0.6794	0.6718
			0.3	0.6469	0.6796	0.6173	0.6470
		7	0	0.6516	0.6820	0.6279	0.6538
			0.3	0.6439	0.6369	0.7455	0.6869
	LASSO	3	0	0.6477	0.6577	0.6834	0.6703
			0.3	0.6542	0.6667	0.6804	0.6735
		5	0	0.6497	0.6510	0.7148	0.6814
			0.3	0.6539	0.6588	0.7043	0.6808
		7	0	0.6481	0.6667	0.6567	0.6617
			0.3	0.6513	0.6794	0.6335	0.6557
Literature (Table 1)	None	3	0	0.5331	0.5535	0.5638	0.5586
			0.3	0.5036	0.5235	0.5870	0.5534
		5	0	0.6325	0.6475	0.6556	0.6515
			0.3	0.5108	0.5251	0.6941	0.5979
		7	0	0.5436	0.5579	0.6218	0.5881
			0.3	0.5240	Predicted only rises		
	SFFS	3	0	0.5123	0.5268	0.6872	0.5963
			0.3	0.5121	0.5274	0.6634	0.5877
		5	0	0.5062	0.5263	0.5788	0.5513
			0.3	0.5103	0.5292	0.5938	0.5596
		7	0	0.5062	0.5240	0.5542	0.5387
			0.3	0.5079	0.5280	0.5744	0.5502
	TS	3	0	0.5094	0.5256	0.6545	0.5830
			0.3	0.5133	0.5251	0.7448	0.6160
		5	0	0.5115	0.5244	0.7284	0.6098
			0.3	0.5128	0.5256	0.7201	0.6077
		7	0	0.5155	0.5249	0.7931	0.6317
			0.3	0.5069	0.5260	0.5974	0.5594
	LASSO	3	0	0.5128	0.5265	0.6990	0.6006
			0.3	0.5200	0.5268	0.8248	0.6430
		5	0	0.5109	0.5258	0.6783	0.5924
			0.3	0.5181	0.5267	0.7931	0.6330
		7	0	0.5121	0.5253	0.7161	0.6060
			0.3	0.4947	0.5256	0.3669	0.4321

forecasting financial series. In their study, ANNs were applied to predict the return of the Nikkei 225 Index using macroeconomic variables such as monetary base, interest rates, trade flow, and industrial production as inputs. Genetic algorithms and simulated annealing were combined with neural networks to improve the prediction accuracy and to overcome the local convergence problem of the backpropagation algorithm.

Despite the approximation capability of a neural network with a single hidden layer, many recent studies have discussed the usage of deep neural networks, given their ability to learn abstract representations from the data that networks with fewer hidden layers may not capture. A list of recent research that applied deep learning models for financial tasks is available in the survey papers of Ozbayoglu, Gudelek, and Sezer (2020) and Sezer, Gudelek, and Ozbayoglu (2020), with the latter focusing on papers that dealt specifically with financial time-series forecasting.

2.3. Technical analysis indicators and machine learning in stock price predictions

In this last subsection of the literature review, we provide an overview of recent applications that conciliate stock price prediction, machine learning models, and technical analysis indicators, addressing the gaps of the current literature on the intersection of these topics and presenting how this paper contributes to filling this gap.

Technical analysis indicators are commonly used tools in investment evaluation, financial trading, and portfolio selection, being categorized as one of the three main groups of machine learning applications in automated financial trading systems by the survey paper of Huang,

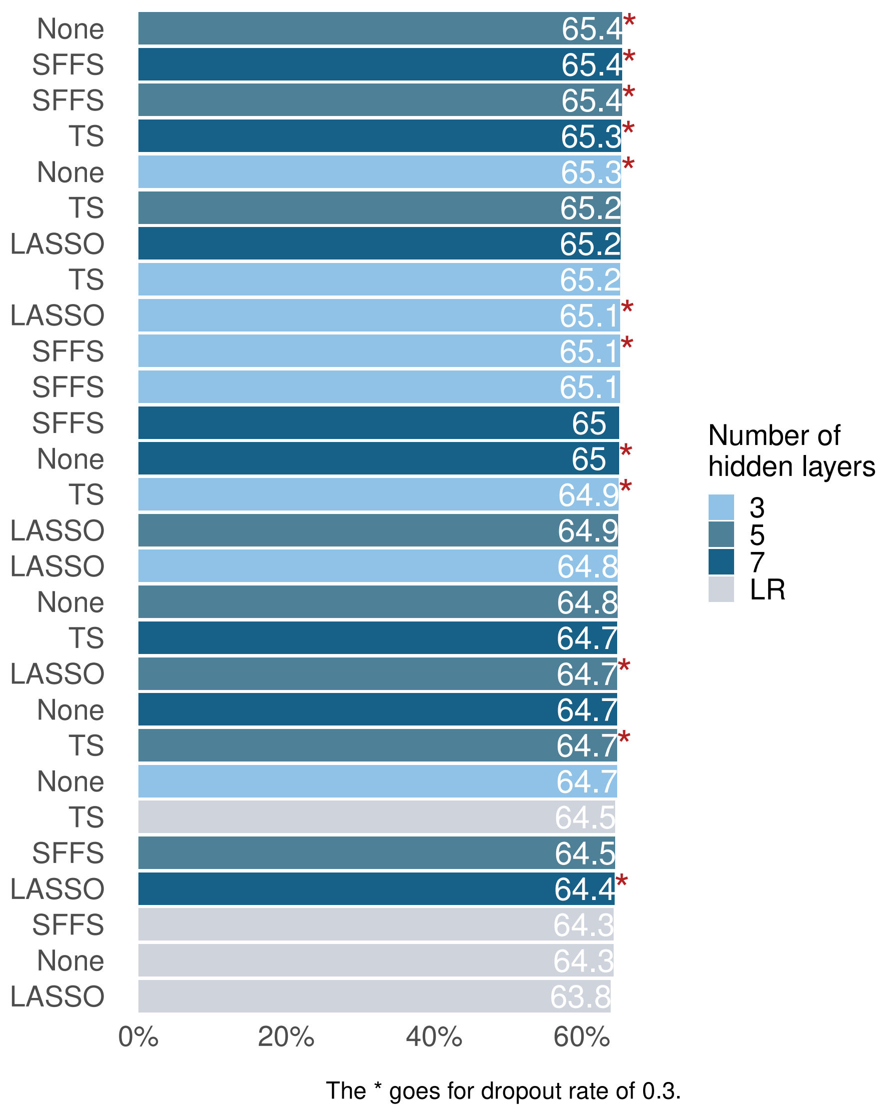

Fig. 3. Out-of-sample accuracy results for assets of S&P 100 Index - Literature + market.

Huan, Xu, Zheng, and Zou (2019). As detailed in Table 1, there is a lengthy list of technical indicators used in the literature as variables to predict a stock's price movement. However, at the same time, many indicators are highly correlated, with some of them being defined by simple mathematical operations or combinations among other technical indicators, which are also candidate criteria for the investor's decision. For example, the Stochastic D% indicator is simply an arithmetic mean of the Stochastic K% indicator for the last n periods, with Stochastic K% itself calculated using the maximum and minimum prices over the last n periods, which are themselves used as separate indicators. In another example, both the Momentum (MOM) and the Rate of Change (ROC) describe the variation of the closing price relative from n periods before, and the only difference between them is that the former states the difference in absolute variation, while the latter gives away the percentage variation.

Given the high degree of "overlapping" among technical analysis indicators, a previous feature selection filtering which variables are most relevant to predict a stock movement is an important procedure to avoid prediction problems arising from redundant variables like multicollinearity and overfitting. Therefore, this paper aimed to observe the improvements in the out-of-sample prediction application of feature selection methods to predict the stock price direction. Besides, we evaluated the effect of feature selection in machine learning models, providing insights about the best combination of machine learning model and feature selection method, as well as the emergence of chaotic behavior of machine learning methods.

The application of technical analysis on stock markets was analyzed in a literature review by Nazário, e Silva, Sobreiro, and Kimura (2017), in which 85 papers published between 1959 and 2016 were classified in terms of the chosen markets, methodologies, consideration of risk and transaction costs, operational tools, among other categories. The analysis showed that artificial neural networks are a widely used tool, mainly due to their consistency for small-range data. In particular, this technique's popularity increased in the last years, as covered by the analysis, coinciding with a growth of interest in machine learning applications in finance.

Given the popularity and empirical effectiveness of machine learning models in financial forecasting, artificial intelligence techniques have been actively applied to the analysis of financial stock prices. As shown in an extensive review by Henrique, Sobreiro, and Kimura (2019), who mapped 57 papers published in high-impact journals, the application of machine learning techniques to the prediction of financial stock prices is still a highly debated research topic in the recent literature, both regarding the forecasting of the market movement direction (a classification problem) and the magnitude of the movement itself (a regression problem). The paper also indicated that the models' input variables are basically divided into fundamentalist and technical indicators, with the most prominent machine learning

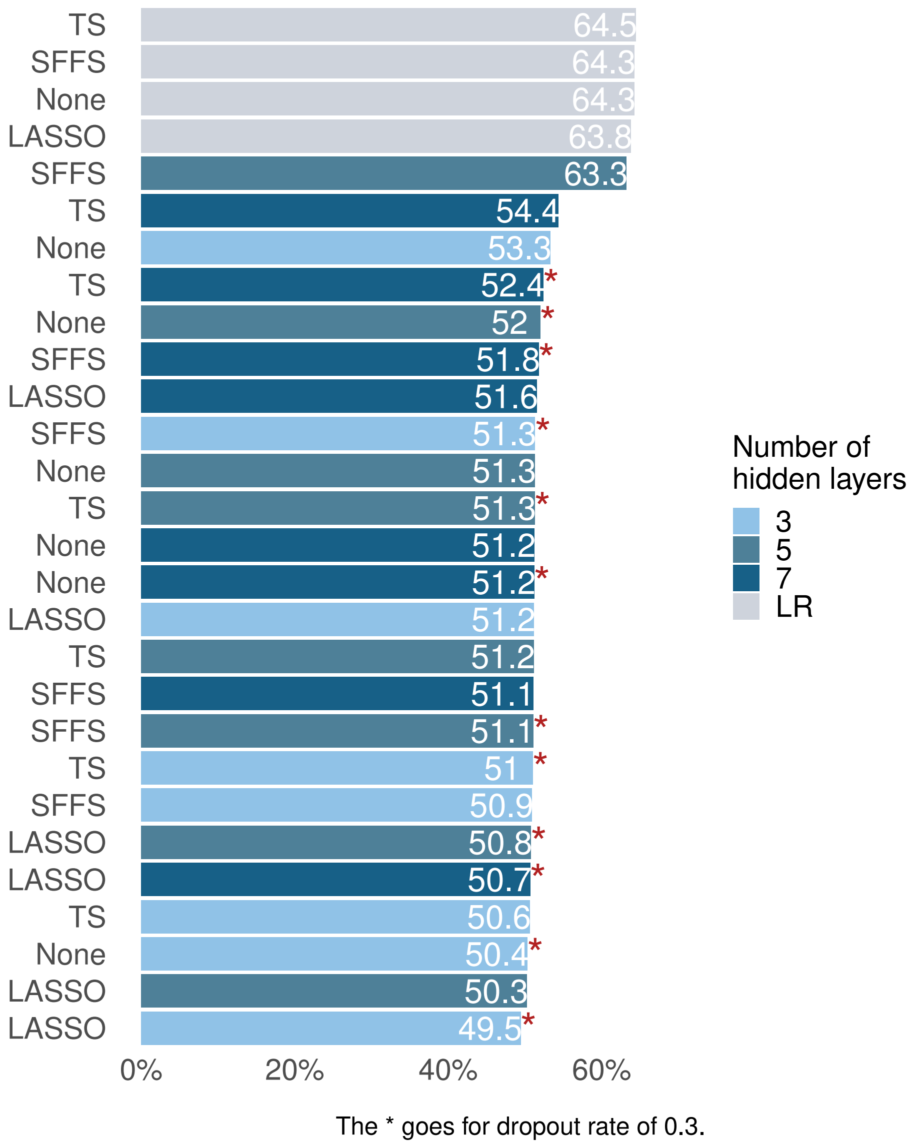

Fig. 4. Out-of-sample accuracy results for assets of S&P 100 Index - Literature.

methods applied to financial market predictions being Artificial Neural Networks, Support Vector Machines and their respective extensions.

For instance, Henrique et al. (2018) used five technical analysis indicators to predict stock price from Brazilian, American, and Chinese financial markets using Support Vector Regression (SVR) for high-frequency data using the Random Walk as the benchmark. The paper argues in favor of the SVR models' predictive power, especially in periods of lower market volatility. da Costa, Nazário, Bergo, Sobreiro, and Kimura (2015) tested the performance of trading strategies in comparison to the buy-and-hold strategy based on technical analysis indicators for 198 stocks of the Brazilian market and analyzed their respective predictability for the market trends under various circumstances of transaction costs. The paper reports that the proposed strategies obtained returns larger than the invested value but have reduced predictive power for future prices.

Żbikowski (2015) applied a volume-weighted extension of a classification SVM to predict the stock price direction for 20 US stocks between 2003 and 2013 using 7 technical analysis indicators. The paper also tested a feature selection method based on Fisher Score ranking, and although the application of this procedure has reached a better performance of trading strategies, the author advised caution and recommended additional research regarding the effectiveness of other feature selection methods. Similarly, Zhu, Jiang, Li, and Zhou (2015) performed an analysis of two Chinese stock exchange indexes between 1991 and 2013. By comparing trading range break, fixed-length moving average, and variable moving average rules, the study's empirical results showed that the former outperformed the others in terms of profitability; specifically, short-term moving average rules worked better than long-term ones. Moreover, White's Reality Check test indicated that the best trading signals from variable moving average and trading range break outperformed the buy-and-hold strategy in a scenario without transactions costs; when they are taken into account, however, there was no statistical evidence indicating the superiority of the technical analysis trading rules.

Alhashel et al. (2018) tested 22 technical analysis trading rules for indexes from 1995 to 2015 of nine financial markets in Asia, namely China, Hong Kong, Indonesia, Japan, Malaysia, Philippines, Singapore, Taiwan, and Thailand. Controlling for transaction costs and strategy risk, the results found evidences of market inefficiency in four of the analyzed markets, implying the existence of predictive power of technical analysis indicators on those markets; on the other five markets, though, the profitability did not outperform the market gains.

Nakano et al. (2018) proposed trading strategies using intraday bitcoin price data applying ANNs for the prediction of the return.

Table 6
Trading profitability and transaction costs of machine learning algorithms for assets of S&P 100 Index.

Technical analysis indicators	Feature selection method	Hidden layers	Dropout	Strategy profitability	Number of transactions	TC_0	TC_{BH}
Literature + market (Tables 1 and 2)	None	3	0	76.3370	91	0.7891	0.4918
			0.3	84.7517	96	0.8742	0.5695
		5	0	−26.2018	125	−0.4878	−0.9811
			0.3	0.2977	213	−0.1306	−0.3258
		7	0	73.7097	89	0.8543	0.5109
			0.3	84.1676	94	0.8464	0.5594
	SFFS	3	0	−33.8689	60	−0.7833	−1.1259
			0.3	−93.9103	127	−0.7350	−0.9511
		5	0	71.9673	98	0.7003	0.4224
			0.3	73.8737	87	0.7715	0.4649
		7	0	−91.4242	72	−1.3961	−1.6452
			0.3	17.7903	210	−0.0991	−0.3066
	TS	3	0	78.5373	96	0.7697	0.4994
			0.3	80.4771	89	0.9240	0.6027
		5	0	−86.9147	73	−1.3871	−1.7970
			0.3	−42.7049	105	−0.4539	−0.6951
		7	0	83.7943	99	0.8832	0.5832
			0.3	77.6707	94	0.8058	0.5031
	LASSO	3	0	61.9372	219	0.2759	0.1193
			0.3	−53.5714	91	−0.6422	−1.0093
		5	0	79.2360	94	0.8422	0.5369
			0.3	65.0487	85	0.8121	0.4136
		7	0	−43.1031	132	−0.3601	−0.5558
			0.3	−31.4264	66	−0.6612	−1.0125
Literature (Table 1)	None	3	0	76.9507	92	0.7869	0.5008
			0.3	72.1548	93	0.7377	0.4497
		5	0	−6.3772	223	−0.0623	−0.2078
			0.3	−18.8899	123	−0.1465	−0.4879
		7	0	77.3609	90	0.8154	0.5065
			0.3	75.7934	93	0.7555	0.4781
	SFFS	3	0	−86.7935	90	−1.5201	−2.0062
			0.3	−87.8024	71	−1.3302	−1.6687
		5	0	76.9180	94	0.8084	0.5005
			0.3	60.8778	91	0.6999	0.3442
		7	0	−35.1803	155	−0.5275	−1.1137
			0.3	−1.5872	76	0.0849	−0.6821
	TS	3	0	71.1070	89	0.7297	0.4288
			0.3	59.3654	86	0.7342	0.3477
		5	0	−43.1454	111	−0.4110	−0.6155
			0.3	−26.8233	70	−0.4755	−0.8923
		7	0	85.1358	91	0.8732	0.5883
			0.3	58.7055	93	0.6590	0.3095
	LASSO	3	0	−26.4434	195	−0.1963	−0.3705
			0.3	−130.7093	1	−130.7092	−160.3766
		5	0	79.8481	89	0.8421	0.5438
			0.3	77.9161	93	0.7701	0.4981
		7	0	−111.1510	89	−1.4438	−1.7723
			0.3	−20.1832	71	−0.1628	−0.9002

Buy-and-Hold strategy profitability over the out-of-sample period: 31.04701

Table 7
Out-of-sample accuracy grouped by target assets and variable sets.

Assets	Literature		Literature + Market	
	Acc. ≤55%	Acc. ≥60%	Acc. ≤55%	Acc. ≥60%
United States (S&P 100 Index)	23	5	0	28
United Kingdom (FTSE 100 Index)	28	0	7	21
France (CAC 40 index)	24	4	3	25
Germany (DAX-30 index)	28	0	5	23
Japan (Top 50 assets from NIKKEI 225 index)	24	4	2	26
China (Top 50 assets from SSE 180 index)	24	4	0	28
Brazil (Bovespa Index)	24	4	0	28

Table 8
Out-of-sample prediction results for assets of FTSE 100 Index.

Technical analysis indicators	Feature selection method	Hidden layers	Dropout	Accuracy	Precision	Recall	F-Score
Literature + market (Tables 1 and 2)	None	3	0	0.6275	0.6108	0.6442	0.6270
			0.3	0.6341	0.6439	0.5532	0.5951
		5	0	0.6181	0.6227	0.5442	0.5808
			0.3	0.6258	0.6562	0.4838	0.5569
		7	0	0.6220	0.6269	0.5498	0.5858
			0.3	0.6332	0.6432	0.5515	0.5938
	SFFS	3	0	0.6357	0.6250	0.6266	0.6258
			0.3	0.6331	0.6406	0.5588	0.5969
		5	0	0.6363	0.6151	0.6732	0.6428
			0.3	0.6266	0.6552	0.4896	0.5605
		7	0	0.6376	0.6340	0.6024	0.6178
			0.3	0.5138	Predicted only drops		
	TS	3	0	0.6452	0.6310	0.6505	0.6406
			0.3	0.6339	0.6651	0.4976	0.5693
		5	0	0.6384	0.6270	0.6327	0.6298
			0.3	0.6335	0.6524	0.5268	0.5829
		7	0	0.6390	0.6420	0.5821	0.6106
			0.3	0.5138	Predicted only drops		
	LASSO	3	0	0.6232	0.6211	0.5768	0.5981
			0.3	0.6380	0.6500	0.5534	0.5978
		5	0	0.6194	0.6330	0.5165	0.5689
			0.3	0.6371	0.6409	0.5765	0.6070
		7	0	0.6256	0.6237	0.5796	0.6009
			0.3	0.5374	0.5209	0.6041	0.5594
Literature (Table 1)	None	3	0	0.5087	0.4963	0.7100	0.5842
			0.3	0.5101	0.4971	0.6708	0.5710
		5	0	0.5155	0.5019	0.4574	0.4786
			0.3	0.5138	Predicted only drops		
		7	0	0.5131	0.4994	0.7265	0.5919
			0.3	0.5138	Predicted only drops		
	SFFS	3	0	0.5158	0.5028	0.3739	0.4289
			0.3	0.5170	0.5034	0.4847	0.4938
		5	0	0.5145	0.5011	0.3259	0.3949
			0.3	0.5151	0.5009	0.7175	0.5899
		7	0	0.5191	0.5044	0.6198	0.5562
			0.3	0.5138	Predicted only drops		
	TS	3	0	0.5170	0.5063	0.2637	0.3468
			0.3	0.5152	0.5045	0.1642	0.2477
		5	0	0.5174	0.5090	0.2064	0.2938
			0.3	0.5138	Predicted only drops		
		7	0	0.5187	0.5077	0.3311	0.4008
			0.3	0.5138	Predicted only drops		
	LASSO	3	0	0.5021	0.4896	0.5723	0.5278
			0.3	0.5088	0.4962	0.6667	0.5689
		5	0	0.5114	0.4968	0.3806	0.4310
			0.3	0.5138	Predicted only drops		
		7	0	0.5112	0.4977	0.6035	0.5455
			0.3	0.5138	Predicted only drops		

Technical analysis indicators and historical return data were used as input variables, and both shallow and deep network architectures were tested. Even considering the effect of transaction costs, the proposed method's risk-adjusted profitability was reported to be significantly greater than the buy-and-hold strategy, especially during a period when bitcoin prices suffered a considerable drawback.

