Expert Systems With Applications 135 (2019) 60–70

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Technical analysis and sentiment embeddings for market trend prediction

Andrea Picasso[a], Simone Merello[a], Yukun Ma[b], Luca Oneto[a,*], Erik Cambria[b]

[a] DIBRIS, University of Genova, Via Opera Pia 11A, Genova I-16145, Italy
[b] School of Computer Science and Engineering, Nanyang Technological University, 50 Nanyang Ave, Singapore

ARTICLE INFO

Article history:
Received 9 November 2018
Revised 6 June 2019
Accepted 6 June 2019
Available online 7 June 2019

Keywords:
Technical analysis
Sentiment embeddings
Market trend prediction
Supervised learning
Time series analysis
Sentiment analysis

ABSTRACT

Stock market prediction is one of the most challenging problems which has been distressing both researchers and financial analysts for more than half a century. To tackle this problem, two completely opposite approaches, namely technical and fundamental analysis, emerged. Technical analysis bases its predictions on mathematical indicators constructed on the stocks price, while fundamental analysis exploits the information retrieved from news, profitability, and macroeconomic factors. The competition between these schools of thought has led to many interesting achievements, however, to date, no satisfactory solution has been found. Our work aims to combine both technical and fundamental analysis through the application of data science and machine learning techniques. In this paper, the stock market prediction problem is mapped in a classification task of time series data. Indicators of technical analysis and the sentiment of news articles are both exploited as input. The outcome is a robust predictive model able to forecast the trend of a portfolio composed by the twenty most capitalized companies listed in the NASDAQ100 index. As a proof of real effectiveness of our approach, we exploit the predictions to run a high frequency trading simulation reaching more than 80% of annualized return. This project represents a step forward to combine technical and fundamental analysis and provides a starting point for developing new trading strategies.

1. Introduction

Stock market prediction is a challenging problem to solve and its complexity is strictly related to multiple factors which could affect price changes. Researchers and practitioners coming from different fields have taken the challenge, so that research units composed of mathematicians, data scientists, philosophers, and financial analysts are widely common. The heterogeneity of this environment has led to important steps forward the market theory. In fact, two theoretical hypotheses have been built to explain the market behavior: Efficient Market Hypothesis (EMH) and Adaptive Market Hypothesis (AMH).

The EMH (Fama, 1991) states that the current market price fully reflects all the recently published news. This results in the past and current information being immediately incorporated into stock prices. Thus, price changes are merely due to new information or news, and independent of existing information. Since news is unpredictable in nature, in theory, stock prices should follow a random walk pattern and the best bet for the next price is the current price. In practice, the EMH states that it is not possible to 'beat the market' because stocks are always traded at their fair value, thus, buying of undervalued stocks or selling them for exaggerated prices should be impossible. However, the AMH (Lo, 2004) tries to connect the rational EMH, with the irrational behavioral finance principles. The AMH applies the principles of evolution and behavior to financial interactions. Behavioral finance attempts to explain stock market anomalies through psychology-based theories. According to AMH, it is possible to exploit weaknesses in the market efficiency to obtain positive returns from a portfolio of stocks.

Another important step in the understanding of the market behavior is the Dow theory (Rhea, 1993). It states that the movements of the market price are organized in trends, specifically three different kinds of trend, depending on their relevance. Thus, practitioners have developed techniques to predict market trends which have resulted in the birth of two different schools of thought: technical and fundamental analysis.

Technical analysts believe price is able to exhaustively explain the market movements; hence, their strategies were based on the stock price and mathematical indicators computed on it, like RSI,

* Corresponding author.
E-mail addresses: andrea.picasso@smartlab.ws (A. Picasso), simone.merello@smartlab.ws (S. Merello), mayu0010@e.ntu.edu.sg (Y. Ma), luca.oneto@unige.it (L. Oneto), cambria@ntu.edu.sg (E. Cambria).

https://doi.org/10.1016/j.eswa.2019.06.014
0957-4174/© 2019 Elsevier Ltd. All rights reserved.

MACD, and Bollinger bands. They performed time series analysis by extracting technical patterns from Candlestick charts and exploiting linear methods like Box-Jenkins auto-regressive integrated moving average (ARIMA) (Box & Jenkins, 1970), which was one of the popular models in time series forecasting. As a subsequent step, the Hurst exponent (Hurst, 1951), a statistical measure used to classify time series, was proved to be useful in understanding the market behavior. In fact, as Qian and Rasheed (2004) discovered, when the Hurst exponent was computed on price values, it provided a measure for trend predictability.

Nowadays, with the rising power of machine learning and deep learning prediction tools, this process has moved from the hands of financial analysts to those of data scientists. Zhang et al. proposed a hybrid model, based on ARIMA and neural networks, for time series forecasting (Zhang, 2003) and machine learning techniques have been applied to market prediction in many research works (Hu, Liu, Bian, Liu, & Liu, 2018; Oancea & Ciucu, 2014; Yao, Tan, & Poh, 1999). Huang, Nakamori, and Wang (2005) have exploited a support vector machine (SVM) to forecast the stock market direction by using a small dataset made up of 676 pairs of observations, achieving an accuracy of nearly 70%. We believe that increasing the dimension of the dataset could have led to more trustworthy performances, as a small dataset limits the generalization of the model. Other researchers have fed a bigger dataset inside a neural network architecture but with the goal of predicting only a specific index of the market (Cristianini & Shawe-Taylor, 2000; Yao et al., 1999). Yao et al. (1999) in their work, have developed a model to forecast only a single index, the Kuala Lumpur Stock Exchange using a dataset of around 2000 samples.

While in technical analysis the strategies are based only on the price time series of a stock, in fundamental analysis (Abarbanell & Bushee, 1998), trading decisions are taken in relation with company's financial conditions and macroeconomic indicators like EBITDA, P/E, income, return on equity, and dividend yield. Therefore, fundamental analysts buy/sell stocks when the intrinsic value is greater/lower than the market price; even though, the proponents of EMH argue that the intrinsic value of a stock is always equal to its current price (Bandy, 2007).

Nowadays, even in this field, machine learning and data science are growing in importance and the outcome of this process is the sentiment analysis application to financial market. Sentiment analysis aims to automatically extract the sentiment from various sources of information like text (Li et al., 2017; Ma, Peng, & Cambria, 2018), images (You, Luo, Jin, & Yang, 2015) and videos (Hazarika et al., 2018; Poria, Cambria, Bajpai, & Hussain, 2017).

The sentiment information extracted from text as an embedding has been leveraged by Peng and Jiang (2015) in a neural network model to predict market movements and it has proven to be effective. Luss and D'Aspremont (2015) applied multiple kernel learning to combine information coming from equity returns with news text. They observed that while the direction of returns is not predictable using either text or returns, their size is predictable. Following their achievements, we have evaluated the performance of our model along different sizes of trend changes, by splitting test samples in different buckets according to the size of the related change in the price. Various textual sources have been exploited to infer the market sentiment and produce predictions, starting from news (Schumaker & Chen, 2006; 2009) and financial blogs (De Choudhury, Sundaram, John, & Seligmann, 2008; Oh & Sheng, 2011) to tweets (Bollen, Mao, & Zeng, 2011; Mittal & Goel, 2012; Rao & Srivastava, 2012; Si et al., 2013).

Recently, the work of Wu, Su, Yu, and Chang (2012) has reported an increase in performance when both news and technical information were exploited in a regression problem. We consider their findings as a starting point for deeper investigations.

Our project aims to develop a robust model able to predict future market trends by exploiting information coming from price time series and sentiment. This will allow technical and fundamental analysts to work together and enhance performances in stock market prediction.

To tackle this problem, as Picasso et al. (2018), we worked on a portfolio of stocks, the twenty most capitalized stocks listed in the NASDAQ100, to avoid liquidity problems during the high frequency trading simulation.

Moreover, we believe in the importance of working on more tickers to prove the statistical relevance of our approach. When working on time series forecasting, it is frequent to work with unbalanced labels. Following the guidelines of Amin et al. (2016) and Picasso et al. (2018), our approach aims to solve this issue. Starting from their work, we have improved the balancing technique through the use of the Geometric Score during model selection. We applied three different models of increasing complexity, namely Random Forest (RF), Support Vector Machines (SVM), and a feed forward Neural Network (NN). A financial benchmark was used to compare the predictors so that their effectiveness was demonstrated in a real trading scenario. Our evaluation was divided in two steps, because, as Xing, Cambria, and Welsch (2018) stated, the evaluation of a data science model might be difficult when applied to the financial domain. The first step took into account metrics more related to the machine learning field to understand the statistical behavior of the model. In the second step, we evaluated the model through the results coming from a trading simulation, which is based on the model predictions, using return, max drawdown, and sharpe ratio as performance indicators. The use of the two steps evaluation makes our approach able to overcome issues and biases related to previous works. With the first step, the power of the proposed classifier is proved. With the second step, the real effectiveness of our predictions is demonstrated. The whole pipeline of our project is depicted in Fig. 1 to make the experiments flow clearer. During the experimental phase, our model proved to be effective both from a statistical and financial evaluation, achieving an annualized return of 85.2% on the portfolio under study.

The remainder of the paper is organized as follows: Section 2 introduces the research question; Section 3 underlines the novelties in our approach to such a question; Section 4 describes available datasets; Section 5 reports the settings and the results for each of our experiments; finally, Section 6 points out the conclusion and future works.

2. Problem formalization

The statistical problem to be solved to forecast the market trend is an auto-recursive classification problem. The input for the classifier X is defined as a sequence of vectors:

$$X = [\mathbf{x}(0), \mathbf{x}(1), .., \mathbf{x}(n-1), \mathbf{x}(n)]$$

where n denotes the number of samples. By selecting a generic sample $\mathbf{x}(t) \in \mathbb{R}^F$ with F as the number of features and t the time stamp of the sample, it can be decomposed in:

$$\mathbf{x}(t) = [x(t)_0, x(t)_1, .., x(t)_F]$$

The target of the classification problem is defined as a sequence of labels:

$$\mathbf{y} = [y(0), y(1), .., y(n)]$$

of the same length of X such that every element of $\mathbf{y} \in \{0, 1\}$.

In the computation of $y(t)$ we introduced the hyper-parameter w which fixes the size of the trend to be classified. Thus, the classification process, given the input X, aims to distinguish between

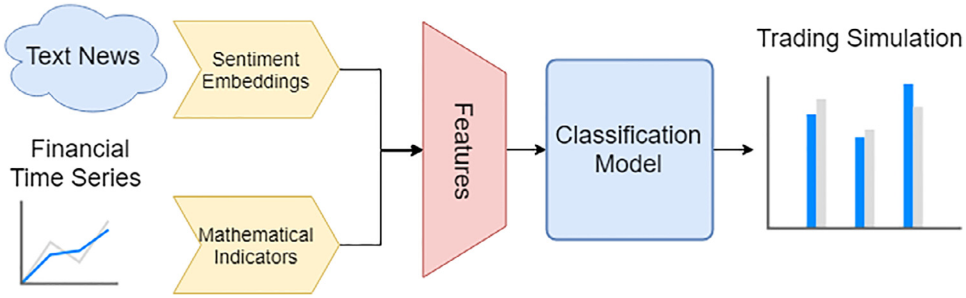

Fig. 1. Data Pipeline.

two classes: positive associated with $y(t) = 1$ and negative associated with $y(t) = 0$. In particular, we consider positive a sample which represents an increase in the closing price pc between time t and $t + w$ and negative the opposite case.

$$\begin{cases} y(t) = 0 & \text{if negative trend in}[t, t + w], \\ y(t) = 1 & \text{if positive trend in}[t, t + w] \end{cases}$$

The mathematics involved in the computation of the explained label is now reported. The step function $\mathbb{1}$ was applied to the price delta moving the result from $\mathbb{R} \to \{0, 1\}$ so that $y(t)$ was computed as:

$$y(t) = \mathbb{1}(pc(t + w) - pc(t))$$

where $pc(t)$ denotes the closing price of the selected stock at time t and w represents the length of the trend to be predicted.

3. Methodology

3.1. Data preprocessing

In this research work, the auto-recursive classification problem previously explained was adopted as approach to forecast the future market trend. Each component x belonging to the vector $\mathbf{x}$ of the input sequence X was normalized such that $x \in \mathbb{R} \to x' \in [0, 1]$ to speed up the convergence. In particular, Min-Max norm was applied following the suggestion of Al Shalabi, Shaaban, and Kasasbeh (2006). Min-Max normalization is a simple technique where the data can be pushed into a pre-defined boundary $[C, D]$ (Patro & Sahu, 2015).

$$x' = \frac{x - min(\mathbf{x})}{max(\mathbf{x}) - min(\mathbf{x})} * (D - C) + C$$

where x' denotes the Min-Max normalized value of x. The sets of features exploited as input in the classification task come from two different sources, namely textual and numerical source. A textual source is composed of the news about the stock under study, while the numerical source refers to its price. Throughout the experiments, the performances of the numerical source ('Price'), the textual source ('News'), and their union ('Price&News') were evaluated and compared. These sets of features were combined to make the most of the sentiment enclosed inside the news together with the trading signals retrieved from the mathematical indicators computed on the price. From textual data, two different sets of features were extracted using the dictionary of Loughran and McDonald (2011) (L&Mc) and AffectiveSpace (Cambria, Fu, Bisio, & Poria, 2015)[1]. In both cases, the news were transformed into sentiment embeddings. The Loughran and McDonald dictionary is specific for financial applications and contains different lists of Constraining, Litigious, Negative, Positive, Uncertainty, Superfluous, and Interesting words. It was chosen because, as Loughran and McDonald (2011) stated, dictionaries developed for other disciplines misclassify common words in financial text. Furthermore, it proved to be effective in many research papers in the financial forecasting field (Jin et al., 2013; Li, Xie, Chen, Wang, & Deng, 2014). On the other hand, AffectiveSpace is a vector space model, built by means of random projection, that allows for reasoning by analogy on natural language concepts. In AffectiveSpace, each concept is mapped to a 100 dimensional vector through a dimensionality reduction of affective common-sense knowledge. This procedure allows semantic features associated with concepts to be generalized and, hence, allows concepts to be intuitively clustered according to their semantic and affective relatedness. Such an affective intuition (because it does not rely on explicit features, but rather on implicit analogies) enables the inference of emotions and polarity conveyed by multiword expressions, thus achieving efficient concept-level sentiment analysis (Cambria et al., 2015). Even if AffectiveSpace is not specific for the financial field, it was chosen because it is able to extract the concept level sentiment from structured texts like news. Both the approaches were applied to extract sentiment embeddings from the text of news summaries. When using the Loughran and McDonald dictionary, the embedding of each news was made up of the counts of Constraining, Litigious, Negative, Positive, Uncertainty, Superfluous, and Interesting words of the dictionary, which were found inside the summary of news. Thus, the news embeddings, published in the same time slot of a quarter of hour, were grouped together and the representative embedding for the slot was computed as the mean between them. An extra feature was included, representing the number of news found in the slot (8 features). In the end, the news related features vector was obtained as the concatenation of the actual embedding with the moving average on the previous samples respectively with a window size of 5,10,15,20 slots. The target of this process was to take into account the influence of the past sentiment, resulting in a 40 (8*5) dimensional vector conveying the sentiment information for the specific stock under study. A similar procedure was done with the embeddings obtained through the SenticNet API[2]. Specifically, the concepts were extracted from news and the representation of each concept was retrieved from AffectiveSpace as a 100 dimensional vector and the embedding of the news was computed as the average within its concepts representation. Eventually, as with the Loughran and McDonald dictionary features, the embedding of a slot was obtained as the mean of the embeddings of the belonging news and the slot representation was obtained as the concatenation of the averages of previous slots (500 dimensional vector). From the price data, retrieved by Google Finance API with a frequency of a quarter of hour (to be aligned with the news slots timing), different technical indicators were calculated using Stockstats library[3] and they were

[1] Publicly available at: https://sentic.net/downloads/.