Weng et al. (2017) combined online data information collected from "knowledge bases" Google and Wikipedia with traditional time-series and financial technical analysis indicators to build a trading expert system that operates on a daily periodicity. Machine learning techniques (decision trees, ANN, and SVM) were used as the proposed system's predictive tool. Even though the sample consisted of only one company, the paper reported an 85% directional accuracy for the predictions and claimed improvement over the results of similar works in the literature.

Patel et al. (2015b) proposed a two-stage fusion model between ANN, Random Forest, and Support Vector Regression (SVR) to predict the value of two Indian stock market indexes (NIFTY 50 and BSE SENSEX), with the first stage estimating the parameters used in the second stage. Using data from 2003 to 2012 and 10 technical analysis indicators as independent variables, the authors reported that the two-stage procedure led to a diminishment of the overall out-of-sample prediction error levels compared to single-step versions of the adopted machine learning models. Among the machine learning models applied individually, ANN and SVR exhibited superior overall performance than Random Forest, while the ANN–SVR hybrid yielded the lowest error metrics between the tested combinations.

Based on a systematic literature review, we found out that, in a similar fashion than the "factor zoo" for asset pricing models described in Section 2.1, the literature also considers a large number of technical

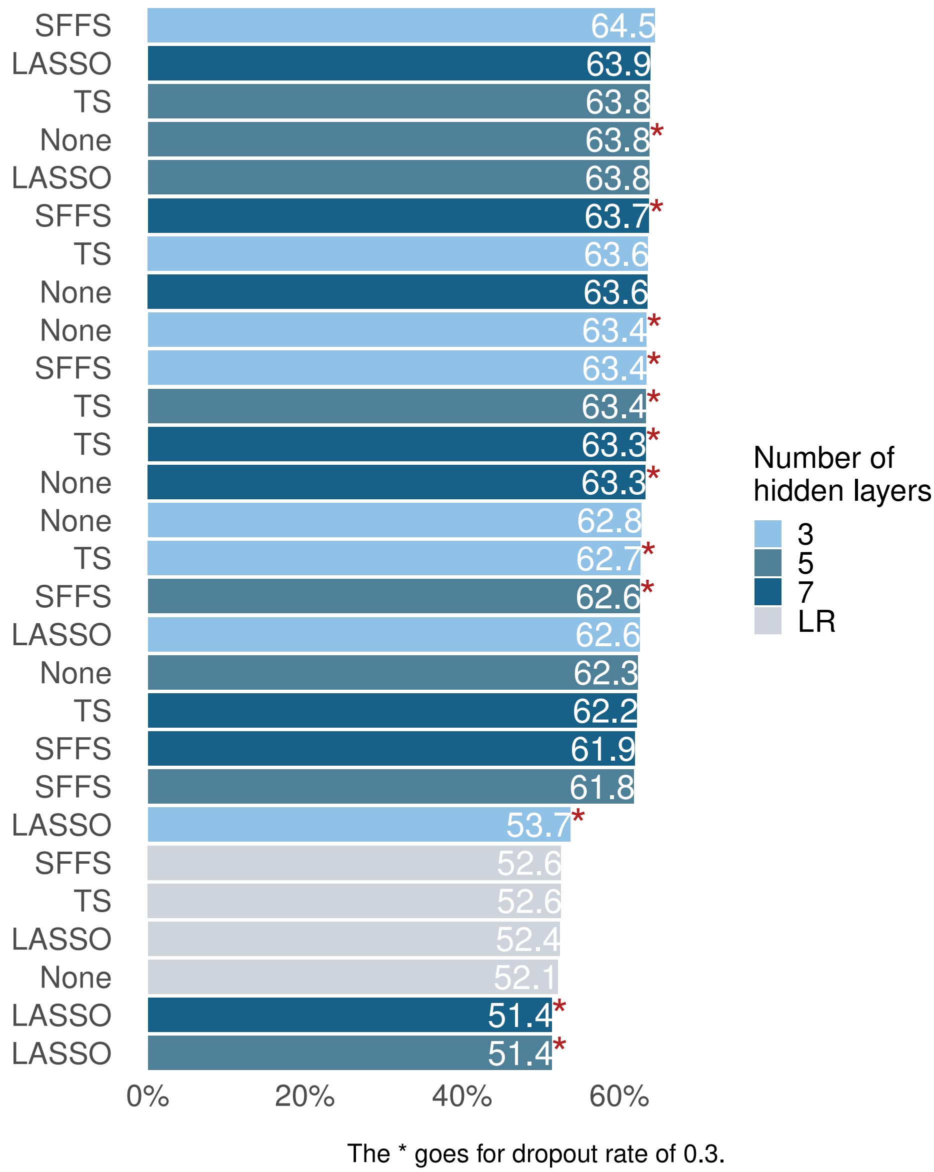

Fig. 5. Out-of-sample accuracy results for assets of FTSE 100 Index - Literature + market.

analysis indicators as predictors to forecast the stock market prices. We searched papers published in high impact journals over the last 20 years (1999–2018) that applied machine learning models to forecast the value or the direction of stock prices of financial market indexes, and we obtained a list of 51 technical analysis indicators used as independent variables in those papers, which is displayed in Table 1. Furthermore, we searched the technical analysis indicators used by four specialized websites that offer financial services and technical analysis softwares (namely: Fidelity Investments, Trading Technologies, StockCharts, and TradingView) to identify variables that are used by market professionals and financial traders. The search resulted in another list, displayed in Table 2, composed by 73 indicators that were not used in any of the academic researchers from Table 1. Therefore, the "factor zoo" of technical analysis indicators has a total size of 124 variables when combining both features used by the literature and the market.

The compilation of Table 2 represents a contribution to the current literature, given that, to the best of our knowledge, no other paper has systematically listed the variables considered in recent scientific works on this research agenda, nor has explored the magnitude of this literature's gap in comparison to indicators applied by the market. For our empirical analysis, feature selection and deep learning model were applied for the whole set of variables (Tables 1 and 2) and the subset of indicators extracted from the literature (Table 1 only) to evaluate the extent of the improvement brought by the indicators used in the market.

3. Method

3.1. Logistic regression, artificial neural networks, and deep neural networks

One of the most well-known classification algorithm is the logistic regression (also known as LR or "logit model"), which is basically a linear regression model for the log-odds of the probability of success of a Bernoulli experiment $p(\boldsymbol{x})$ conditioned to a vector of observed independent variables $\boldsymbol{x} \in \mathbb{R}^k$, such that:

$$log\left(\frac{p_i(\boldsymbol{x})}{1-p_i(\boldsymbol{x})}\right) = w_0 + w_1 x_{1,i} + \cdots + w_k x_{k,i} \quad (1)$$

which can be rewritten as:

$$p_i(\boldsymbol{x}) = \frac{1}{1+e^{-(w_0+w_1 x_{1,i}+\cdots+w_k x_{k,i})}} = \frac{e^{w_0+w_1 x_{1,i}+\cdots+w_k x_{k,i}}}{1+e^{w_0+w_1 x_{1,i}+\cdots+w_k x_{k,i}}} \quad (2)$$

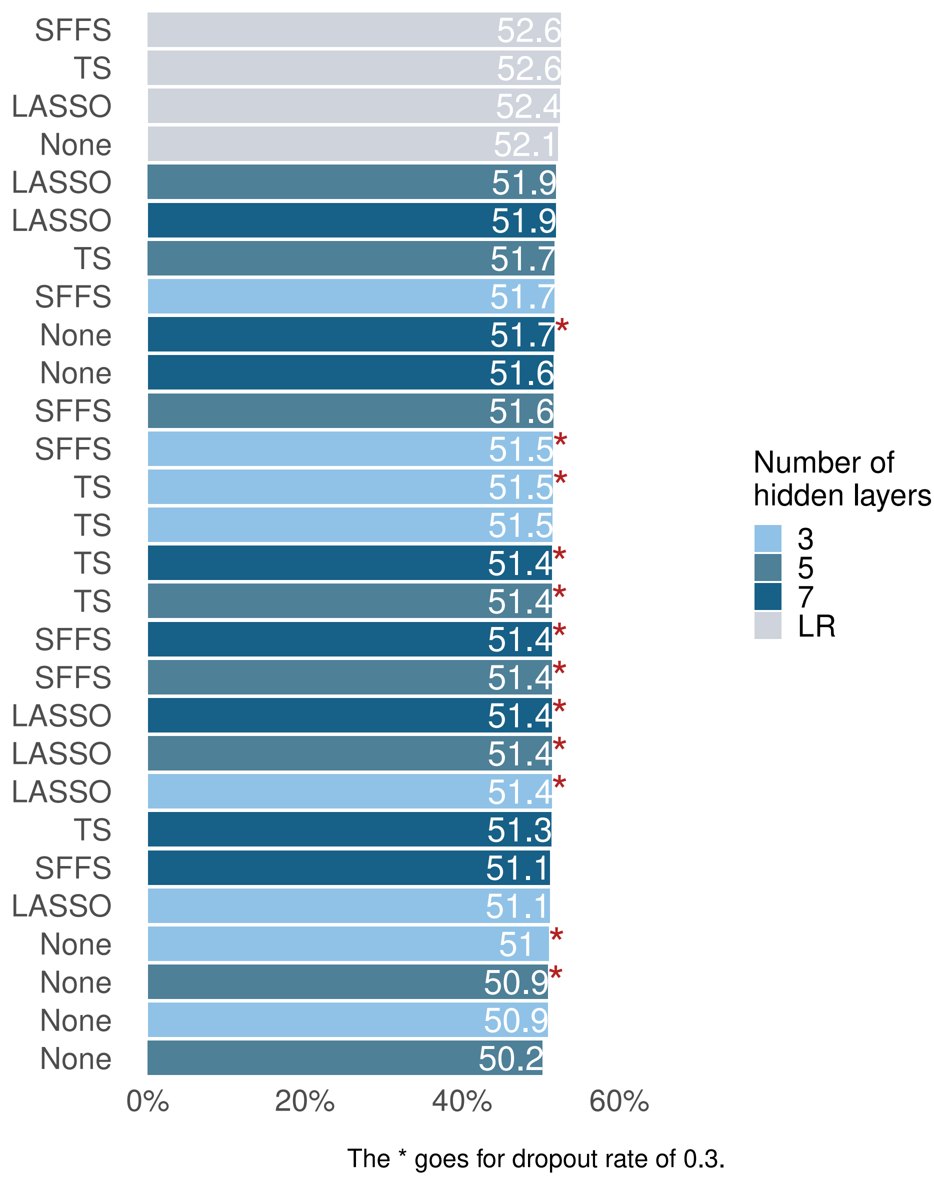

Fig. 6. Out-of-sample accuracy results for assets of FTSE 100 Index - Literature.

where $w_0, \ldots, w_k$ are the parameters associated with each independent variable model and $\sigma(x) = \frac{1}{1+e^{-x}}$ is known as the sigmoid function (or standard logistic function). In summary, the logistic regression is a linear regression model whose output is "squashed" into the range $[0, 1]$ through the sigmoid function. Given that the log-odd can be interpreted as the ratio between the probability of success and the probability of failure of the Bernoulli experiment, usually the cutoff $p_i(\boldsymbol{x}) = 0.5$ is used as a classifying rule for binary dependent variables, with the prediction being "class 1" if $p_i(\boldsymbol{x}) > 0.5$ and "class 0" if otherwise.

While being simple and providing a straightforward interpretation, the logistic regression has, by construction, an assumption of linearity, which makes a major limitation of this model. Over recent years, many other nonparametric models showed better empirical performance in classification tasks, hence becoming increasingly popular among researchers due to their flexibility, notably in financial contexts: as discussed in Hsu, Lessmann, Sung, Ma, and Johnson (2016), machine learning methods were shown to consistently outperform traditional econometrics models in a various range of applications, including problems in finance such as stock market prediction, portfolio analysis, asset pricing, and risk management. One of the most used machine learning methods is Artificial Neural Networks (ANN) and their many extensions, such as deep networks, recurrent networks, and convolutional networks. Their applications range from image processing, text translating to chromosome mapping, and financial forecasting; specifically in finance, references that used ANNs were briefly summarized in Section 2.

While the big variety of ANN extensions differ in functional forms and complexities, in essence, an ANN is a recursive application of linear models and "chunks" of nonlinearity. Algebraically, while a linear regression model can be expressed as $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w}$, with $\boldsymbol{y}$ being a vector of dependent variables, $\boldsymbol{X}$ a matrix of observed independent variables and $\boldsymbol{w}$ a vector of parameters, an ANN can be generally expressed as:

$$\boldsymbol{y} = \boldsymbol{\psi}_{\ell}(\ldots\boldsymbol{\psi}_2(\boldsymbol{\psi}_1(\boldsymbol{X}\boldsymbol{W}_1)\boldsymbol{W}_2)\ldots\boldsymbol{W}_{\ell})\boldsymbol{w}_o \tag{3}$$

where $\boldsymbol{W}_1, \ldots, \boldsymbol{W}_{\ell}$ are the parameters associated with each hidden layer, $\boldsymbol{w}_o$ are the parameters associated with the output layer, $\psi_i(\cdot), i = 1, 2, \ldots, \ell$ are activation functions — where the nonlinearity is introduced — and ℓ is the number of hidden layers. In other words, an ANN is simply a sequence of linear regressions, and the linear regression itself is actually an ANN with only one layer and $\psi(\boldsymbol{x}) = \boldsymbol{x}$, while the logistic regression is also an ANN with one layer and $\psi(\boldsymbol{x})$ equal to

Table 9
Out-of-sample prediction results for assets of CAC 40 Index.

Technical analysis indicators	Feature selection method	Hidden layers	Dropout	Accuracy	Precision	Recall	F-Score
Literature + market (Tables 1 and 2)	None	3	0	0.6174	0.6347	0.5961	0.6148
			0.3	0.6424	0.6576	0.6298	0.6434
		5	0	0.6187	0.6197	0.6617	0.6600
			0.3	0.6295	0.7038	0.4777	0.5691
		7	0	0.6213	0.6215	0.6664	0.6432
			0.3	0.4877	Predicted only drops		
	SFFS	3	0	0.6564	0.6780	0.6270	0.6515
			0.3	0.6406	0.6690	0.5904	0.6273
		5	0	0.6378	0.7135	0.4859	0.5781
			0.3	0.6360	0.6754	0.5572	0.6106
		7	0	0.6452	0.6797	0.5812	0.6266
			0.3	0.4877	Predicted only drops		
	TS	3	0	0.6507	0.6480	0.6966	0.6714
			0.3	0.6376	0.6773	0.5586	0.6122
		5	0	0.6501	0.6598	0.6543	0.6571
			0.3	0.6406	0.6587	0.6192	0.6383
		7	0	0.6459	0.6416	0.6994	0.6693
			0.3	0.4877	Predicted only drops		
	LASSO	3	0	0.6145	0.6518	0.5312	0.5853
			0.3	0.6382	0.6713	0.5753	0.6196
		5	0	0.6225	0.6197	0.6808	0.6488
			0.3	0.6375	0.6936	0.5234	0.5966
		7	0	0.6196	0.6218	0.6567	0.6388
			0.3	0.6378	0.6902	0.5314	0.6005
Literature (Table 1)	None	3	0	0.5010	0.5178	0.3765	0.4360
			0.3	0.5192	0.5218	0.7329	0.6096
		5	0	0.5111	0.5144	0.8079	0.6286
			0.3	0.5169	0.5210	0.7044	0.5989
		7	0	0.5242	0.5238	0.7817	0.6273
			0.3	0.5064	0.5277	0.3451	0.4173
	SFFS	3	0	0.4877	Predicted only drops		
			0.3	0.4877	Predicted only drops		
		5	0	0.4877	Predicted only drops		
			0.3	0.4877	Predicted only drops		
		7	0	0.4877	Predicted only drops		
			0.3	0.4877	Predicted only drops		
	TS	3	0	0.5141	0.5226	0.5923	0.5534
			0.3	0.4998	0.5361	0.1750	0.2639
		5	0	0.4879	0.5002	0.2743	0.3543
			0.3	0.4877	Predicted only drops		
		7	0	0.4968	0.5196	0.2340	0.3227
			0.3	0.4877	Predicted only drops		
	LASSO	3	0	0.5074	0.5105	0.6577	0.5777
			0.3	0.5099	0.5152	0.7284	0.6035
		5	0	0.5134	0.5153	0.8372	0.6380
			0.3	0.5083	0.5131	0.77990	0.6190
		7	0	0.5151	0.5169	0.8136	0.6322
			0.3	0.4877	Predicted only drops		

the logistic function $\sigma(\boldsymbol{x}) = \frac{1}{1+e^{-x}}$, which is one of the most popular activation functions in ANN applications.

Instead of using only one hidden layer, an ANN can be specified with an arbitrary degree of "deepness" by stacking more layers, which allows the algorithm to learn more abstract knowledge representations. For instance, in image recognition, the first layers focus on simpler tasks like identifying the contrast of the pixels to preliminary define the contours of the objects in the image; as the inputs go deeper into the layers, more complex patterns like edges and lines are learned, and in the deepest layers the neurons specialize in identifying actual objects, such as eyes and ears (Goodfellow, Bengio, Courville, & Bengio, 2016). In financial applications, the same reasoning is valid, as discussed by Heaton, Polson, and Witte (2017): by using an ANN with more hidden layers, the algorithm becomes able to learn stylized facts from financial data such as volatility clustering and the leverage effect; in factor models, an ANN generalizes cross-interactions between the factor as a hierarchical nonlinear factor model. Regarding technical analysis indicators, it is expected for ANNs to learn the trading rules that derive from them — for instance, a buy or sell signal arising from the crossing of a short-term and a long-term moving average, as summarized in the MACD indicator. Moreover, a deep ANN can theoretically provide a joint analysis for different indicators that may give away different actions for investors — for example, if some indicators give a buy signal while others give a sell signal, a deep ANN is able to consider more abstract market dynamics that each indicator is modeling and their combined effect on the investor's ultimate decision.