[2] sentic.net/api.

[3] github.com/jealous/stockstats.

Table 1
Indicators.

Name	Formula
Simple Moving Average $SMA(t, N)$	$\frac{\sum_{k=1}^{N} pc(t-k)}{N}$
Exponential Moving Average $EMA(t, \Delta)$	$(pc(t) - EMA(t-1)) * mult + EMA(t-1)\ mult = \frac{2}{\Delta+1}, \quad \Delta = timeperiodEMA$
$MACD(t)$	$EMA(t, \Delta = 12) - EMA(t, \Delta = 16)$
Relative Strength Index $RSI(t)$	$\frac{100}{1+RS(t)}, \quad RS(t) = \frac{AvgGain(t)}{AvgLoss(t)}$
Bollinger Bands	$UpperBand(t) = 20 * SMA(t, N) + (40 * std(pc))\ MiddleBand(t) = 20 * SMA(t, N)\ LowerBand(t) = 20 * SMA(t)(40 * std(pc))$
Stochastic Oscillator $KDJ(t)$	$100\frac{(pc(t)-MIN(pl)}{MAX(ph)-MIN(pl)}$
True Range $TR(t)$	$MAX(ph(t) - pl(t); ph(t) - pc(t-1); pc(t-1) - pl(t))$
Average True Range $ATR(t)$	$\frac{13*ATR(t-1)+TR(t)}{14}$
Williams Indicator $WR(t)$	$\frac{MAX(ph)-pc(t)}{MAX(ph)-MIN(pl)*(-100)}$
CR indicator $CR(t)$	$100\frac{SMA(ph(t)-MIN(m,ph(t)))}{SMA(m-MIN(m(t),pl(t)))} \quad m(t) = \frac{pl(t)+ph(t)+pc(t)}{3}$

With pc=close price, po=open price, pl=low price, ph=high price and std=standard deviation.

concatenated with the price values. The result was a 111 dimensional vector constructed by the concatenation of the price values and the set of indicators selected. The indicators to be computed coincide with the set used by Picasso et al. (2018) and they were chosen following previous research works (Choudhry & Garg, 2008; Huang, Yang, & Chuang, 2008; Kim & Han, 2000; Mizuno, Kosaka, Yajima, & Komoda, 1998). Their mathematical formulation is reported in Table 1.

The classification task falls under the time series prediction problems and, as commonly happens, it is affected by labels unbalance, caused by the trending tendency of price movements. To solve this weakness in the dataset, a proper balancing technique was adopted to avoid developing a biased classifier able to predict only trends analogue to the ones represented in the training and validation sets. The Synthetic Minority Over-sampling Technique (SMOTE) was specifically chosen because it has been proved to be the most effective from previous research works (Amin et al., 2016; Chawla, Bowyer, Hall, & Kegelmeyer, 2002; Lusa & Blagus, 2010). SMOTE is an over-sampling approach in which the minority class is over-sampled by creating "synthetic" examples (Chawla et al., 2002). The balancing technique was applied separately on the train and validation sets to make their labels respectively balanced. The test set was left unbalanced because the trend to be predicted represents future values of the price which are unavailable for modifications.

Another issue, common in time series prediction tasks, is the dependency between samples inside the dataset. In fact, when mathematical indicators like Simple Moving Average (SMA) (refer to Table 1 for mathematical details) are computed on the price with a window of D elements at time t, two neighbor inputs, $x(t)$ and $x(t+1)$, are no longer independent. During the computation of the SMA feature, defined as $x(t)_i$, the sets of closing price values, $pc(t)$, which are taken into account, differ only in one element. To illustrate this concept the computation for $x(t)_i$ and $x(t+1)_i$ is given below:

$$x(t)_i = \frac{\sum_{k=1}^{D} pc(t-k)}{D} \tag{1}$$

$$x(t+1)_i = \frac{\sum_{k=1}^{D} pc(t+1-k)}{D} \tag{2}$$

Because of the overlapping sets prevent the use of k-fold cross validation, thus it is not possible to shuffle the data and randomly pick up train and validation portions. Otherwise the samples from the validation set will be strongly dependent on the training ones. To circumvent this issue, a specific kind of cross-validation technique, called 'increasing window cross-validation' was adopted in Picasso et al. (2018). The splitting of the dataset executed with this technique is reported in Fig. 2.

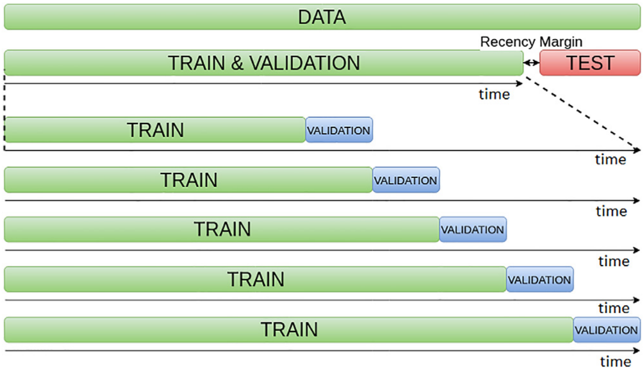

Fig. 2. Increasing-window CrossValidation.

This technique has been proved to be effective in the cross-validation phase of time series problems. Firstly the data is sorted over time, subsequently the train and validation phase is split from the test set so that the test set represents the 'future' of the train set. The training phase is divided in different folds and in each one the training section is increased and the validation set is moved forward in time. The outcome is a training procedure based on folds but without shuffling the samples. Furthermore, the 'increasing window cross-validation' technique adopts a margin between the training-validation and the test set to overcome the recency problems underlined by Yao and Poh (1995). After the split was executed on the dataset, the available samples were treated as input to different classification models to have a comparison between them. Specifically, a Random Forest (RF), a SVM, and a feed forward Neural Network were adopted. In the next subsection, a brief description of the applied models is reported to have a better understanding of the experimental phase.

3.2. Models

The first model experimented was a RF in order to have a benchmark and obtain some insights on the trend prediction task. RF is an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and predicting the class which represents the mode of the classes. A decision tree is a structure where each node represents a feature, each link represents a decision, and each leaf represents a label. In this work, RF was pruned through the Gini impurity metric. Gini impurity is a measure of how often a randomly chosen element from the set would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset. To compute Gini impurity for a set of items with J classes, suppose $i \in \{1, 2, \ldots, J\}$, and let p_i be the fraction of items labeled with class i in the set.

$$I_G(p) = 1 - \sum_{i=1}^{J} p_i^2$$

The RF, although simple, was chosen because of its significant performance when applied to classification tasks (Khaidem, Saha, & Dey, 2016; Kumar & Thenmozhi, 2006).

As a second step, a SVM (Cristianini & Shawe-Taylor, 2000) was exploited in the trend prediction task, specifically, a SVM with Gaussian kernel (KSVM), which has achieved interesting performance in previous research works (Choudhry & Garg, 2008; Huang et al., 2005). The SVM is also called maximum margin classifier because it tries to maximize the margin between the decision boundaries. An SVM classifier is created through the optimization of the following objective function:

$$\left[\frac{1}{n}\sum_{i=1}^{n}\max(0, 1 - y(i)(\mathbf{w} * \mathbf{x}(i) - b)\right] + \lambda||\mathbf{w}||^2$$

where $\mathbf{w}$ denotes the weights and b the bias. The first part of the objective is a hinge-loss function, where y_i is the true label (-1 or 1 in our setting) (Gentile & Warmuth, 1999) whereas the second term is a L-2 norm which controls the dimensionality of the margin.

In addition to only performing a linear classification, SVM can also efficiently perform a non-linear classification with the use of a non-linear kernel function used to map the inputs $\mathbf{x}$ into a new feature space $\mathbf{x} \to \tilde{\mathbf{x}}$. The SVM performs better than other classification models because it is designed to minimize the structural risk, while alternative techniques are based on minimization of empirical risk. In other words, SVM seeks to minimize an upper bound of the generalization error rather than minimizing training error. Hence, it is less vulnerable to the over-fitting problem. Furthermore, the solution of the optimization problem is unique and absent from local minima.

As a last step, a subsymbolic approach was experimented through the use of a feed forward neural network. Neural networks are models composed by a stack of different layers. Each unit of a layer (neuron) applies a non-linearity to a linear combination of its input. The neuron itself is not powerful but, once more neurons are combined together in more layers, the network becomes both an extremely powerful classifier (Ruck, Rogers, Kabrisky, Oxley, & Suter, 1990) and regressor (Cybenko, 1989). The training is executed through the use of the back-propagation algorithm (Rumelhart, Hinton, & Williams, 1986) with the target to minimize a predefined objective function, this process is not assured to reach an optimal minimum but it has been empirically proved to reach one of the best local minima of the objective. In this work, the proposed network, whose structure is reported in Fig. 3, is divided in four layers. The number of neurons in the various layers is decreasing; starting from the first one with Ne neurons, the second with $\frac{Ne}{2}$, the third with $\frac{Ne}{4}$, to the last one with only one neuron. The Adam optimizer (Kingma & Ba, 2014) is exploited to minimize the binary cross entropy $H_{y'}(y)$ objective:

$$H_y(y') = -\sum_{i=1}^{n}(y_i * log(y_i') + (1 - y_i) * log(1 - y_i'))$$

where y_i' and y_i denote respectively predicted and truth labels.

In each layer, a batch normalization (Ioffe & Szegedy, 2015) was inserted to avoid the covariate shift (Shimodaira, 2000) and speed up the training phase. Moreover, a Leaky Relu was adopted as activation function in each layer with the exception of the last one where a sigmoid function is used to perform the classification task.

To avoid overfitting and improve the generalization power of the network, Dropout (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014) was applied on the internal/hidden output. Since it has the side effect of increasing the values of the weights $\mathbf{w}$, following the suggestion of Srivastava et al., two different type of regularization were performed. In particular, a max norm regularization (Srebro, Rennie, & Jaakkola, 2005) was applied on the weights $\mathbf{w}$ and a L2-norm regularization was performed on the hidden output of the dense layers. The fine tuning of the network structure was carried out with the observation of the learning curves and the definitive structure is reported in Fig. 3.

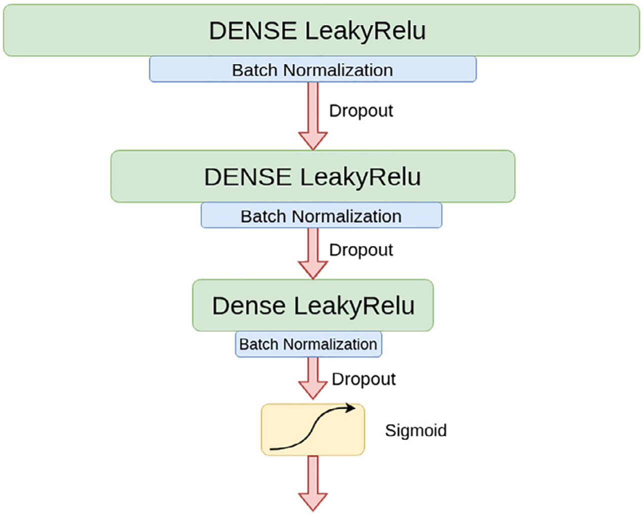

Fig. 3. Feed forward neural network structure.

3.3. Model selection and evaluation

The goal of this work is to develop an approach that is able to perform trend prediction on a portfolio of stocks. Thus, every result and metrics considered was computed as the mean of the results achieved on each of the tickers belonging to the portfolio under study. During training and validation phase each model was trained more times, following the previously explained cross-validation technique and different parameters were experimented. Subsequently, a model selection step was carried out to adopt the best parameters according to the training and validation sets. Before illustrating model selection, it is important to recall that the presence of unbalanced labels is a typical issue of market time series forecasting. In fact, the market during the period under study can be mostly positive (bullish) or negative (bearish). Thus, the main target during this phase is to choose a model which is able to make correct predictions both on positive and negative samples and not only achieve a high directional accuracy (correct classification of the direction of the market: positive $y = 1$ or negative $y = 0$). To reach this target, during model selection, the Geometric score metric was introduced. The Geometric-score, also called G-score, takes into account both recall on positive and negative samples balancing the proportion between them. To have a better understanding of its power, the G-score calculation is reported:

$$G - score = \sqrt{TPR * TNR}$$

As the formula shows, to achieve a high value of G-score metrics the *TPR*, True Positive Rate, and *TNR*, the True Negative Rate, must be high and balanced. Thus, the model selected in this phase is able to make correct predictions both in a bullish and bearish market. Summarizing, this metric was adopted to improve the generalization and the predictive power of our model on both the positive and negative trends, removing the bias coming from the distribution of the training set.

Table 2
Details of buckets used for evaluation computed on the Test set.

Bucket	Min value	Max value	Number of samples	Imbalance Pos/neg
1	0%	2.0%	9217 (45.4%)	55.4%
2	2.0%	4.1%	6009 (29.6%)	73.0%
3	4.1%	6.1%	3028 (14.9%)	73.6%
4	6.1%	8.1%	1384 (6.8%)	68.6%
5	8.1%	10.2%	654 (3.3%)	63.4%

Values are averaged among the stocks. Min value, Max value: boundaries of market fluctuations for each bucket.

Once the predictive model is developed, it has to be evaluated on out-of-sample data to understand its real performance. As pointed out by Frank et al., when evaluating a data science model applied to the financial world, considering only statistical metrics is not exhaustive (Xing et al., 2018). Our proposal is a combined and innovative evaluation phase divided in two steps. The first involves the computation of data science metrics, namely directional accuracy and recall values, but the approach to their computation is original. In fact, before the computation of the metrics values, the labels belonging to the test set were divided into five buckets. The first bucket was filled with the input related with the labels representing the smallest changes in the market price. The other four buckets were filled with samples representing increasing percentage changes in the market price. More details about the elements belonging to each bucket are reported in Table 2.

The goal of this 'bucketization' process is to understand the behavior of the model with the different predicted trends. In fact, achieving high performance on the 'most relevant' samples, i.e., the ones representing big changes in the market price, is more valuable than obtaining the same results on the less relevant samples. This is due to the fact that, from a trader's perspective, the returns coming from a small delta in the price are often nullified by the transaction cost, while the biggest deltas in the market price are very valuable and they produce positive returns, even after the deduction of the transaction cost. In particular, the samples were split in a linear manner following the subsequent rule. Defined $\Delta_i = pc(t_i + w) - pc(t_i)$ the changes of the market for the sample i and $\Delta_{\max}$, $\Delta_{\min}$ respectively the maximum and minimum price delta of the test set; sample i belongs to bucket $j \in \mathbb{N}$ with $j \in [1;5]$ if:

$$step * (j-1) < \Delta_i \leq step * (j+1)$$

where, using five buckets, *step* is defined as:

$$step = \frac{\Delta_{max} - \Delta_{min}}{5}$$

Thus, in the first step of the model evaluations, accuracy and recall values were computed for each bucket. Upon the execution of the first step, the second step was to run a trading simulation to examine the performance of the predictions generated from the model.