Nonetheless, a deeper network structure, while allowing the algorithm to learn more complex structures, also makes it more prone to overfitting — that is, the ANN may simply "memorize" the in-sample data alongside the noisy information specific to those observations,

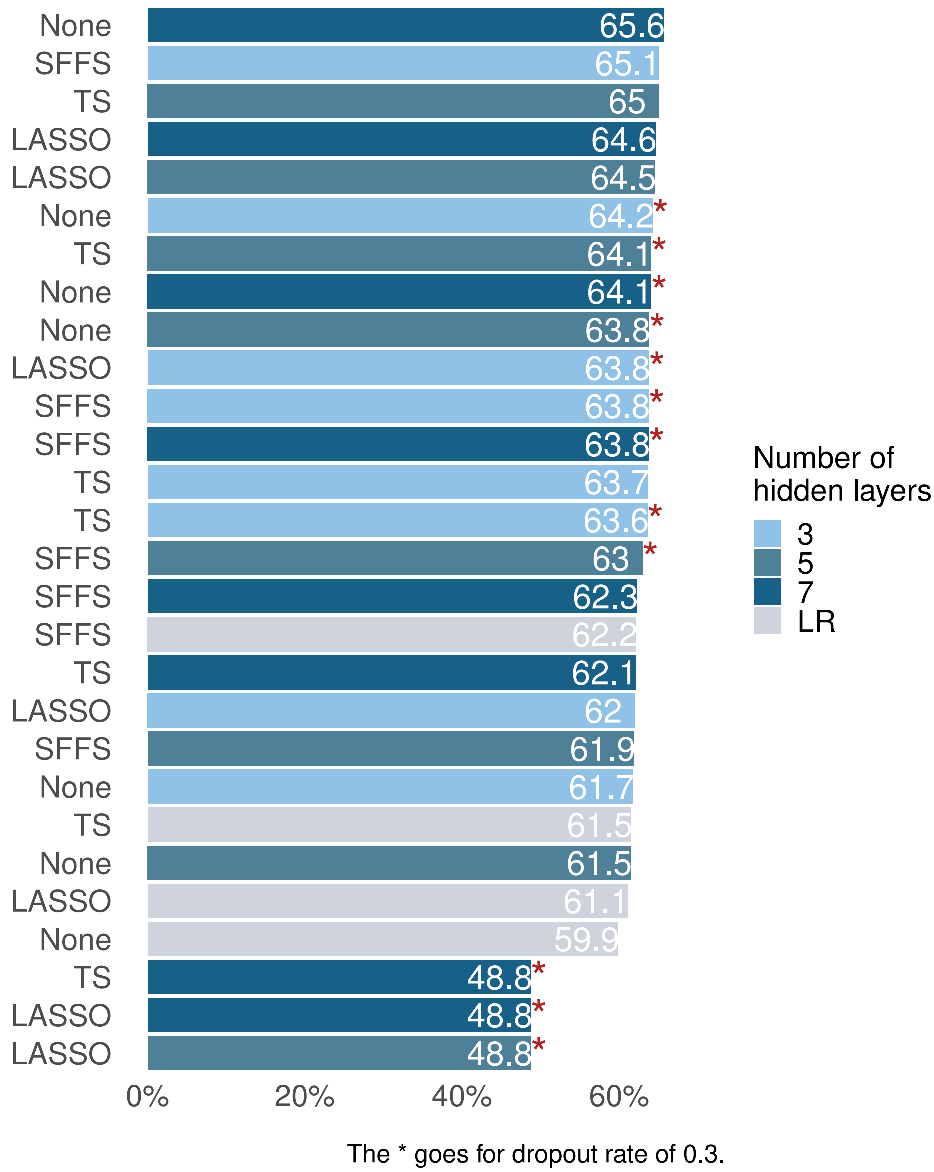

Fig. 7. Out-of-sample accuracy results for assets of CAC 40 Index - Literature + market.

making it worse for generalizations. When dealing with financial data, this issue is particularly relevant, as pointed out in Heaton et al. (2017). In this sense, there is no consensus in the literature of financial applications concerning the effects of introducing additional hidden layers in ANNs on the model's out-of-sample predictive performance. For example, in Nakano et al. (2018)'s experiments on the profitability of trading strategies for bitcoin, deeper ANNs yielded better results, indicating that more levels of interaction between the variables can help to reveal more complex patterns in the data. On the other hand, in Gu et al. (2020)'s application on risk premia of US Stocks assets, neural networks with different numbers of hidden layers were tested, and the deep architecture neural networks showed worse results than shallower networks, also evidencing the possibility of overfitting when dealing with data with high levels of noise. Therefore, in this paper, we tested for ANNs with different numbers of hidden layers and analyzed the effects of previously applying feature selection methods in the original set of technical analysis indicators.

3.2. Regularization and dropout

Dropout, introduced by Srivastava, Hinton, Krizhevsky, Sutskever, and Salakhutdinov (2014), is a regularization method commonly used in neural network applications to avoid overfitting in the training process. In each iteration, instead of computing every possible parameter throughout the network in the backpropagation, each neuron is activated with a probability $1 - p$, with p being the dropout rate. The motivation of doing so is to force the neural network to learn from a broader range of "paths" instead of attributing higher weights to interactions between specific neurons, consequently ignoring other possible interactions. In this sense, dropout can be regarded as an indirect form of feature selection for nonlinear interactions of the original variables, making the ANN converge to a set of weights that are larger for important interactions and closer to zero for irrelevant ones.

In each training epoch, the dropout randomly assigns a percentage of features as zero, such that no further interactions will be derived from the zeroed neurons. The mechanism avoids the excessive exploitation of a previously neuron path, thus decreasing the chance of memorizing a path that conveniently yielded a lower error rate. When one neuron of this path is removed, the network will be forced to consider other potential paths that minimize the in-sample error, thus allowing it to learn other potentially useful patterns for predicting using out-of-sample data.

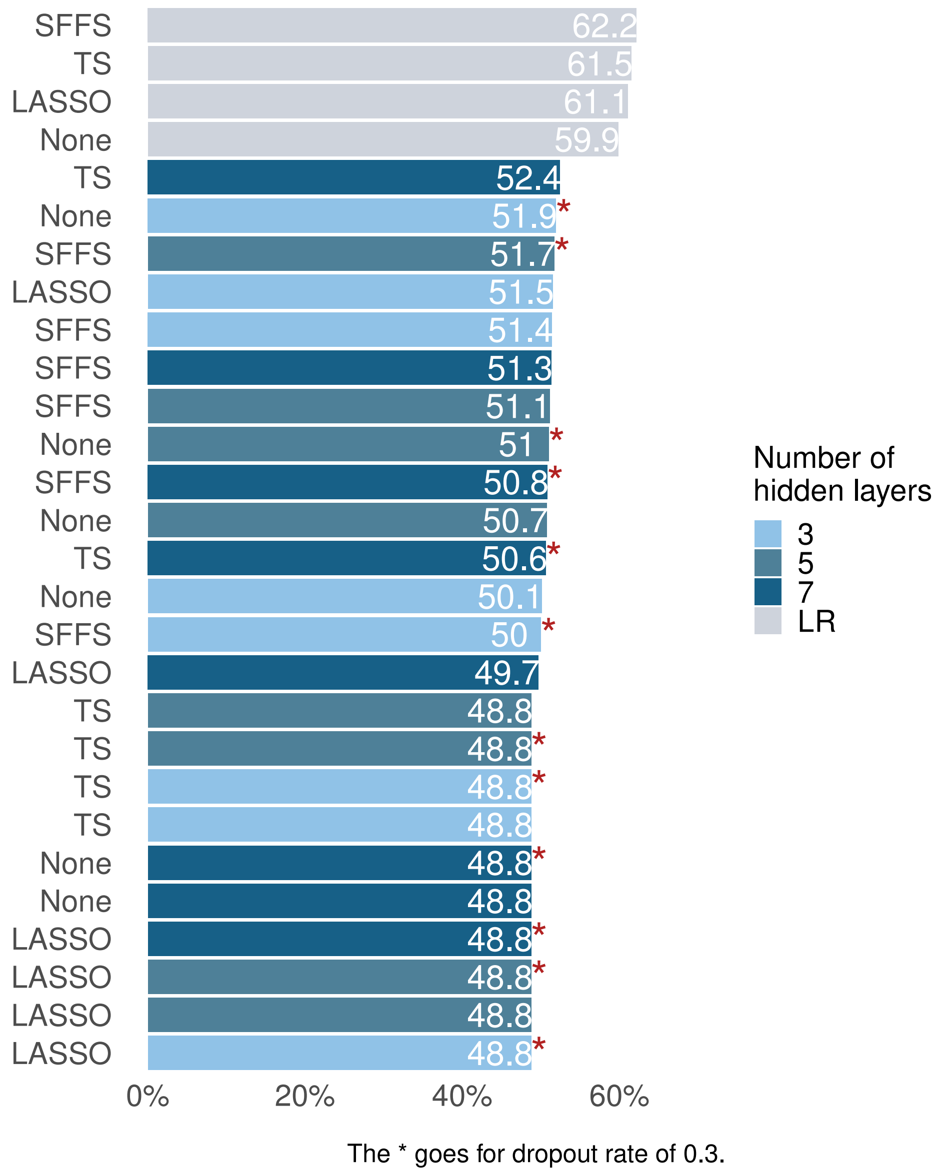

Fig. 8. Out-of-sample accuracy results for assets of CAC 40 Index - Literature.

While dropout is already a well-known procedure in recent machine learning studies, its use as a regularization tool is still scarce in financial applications: while this technique was employed in Heaton et al. (2017) for the construction of "deep portfolios", specifically concerning applications that used technical analysis indicators and ANNs to stock prices prediction, dropout was not applied in any of the papers analyzed in review papers Henrique et al. (2019) and Nazário et al. (2017). Bearing in mind the large number of features present in our application, an additional mechanism to control overfitting and the ANNs' complexity is desirable in terms of prediction effectiveness. Therefore, in this paper, we applied this method and verified whether this technique managed to further improve the predictive performance of both the original feature set and the refined set after applying feature selection.

3.3. Feature selection methods

The survey paper of Chandrashekar and Sahin (2014) classified feature selection methods in three broad categories: filter methods, in which the features are ranked according to their respective conditional dependencies to the class labels; wrapper methods, where an algorithm searches for the feature subset with the best predictive performance in a validation dataset; and embedded methods, in which the feature selection occurs simultaneously with the training process without splitting the dataset. However, the authors pointed out that filter methods like feature ranking by criteria like covariance and mutual information tend to ignore inter-feature dependencies, notably highly nonlinear ones. Furthermore, as shown in the experiments of John, Kohavi, and Pfleger (1994), the subsets yielded from wrapper selection models generated more parsimonious classification decision trees in comparison to filter selection models. Therefore, since we applied ANNs in this paper, a class of models that focuses precisely on the introduction of nonlinearities and the learning of abstract dependency structures between the input features, we did not consider filter feature selection methods and tested only wrapper and embedded methods instead. In this sense, this paper tested three feature selection methods for our empirical experiments, namely: 1) Sequential Forward Floating Selection algorithm (SFFS); 2) Tournament Screening algorithm (TS); and 3) Least Absolute Shrinkage and Selection Operator (LASSO).

The Sequential Forward Floating Selection (SFFS) algorithm, proposed by Pudil, Novovičová, and Kittler (1994), is a sequential selection algorithm that combines a forward wrapper selection method (Sequential Feature Selection — SFS) with a backward one (Sequential

Table 10
Out-of-sample prediction results for assets of DAX-30 Index.

Technical analysis indicators	Feature selection method	Hidden layers	Dropout	Accuracy	Precision	Recall	F-Score
Literature + market (Tables 1 and 2)	None	3	0	0.4997	0.5328	0.4750	0.5023
			0.3	0.5259	0.5329	0.8713	0.6614
		5	0	0.4881	0.5313	0.3115	0.3928
			0.3	0.5298	0.5362	0.8514	0.6580
		7	0	0.5269	0.5326	0.8969	0.6683
			0.3	0.5313	Predicted only rises		
	SFFS	3	0	0.6226	0.6349	0.6816	0.6574
			0.3	0.5370	0.5390	0.8887	0.6710
		5	0	0.6192	0.6201	0.7314	0.6712
			0.3	0.5344	0.5362	0.9167	0.6766
		7	0	0.6164	0.6022	0.8190	0.6941
			0.3	0.5313	Predicted only rises		
	TS	3	0	0.6197	0.6209	0.7300	0.6711
			0.3	0.5419	0.5417	0.8945	0.6748
		5	0	0.6197	0.6209	0.7300	0.6711
			0.3	0.5419	0.5417	0.8945	0.6748
		7	0	0.6282	0.6473	0.6598	0.6535
			0.3	0.5426	0.5426	0.8863	0.6731
	LASSO	3	0	0.4938	0.5375	0.3391	0.4159
			0.3	0.5293	0.5351	0.8684	0.6622
		5	0	0.4920	0.5460	0.2612	0.3534
			0.3	0.5354	0.5381	0.8877	0.6700
		7	0	0.5239	0.5317	0.8713	0.6604
			0.3	0.5452	0.5468	0.8408	0.6627
Literature (Table 1)	None	3	0	0.5370	0.5362	0.9521	0.6860
			0.3	0.5336	0.5341	0.9579	0.6858
		5	0	0.5331	0.5329	0.9816	0.6908
			0.3	0.5359	0.5358	0.9467	0.6843
		7	0	0.5344	0.5335	0.9854	0.6922
			0.3	0.5313	Predicted only rises		
	SFFS	3	0	0.5300	0.5332	0.9264	0.6769
			0.3	0.5323	0.5331	0.9646	0.6867
		5	0	0.5259	0.5330	0.8688	0.6607
			0.3	0.5326	0.5344	0.9332	0.6797
		7	0	0.5275	0.5311	0.9448	0.6800
			0.3	0.5313	Predicted only rises		
	TS	3	0	0.5344	0.5359	0.9221	0.6779
			0.3	0.5329	0.5332	0.9685	0.6878
		5	0	0.5347	0.5364	0.9158	0.6765
			0.3	0.5326	0.5350	0.9187	0.6762
		7	0	0.5365	0.5360	0.9492	0.6851
			0.3	0.5313	Predicted only rises		
	LASSO	3	0	0.5305	0.5340	0.9143	0.6742
			0.3	0.5341	0.5353	0.9332	0.6804
		5	0	0.5318	0.5334	0.9496	0.6831
			0.3	0.5357	0.5367	0.9225	0.6786
		7	0	0.5344	0.5355	0.9332	0.6805
			0.3	0.5313	Predicted only rises		

Backward Selection — SBS). The SFS starts with an empty set of features and adds recursively the variable that gives away the most significant improvement in the classification performance until the refined subset reaches a user-defined size parameter d. SBS works in a similar way, starting from the full set of all variables and removes features that give away the lowest decrease in prediction performance.

As both SFS and SBS are greedy approaches, variable subsets that present a big improvement when jointly considered can be missed by these algorithms. SFFS, on the other hand, adds more flexibility to the basic SFS by adding a step to exclude already included features instead of keeping them permanently. Thus, the SFFS alternates a forward step from SFS with a backward step in which previously added features are excluded while a new best feature subset is obtained through the exclusion. The superiority of SFFS over SFS is discussed in Reunanen (2003), who compared both frameworks for datasets with classification tasks from various knowledge fields collected from the UCI Machine Learning Repository. In this paper, we performed the SFS and SBS routines using logistic regressions as fitting models and the Akaike information criterion (Akaike, 1974) to evaluate the features.

Concerning stock direction prediction, Lee (2009) applied the SFFS algorithm to perform wrapper feature selection on a set of 29 variables composed of financial indexes, currency quotations, and commodities futures to predict the NASDAQ index direction; the selected feature subset was then fitted into a classification SVM and a standard ANN. Moreover, the study compared the effectiveness of other filter feature selection methods based on three criteria (information gain, symmetrical uncertainty, and correlation), and SFFS was the algorithm that yielded the greatest improvements over its full feature set counterpart for both models.

The steps of the SFFS algorithm can be summarized in algorithm 1 below:

Algorithm 1 Sequential Forward Floating Selection

procedure SFFS(Feature set $\mathcal{F}$, number of intended relevant features K)
  Initialize $D = 0$, $p_{best} = 0$, $\mathcal{F}^* = \{\}$
  while $D < K$ **do**
    Execute 1-step SFS based on information criterion
      $\triangleright$ select best feature $f_{forward}$ from $\mathcal{F}$
    Store SFS classification accuracy $p_{forward}$
    Add $f_{forward}$ to $\mathcal{F}^*$
    Update $D = D + 1$
    if $p_{forward} > p_{best}$ **then**
      $p_{best} = p_{forward}$
    $p_{backward} = \infty$
    while $p_{backward} > p_{best}$ **do**
      Execute 1-step SBS based on information criterion
        $\triangleright$ remove worst feature $f_{backward}$ from $\mathcal{F}$
      Store SBS classification accuracy $p_{backward}$
      Remove $f_{backward}$ from $\mathcal{F}^*$
      Update $D = D - 1$
      if $p_{backward} > p_{best}$ **then**
        $p_{best} = p_{backward}$
      else
        Add $f_{backward}$ back to $\mathcal{F}^*$
        Update $D = D + 1$
  return Set of relevant features $\mathcal{F}^*$

On the other hand, the tournament screening (TS) algorithm, proposed by Chen and Chen (2009), is a heuristic search algorithm that generates candidate features based on the best features of mutually exclusive subsets. The variables are recursively split into smaller groups, and a "tournament" takes place inside each subset, which the features that "survive" the contests being classified as the best ones. Therefore, the main idea of tournament screening is analogous to a genetic algorithm, in which the "strongest" offsprings crossover amongst themselves while the weakest are gradually eliminated (Chandrashekar & Sahin, 2014).

In the TS algorithm, the set of original variables are randomly subdivided into disjoint subsets, and inside each subset, a verification model is fitted, and the variable with the least contribution or statistical significance is excluded from the group. The remaining variables from all subsets are aggregated and attributed again to new mutually exclusive subsets, repeating the process of recursive elimination until the number of remaining is reduced to a user-specified number. As pointed out by Abdulla and Khasawneh (2020) and Ferreira Filho et al. (2019), the tournament screening is a good alternative for parametric models of high dimensionality, in particular when the number of features is so high to the point where the number of degrees of freedom is not sufficient for the joint estimation of all parameters specified in the model. In those cases, the application of tournament screening allows the parameters to be estimated within each subgroup, assuming that the null parameters will ultimately be estimated as non-influential values inside those subgroups.

For the selection of the most significant features, we applied ANOVA (analysis of variance) on each subset at each round, testing the relative impact of removing each technical analysis indicator against the model with all indicators from the subset, using Wald tests, which is asymptotically equivalent to a likelihood-ratio test (Engle, 1984), and using logistic regression to fit each model. In this sense, the p-values associated with each feature's coefficients were ranked to provide a measure of significance, and the feature with the lowest p-value was considered the least significant in terms of explanatory power, thus being removed from the feature subset. The steps of the TS algorithm are summarized in algorithm 2 below:

Algorithm 2 Tournament Screening

procedure TS(Feature set $\mathcal{F}$, number of intended relevant features K, number of subsets P)
  Initialize $\mathcal{F}^* = \mathcal{F}$
  Split $\mathcal{F}^*$ into P mutually exclusive subsets $\mathcal{F}^{(1)}, \ldots, \mathcal{F}^{(P)}$
  for $i \in \{1, \ldots, P\}$ **do**
    while $length(\mathcal{F}^{(i)}) > \frac{length(\mathcal{F}^*)}{P}$ **do**
      Fit base classifier using features from subset $\mathcal{F}^{(i)}$
      Rank features significance based on ANOVA tests
      Update subset $\mathcal{F}^{(i)}$ removing least significant feature
  Update $\mathcal{F}^* = \bigcup_{i=1}^{P} \mathcal{F}^{(i)}$
  return Set of relevant features $\mathcal{F}^*$

Finally, the Least Absolute Shrinkage and Selection Operator (LASSO) (Tibshirani, 1996) is a regularization method in which a penalty term is added to the likelihood function optimized in linear regression. The unconstrained OLS estimates $\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$ can be vulnerable to high variance, which in turn can affect inference negatively. Thus, a penalty term for the magnitude of the coefficients can control the variance. Similarly to ridge regression, in which the penalty term is the ℓ_2 norm for the $\boldsymbol{\beta}$ parameters, in LASSO the penalty is the ℓ_1 norm. The main difference is that the LASSO can yield a set of sparse solutions for the betas, making LASSO an embedded feature selection method, as the algorithm training process is done simultaneously as the feature selection.

The coefficients of the LASSO regression are the solutions of the following constrained optimization problem:

$$\beta(\lambda) = \underset{\beta}{\operatorname{argmin}} \left\{ \frac{1}{N}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^{\mathrm{T}}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) \right\}, \text{ subject to } \sum_{j=1}^{p} |\beta_j| < t \tag{4}$$

which is equivalent to:

$$\beta(\lambda) = \underset{\beta}{\operatorname{argmin}} \left\{ \frac{1}{N}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^{\mathrm{T}}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) + \lambda\|\boldsymbol{\beta}\|_1 \right\} \tag{5}$$

where λ is a free regularization parameter that controls the degree of shrinkage of the betas. Therefore, sufficiently large values for λ will effectively force some betas to be zero, producing a sparse solution for the LASSO estimator. In general, the optimal λ that minimizes the out-of-sample error is found by manually tuning through K-fold cross-validation.

For classification problems, the ℓ_1-regularization is analogously introduced to the likelihood function optimized in logistic regression, which leads to the LASSO logistic regression, whose coefficients are obtained solving the following optimization problem:

$$\hat{\beta(\lambda)} = \underset{\beta}{\operatorname{argmin}} \left\{ \frac{1}{N} \sum_{i=1}^{N} \rho_{(\beta)}(X_i, Y_i) + \lambda\|\boldsymbol{\beta}\|_1 \right\} \tag{6}$$

where $\rho_{(\beta)}(X_i, Y_i) = -y\left\{\sum_{j=0}^{k} \beta_j x^{(j)} + log\left[1 + exp\left(\sum_{j=0}^{k} \beta_j x^{(j)}\right)\right]\right\}$ is the likelihood function optimized to obtain the beta coefficients in Eq. (2).

3.4. Data and empirical analysis

We collected daily data between January 1st, 2008 and March 1st, 2019 from firms that composed financial market indexes from seven markets, namely: United States (S&P 100 Index), United Kingdom (FTSE 100 Index), France (CAC 40 Index), Germany (DAX-30 Index),

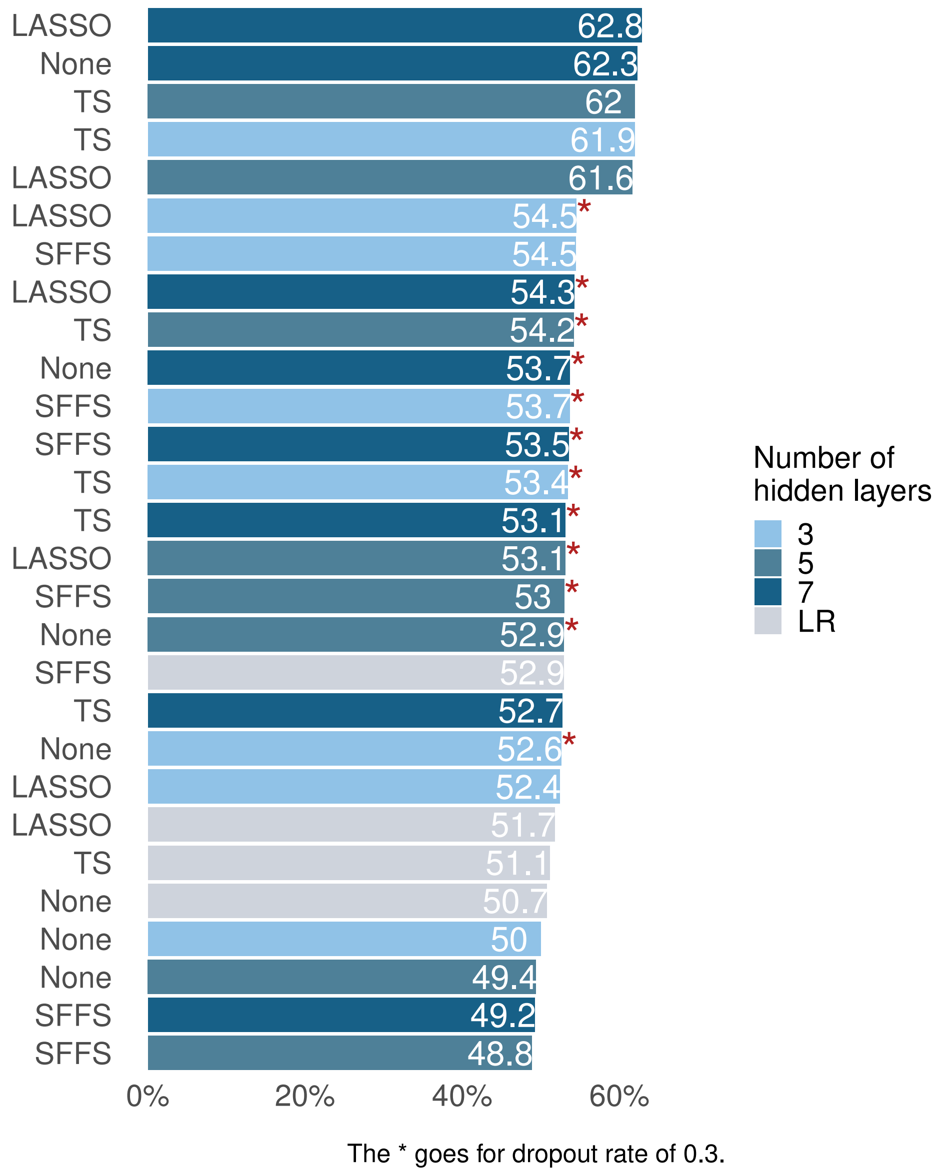

Fig. 9. Out-of-sample accuracy results for assets of DAX-30 Index - Literature + market.

Japan (Top 50 assets from NIKKEI 225 Index), China (Top 50 assets from SSE 180 Index) and Brazil (Bovespa Index). The independent variables are the technical analysis indicators listed on Tables 1 and 2 for period t, and the dependent variable is the price direction movement between periods t and $t+1$. As a pre-processing step, we first split the collected datasets into two sequential and mutually exclusive subsets: the first one being composed by observations between January 1st, 2008 and December 31th, 2010, which was used to apply the feature selection methods; and the second one with the observations between January 1st, 2011 to March 1st, 2019, which was used to train the deep neural network models. For both subsets, we used a training–testing proportion of 75% to 25%.

Using data from the first subset and a training–testing proportion of 75% to 25%, we applied the feature selection methods detailed in Section 3.3 for each of the seven analyzed markets. For the wrapper feature selection methods (SFFS and TS), we recursively fitted logistic regressions (LR) with the candidate features and perform additions/removals based on its performance on the first testing set, using the accuracy as the evaluation metric. For the LASSO, we tuned the λ hyperparameter using grid-search and 10-fold cross-validation, taking the value for λ that minimizes the classification error (measured by the binomial deviance) as the optimal hyperparameter, using this value to fit the model in the validation set, then taking the variables for which the LASSO estimates for this model are non-zero as the refined variable subset. We applied the feature selection methods on the list of technical indicators from Table 1, which are the variables already used by the scientific community, and also for the indicators listed in both Tables 1 and 2, to assess the gains in predictive power and profitability derived from the addition of variables used by market professionals but not yet considered by high-impact papers.

After reaching a refined variable subset using SFFS, TS, and LASSO for all seven markets, all subsets were then used to fit the Deep Neural Networks using the data from the second subset (January 1st, 2011 to March 1st, 2019), also using a training–testing proportion of 75% to 25%. Different numbers of hidden layers and dropout rates were tested to analyze further the effects of deep network architectures and degree of regularization. The optimal weights obtained in this step were finally applied in the second testing set to verify the models' out-of-sample predictive performance, which were measured not only by accuracy but also by precision, recall, and F-Score. We tested neural networks with 3,

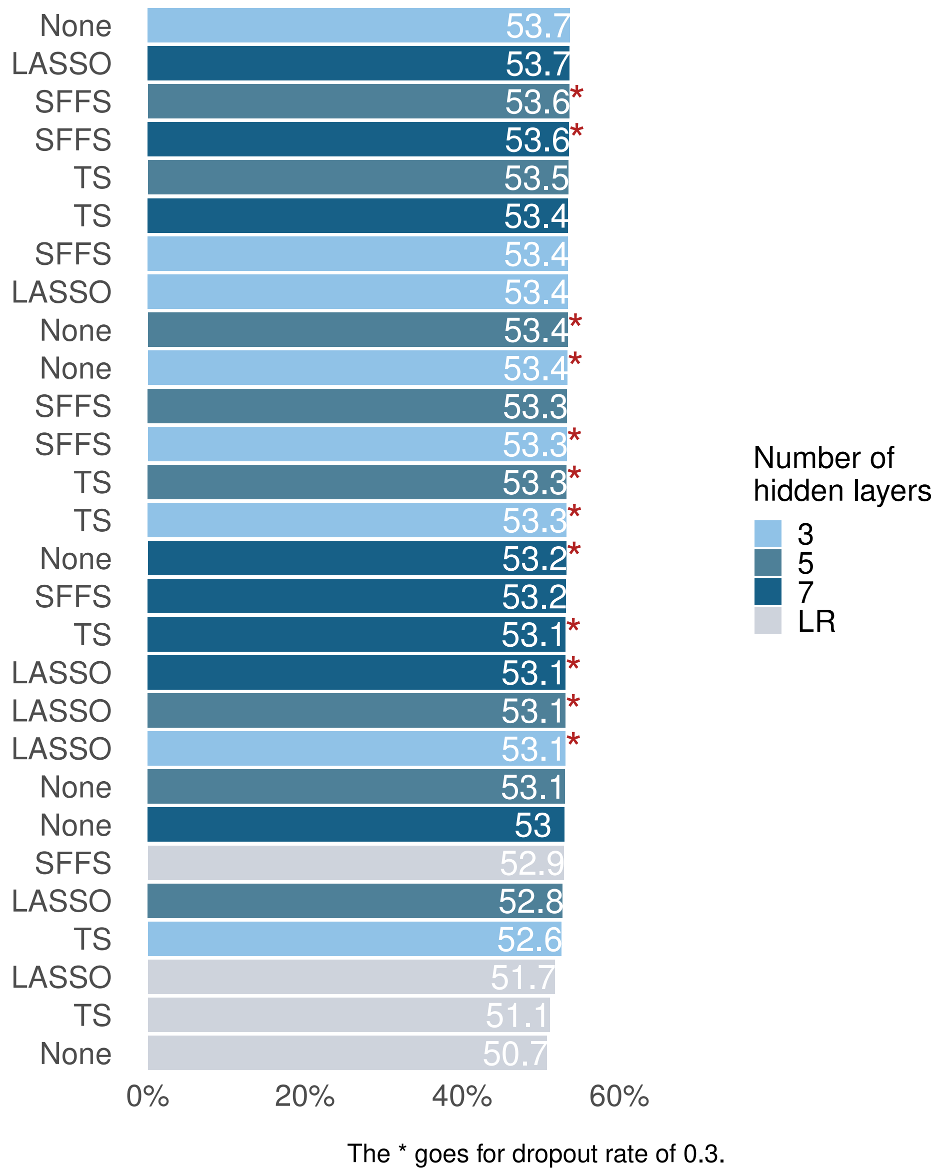

Fig. 10. Out-of-sample accuracy results for assets of DAX-30 Index - Literature.

5, and 7 hidden layers, with the Sigmoid function $\sigma(\cdot)$ as the activation function for all cases; we tested two parameter values for Dropout (0 and 0.3). For the training of the networks, we used the Adam optimization algorithm (Kingma & Ba, 2014), 400 training epochs, and mini-batches of size 128 for all cases. All tests were replicated for both Tables 1 and 2 (Literature + Market) and only for Table 1 (Literature only).

Given the existence of a number of stylized facts for financial time-series, different settings of regularization hyperparameters can generate significantly different predictive performance for stock price movements and trading profitability (Shynkevich, McGinnity, Coleman, Belatreche, & Li, 2017). Particularly for experiments using machine learning models, due to the larger flexibility of these models to approximate complex patterns for high-dimensional data, the sensibility of the out-of-sample predictions to small variations in their hyperparameters also tend to be high, as pointed out by works like Claesen and De Moor (2015). In applications of neural networks in financial predicting tasks, the importance of testing different settings for parameters like the number of hidden layers ("deep" or "shallow" architecture) and regularization factors, such as the dropout rate, are discussed in papers like Ravi, Pradeepkumar, and Deb (2017), which proposed a hybrid model combining chaos theory, multi-layer perceptrons and multi-objective evolutionary algorithms to predict financial time-series; Chatzis, Siakoulis, Petropoulos, Stavroulakis, and Vlachogiannakis (2018), which analyzed the propagation of crash events across international financial markets using a wide class of machine learning models, with the deep neural networks contributing with significant boosts in classification accuracy; and Nakano et al. (2018), which tested neural networks with different depths to generate trading signals for high-frequency Bitcoin data, comparing the resultant profitability with a buy-and-hold strategy.

Concerning evaluation metrics, as shown in Henrique et al. (2019), the vast majority of studies that applied machine learning methods in stock price direction prediction uses accuracy for performance measurement; however, this metric does not take into account the proportion between true positives and false positives (Type I Error) nor the proportion between true positives and false negatives (Type II Error), making the accuracy rate potentially misleading, especially when the classes are unbalanced.

Table 11
Out-of-sample prediction results for top 50 assets of NIKKEI 225 Index.

Technical analysis indicators	Feature selection method	Hidden layers	Dropout	Accuracy	Precision	Recall	F-Score
Literature + market (Tables 1 and 2)	None	3	0	0.6254	0.6014	0.6219	0.6115
			0.3	0.6346	0.6057	0.6566	0.63020
		5	0	0.6245	0.5919	0.6694	0.6283
			0.3	0.6333	0.6015	0.6710	0.6343
		7	0	0.6283	0.6074	0.6103	0.6088
			0.3	0.6246	0.5824	0.7354	0.6500
	SFFS	3	0	0.6432	0.6187	0.6444	0.6331
			0.3	0.6389	0.6154	0.6347	0.6249
		5	0	0.6418	0.6249	0.6112	0.6180
			0.3	0.6391	0.6200	0.6167	0.6183
		7	0	0.6392	0.6492	0.5198	0.5773
			0.3	0.5259	Predicted only drops		
	TS	3	0	0.6430	0.6250	0.6168	0.6209
			0.3	0.6380	0.6261	0.5863	0.6056
		5	0	0.6399	0.6293	0.5848	0.6062
			0.3	0.6329	0.6089	0.6308	0.6197
		7	0	0.6414	0.6414	0.5522	0.5935
			0.3	0.5259	Predicted only drops		
	LASSO	3	0	0.6295	0.6124	0.5946	0.6034
			0.3	0.6382	0.6150	0.6327	0.6238
		5	0	0.6293	0.6030	0.6381	0.6200
			0.3	0.6271	0.5897	0.7007	0.6404
		7	0	0.6290	0.5998	0.6529	0.6252
			0.3	0.6241	0.5808	0.7438	0.6523
Literature (Table 1)	None	3	0	0.5142	0.4898	0.5971	0.5382
			0.3	0.5101	0.4853	0.5518	0.5164
		5	0	0.5227	0.4959	0.4134	0.4509
			0.3	0.4990	0.4818	0.7556	0.5884
		7	0	0.5186	0.4910	0.4215	0.4536
			0.3	0.5259	Predicted only drops		
	SFFS	3	0	0.5219	0.4944	0.3817	0.4308
			0.3	0.5053	0.4817	0.5744	0.5240
		5	0	0.5213	0.4945	0.4417	0.4668
			0.3	0.5166	0.4871	0.3726	0.4222
		7	0	0.5226	0.4947	0.3322	0.3975
			0.3	0.5259	Predicted only drops		
	TS	3	0	0.5043	0.4825	0.6301	0.5465
			0.3	0.4906	0.4788	0.8435	0.6108
		5	0	0.5165	0.4871	0.3789	0.4262
			0.3	0.5156	0.4876	0.4320	0.4581
		7	0	0.5106	0.4834	0.4712	0.4773
			0.3	0.5259	Predicted only drops		
	LASSO	3	0	0.5184	0.4911	0.4436	0.4662
			0.3	0.4988	0.4810	0.7278	0.5792
		5	0	0.5319	0.5091	0.3475	0.4131
			0.3	0.5010	0.4818	0.6995	0.5706
		7	0	0.5229	0.4937	0.2555	0.3368
			0.3	0.5259	Predicted only drops		

Accuracy, precision, recall, and F-Score are given, respectively, by:

$$\begin{aligned} \text{Accuracy} &= \frac{TP+TN}{TP+FP+TN+FN} \\ \text{Precision} &= \frac{TP}{TP+FP} \\ \text{Recall} &= \frac{TP}{TP+FN} \\ \text{F}-\text{Score} &= \left(\frac{\frac{1}{\text{Precision}}+\frac{1}{\text{Recall}}}{2}\right)^{-1} \end{aligned} \tag{7}$$

where TP is the number of true positives, FP is the number of false positives, TN is the number of true negatives, and FN is the number of false negatives, with the positive class being associated with an increase in the stock's price between t and $t-1$ and the negative class being assigned upon a price decrease.

Precision is a metric that penalizes the Type I Error — in this case, stocks whose prices dropped when a rise was predicted; while recall yields lower values with the presence of the Type II Error — stocks that gained value but were classified as not worth buying. Therefore, the precision can be an indicator of potential interest for investors with a high degree of risk aversion, which prioritizes avoiding bad investment choices; on the other hand, the recall gives an indication of how many potentially profitable investment opportunities are being missed, which is proportionally more costly for investors with a higher appetite for risk. Finally, the F-Score – which is the harmonic mean of precision and recall – gives a conservative middle-ground between the two types

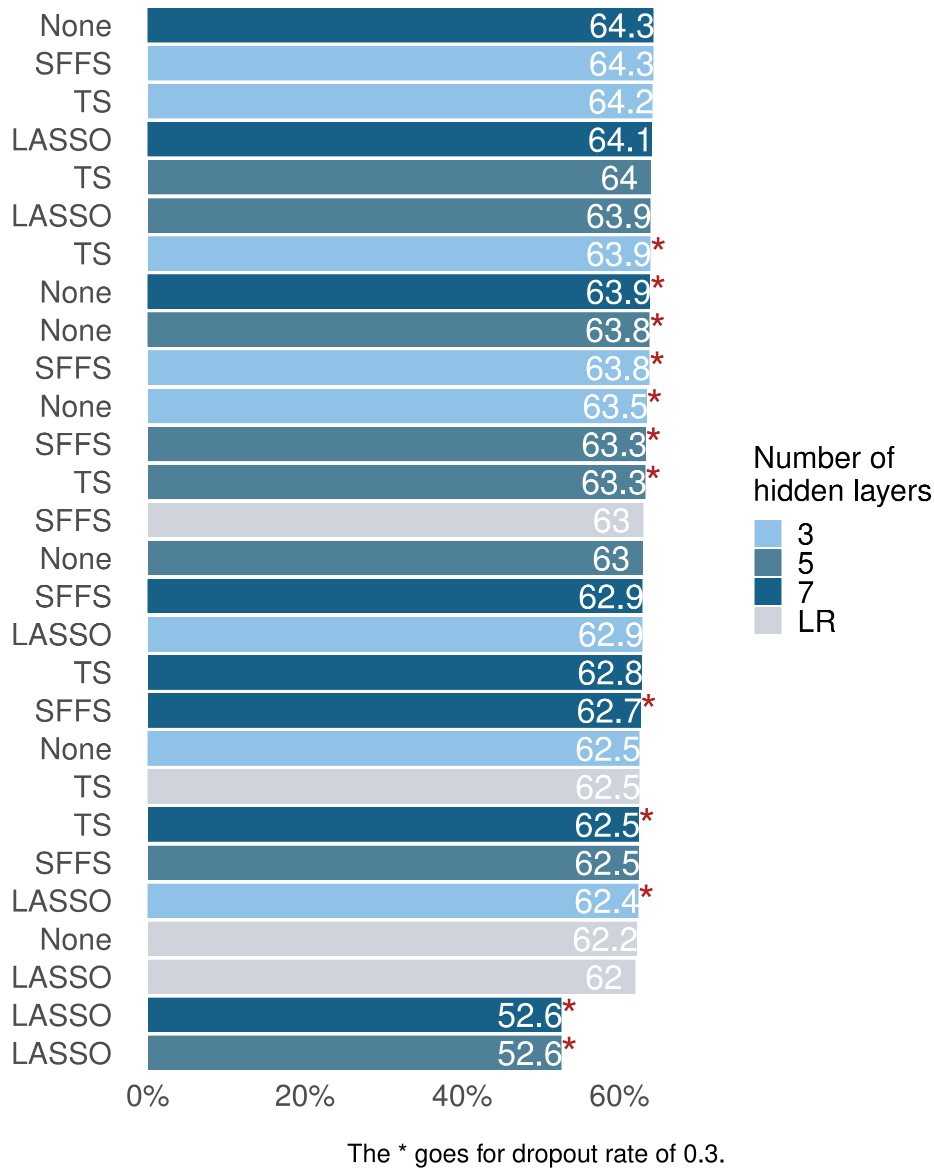

Fig. 11. Out-of-sample accuracy results for assets of NIKKEI 225 Index - Literature + market.

of error, in the sense of yielding a high value only if both precision and recall are high, being sensible to low values from both indicators.