The trading strategy was developed according to a trading threshold *TSH* on the neural network output,which is based on empirical findings obtained via the observation of the predictions generated.

$$\begin{cases} \boldsymbol{BUY} & if prediction > 0.5 + TSH, \\ \boldsymbol{SELL} & if prediction < 0.5 - TSH \end{cases}$$

With the use of a threshold we aim to trade only when our model prediction has a certain confidence, quantified by the value of the sigmoid output, and the sample predicted represents a significant change in the market price. In fact, when trading on samples representing small changes in the market price, the gain is nullified by the transaction cost.

The process to determine the value of the threshold is now explained. Firstly, the predictions on the validation sets were plotted to have insights on their distribution. Subsequently, the value of the threshold was set to trade only when the predictions were enough 'reliable' according to the following computation. Given the distribution of the prediction as the probability mass function (discrete probability distribution) $p(i) : \mathbb{N} \to \mathbb{N}$ divided in 100 bins of size 0.01 in the interval [0,1] and *nv* as the number of validation samples, the $TSH \in \mathbb{R}$ is computed as:

$$TSH = \frac{ni}{100} \quad s.t. \quad \sum_{i=1}^{ni} (p(i) + p(100 - i)) = 0.75 * nv$$

where *ni* is used as integer index in the probability mass function. This computation allows us to consider the 75% of our predictions which are the most 'reliable' according to the sigmoidal output.

The BackTrader[4] python library was used to run the simulations and the annualized return, the sharpe ratio, and the maximum drawdown were computed as performance metrics. The annualized return, *aR*, is the geometric average amount of money earned by an investment each year over a given time period. It is calculated as a geometric average to show what an investor would earn over a period of time if the annual return was compounded:

$$aR = (1 + R)^{\frac{365}{d}} - 1$$

where *R* denotes the cumulative return during the period of *d* days under consideration. The annualized return provides only a snapshot of an investment performance and does not give investors any indication of its volatility. Thus, to have an insight of the volatility on the portfolio value, sharpe ratio and maximum drawdown were exploited. The sharpe ratio, *Sr*, is the average return earned in excess of the risk-free rate per unit of volatility or total risk, represented by the portfolio standard deviation σ_p.

$$Sr = \frac{R - r}{\sigma_p}$$

Subtracting the risk-free rate *r* from the return *R*, the performance associated with risk-taking activities can be isolated. One intuition of this calculation is that a portfolio engaging in "zero risk" investment, such as the purchase of U.S. Treasury bills, for which the expected return is the risk-free rate, has a sharpe ratio of exactly zero. On the other hand, the maximum drawdown (MDD) is the maximum loss from a peak to a trough of a portfolio, before a new peak is attained. The MDD is an indicator of downside risk over a specified time period.

The use of the two steps evaluation makes our approach able to overcome the issues and biases related to previous works. In fact, with the first step, the power of the classifier proposed is proved and, with the second step, the real effectiveness of our predictions is demonstrated.

4. Available data

The dataset used to perform the market trend prediction regards the NASDAQ100 index. It is made up of twenty tickers to make the results of this research statistically significant. In particular, the twenty most capitalized stocks were chosen to avoid liquidity problems during the trading simulation. The time span of the dataset is from 03/07/2017 to 14/06/2018 and the list of the twenty stocks under study with related capitalization is reported in Table 3. For each ticker, the news, related to the specific stock, were retrieved from Intrinio API,[5] while the time series of the price values, with a frequency of 15 min, were downloaded from Google Finance API.[6] The time span of our dataset is limited due to the limitations imposed by the free version Intrinio API. Thus, even if

[4] backtrader.com.

[5] intrinio.com.

[6] pypi.org/project/googlefinance.client.

Table 3
Stocks under study.

Stock	Ticker	Capitalization	Gain
Apple Inc.	AAPL	$1026.62B	32.96%
Amazon.com Inc	AMZN	$923.45B	80.76%
Google	GOOGL	$868.65B	28.20%
Microsoft Corporation	MSFT	$830.73B	48.78%
FaceBook Inc	FB	$519.84B	32.59%
Intel Corporation	INTC	$225.78B	65.99%
Cisco Syestems Inc.	CSCO	$205.75B	42.64%
Comcast Corporation	CMCSA	$160.55B	−15.75%
Pepsico Inc	PEP	$159.85B	−8.73%
NVIDIA Corporation	NVDA	$155.46B	91.57%
Netflix Inc	NFLX	$148.63B	168.78%
Amgen Inc	AMGN	$126.95B	7.52%
Adobe Systems Incorporated	ADBE	$124.5B	86.47%
Texas Instrument Incorporated	TXN	$107.12B	49.20%
PayPal Holdings Inc	PYPL	$102.99B	61.93%
Gilead Sciences Inc	GILD	$100.01B	0.64%
Costco Wholesale Corporation	COST	$96.3B	29.15%
QUALCOMM Incorporated	QCOM	$95.26	8.21%
Broadcom Inc	AVGO	$91.12B	16.84%
Booking Holdings Inc	BKNG	$89.33B	15.19%

Gain: Fluctuation of stock's value during the timespan of the experiment.

the finance industry is not lacking prices historical data, our experiment is constrained into the reported period because of the news data source.

The Google Finance data is constituted by the open, close, mid, high, low price, and volume values without missing samples. On the other hand, the Intrinio API's dataset comprises of news coming from press agency like Reuters or Bloomberg and the publications frequency is not as regular as the price, thus missing samples are possible. To make the dataset continuous, the sample representing the news in the previous 15 minutes slot is replicated in the next one in the absence of published news for those 15 minutes. This process follows the rule of thumb according to which the sentiment is persistent along the time if there is not new information. Since the price values are available only during the opening time of the market, overnight news, published after the closure of the trading session, are collapsed in the first slot of the next trading day. The embedding vectors (explained in Section 3) were constructed on the available datasets and, subsequently, were fed into the models as input.

5. Experiments discussion

In the experimental phase, the approach of our work was developed and the models were implemented with the use of Python. Specifically, the RF and SVM implementation were retrieved from Sklearn, while the neural network structure was built with the use of Keras and Tensorflow as back-end. Three different datasets were created. The first is constituted by the features extracted from the price and the mathematical indicators computed on it. The second is composed by the sentiment embeddings retrieved from the news. Lastly, the third is made up with the combination of the previous two through a concatenation of the features vectors.

In the trend classification task, the length of the trend to be predicted w was set as 140 time steps, which, in 15 min frequency data, represents about one week. This value was chosen according to our previous experiments and the findings of the literature (Merello, Picasso, Oneto, & Cambria, 2019; Tay & Cao, 2001; Thomason, 1999; Xu & Cohen, 2018). As reported in Table 2, during the timespan of the experiment every bucket has a positive/negative samples percentage ratio which is between 55% and 74%, thus the overall trend of the market was bullish. To reduce this imbalance factor, the usage of a proper balancing technique was considered. During the preprocessing phase, which is common to every model, the training and validation set were balanced through the use of SMOTE algorithm with number of neighbors set to 5, following the settings already introduced by Lusa and Blagus (2010) and Chawla et al. (2002). At last, the whole input dataset was normalized with the Min-Max normalization.