In a scenario where all predictions are correct, all four metrics would exhibit a perfect score of 1; however, in a realistic mixed scenario between misclassifications from both classes (*i.e.*, both uptrends predicted as price drops and downtrends predicted as price rises), the traditionally more used accuracy rate can be misleading, especially if the predictions are heavily unbalanced (concentrated in one of the two classes). In those cases, also observing the precision and the recall can reveal more details about the real quality of the predictions, as well as providing a quick overview of the model's propensity to Type I or Type II Errors. Finally, the F-Score provides a practical way to see the average consistency of the model to both error types, which in this application would represent inadequate resource allocations, both when buying an overly expansive asset and when selling a holding asset for a price too small.

4. Results and discussion

4.1. Feature selection of technical analysis indicators

Concerning the "factor zoo" of technical analysis indicators described in Tables 1 and 2, the first step was to apply feature selection methods detailed in Section 3.3 (SFFS, TS, and LASSO) for the first set of training–testing periods (using data between 2008 and 2010). In a scenario where all technical analysis indicators are equally relevant in terms of predictive power, one could expect that, on average, all columns were picked a similar number of times, such that, conversely, a non-uniform distribution on the incidence of specific indicators being more frequently chosen can be an indication of importance. In this sense, the number of time that each indicator – from both Literature researches and services used by investors to operate in the Markets – was picked by any of the three feature selection methods was aggregated across all seven analyzed financial markets, and the distribution of times chosen and its histogram are displayed, respectively, in Table 3 and Fig. 1.

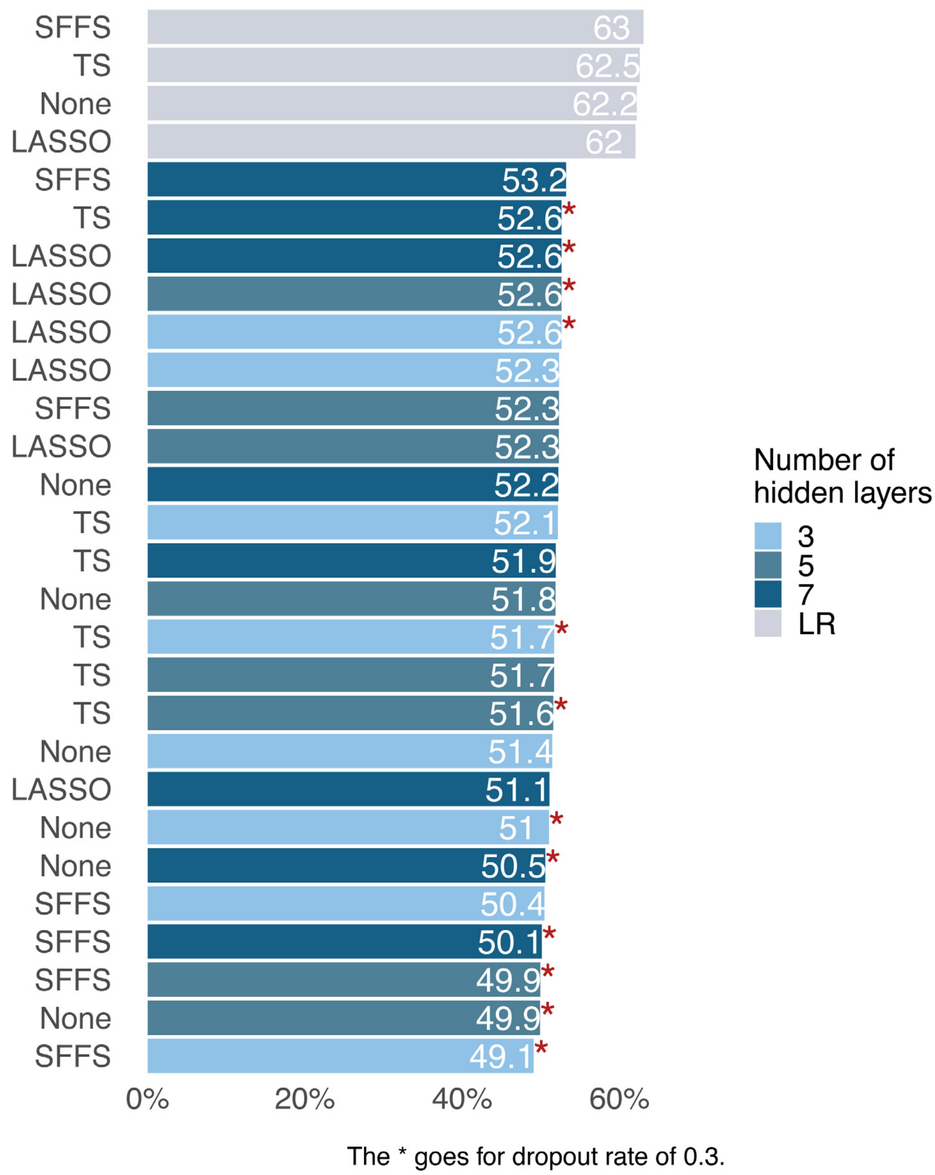

Fig. 12. Out-of-sample accuracy results for assets of NIKKEI 225 Index - Literature.

As seen in Table 3, the empirical distribution presented a fairly asymmetrical behavior, with indicators like DPO (Detrended Price Oscillator), HULL (Hull Moving Average), and MFM (Money Flow Multiplier) being chosen 16 times or more, while indicators widely used in scientific researches like SMA (Simple Moving Average) and WMA (Weighted Moving Average) were picked in a very small number of occasions. Analogously, Fig. 1 shows that out of the 68 technical analysis indicators that were chosen more times than the average of all 124 columns in Tables 1 and 2, 38 belong to the "market side", whilst only 30 were already being considered by academic papers. Given that many technical analysis indicators bear similar ways of calculation, formulas that combined more sources of information seemed to have been prioritized over "simpler" indicators — for instance, the Hull Moving Average is a combination of Weighted Moving Averages, and the feature selection methods, by identifying this combination as informative, probably interpreted the simpler WMA as a redundant source of information. Similar results were found using only the indicators from Table 1, as shown below in Table 4 and Fig. 2.

Indeed, out of the 51 technical analysis indicators used in recent scientific articles, 29 were picked more than the average, also exhibiting an asymmetrical behavior in which some columns like ADO (Accumulation/Distribution Oscillator), BIAS (Bias), DIU (Directional Indicator Up) and MACD (Moving Average convergence–divergence) were picked a high amount of times, while "simpler" indicators like EMA (Exponential Moving Average) and OPEN (Opening price of the day) were less picked. Just like the observed pattern for "Literature + Market", combinations of simpler indicators were identified as "informative", while the constitutes of those indicators were regarded as "redundant" – indeed, BIAS is a combination of CLOSE (Closing price) and SMA, and MACD is a difference of EMAs, all of those were chosen as separate features a small number of times in comparison (see Figs. 3 and 4).

4.2. Predictive performance, profitability of strategies and transaction costs

As shown in Tables 5 to 13, the predictive performance for all seven markets and all 48 combinations of hyperparameters in each market (Literature + Market/Only literature, technical analysis indicators chosen by feature selection, number of hidden layers, and dropout rate) were basically around two key values: the accuracy of all cases

Table 12
Out-of-sample prediction results for top 50 assets of SSE 180 Index.

Technical analysis indicators	Feature selection method	Hidden layers	Dropout	Accuracy	Precision	Recall	F-Score
Literature + market (Tables 1 and 2)	None	3	0	0.6165	0.5804	0.6657	0.6201
			0.3	0.6328	0.6263	0.5435	0.5819
		5	0	0.6291	0.6115	0.5793	0.5950
			0.3	0.6388	0.6337	0.5493	0.5885
		7	0	0.6212	0.5921	0.6252	0.6082
			0.3	0.6377	0.6354	0.5386	0.5831
	SFFS	3	0	0.6513	0.6331	0.6146	0.6237
			0.3	0.6412	0.6541	0.5028	0.5686
		5	0	0.6416	0.6083	0.6678	0.6366
			0.3	0.6391	0.6604	0.4787	0.5551
		7	0	0.6505	0.6292	0.6252	0.6272
			0.3	0.6377	0.6690	0.4545	0.5413
	TS	3	0	0.6492	0.6396	0.5818	0.6093
			0.3	0.6449	0.6375	0.5674	0.6004
		5	0	0.6501	0.6414	0.5803	0.6093
			0.3	0.6436	0.6422	0.5467	0.5906
		7	0	0.6466	0.6379	0.5746	0.6046
			0.3	0.6437	0.6292	0.5898	0.6089
	LASSO	3	0	0.6347	0.6288	0.5443	0.5835
			0.3	0.6379	0.6544	0.4876	0.5588
		5	0	0.6152	0.5909	0.5904	0.5906
			0.3	0.6400	0.6608	0.4818	0.5573
		7	0	0.6164	0.5869	0.6220	0.6040
			0.3	0.6338	0.6751	0.4267	0.5229
Literature (Table 1)	None	3	0	0.5155	0.4854	0.5067	0.4958
			0.3	0.5262	0.4918	0.2246	0.3083
		5	0	0.5253	0.4885	0.1996	0.2834
			0.3	0.5280	0.4966	0.27220	0.3517
		7	0	0.5266	0.4909	0.1781	0.2613
			0.3	0.5269	0.4997	0.3540	0.4144
	SFFS	3	0	0.5342	0.5150	0.1616	0.2460
			0.3	0.5352	0.5152	0.1970	0.2850
		5	0	0.5310	0.5022	0.2893	0.3671
			0.3	0.5341	0.5089	0.2643	0.3480
		7	0	0.5372	0.5118	0.3428	0.4106
			0.3	0.5297	Predicted only drops		
	TS	3	0	0.5341	0.5116	0.2043	0.2920
			0.3	0.5343	0.5100	0.2480	0.3337
		5	0	0.5306	0.5031	0.1391	0.2180
			0.3	0.5349	0.5101	0.2747	0.3571
		7	0	0.5323	0.5083	0.1683	0.2529
			0.3	0.5365	0.5128	0.2882	0.3690
	LASSO	3	0	0.5208	0.4876	0.3766	0.4250
			0.3	0.5256	0.4879	0.1767	0.2595
		5	0	0.5235	0.4807	0.1655	0.2463
			0.3	0.5269	0.4928	0.2095	0.2940
		7	0	0.5222	0.4798	0.1909	0.2731
			0.3	0.53016	0.5006	0.3468	0.4098

was concentrated around 50% and 65%. The first value is consistent with the scenario postulated by the Efficient Markets Hypothesis in its weak form, which implies that no strategy can systematically beat the Random Walk simply using past data and generate abnormal gains over the market; indeed, a fairly large proportion of the results estimated in this study converged to this state for different settings of neural network architectures and regularization parameters, notably for the set of features used only by the literature (Table 1). Tables 8 to 13 are displayed in Appendix A to facilitate the reading of this paper.

However, on the other hand, parallel to the theoretically intuitive accuracy of 50%, many cases converged to a kind of "strange attractor" of 65% of accuracy, which is, in turn, a measure that argues favorably towards the existence of profit margins above the market level. For all cases in which the accuracy rate did not lie at the surroundings of 50%, they converged systematically to 65%; computational experiments made using more training epochs showed that those cases do indeed reach a species of "stationary state" in 65%. This pattern appeared in all analyzed markets (except for the German one) and for both information sources (technical analysis indicators from Literature + Market or only from literature). Those two scenarios were observed for all feature selection methods (in the "None" case, all columns were used for the training procedures).

Moreover, the emergence of the 65% accuracy value revealed some patterns amongst its occurrences: firstly, it occurred much more frequently when using technical analysis indicators used by market operators listed in Table 2 in addition to those used by the literature, suggesting the predictive gain arising from the additional information of those variables; secondly, for the cases using only features used in the

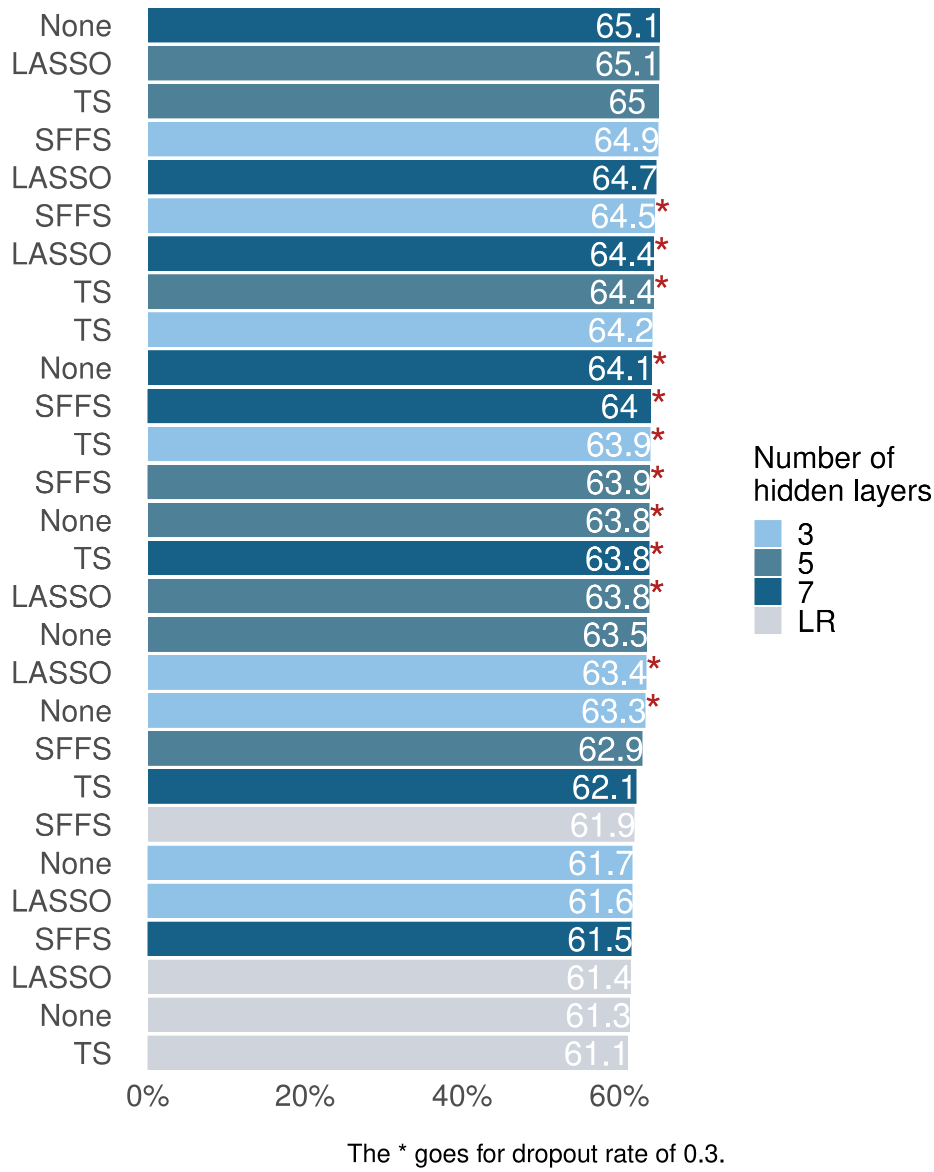

Fig. 13. Out-of-sample accuracy results for assets of SSE 180 Index - Literature + market.

literature, the majority of the models that exhibited 65% accuracy were shallow networks (logistic regression), suggesting that more complex models for this smaller set of explanatory variables actually performed worse than a simpler framework; thirdly, no distinguishable patterns were identified regarding the influence of the number of hidden layers and dropout rates for the emergence of the 65% accuracy "attractor" – in comparison to the case without dropout, turning off 30% of the neurons at each training epoch seemed to make a small effect (sometimes positive and sometimes negative) in the out-of-sample accuracy rates, while in some cases the performance metrics had a notable worsening with the presence of dropout, especially for the case with 7 hidden layers, in which neuron paths that yielded good performance were apparently "blocked". The aforementioned findings can be relevant for future financial applications of machine learning algorithms to further explore potentially informative financial indicators and better understand the potentially chaotic behavior of the hyperparameter combinations in deep neural networks.

Concerning the other classification metrics, on average, they were close to the accuracy rate, as the predictions were approximately balanced for the majority of hyperparameter combinations. Throughout the 336 combinations across the seven markets, in 41 the predictions yielded only one class – 34 cases that predicted that the prices would only drop, and 7 cases predicting that the prices would only rise. In a "only drop" case, the precision and recall would be zero, as no predictions were made for the "positive class"; similarly, in a "only rise" case, the recall would be equal to one, as a false negative would not exist since no predictions were made towards the "negative class". Both cases were highlighted in Tables 5, 8 to 13 for clarity purposes.

Besides the classification metrics discussed in the previous sections, we evaluated the profitability of the strategies based on the predictions made by the deep learning models and the maximum value for the transaction cost in the respective market for the machine learning algorithms to be able to break-even (TC_0) and to beat the Buy-and-Hold strategy (TC_{BH}). The profitability of the Buy-and-Hold was computed as the average profitability of buying all assets of the respective market at the first day of the out-of-sample testing period and selling them at the day of this period — which is equivalent to the gains of the uniform ($\frac{1}{N}$) portfolio during this period. The results are displayed in Tables 6, 14 to 19 (Tables 14 to 19 are displayed in Appendix B).