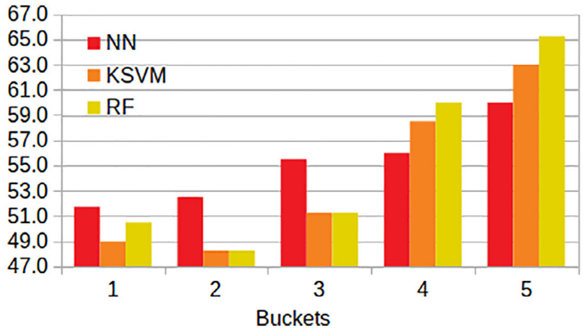

Fig. 4. Accuracy along the buckets with L&Mc dictionary.

After this preliminary phase, the combined dataset (Price&News) of Loughran and McDonald dictionary was fed into the three different models with the subsequent settings to have an initial insight of the effectiveness of our approach. In the RF, the computation of the optimal feature/split combination was based on the Gini Impurity, while the SVM with Gaussian kernel was trained searching for the values of $C = \frac{1}{\lambda}$ and γ in $\{10^{-4}, 10^{-3.5}, \ldots, 10^{6}\}$. Instead, the hyper-parameters explored with the neural network were:

- activity regularization L2 0.01
- kernel regularization max-norm with max_value=1
- dropout 0.5
- number of neurons in the first layer $Ne = [64, 128, 256]$
- learning rate [0.001, 0.0001]
- Adam was used as optimizer with $\epsilon = 10^{-8}, \beta_1 = 0.9, \beta_2 = 0.999$ as suggested by Kingma and Ba (2014)

After the training and validation phase, the models were tested on out-of-sample data and the values of accuracy along the buckets are reported in Fig. 4.

In the initials buckets, the performance of our neural network overcomes the results obtained with the RF and the KSVM. Instead, in the last buckets, the accuracy values for RF and KSVM are higher but, once the recall values for RF and KSVM are computed, they result to be biased. In fact the recall on the positive and negative samples differs for more than 20%; precisely, in the RF, the positive and negative recalls are respectively, 69% and 38%. This is an issue for the generalization of the classifier, which must be taken into account and it makes the values of accuracy not trustworthy. Instead, the NN recall values are much more balanced, thus, to properly evaluate the behavior of the NN classifier, its values of recall were plotted in Fig. 5.

This proves the ability of our NN model to fairly classify both the positive and the negative samples. In fact, the delta between the orange and the red chart, respectively positive and negative recall, is well below 20%, proving the positive effect of the G-score introduction in the NN model selection phase. Thus, even though the labels were unbalanced, the combined use of a proper balancing technique and G-Score metric during the training and validation phase has led to an unbiased model. In addition, from the experimental results, a positive trend in the accuracy values along the buckets is illustrated in every model, which demonstrates the power of our approach to identify the most relevant changes in the market price.

After the evaluation of the Loughran and McDonald dictionary features, a comparison with the use of AffectiveSpace was

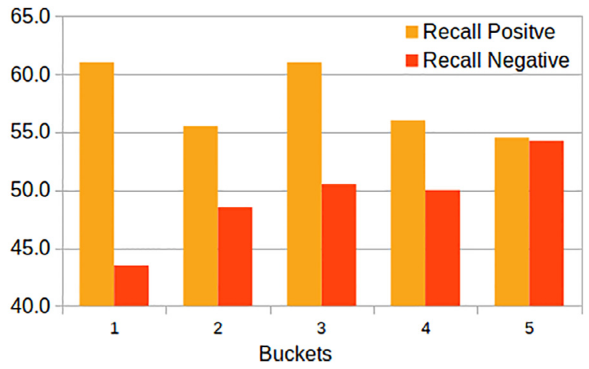

Fig. 5. Recall along the buckets: NN with L&Mc.

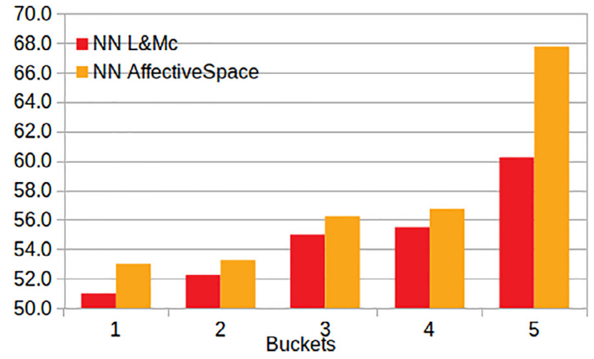

Fig. 6. Accuracy along the buckets: L&Mc vs AffectiveSpace.

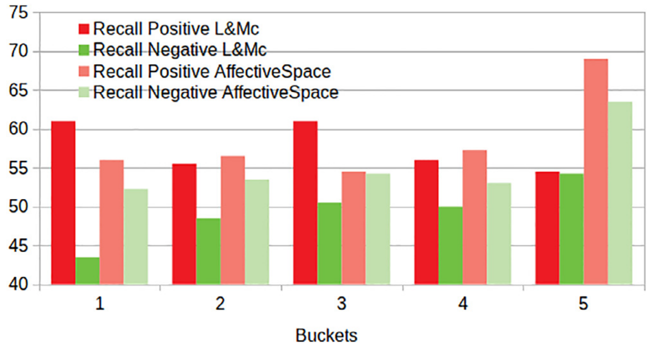

Fig. 7. Recall along the buckets: L&Mc vs AffectiveSpace.

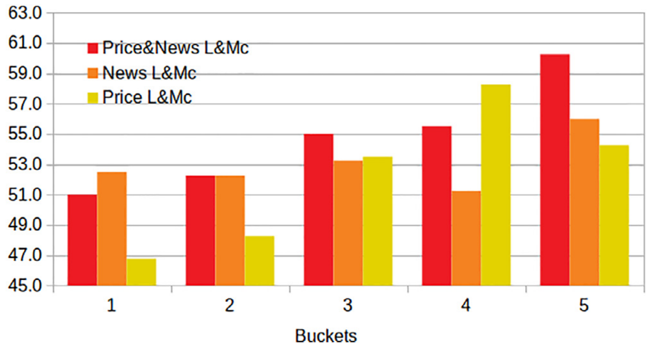

Fig. 8. Accuracy along the buckets with the different features sets L&Mc.

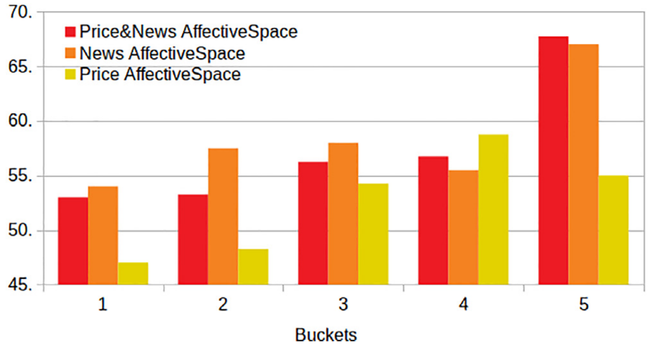

Fig. 9. Accuracy along the buckets with the different features sets AffectiveSpace.

experimented. The NN was trained and validated with the same set of hyper-parameters with the exception of the number of neurons range *Ne*, which was moved from [64,128,256] to [256,512,756]. This is because, following the insights given by Panchal and Panchal (2014), the appropriate number of neurons in a NN is related to the dimensionality of the input vector which, in this case, increases from 151 to 611. Respectively, the dimension 151 is the concatenation of Price and News features obtained with Loughran and McDonald dictionary (111+40) while 611 is related to the concatenation of the Price and News features coming from AffectiveSpace (111+500). Subsequently to the training and validation phase, the model was tested on out-of-sample data. The values of accuracy and recall along the buckets, together with the previous values obtained with Loughran and McDonald dataset, were reported respectively, in Figs. 6 and 7.

The comparison reports a further improvement both in the accuracy values and positive/negative recall delta. Thus, the features extracted with AffectiveSpace API were more effective in representing the sentiment embedded in the text of financial news. This is the proof of the power of AffectiveSpace to extract the concept level sentiment from structured texts.

In a second phase of our experiment, the relation between the different features sets was investigated to understand if including the sentiment in a market trend classification task leads to higher performance. To achieve this goal, the neural network was trained and validated with the same hyper-parameters range as before but on the three different features sets extracted from our data and the results were grouped in two charts. The performance obtained exploiting the Loughran and McDonald and the AffectiveSpace features were showed respectively in Figs. 8 and 9.

From careful evaluation of the plots, an over performance of the Price&News set in comparison with the Price set was found but, in the meanwhile, the over performance of the Price&News set in comparison with the News set was not outstanding. This behavior enlightens that adding the sentiment representation to the price features leads to important improvements in the performance. On the other hand, these results suggest that, instead of using a simple concatenation, a process of features fusion between the Price and News sets would allow the model to exploit even better the representative power of the combined features.

In the last part of the experimental phase, the second step in the evaluation process, which represents one of the novelties of our approach, was performed to proof the power of our prediction when exploited in a trading simulation. In this stage, a trading simulation was set up as follows. The initial account value was set as $1,000,000 and the transaction cost was fixed at $0.001 as retrieved from online brokers[7,8]. During the computation of the Sharpe ratio, the value of the risk free return *r* was set as 2.8% according to the T-Bond quotation. The trading simulation was executed after the computation of the threshold *TSH* as explained in

[7] yadix.com/hft.

[8] ig.com.

Table 4
Trading simulation results.

L&Mc		Annualized return	Maximum drawdown	Sharpe ratio
	Price	−46.5%	12.9%	-3.3%
	News	78.3%	**1.52%**	**7.78%**
	News&Price	**85.2%**	3.9%	4.76%
Affective		Annualized return	Maximum drawdown	Sharpe ratio
	Price	−45.6%	11,2%	−3.3%
	News	42.6%	3.82%	3.4%
	News&Price	5.05%	5.34%	0.3%
Buy & Hold		Annualized return	Maximum drawdown	Sharpe ratio
	Price	43.5%	1.59%	5.6%

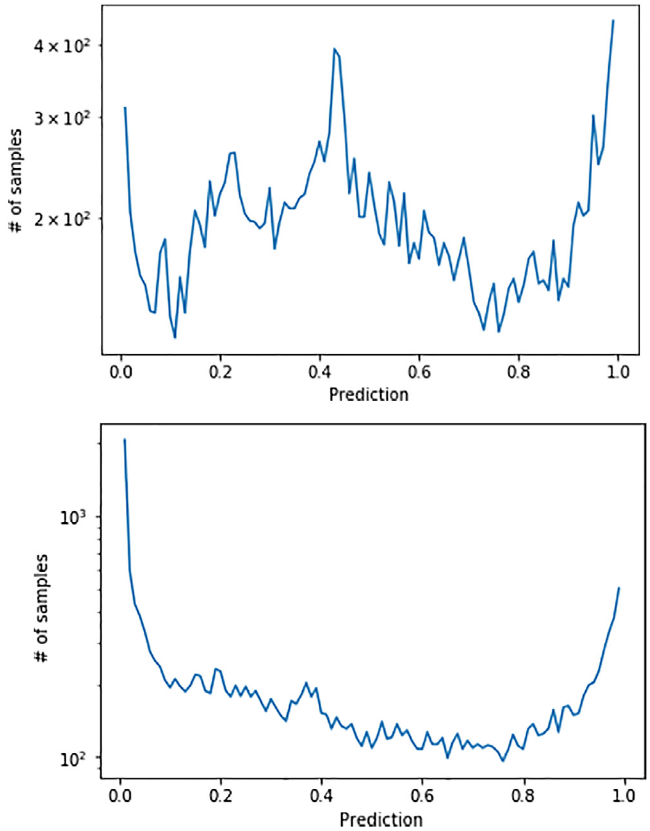

Fig. 10. Predictions obtained with L&Mc dataset: News (up) and News&Price (down) features.

the Section 3. To have an insight on this process two discrete probability distributions are showed in Fig. 10.

The reported charts also enlighten that the distribution of the predictions generated with the features combination of Price&News results to be more concentrated on the extreme values of the Sigmoid activation function in the output layer. This fact underlines an higher confidence on the predictions in the model trained with the Price&News set in comparison with the News only.

After the threshold computation, the trading strategy was performed exploiting the prediction of the out-of-sample data. As Table 4 reports, our model, when applied to the combined features set, achieves a high and positive annualized gain, demonstrating the effectiveness and the strong profitability of our prediction approach. The Buy & Hold strategy is inserted as a financial benchmark to our achievements.

Summarizing, with this second evaluation step, our approach is proved to be effective from both the correct behavior of the classifiers developed and the performance obtained during the trading simulation. Moreover, other important achievements, regarding the predictive power of our available features sets, can be drawn from the reported performance.

In the case of the Loughran and McDonald dictionary dataset, the gain obtained with the exploitation of the combined features set is the highest, while, with the use of AffectiveSpace 2, the news set outperforms among the others.

This is a further point in favor of our previous conclusion related to the importance of a features fusion technique. In fact, when working with the 151 dimensional input of the first dataset, our model is able to combine the features and obtain higher results even if the features sets are simply concatenated.

On the other hand, when the 611 dimensional vector of Loughran and McDonald dictionary was fed as input, our model was not able to properly exploit the features combination power of this wider vector. Thus, in the second case, a features fusion process would improve the predictions leading to higher returns. The values of the other performance indicators led to the same conclusion. In fact, both the maximum drawdown and the sharpe ratio enlightened more volatility in the portfolio values when the Price&News dataset was used. So, even if the combination of the features has achieved very high returns, passing through a feature fusion step could improve our interesting achievements leading to a decrease in the volatility of the portfolio value.

6. Conclusion and future directions

The target of this work was to combine the technical and fundamental analysts approaches to market trend forecasting through the use of machine learning techniques applied to time series prediction and sentiment analysis. Furthermore, we aimed to develop a robust model able to predict the trends of a portfolio of stocks and to exploit its predictions in a trading strategy. Thus, we propose the exploitation of a feed forward neural network architecture into a trend classification problem and we perform an evaluation divided into two steps. The first step is executed to evaluate the statistical behavior of the classifier, while the second is focused in testing the effectiveness of the models' predictions when exploited in a trading simulation.

The outcome is a robust model which is able to effectively and fairly classify both positive and negative trends in the portfolio of stocks under study. The correct behavior of the classifier is demonstrated with the evaluation of the gap between the recall values of the positive and negative samples. Moreover, our model is able to recognize the most meaningful changes in the market trend and to achieve positive returns during the trading simulation. In this work, two different approaches were used to extract sentiment embeddings from the news: the Loughran and McDonald dictionary and AffectiveSpace 2. The use of AffectiveSpace features as input to the neural network architecture resulted to be more effective in achieving high accuracy values, while the exploitation of the features computed with the use of Loughran and

McDonald dictionary led to higher values of annualized returns. A comparison between three feature sets, namely the Price, News, and Price&News sets, was performed and the combination of the sentiment embeddings with the price technical indicators results to out-perform the use of the Price set only. On the other hand, the over performance in comparison with the use of the News set alone is not outstanding. Thus, we believe that, to solve this weakness, the use of a proper features fusion technique is an interesting future direction to investigate. Furthermore, we are aware that the time span of our data is limited and we are actively working to retrieve more news data and to build on top of it a proper backtesting phase to test our model. Summarizing, this work establishes a solid basement for future collaborations between technical and fundamental approaches to the market prediction. Moreover, it reconciles opposing approaches, which have divided the market analysts for more than half a century, with the exploitation of machine learning and data science techniques.