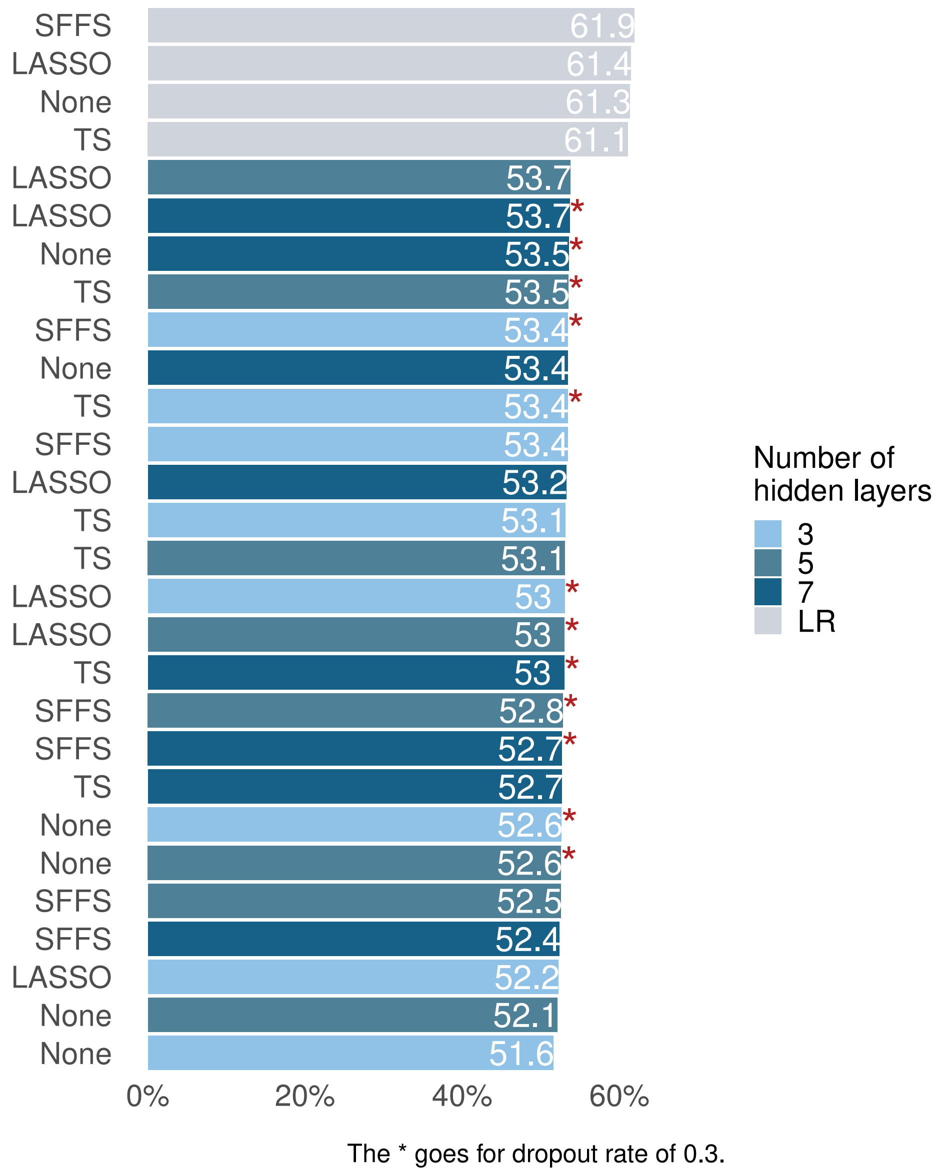

Fig. 14. Out-of-sample accuracy results for assets of SSE 180 Index - Literature.

While the predictive performance was "split" into the macro classes of 50% accuracy and 65% accuracy, the actual profitability of the machine learning-based strategies was more homogeneous: the profitability oscillated between the value of the Buy-and-Hold strategy, albeit with a big variance, and consistently negatively concentrated – *i.e.:* the models did not manage to yield profits much larger than the average market level, while they did manage to register fairly high levels of loss, especially for the British and the German markets. Even some cases with 65% out-of-sample accuracy ended up with non-profitable strategies.

Especially when considering the existence of transaction costs, the profitability of the strategies become even less desirable: in many cases, a small profit is attainable using a big number of operations, thus demanding the transaction costs TC_0 and TC_{BH} to be proportionally smaller for the strategy to become actually worth executing to generate some gain. Besides, many strategies had negative profitability to start with, such that the transaction cost would also have to be negative for those strategies to be worthwhile. Therefore, on average, the economic gains of the strategies yielded from machine learning techniques tested in this paper were statistically close to zero, reinforcing the implications of the Efficient Market Hypothesis. The strategies' profits were especially bad for the British market, where even the Buy-and-Hold gain was negative, possibly due to the period of relative political and economic instability forthcoming the events of the Brexit referendum in recent years. For the cases in which the algorithm predicted only one class (the "only rise" and "only drop" cases), the profitability was simply zero (in this case, the investor never entered the market, with the number of transactions equal to zero) or a single negative value (in this case the investor only bought the asset on day one and predicted that this price would go up all the way to the last day, in which he would still be expecting a price boost, so this investor only operated once, which was buying the asset on the first day).

4.3. Discussion of the results and practical contributions

In light of the presented results, in this subsection we will discuss the practical contributions of this paper, according to the three points listed in the introduction:

Table 13
Out-of-sample prediction results for assets of Bovespa Index.

Technical analysis indicators	Feature selection method	Hidden layers	Dropout	Accuracy	Precision	Recall	F-Score
Literature + market (Tables 1 and 2)	None	3	0	0.6241	0.5955	0.7190	0.6514
			0.3	0.6436	0.6247	0.6775	0.6500
		5	0	0.6263	0.6159	0.6241	0.6200
			0.3	0.6404	0.6130	0.7158	0.6604
		7	0	0.6253	0.5967	0.7187	0.6521
			0.3	0.6435	0.6295	0.6566	0.6428
	SFFS	3	0	0.6478	0.6341	0.6598	0.6467
			0.3	0.6426	0.6330	0.6385	0.6357
		5	0	0.6452	0.6228	0.6940	0.6565
			0.3	0.6408	0.6446	0.5901	0.6161
		7	0	0.6455	0.6232	0.6937	0.6566
			0.3	0.6428	0.6353	0.6308	0.6331
	TS	3	0	0.6466	0.6314	0.6645	0.6475
			0.3	0.6417	0.6258	0.6632	0.6439
		5	0	0.6478	0.6299	0.6762	0.6523
			0.3	0.6408	0.6198	0.6849	0.6507
		7	0	0.6377	0.5991	0.7811	0.781
			0.3	0.6389	0.6184	0.6808	0.6481
	LASSO	3	0	0.6303	0.6122	0.6631	0.6366
			0.3	0.6417	0.6262	0.6613	0.6433
		5	0	0.6359	0.6111	0.7004	0.6527
			0.3	0.6413	0.6186	0.6929	0.6537
		7	0	0.6312	0.6116	0.6710	0.6400
			0.3	0.6388	0.6118	0.7128	0.6585
Literature (Table 1)	None	3	0	0.5092	0.4977	0.5218	0.5095
			0.3	0.5057	0.4939	0.4848	0.4893
		5	0	0.5096	0.4229	0.4681	0.4802
			0.3	0.5066	0.4945	0.4502	0.4713
		7	0	0.5101	0.4985	0.4773	0.4877
			0.3	0.5114	Predicted only drops		
	SFFS	3	0	0.5163	0.5134	0.1905	0.2779
			0.3	0.5197	0.5389	0.1165	0.1917
		5	0	0.5175	0.5146	0.2163	0.3046
			0.3	0.5114	Predicted only drops		
		7	0	0.5159	0.5253	0.0947	0.1605
			0.3	0.5114	Predicted only drops		
	TS	3	0	0.5147	0.5100	0.1691	0.2540
			0.3	0.5148	0.5300	0.0611	0.1096
		5	0	0.5109	0.4989	0.2499	0.3330
			0.3	0.5114	Predicted only drops		
		7	0	0.5164	0.5221	0.1193	0.1943
			0.3	0.5114	Predicted only drops		
	LASSO	3	0	0.5123	0.5013	0.3460	0.4094
			0.3	0.5108	0.4989	0.3293	0.3968
		5	0	0.5110	0.4992	0.3148	0.3861
			0.3	0.5034	0.4910	0.4521	0.4708
		7	0	0.5041	0.4919	0.4594	0.4751
			0.3	0.5114	Predicted only drops		

- **Contribution 1**: Provide a comprehensive compendium of technical analysis indicators considered in recent scientific articles on stock prices prediction, as well as other variables that are used by market professionals that can serve as additional sources of information for researches about stock price prediction.

Concerning contribution 1, as highlighted in Section 2.3, this paper first identified a big number of technical analysis indicators that have not been applied in recent scientific papers published in top-journals, summarized in Table 2. Empirically, the overall impact of adding the variables from Table 2 to the predictive performance of stock price direction can be observed by comparing the out-of-sample performance of the deep neural network models for "Tables 1 + Table 2" and its counterpart only using variables from Table 1.

Regarding this comparison, our results showed that the addition of the variables from Table 2 managed to improve the predictive performance for all seven tested markets. As displayed in Table 7, the out-of-sample accuracy across all markets was either lesser or equal to 55% or larger or equal to 60%, and the proportion of model combinations with larger out-of-sample accuracy was much higher when the technical analysis indicators from the "market side" were considered. This result evidences the gap in the current literature and provides a list of new candidate variables which may be incorporated by stock

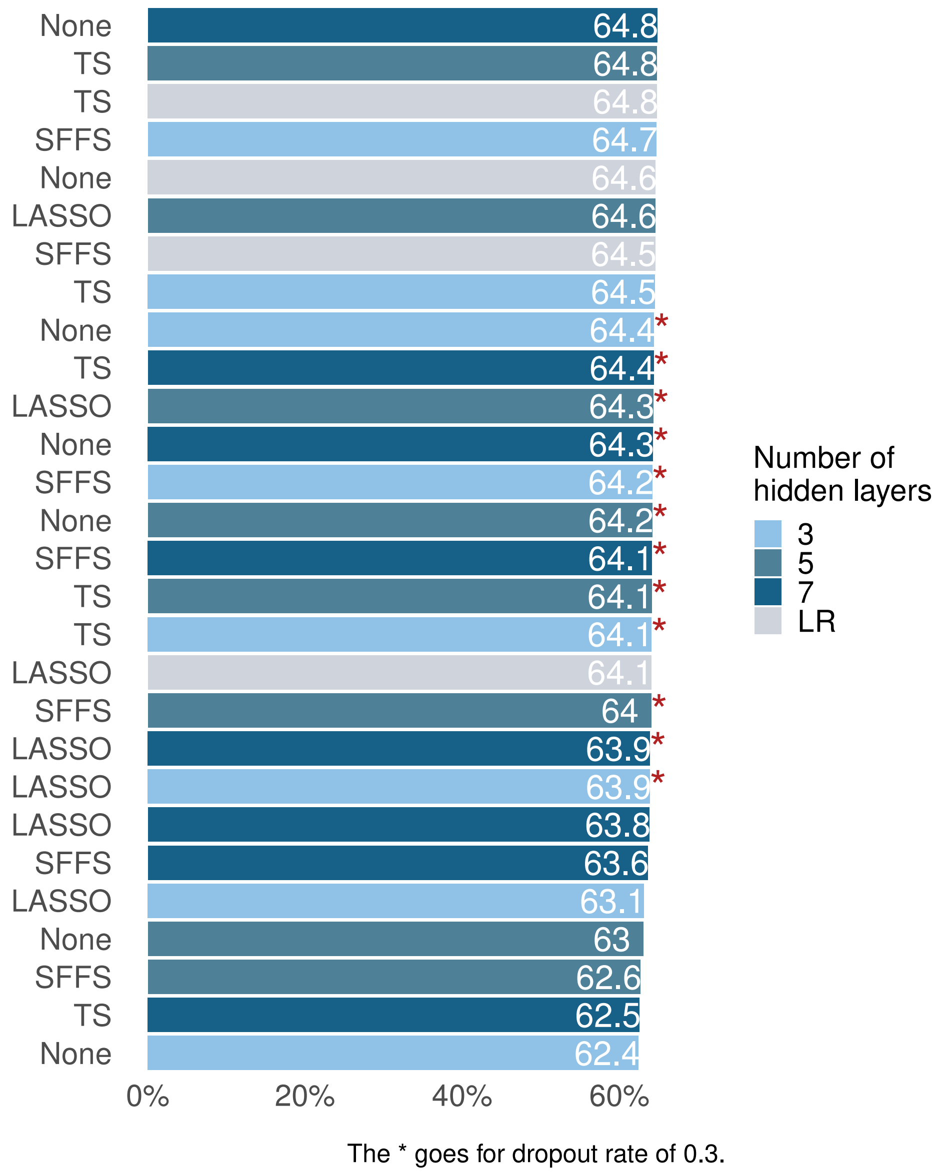

Fig. 15. Out-of-sample accuracy results for assets of Bovespa Index - Literature + market.

price prediction models on future academic researches that intend to perform similar tasks.

- **Contribution 2**: Evaluate the relative importance of each technical analysis indicator used by the literature and the market using various feature selection methods.

About contribution 2, it was observed in Section 4.1 that the technical analysis indicators were selected in a non-uniform way by the feature selection methods, indicating that some variables may contain redundant information that was already captured by other indicators, thus having a smaller relevance for predicting purposes, since their addition would represent a small potential gain of performance at the expense of a larger increment of noise.

Since the applications of feature selection methods yielded similar results in comparison to fitting the models with all features for our experiments, the findings from Section 4.1 can provide a way to shrink down the "factor zoo" of technical analysis indicators presented in Section 2.3 to a smaller number of most relevant features, without significant losses on out-of-sample prediction performance.

Moreover, out of the 72 features used by the "market side" but not yet considered by academic papers, 38 were amongst the most picked as "relevant" by the feature selection methods (on the right side of the red line in Fig. 1): the smaller subset composed by those 38 features represents an abridged version of Table 2 and can serve as a priority list for future researches in the literature of machine learning applications in stock price prediction. Cross-interactions between those variables and their respective impacts on predictive performance can also be explored in future papers from this agenda.

- **Contribution 3**: Test the empirical performance of deep neural networks for seven markets by applying different settings of architecture and regularization, evaluating not only the classification metrics but also estimating the maximum bearable transaction cost from the perspective of an investor in order to obtain actual profitability from the yielded strategies.

Finally, regarding contribution 3, our empirical analysis took into consideration not only whether the deep neural network models succeed in predicting the direction of the stock prices but also to which

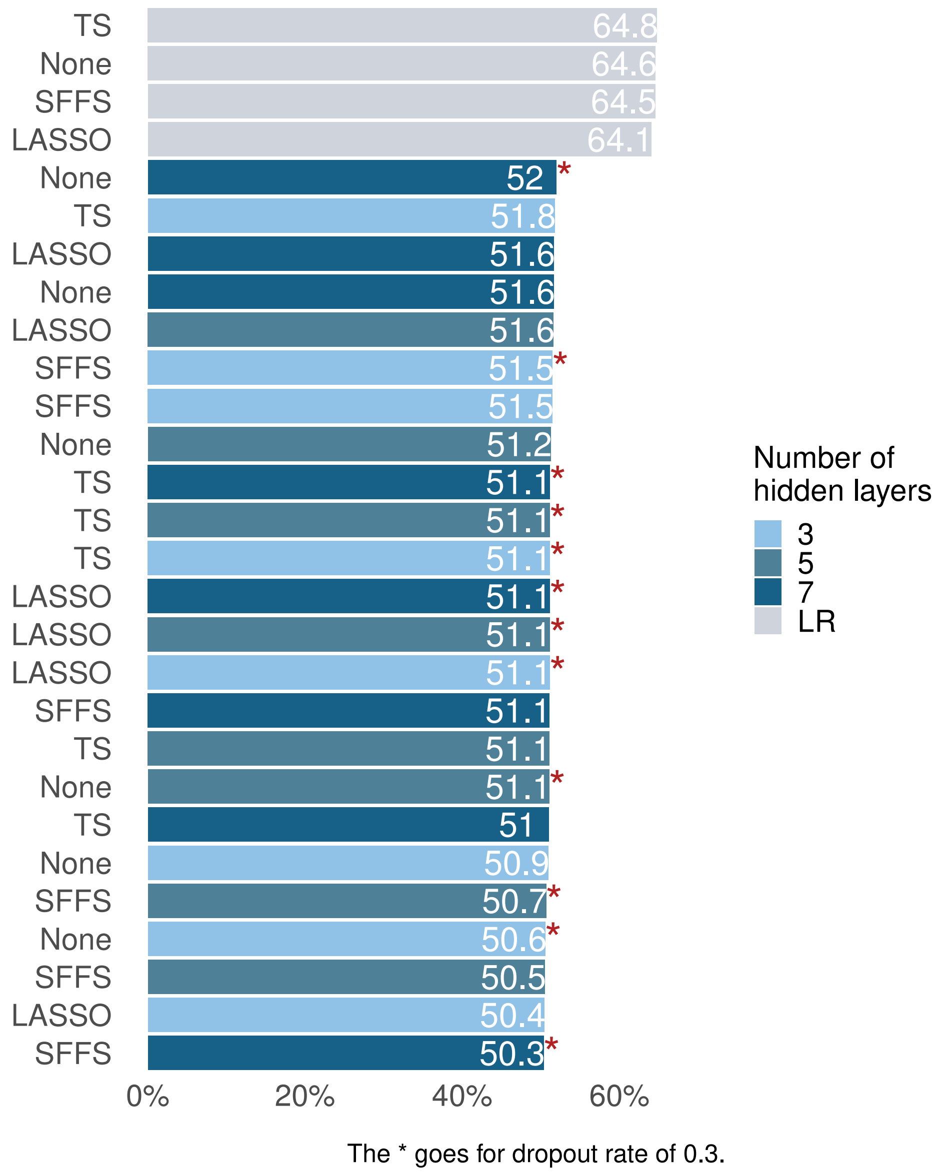

Fig. 16. Out-of-sample accuracy results for assets of Bovespa Index - Literature.

extent those predictions can lead to profitable trading strategies for an investor that applies it in practice. For this purpose, we paired the profitability analysis with the maximum affordable transaction cost that allows each model combination to generate gains and to outperform the buy-and-hold strategy, which is considered a conservative trading strategy (Dichtl, 2020; Sanderson & Lumpkin-Sowers, 2018). However, transaction costs are often not considered in studies that analyze the profitability of machine learning-based trading strategies. For instance, Long, Lu, and Cui (2019) proposed a neural network model with multi-filters architecture and reported that this model outperformed convolutional and recurrent neural networks in terms of profitability; however, while the authors considered a risk-adjusted evaluation metric (Sharpe ratio) to evaluate the results, a key assumption was the absence of transaction costs. Similarly, Vijh, Chandola, Tikkiwal, and Kumar (2020) claimed that random forest and artificial neural network showed to be efficient in predicting stock closing price testing for a small number of assets and not considering transaction costs. Other works that disregarded the existence of transaction costs in machine learning applications for stock price prediction include Nelson, Pereira, and de Oliveira (2017) and Nikou, Mansourfar, and Bagherzadeh (2019).

Although the absence of transaction costs is a common assumption in papers that perform similar experiments (Long et al., 2019), this is an unrealistic assumption in actual financial trading; therefore, by assessing the strategies' profitability as a function of the maximum transaction cost under which gains are possible, our paper adds to the literature of machine learning applications in stock price prediction by determining the effectiveness of the models based on the actual transaction costs from the respective markets, hence providing a more accurate measure of the models' impact on real-world decision-making.

In this sense, the results presented in Section 4.2 are consistent with previous studies like Fischer and Krauss (2018), which reported that the profitability of strategies based on Long Short-Term Memory (LSTM) neural networks oscillated around zero for S&P 500 stocks from 2010 to 2015 when the presence of transaction costs are considered, even though said strategy managed to yield significant excess returns over the market from 1992 to 2009. Similar results were reported in Paiva, Cardoso, Hanaoka, and Duarte (2019), which applied Support

Table 14
Trading profitability and transaction costs of machine learning algorithms for assets of FTSE 100 Index.

Technical analysis indicators	Feature selection method	Hidden layers	Dropout	Strategy profitability	Number of transactions	TC_0	TC_{BH}
Literature + market (Tables 1 and 2)	None	3	0	−1231.6579	204	−6.6718	−2.5560
			0.3	−876.1521	197	−4.9394	−0.7267
		5	0	−679.8966	222	−3.4837	0.2314
			0.3	−342.2837	180	−2.1448	2.4721
		7	0	−818.2142	198	−4.2889	−0.4163
			0.3	−1022.4958	191	−6.1411	−1.6068
	SFFS	3	0	−728.0858	221	−3.7490	0.0032
			0.3	−483.8384	200	−2.6855	1.3802
		5	0	−1001.1223	197	−5.6457	−1.4699
			0.3	−977.1600	186	−5.9200	−1.4305
		7	0	−872.9466	148	−5.6816	−0.8380
			0.3	0.0000	0	0.0000	0.0000
	TS	3	0	−905.6980	197	−4.8306	−0.8758
			0.3	−953.5090	182	−5.9360	−1.2925
		5	0	−795.1669	213	−4.1427	−0.2757
			0.3	−294.7987	203	−1.7610	2.4298
		7	0	−1010.0929	206	−5.0971	−1.3522
			0.3	−1093.3283	198	−6.0597	−1.9353
	LASSO	3	0	−720.3165	209	−4.0450	−0.0097
			0.3	0.0000	0	0.0000	0.0000
		5	0	−558.4443	202	−2.8833	0.9546
			0.3	−1102.4661	190	−6.3286	−2.0084
		7	0	−1068.2750	158	−8.1302	−2.5532
			0.3	−486.8248	152	−4.0796	1.9143
Literature (Table 1)	None	3	0	−880.1103	200	−4.5685	−0.7427
			0.3	−958.1202	194	−5.4884	−1.2621
		5	0	−1089.2449	135	−8.6373	−2.9405
			0.3	0.0000	0	0.0000	0.0000
		7	0	−951.0772	205	−5.1854	−0.9388
			0.3	−1067.3949	187	−6.1045	−1.7920
	SFFS	3	0	−772.5879	203	−3.8355	−0.1945
			0.3	−1114.6619	122	−10.7339	−3.6361
		5	0	−630.8446	205	−3.5275	0.6870
			0.3	0.0000	0	0.0000	0.0000
		7	0	−261.1525	209	−1.4958	2.3081
			0.3	0.0000	0	0.0000	0.0000
	TS	3	0	−759.3180	205	−3.6857	−0.0713
			0.3	0.0000	0	0.0000	0.0000
		5	0	−376.9710	182	−2.2109	2.3533
			0.3	0.0000	0	0.0000	0.0000
		7	0	−1007.0923	198	−5.4493	−1.4586
			0.3	−1064.7026	182	−6.7676	−1.9861
	LASSO	3	0	−846.9454	183	−4.9279	−0.8298
			0.3	0.0000	0	0.0000	0.0000
		5	0	−1138.4845	203	−6.0562	−2.0636
			0.3	−1013.1588	186	−5.8574	−1.4905
		7	0	−274.0323	206	−1.4215	2.5372
			0.3	0.0000	0	0.0000	0.0000

Buy-and-Hold strategy profitability over the out-of-sample period: −34.75736

Vector Machine for portfolio selection and achieved good predicting results, but the portfolios' profitability was strongly diminished when brokerage costs are introduced.

In terms of the profitability of the trading strategies, the effect of the number of hidden layers was also not apparent, as the same pattern of similarity between the cases with 3 and 7 hidden layers persisted. Our results differ from the reports of Nakano et al. (2018), in which deeper neural networks yielded strategies with better profitability; on the other hand, our results were similar to those found by Lv, Yuan, Li, and Xiang (2019), in whose experiments the deep learning methods did not statistically outperform the logistic regression, and the majority of cases exhibited significantly different performances with the introduction of transaction costs. In our results, the effect of dropout also seems heterogeneous across different hyperparameters and markets. Concerning the choice of candidate features, the "None" case (no feature selection method) showed fairly good profitability when considering only technical analysis indicators from the "Literature side", while for the more extensive indicator set composed by both Literature and Market experiences, the application of feature selection algorithms yielded a slight overall improvement, an expected result due to the size of the feature set and the high degree of correlation between them.

Table 15
Trading profitability and transaction costs of machine learning algorithms for assets of CAC 40 Index.

Technical analysis indicators	Feature selection method	Hidden layers	Dropout	Strategy profitability	Number of transactions	TC_0	TC_{BH}
Literature + market (Tables 1 and 2)	None	3	0	30.9116	88	0.352	0.278
			0.3	19.7555	80	0.249	0.183
		5	0	−17.9450	128	−0.148	−0.200
			0.3	−11.6289	73	−0.226	−0.346
		7	0	30.9218	101	0.294	0.225
			0.3	31.4456	83	0.412	0.358
	SFFS	3	0	−6.6505	93	−0.1185	−0.2590
			0.3	−9.4849	55	−1.0847	−0.9257
		5	0	22.9734	82	0.2824	0.2045
			0.3	28.5655	90	0.3412	0.2763
		7	0	−21.7477	109	−0.2165	−0.2839
			0.3	5.6222	72	0.0772	−0.0297
	TS	3	0	28.8666	94	0.3341	0.2667
			0.3	26.2245	82	0.4768	0.4215
		5	0	−11.0071	99	−0.1324	−0.2043
			0.3	0.3697	42	−0.4944	−0.5041
		7	0	32.5566	96	0.3535	0.2933
			0.3	31.1308	77	0.3667	0.3063
	LASSO	3	0	−0.8673	120	0.0043	−0.0613
			0.3	−0.8773	72	−0.0132	−0.1185
		5	0	24.0689	100	0.2615	0.1867
			0.3	37.7236	78	0.4752	0.4129
		7	0	−7.6266	63	−0.2898	−0.3378
			0.3	−9.6566	50	−1.1598	−1.0225
Literature (Table 1)	None	3	0	23.7956	84	0.2935	0.2220
			0.3	19.3818	87	0.2240	0.1636
		5	0	2.7511	99	0.0322	−0.0640
			0.3	5.6516	72	0.0830	−0.0321
		7	0	29.1989	92	0.3373	0.2611
			0.3	31.7185	88	0.5304	0.4733
	SFFS	3	0	−8.7979	97	−0.0866	−0.1614
			0.3	−0.8606	40	−0.7441	−0.7401
		5	0	23.4107	92	0.2448	0.1647
			0.3	0.0000	0	0.0000	0.0000
		7	0	−23.2653	75	−0.4770	−0.5796
			0.3	−9.5067	86	−0.1387	−0.2203
	TS	3	0	28.9731	106	0.2853	0.2142
			0.3	0.0000	0	0.0000	0.0000
		5	0	−9.2055	58	−0.5575	−0.5791
			0.3	0.0000	0	0.0000	0.0000
		7	0	20.1151	80	0.2549	0.1755
			0.3	0.0000	0	0.0000	0.0000
	LASSO	3	0	−4.3156	74	−0.0990	−0.1910
			0.3	0.0000	0	0.0000	0.0000
		5	0	27.1119	87	0.3367	0.2611
			0.3	0.0000	0	0.0000	0.0000
		7	0	0.0000	0	0.0000	0.0000
			0.3	0.0000	0	0.0000	0.0000

Buy-and-Hold strategy profitability over the out-of-sample period: 7.426471

5. Conclusion and remarks

This paper analyzed the performance of deep neural network algorithms to predict the stock price movement based on technical analysis indicators selected from recent scientific articles and specialized trading websites. Using daily data from financial assets that compose seven market indexes around the world between 2008 and 2019, we tested different settings of hyperparameters, namely the number of hidden layers in each neural network and the dropout rate. We applied three feature selection methods (Sequential Forward Floating Selection, Tournament Screening, and LASSO) on the feature set of technical analysis indicators, using their filtered counterparts to be used as explanatory variables in the training process.

The results indicated that the out-of-sample accuracy rate of the prediction converged to two values: the 50% value, which would reflect market efficiency, and a "strange attractor" of 65%, which was also was achieved consistently, but does seem to reflect a theoretical concept, and could be investigated in further studies. Nonetheless, when applying the prediction into a real trading experiment, the strategies' profitability did not manage to significantly outperform

Table 16
Trading profitability and transaction costs of machine learning algorithms for assets of DAX-30 Index.

Technical analysis indicators	Feature selection method	Hidden layers	Dropout	Strategy profitability	Number of transactions	TC_0	TC_{BH}
Literature + market (Tables 1 and 2)	None	3	0	−7.0142	108	−0.0999	0.3064
			0.3	−95.3067	30	−51.1154	−51.0662
		5	0	−111.6834	19	−55.6554	−56.4373
			0.3	−118.0445	15	−50.0377	−49.1479
		7	0	−57.8716	116	−0.5213	−0.1559
			0.3	−58.3453	28	−50.3453	−50.3112
	SFFS	3	0	−116.7232	26	−53.7583	−53.6025
			0.3	−116.6538	14	−57.9246	−59.4331
		5	0	5.3684	107	0.0982	0.5336
			0.3	−74.4657	13	−25.3718	−31.0785
		7	0	−50.3786	119	−0.4876	−0.1002
			0.3	−57.6454	88	−0.6801	−0.2187
	TS	3	0	−32.4863	114	−0.2170	0.3989
			0.3	−47.2818	28	−56.1255	−58.3279
		5	0	−101.1019	9	−75.3873	−80.9304
			0.3	−128.2447	15	−79.7961	−91.0733
		7	0	−37.3058	31	−48.4516	−48.0509
			0.3	−69.3260	28	−29.4667	−35.9161
	LASSO	3	0	−118.7176	13	−31.7819	−41.8065
			0.3	−117.4401	13	−50.3404	−49.7330
		5	0	−24.7099	81	−0.0299	0.4779
			0.3	−57.4242	22	−51.2784	−51.4561
		7	0	−109.0390	15	−79.5897	−90.8774
			0.3	−116.5046	11	−59.4268	−62.0856
Literature (Table 1)	None	3	0	21.9347	101	0.0947	0.4276
			0.3	−52.9348	12	−29.9100	−37.5991
		5	0	−119.4156	42	−2.6834	−2.8865
			0.3	−49.0859	91	−0.5618	−0.1290
		7	0	−66.6646	29	−57.8594	−60.1799
			0.3	−81.8274	19	−38.8543	−45.0517
	SFFS	3	0	−124.6436	39	−45.9953	−43.6830
			0.3	−45.0745	71	−40.0923	−36.5127
		5	0	−12.4590	115	−0.1727	0.1668
			0.3	−80.3769	16	−53.0410	−55.4448
		7	0	−126.0814	18	−37.6227	−33.7698
			0.3	−129.2861	1	−129.2861	−91.7557
	TS	3	0	−75.2838	26	−26.9383	−32.0525
			0.3	−71.8876	29	−50.4763	−50.5510
		5	0	−103.2904	17	−70.8163	−80.5473
			0.3	−129.2861	1	−129.2861	−91.7557
		7	0	−65.2524	17	−40.2059	−47.0944
			0.3	−129.2861	1	−129.2861	−91.7557
	LASSO	3	0	−108.2516	35	−4.9690	−5.7337
			0.3	−129.2861	1	−129.2861	−91.7557
		5	0	−119.3640	27	−36.5592	−41.9525
			0.3	−129.2861	1	−129.2861	−91.7557
		7	0	−91.3948	43	−43.9058	−41.1566
			0.3	−129.2861	1	−129.2861	−91.7557

Buy-and-Hold strategy profitability over the out-of-sample period: 13.75618

the Buy-and-Hold strategy while showing more consistent losses in markets that presented higher levels of volatility during the testing period.

The findings of this paper can be of potential interest for scholars for future inquiries in similar lines of research, as many technical analysis indicators that were most picked by feature selection methods were not considered by authors in recent applications on stock price prediction. Instead, many of these indicators are commonly used by investors in their real-world trading. In addition, we find that some indicators composed of the combination of other indicators can be taken into account instead of their constitute counterparts. This procedure can diminish the levels of redundant information taken into account for the models and potentially yield better predictive results and asset allocations. Moreover, the values for the maximum transaction cost levels for an investor to reach some economic gain or to outperform the Buy-and-Hold strategy can be used to analyze the overall attractiveness of different financial markets, with an investor potentially willing to operate in markets in which the transaction costs are lower than the thresholds found in this paper.

Table 17
Trading profitability and transaction costs of machine learning algorithms for the top 50 assets of NIKKEI 225 Index.

Technical analysis indicators	Feature selection method	Hidden layers	Dropout	Strategy profitability	Number of transactions	TC_0	TC_{BH}
Literature + market (Tables 1 and 2)	None	3	0	7803.1598	89	77.4127	64.7076
			0.3	7920.0890	93	88.3608	74.6872
		5	0	−5529.4543	51	−738.2236	−904.1374
			0.3	−2406.1468	151	−19.3880	−28.6744
		7	0	6946.7547	90	70.3862	56.6876
			0.3	7288.4576	97	74.6892	61.4661
	SFFS	3	0	−4673.4202	92	−82.4101	−106.5526
			0.3	−2840.2886	138	−23.0337	−33.5978
		5	0	7205.2919	93	69.5090	56.9625
			0.3	7393.7619	87	84.1624	69.9723
		7	0	−6433.7873	43	−5087.6461	−6156.3638
			0.3	1778.6500	191	9.6339	1.7947
	TS	3	0	7240.3170	91	70.0058	57.5585
			0.3	8157.9939	78	94.5646	80.0575
		5	0	−4951.7845	79	−629.9660	−832.6242
			0.3	−2842.2710	137	−21.3855	−30.9577
		7	0	7431.5764	92	79.4053	65.0827
			0.3	6658.2396	90	68.4016	54.5935
	LASSO	3	0	−2633.1845	128	−24.4690	−36.4269
			0.3	−5750.1541	53	−320.4699	−423.5817
		5	0	7685.6084	98	80.9512	67.6100
			0.3	7178.1310	92	71.1044	58.2038
		7	0	−3598.0590	134	−35.1086	−48.4995
			0.3	−4307.3004	84	−49.9346	−63.9898
Literature (Table 1)	None	3	0	6986.1535	81	91.9816	74.8566
			0.3	7116.7184	92	69.7707	57.0395
		5	0	−7329.1988	9	−4835.3665	−5758.0627
			0.3	0.0000	0	0.0000	0.0000
		7	0	7727.0345	87	83.7075	69.7626
			0.3	7345.8764	84	72.4738	59.9154
	SFFS	3	0	−2425.7614	128	−18.3464	−29.3510
			0.3	−3759.1445	66	−136.4005	−189.0367
		5	0	7066.3257	92	72.9686	59.8742
			0.3	6722.3498	93	66.8265	53.3423
		7	0	−3214.7343	142	−24.9785	−35.0148
			0.3	0.0000	0	0.0000	0.0000
	TS	3	0	6952.5557	96	69.0497	56.3109
			0.3	0.0000	0	0.0000	0.0000
		5	0	−4545.8384	132	−38.3434	−48.6162
			0.3	0.0000	0	0.0000	0.0000
		7	0	7665.0296	83	84.9334	71.1548
			0.3	6415.0398	89	70.5765	56.1770
	LASSO	3	0	−6435.8059	89	−107.3741	−127.7128
			0.3	0.0000	0	0.0000	0.0000
		5	0	8380.3421	83	94.5106	80.9789
			0.3	5719.4074	85	60.6517	46.9317
		7	0	633.1867	122	7.3991	−5.8053
			0.3	0.0000	0	0.0000	0.0000

Buy-and-Hold strategy profitability over the out-of-sample period: 1396.319

The combinations of hyperparameters considered in this paper are not exhaustive, as many improvements and additional cases could be executed. For instance, the only activation function that we applied was the Sigmoid function, whilst there are many other candidate functions such as the Hyperbolic Tangent and the ReLU (Rectified Linear Unit), both very popular in neural network applications. As discussed in Yaohao and Albuquerque (2019), the choice of the function that defines the structure of non-linear interactions of the data has a decisive impact on the results. In this sense, we recommend further investigations about the implications of different activation functions in this paper's application.

Other potential improvements include using more training epochs and testing for more values for the number of hidden layers other than 3, 5, and 7, as well as testing other cases for the dropout rate of the networks and other feature selection methods. An analysis of the sensibility of the models to alterations in those hyperparameters could also contribute to further understanding the behavior of financial assets. Replications of this study considering other time periods and financial assets also reflect a topic for future developments. In addition, one could also investigate the use of rolling windows to re-calibrate the models with a larger periodicity and further analyze whether the strategies' profitability can be better in a smaller period, i.e., using high-frequency data. Finally, another future topic would be to assess

Table 18
Trading profitability and transaction costs of machine learning algorithms for the top 50 assets of SSE 180 Index.

Technical analysis indicators	Feature selection method	Hidden layers	Dropout	Strategy profitability	Number of transactions	TC_0	TC_{BH}
Literature + market (Tables 1 and 2)	None	3	0	19.2356	82	0.1900	0.0866
			0.3	23.8928	106	0.2329	0.1130
		5	0	−21.5139	65	−0.7373	−0.7669
			0.3	−22.6591	131	−0.5747	−0.5208
		7	0	22.2560	84	0.2135	0.0969
			0.3	24.8313	106	0.2217	0.1097
	SFFS	3	0	−0.9213	58	−0.1404	−0.2672
			0.3	9.0263	100	−0.1058	−0.1382
		5	0	23.2658	101	0.1868	0.0908
			0.3	22.9580	91	0.2302	0.1035
		7	0	4.1075	85	0.0357	−0.0897
			0.3	7.7811	117	0.0648	−0.0461
	TS	3	0	22.9278	92	0.2105	0.1522
			0.3	24.3774	99	0.2448	0.2033
		5	0	2.1913	57	0.0881	−0.2479
			0.3	−1.1676	87	0.3269	−2.7080
		7	0	25.3481	102	0.2036	0.1042
			0.3	23.9257	86	0.2117	0.1226
	LASSO	3	0	−17.9146	114	−0.1225	−0.1958
			0.3	3.2801	74	0.0421	−0.1180
		5	0	19.5945	110	0.1595	0.0603
			0.3	20.0520	89	0.1776	0.0747
		7	0	1.8168	74	−0.0242	−0.1168
			0.3	3.5199	66	0.0481	−0.1276
Literature (Table 1)	None	3	0	41.0052	92	0.3647	0.2543
			0.3	24.1797	99	0.1969	0.1000
		5	0	3.1839	66	0.0227	−0.1108
			0.3	5.4558	84	0.0528	−0.0788
		7	0	24.6735	96	0.2549	0.1234
			0.3	15.8630	84	0.1565	0.0598
	SFFS	3	0	−25.7442	94	−0.2258	−0.3397
			0.3	−0.8207	60	0.0592	−0.9790
		5	0	25.1208	96	0.2581	0.1699
			0.3	22.1315	82	0.2281	0.1447
		7	0	2.1438	118	−0.4112	−0.3753
			0.3	−8.4498	73	−0.3854	−1.0176
	TS	3	0	13.3439	97	0.1403	−0.0104
			0.3	23.2204	101	0.2099	0.0990
		5	0	−12.8620	100	−0.0965	−0.1810
			0.3	−14.9783	82	−0.3454	−0.6642
		7	0	21.6362	106	0.1904	0.0784
			0.3	25.4432	101	0.1980	0.1064
	LASSO	3	0	−15.3891	83	−0.2678	−0.5097
			0.3	0.0000	0	0.0000	0.0000
		5	0	25.7859	97	−0.1525	−0.1433
			0.3	17.3476	91	0.1547	0.0472
		7	0	−28.9513	92	−0.2296	−0.3321
			0.3	0.0000	0	0.0000	0.0000

Buy-and-Hold strategy profitability over the out-of-sample period: 13.25879

to what extent the gains can be higher when using a dynamic model to incorporate sudden changes over the historical pattern.

CRediT authorship contribution statement

Yaohao Peng: Conceptualization, Methodology, Software, Formal analysis, Writing - original draft, Writing - review & editing. **Pedro Henrique Melo Albuquerque:** Conceptualization, Methodology, Validation, Formal analysis, Writing - review & editing. **Herbert Kimura:** Methodology, Validation, Formal analysis, Writing - review & editing. **Cayan Atreio Portela Bárcena Saavedra:** Software, Visualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Disclaimer

The views expressed in this work are of entire responsibility of the authors and do not necessarily reflect those of their respective affiliated institutions nor those of its members.

Table 19
Trading profitability and transaction costs of machine learning algorithms for assets of Bovespa Index.

Technical analysis indicators	Feature selection method	Hidden layers	Dropout	Strategy profitability	Number of transactions	TC_0	TC_{BH}
Literature + market (Tables 1 and 2)	None	3	0	35.6638	95	0.3841	0.2829
			0.3	39.0711	88	0.4365	0.3410
		5	0	−4.8916	122	−0.2552	−0.6701
			0.3	2.5210	83	0.0390	−0.3591
		7	0	32.5512	86	0.3893	0.2914
			0.3	34.9669	89	0.1855	−0.2500
	SFFS	3	0	−2.5505	116	−0.0255	−0.0989
			0.3	−2.6532	94	−0.0343	−0.2762
		5	0	37.1818	86	0.4339	0.3332
			0.3	37.4408	95	0.3885	0.3011
		7	0	−1.2046	58	−0.0498	−0.2126
			0.3	−0.3102	77	−0.0128	−0.1337
	TS	3	0	37.2284	81	0.4705	0.3657
			0.3	38.9214	89	0.4267	0.3356
		5	0	−3.0324	110	−0.0308	−0.1212
			0.3	0.0000	0	0.0000	0.0000
		7	0	32.4841	86	0.3875	0.2822
			0.3	37.9563	93	0.4033	0.3105
	LASSO	3	0	−5.4923	108	−0.0915	−0.2472
			0.3	0.0000	0	0.0000	0.0000
		5	0	32.5362	84	0.3538	0.1858
			0.3	37.1518	90	0.4047	0.3146
		7	0	−4.4870	103	−0.0410	−0.1209
			0.3	−0.5910	91	−0.0162	−0.1006
Literature (Table 1)	None	3	0	35.7799	84	0.3985	0.2526
			0.3	36.8370	90	0.4015	0.3110
		5	0	2.5242	141	0.0171	−0.0463
			0.3	0.0000	0	0.0000	0.0000
		7	0	39.6499	82	0.4834	0.3831
			0.3	38.0610	96	0.3865	0.3029
	SFFS	3	0	−3.8389	76	−0.0422	−0.1778
			0.3	0.0000	0	0.0000	0.0000
		5	0	31.2168	89	0.3721	0.2775
			0.3	39.3754	89	0.4322	0.3377
		7	0	−2.5622	113	−0.0459	−0.1641
			0.3	0.0000	0	0.0000	0.0000
	TS	3	0	36.0714	88	0.4155	0.3142
			0.3	31.2031	85	0.3155	0.1067
		5	0	−8.7033	117	−0.0796	−0.1517
			0.3	0.0000	0	0.0000	0.0000
		7	0	30.6606	89	0.3504	0.2434
			0.3	33.2796	89	0.3827	0.2793
	LASSO	3	0	−1.2243	112	−0.0179	−0.1001
			0.3	0.0000	0	0.0000	0.0000
		5	0	38.2578	82	0.4765	0.3721
			0.3	37.5638	93	0.3970	0.3105
		7	0	−1.7949	128	−0.0134	−0.0870
			0.3	0.0000	0	0.0000	0.0000

Buy-and-Hold strategy profitability over the out-of-sample period: 8.71314

Appendix A. Predictive performance for the remaining markets

See Tables 8–13 and Figs. 5–16.

Appendix B. Profitability of strategies and transaction cost for the remaining markets

See Tables 14–19.

References

Abdulla, M., & Khasawneh, M. T. (2020). G-forest: An ensemble method for cost-sensitive feature selection in gene expression microarrays. *Artificial Intelligence in Medicine*, *108*, Article 101941.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, *19*(6), 716–723.

Alhashel, B. S., Almudhaf, F. W., & Hansz, J. A. (2018). Can technical analysis generate superior returns in securitized property markets? Evidence from east asia markets. *Pacific-Basin Finance Journal*, *47*, 92–108.

Ang, K. K., & Quek, C. (2006). Stock trading using rspop: A novel rough set-based neuro-fuzzy approach. *IEEE Transactions on Neural Networks*, *17*(5), 1301–1315.

Armano, G., Marchesi, M., & Murru, A. (2005). A hybrid genetic-neural architecture for stock indexes forecasting. *Information Sciences*, *170*(1), 3–33.

Cavendish Astrophysics (2011). Technical indicators and overlays. https://www.mrao.cam.ac.uk/~mph/Technical_Analysis.pdf.

Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers and Electrical Engineering*, *40*(1), 16–28.

Chang, P.-C., & Fan, C.-Y. (2008). A hybrid system integrating a wavelet and TSK fuzzy rules for stock price forecasting. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *38*(6), 802–815.

Chang, P.-C., Liu, C.-H., Lin, J.-L., Fan, C.-Y., & Ng, C. S. (2009). A neural network with a case based dynamic window for stock trading prediction. *Expert Systems with Applications*, *36*(3), 6889–6898.

Chang, P.-C., et al. (2012). A novel model by evolving partially connected neural network for stock price trend forecasting. *Expert Systems with Applications*, *39*(1), 611–620.

Chatzis, S. P., Siakoulis, V., Petropoulos, A., Stavroulakis, E., & Vlachogiannakis, N. (2018). Forecasting stock market crisis events using deep and statistical machine learning techniques. *Expert Systems with Applications*, *112*, 353–371.

Chavarnakul, T., & Enke, D. (2009). A hybrid stock trading system for intelligent technical analysis-based equivolume charting. *Neurocomputing*, *72*(16–18), 3517–3528.

Chen, Z., & Chen, J. (2009). Tournament screening cum ebic for feature selection with high-dimensional feature spaces. *Science in China Series A: Mathematics*, *52*(6), 1327–1341.

Chen, Y.-S., Cheng, C.-H., & Tsai, W.-L. (2014). Modeling fitting-function-based fuzzy time series patterns for evolving stock index forecasting. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, *41*(2), 327–347.

Chen, C.-C., Kuo, Y.-C., Huang, C.-H., & Chen, A.-P. (2014). Applying market profile theory to forecast Taiwan index futures market. *Expert Systems with Applications*, *41*(10), 4617–4624.

Chen, H., Xiao, K., Sun, J., & Wu, S. (2017). A double-layer neural network framework for high-frequency forecasting. *ACM Transactions on Management Information Systems (TMIS)*, *7*(4), 11.

Chiang, W.-C., Enke, D., Wu, T., & Wang, R. (2016). An adaptive stock index trading decision support system. *Expert Systems with Applications*, *59*, 195–207.

Chicco, D. (2017). Ten quick tips for machine learning in computational biology. *BioData Mining*, *10*(1), 35.

Choudhury, S. J., & Pal, N. R. (2019). Imputation of missing data with neural networks for classification. *Knowledge-Based Systems*, *182*, Article 104838.

Claesen, M., & De Moor, B. (2015). Hyperparameter search in machine learning. arxiv preprint arxiv:1502.02127.

Cochrane, J. H. (2011). Presidential address: Discount rates. *The Journal of Finance*, *66*(4), 1047–1108.

da Costa, T. R. C. C., Nazário, R. T., Bergo, G. S. Z., Sobreiro, V. A., & Kimura, H. (2015). Trading system based on the use of technical analysis: A computational experiment. *Journal of Behavioral and Experimental Finance*, *6*, 42–55.

Creamer, G. (2012). Model calibration and automated trading agent for euro futures. *Quantitative Finance*, *12*(4), 531–545.

Dichtl, H. (2020). Investing in the s&p 500 index: Can anything beat the buy-and-hold strategy?. *Review of Financial Economics*, *38*(2), 352–378.

Duriez, T., Brunton, S. L., & Noack, B. R. (2017). *Machine learning control-taming nonlinear dynamics and turbulence, Vol. 116*. Springer.

Engle, R. F. (1984). Wald, likelihood ratio, and Lagrange multiplier tests in econometrics. *Handbook of Econometrics*, *2*, 775–826.

Fama, E. F., & French, K. R. (1992). The cross-section of expected stock returns. *The Journal of Finance*, *47*(2), 427–465.

Fama, E. F., & French, K. R. (1993). Common risk factors in the returns on stocks and bonds. *Journal of Financial Econometrics*, *33*(1), 3–56.

Fama, E. F., & French, K. R. (1996). Multifactor explanations of asset pricing anomalies. *The Journal of Finance*, *51*(1), 55–84.

Fama, E. F., & French, K. R. (2015). A five-factor asset pricing model. *Journal of Financial Econometrics*, *116*(1), 1–22.

Feng, G., Giglio, S., & Xiu, D. (2020). Taming the factor zoo: A test of new factors. *The Journal of Finance*, *75*(3), 1327–1370.

Ferreira Filho, D., Bueno Filho, J. S. d. S., Regitano, L. C. d. A., Alencar, M. M. d., Alves, R. R., & Meirelles, S. L. C. (2019). Tournaments between markers as a strategy to enhance genomic predictions. *PLoS One*, *14*(6), Article e0217283.

Fidelity Investments (2019). Technical indicator guide. https://www.fidelity.com/learning-center/trading-investing/technical-analysis/technical-indicator-guide/overview.

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, *270*(2), 654–669.

Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning, Vol. 1*. MIT press Cambridge.

Gorenc Novak, M., & Velušček, D. (2016). Prediction of stock price movement based on daily high prices. *Quantitative Finance*, *16*(5), 793–826.

Gu, S., Kelly, B., & Xiu, D. (2020). Empirical asset pricing via machine learning. *Review of Financial Studies*, *33*(5), 2223–2273.

Guresen, E., Kayakutlu, G., & Daim, T. U. (2011). Using artificial neural network models in stock market index prediction. *Expert Systems with Applications*, *38*(8), 10389–10397.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, *3*(Mar), 1157–1182.

Harvey, C. R., Liu, Y., & Zhu, H. (2016). …and the cross-section of expected returns. *Review of Financial Studies*, *29*(1), 5–68.

Heaton, J., Polson, N., & Witte, J. H. (2017). Deep learning for finance: deep portfolios. *Applied Stochastic Models in Business and Industry*, *33*(1), 3–12.

Henrique, B. M., Sobreiro, V. A., & Kimura, H. (2018). Stock price prediction using support vector regression on daily and up to the minute prices. *The Journal of Finance and Data Science*, *4*(3), 183–201.

Henrique, B. M., Sobreiro, V. A., & Kimura, H. (2019). Literature review: Machine learning techniques applied to financial market prediction. *Expert Systems with Applications*.

Hsu, M.-W., Lessmann, S., Sung, M.-C., Ma, T., & Johnson, J. E. (2016). Bridging the divide in financial market forecasting: machine learners vs. financial economists. *Expert Systems with Applications*, *61*, 215–234.

Huang, B., Huan, Y., Xu, L. D., Zheng, L., & Zou, Z. (2019). Automated trading systems statistical and machine learning methods and hardware implementation: a survey. *Enterprise Information Systems*, *13*(1), 132–144.

Huang, C.-L., & Tsai, C.-Y. (2009). A hybrid sofm-svr with a filter-based feature selection for stock market forecasting. *Expert Systems with Applications*, *36*(2), 1529–1539.

Huang, Y.-P., & Yen, M.-F. (2019). A new perspective of performance comparison among machine learning algorithms for financial distress prediction. *Applied Soft Computing*, *83*, Article 105663.

Hwang, S., & Rubesam, A. (2019). *Searching the factor zoo*: *IÉSEG working paper series 2018-ACF-03*.

Jadhav, S., He, H., & Jenkins, K. (2018). Information gain directed genetic algorithm wrapper feature selection for credit rating. *Applied Soft Computing*, *69*, 541–553.

John, G. H., Kohavi, R., & Pfleger, K. (1994). Irrelevant features and the subset selection problem. In *Machine learning proceedings 1994* (pp. 121–129). Elsevier.

Kara, Y., Boyacioglu, M. A., & Baykan, Ö. K. (2011). 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*(5), 5311–5319.

Kim, K.-j. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, *55*(1–2), 307–319.

Kim, K.-j., & Han, I. (2000). Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert Systems with Applications*, *19*(2), 125–132.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arxiv preprint arxiv:1412.6980.

Kofman, P., & Sharpe, I. G. (2003). Using multiple imputation in the analysis of incomplete observations in finance. *Journal of Financial Econometrics*, *1*(2), 216–249.

Kozak, S., Nagel, S., & Santosh, S. (2020). Shrinking the cross-section. *Journal of Financial Econometrics*, *135*(2), 271–292.

Kwon, Y.-K., & Moon, B.-R. (2007). A hybrid neurogenetic approach for stock forecasting. *IEEE Transactions on Neural Networks*, *18*(3), 851–864.

Laloux, L., Cizeau, P., Bouchaud, J.-P., & Potters, M. (1999). Noise dressing of financial correlation matrices. *Physical Review Letters*, *83*(7), 1467.

Lee, M.-C. (2009). Using support vector machine with a hybrid feature selection method to the stock trend prediction. *Expert Systems with Applications*, *36*(8), 10896–10904.

Long, W., Lu, Z., & Cui, L. (2019). Deep learning-based feature engineering for stock price movement prediction. *Knowledge-Based Systems*, *164*, 163–173.

Lv, D., Yuan, S., Li, M., & Xiang, Y. (2019). An empirical study of machine learning algorithms for stock daily trading strategy. *Mathematical Problems in Engineering*, *2019*.

Moghaddam, A. H., Moghaddam, M. H., & Esfandyari, M. (2016). Stock market index prediction using artificial neural network. *Journal of Economics, Finance and Administrative Science*, *21*(41), 89–93.

Mullainathan, S., & Spiess, J. (2017). Machine learning: an applied econometric approach. *Journal of Economic Perspectives*, *31*(2), 87–106.

Nakano, M., Takahashi, A., & Takahashi, S. (2018). Bitcoin technical trading with artificial neural network. *Physica A. Statistical Mechanics and its Applications*, *510*, 587–609.

Nayak, A., Pai, M. M., & Pai, R. M. (2016). Prediction models for indian stock market. *Procedia Computer Science*, *89*, 441–449.

Nazário, R. T. F., e Silva, J. L., Sobreiro, V. A., & Kimura, H. (2017). A literature review of technical analysis on stock markets. *The Quarterly Review of Economics and Finance*, *66*, 115–126.

Nelson, D. M., Pereira, A. C., & de Oliveira, R. A. (2017). Stock market's price movement prediction with lstm neural networks. In *2017 international joint conference on neural networks (IJCNN)* (pp. 1419–1426). IEEE.

Nikou, M., Mansourfar, G., & Bagherzadeh, J. (2019). Stock price prediction using deep learning algorithm and its comparison with machine learning algorithms. *Intelligent Systems in Accounting, Finance and Management*, *26*(4), 164–174.

Nobi, A., Maeng, S. E., Ha, G. G., & Lee, J. W. (2013). Random matrix theory and cross-correlations in global financial indices and local stock market indices. *Journal of the Korean Physical Society*, *62*(4), 569–574.

de Oliveira, F. A., Nobre, C. N., & Zárate, L. E. (2013). Applying artificial neural networks to prediction of stock price and improvement of the directional prediction index–case study of petr4, petrobras, brazil. *Expert Systems with Applications*, *40*(18), 7596–7606.

Ozbayoglu, A. M., Gudelek, M. U., & Sezer, O. B. (2020). Deep learning for financial applications: A survey. *Applied Soft Computing*, Article 106384.

Paiva, F. D., Cardoso, R. T. N., Hanaoka, G. P., & Duarte, W. M. (2019). Decision-making for financial trading: A fusion approach of machine learning and portfolio selection. *Expert Systems with Applications*, *115*, 635–655.

Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015a). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications*, *42*(1), 259–268.

Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015b). Predicting stock market index using fusion of machine learning techniques. *Expert Systems with Applications*, *42*(4), 2162–2172.

Peng, Y., Albuquerque, P. H. M., do Nascimento, I. F., & Machado, J. V. F. (2019). Between nonlinearities, complexity, and noises: an application on portfolio selection using kernel principal component analysis. *Entropy*, *21*(4), 376.

Peng, Y., & Nagata, M. H. (2020). An empirical overview of nonlinearity and overfitting in machine learning using covid-19 data. *Chaos, Solitons & Fractals*, *139*, Article 110055.

Pudil, P., Novovičová, J., & Kittler, J. (1994). Floating search methods in feature selection. *Pattern Recognition Letters*, *15*(11), 1119–1125.

Qiu, M., Song, Y., & Akagi, F. (2016). Application of artificial neural network for the prediction of stock market returns: The case of the japanese stock market. *Chaos, Solitons & Fractals*, *85*, 1–7.

Ramezanian, R., Peymanfar, A., & Ebrahimi, S. B. (2019). An integrated framework of genetic network programming and multi-layer perceptron neural network for prediction of daily stock return: An application in tehran stock exchange market. *Applied Soft Computing*, *82*, Article 105551.

Ravi, V., Pradeepkumar, D., & Deb, K. (2017). Financial time series prediction using hybrids of chaos theory, multi-layer perceptron and multi-objective evolutionary algorithms. *Swarm and Evolutionary Computation*, *36*, 136–149.

Reunanen, J. (2003). Overfitting in making comparisons between variable selection methods. *Journal of Machine Learning Research*, *3*(Mar), 1371–1382.

Sanderson, R., & Lumpkin-Sowers, N. L. (2018). Buy and hold in the new age of stock market volatility: A story about etfs. *International Journal of Financial Studies*, *6*(3), 79.

Sensoy, A., Yuksel, S., & Erturk, M. (2013). Analysis of cross-correlations between financial markets after the 2008 crisis. *Physica A. Statistical Mechanics and its Applications*, *392*(20), 5027–5045.

Sezer, O. B., Gudelek, M. U., & Ozbayoglu, A. M. (2020). Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied Soft Computing*, *90*, Article 106181.

Shynkevich, Y., McGinnity, T. M., Coleman, S. A., Belatreche, A., & Li, Y. (2017). Forecasting price movements using technical indicators: Investigating the impact of varying input window length. *Neurocomputing*, *264*, 71–88.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(1), 1929–1958.

StockCharts (2019). Technical indicators and overlays. https://stockcharts.com/school/doku.php?id=chart_school:technical_indicators.

Tay, F. E., & Cao, L. (2001). Application of support vector machines in financial time series forecasting. *Omega*, *29*(4), 309–317.

Thawornwong, S., Enke, D., & Dagli, C. (2003). Neural networks as a decision maker for stock trading: a technical analysis approach. *International Journal of Smart Engineering System Design*, *5*(4), 313–325.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, 267–288.

Ticknor, J. L. (2013). A bayesian regularized artificial neural network for stock market forecasting. *Expert Systems with Applications*, *40*(14), 5501–5506.

Trading Technologies (2019). X_Study list of technical indicators. www.tradingtechnologies.com/help/x-study/technical-indicator-definitions/list-of-technical-indicators.

TradingView (2019). Indicators and overlays. https://www.tradingview.com/wiki/Category:Indicators_and_overlays.

Vanstone, B., & Finnie, G. (2010). Enhancing stockmarket trading performance with anns. *Expert Systems with Applications*, *37*(9), 6602–6610.

Vijh, M., Chandola, D., Tikkiwal, V. A., & Kumar, A. (2020). Stock closing price prediction using machine learning techniques. *Procedia Computer Science*, *167*, 599–606.

Weng, B., Ahmed, M. A., & Megahed, F. M. (2017). Stock market one-day ahead movement prediction using disparate data sources. *Expert Systems with Applications*, *79*, 153–163.

Xue, Y., Tang, Y., Xu, X., Liang, J., & Neri, F. (2021). Multi-objective feature selection with missing data in classification. arxiv preprint arxiv:2104.08747.

Xue, Y., Xue, B., & Zhang, M. (2019). Self-adaptive particle swarm optimization for large-scale feature selection in classification. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *13*(5), 1–27.

Xue, B., Zhang, M., Browne, W. N., & Yao, X. (2015). A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation*, *20*(4), 606–626.

Yaohao, P., & Albuquerque, P. H. M. (2019). Non-linear interactions and exchange rate prediction: Empirical evidence using support vector regression. *Applied Mathematical Finance*, *26*(1), 69–100.

Yu, L., Chen, H., Wang, S., & Lai, K. K. (2009). Evolving least squares support vector machines for stock market trend mining. *IEEE Transactions on Evolutionary Computation*, *13*(1), 87–102.

Żbikowski, K. (2015). Using volume weighted support vector machines with walk forward testing and feature selection for the purpose of creating stock trading strategy. *Expert Systems with Applications*, *42*(4), 1797–1805.

Zhu, H., Jiang, Z.-Q., Li, S.-P., & Zhou, W.-X. (2015). Profitability of simple technical trading rules of chinese stock exchange indexes. *Physica A. Statistical Mechanics and its Applications*, *439*, 75–84.