Declaration of Competing Interest

We declare no conflicts of interest.

Credit authorship contribution statement

Andrea Picasso: Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing - original draft, Writing - review & editing, Visualization. **Simone Merello:** Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing - original draft, Writing - review & editing, Visualization. **Yukun Ma:** Supervision, Validation, Formal analysis, Investigation, Writing - original draft. **Luca Oneto:** Supervision, Methodology, Formal analysis, Resources, Writing - original draft, Writing - review & editing. **Erik Cambria:** Supervision, Methodology, Resources, Writing - original draft, Writing - review & editing.

References

Abarbanell, J. S., & Bushee, B. J. (1998). Abnormal returns to a fundamental analysis strategy. *Accounting Review*, 19–45.

Al Shalabi, L., Shaaban, Z., & Kasasbeh, B. (2006). Data mining: A preprocessing engine. *Journal of Computer Science, 2*(9), 735–739.

Amin, A., Anwar, S., Adnan, A., Nawaz, M., Howard, N., Qadir, J., et al. (2016). Comparing oversampling techniques to handle the class imbalance problem: A customer churn prediction case study. *IEEE Access, 4*, 7940–7957.

Bandy, H. (2007). *Quantitative trading systems*.

Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of Computational Science, 2*(1), 1–8.

Box, G., & Jenkins, G. (1970). *Time series analysis-forecasting and control.*.

Cambria, E., Fu, J., Bisio, F., & Poria, S. (2015). Affectivespace 2: Enabling affective intuition for concept-level sentiment analysis.. *AAAI conference on artificial intelligence*.

Chawla, N. V., Bowyer, K., Hall, L., & Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*, 321–357.

Choudhry, R., & Garg, K. (2008). A hybrid machine learning system for stock market forecasting. *World Academy of Science, Engineering and Technology, 39*(3), 315–318.

Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems, 2*(4), 303–314.

De Choudhury, M., Sundaram, H., John, A., & Seligmann, D. D. (2008). Can blog communication dynamics be correlated with stock market activity? *ACM conference on hypertext and hypermedia*.

Fama, E. F. (1991). Efficient capital markets: II. *The Journal of Finance, 46*(5), 1575–1617.

Gentile, C., & Warmuth, M. K. (1999). Linear hinge loss and average margin. *Advances in neural information processing systems*.

Hazarika, D., Poria, S., Zadeh, A., Cambria, E., Morency, L. P., & Zimmermann, R. (2018). Conversational memory network for emotion recognition in dyadic dialogue videos. In *Conference of the north american chapter of the association for computational linguistics: Human language technologies*.

Hu, Z., Liu, W., Bian, J., Liu, X., & Liu, T. (2018). Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction. *Acm international conference on web search and data mining*.

Huang, C., Yang, D., & Chuang, Y. (2008). Application of wrapper approach and composite classifier to the stock trend prediction. *Expert Systems with Applications, 34*(4), 2870–2878.

Huang, W., Nakamori, Y., & Wang, S. (2005). Forecasting stock market movement direction with support vector machine. *Computers & Operations Research, 32*(10), 2513–2522.

Hurst, H. E. (1951). Long-term storage capacity of reservoirs. *Transactions of the American Society of Civil Engineers, 116*, 770–799.

Ioffe, S., & Szegedy, C. (2015). *Batch normalization: Accelerating deep network training by reducing internal covariate shift*. arXiv: 1502.03167.

Jin, F., Self, N., Saraf, P., Butler, P., Wang, W., & Ramakrishnan, N. (2013). Forex-foreteller: Currency trend modeling using news articles. *ACM sigkdd international conference on knowledge discovery and data mining*.

Khaidem, L., Saha, S., & Dey, S. R. (2016). *Predicting the direction of stock market prices using random forest*. arXiv: 1605.00003.

Kim, K., & 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: 1412.6980.

Kumar, M., & Thenmozhi, M. (2006). Forecasting stock index movement: A comparison of support vector machines and random forest. *Journal of Engineering Research and Applications*.

Li, X., Xie, H., Chen, L., Wang, J., & Deng, X. (2014). News impact on stock price return via sentiment analysis. *Knowledge-Based Systems, 69*, 14–23.

Li, Y., Pan, Q., Yang, T., Wang, S., Tang, J., & Cambria, E. (2017). Learning word representations for sentiment analysis. *Cognitive Computation, 9*(6), 843–851.

Lo, A. W. (2004). The adaptive markets hypothesis: Market efficiency from an evolutionary perspective. *Journal of Portfolio Management, Forthcoming*.

Loughran, T., & McDonald, B. (2011). When is a liability not a liability? Textual analysis, dictionaries, and 10-ks. *The Journal of Finance, 66*(1), 35–65.

Lusa, L., & Blagus, R. (2010). Class prediction for high-dimensional class-imbalanced data. *BMC Bioinformatics, 11*(1), 523.

Luss, R., & D'Aspremont, A. (2015). Predicting abnormal returns from news using text classification. *Quantitative Finance, 15*(6), 999–1012.

Ma, Y., Peng, H., & Cambria, E. (2018). Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive lstm. *AAAI conference on artificial intelligence*.

Merello, S., Picasso, A., Oneto, L., & Cambria, E. (2019). Predicting future market trends: which is the optimal window? *International neural network society big data and deep learning*.

Mittal, A., & Goel, A. (2012). Stock prediction using twitter sentiment analysis. *Standford University, 15*.

Mizuno, H., Kosaka, M., Yajima, H., & Komoda, N. (1998). Application of neural network to technical analysis of stock market prediction. *Studies in Informatic and Control, 7*(3), 111–120.

Oancea, B., & Ciucu, S. C. (2014). *Time series forecasting using neural networks*. arXiv: 1401.1333.

Oh, C., & Sheng, O. (2011). Investigating predictive power of stock micro blog sentiment in forecasting future stock price directional movement.. *International conference on information systems*.

Panchal, G., & Panchal, M. (2014). Review on methods of selecting number of hidden nodes in artificial neural network. *International Journal of Computer Science and Mobile Computing, 3*(11), 455–464.

Patro, S., & Sahu, K. K. (2015). *Normalization: A preprocessing stage*. arXiv: 1503.06462.

Peng, Y., & Jiang, H. (2015). *Leverage financial news to predict stock price movements using word embeddings and deep neural networks*. arXiv: 1506.07220.

Picasso, A., Merello, S., Ma, Y., Malandri, L., Oneto, L., & Cambria, E. (2018). Ensemble of technical analysis and machine learning for market trend prediction. *IEEE symposium series on computational intelligence*.

Poria, S., Cambria, E., Bajpai, R., & Hussain, A. (2017). A review of affective computing: From unimodal analysis to multimodal fusion. *Information Fusion, 37*, 98–125.

Qian, B., & Rasheed, K. (2004). Hurst exponent and financial market predictability. *Iasted international conference on financial engineering and applications*.

Rao, T., & Srivastava, S. (2012). Analyzing stock market movements using twitter sentiment analysis. *International conference on advances in social networks analysis and mining*.

Rhea, R. (1993). *The Dow theory: An explanation of its development and an attempt to define its usefulness as an aid in speculation*. Fraser Publishing Company.

Ruck, D. W., Rogers, S. K., Kabrisky, M., Oxley, M. E., & Suter, B. W. (1990). The multilayer perceptron as an approximation to a Bayes optimal discriminant function. *IEEE Transactions on Neural Networks, 1*(4), 296–298.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature, 323*(6088), 533.

Schumaker, R., & Chen, H. (2006). Textual analysis of stock market prediction using financial news articles. *Americas conference on information system*.

Schumaker, R. P., & Chen, H. (2009). Textual analysis of stock market prediction using breaking financial news: The azfin text system. *ACM Transactions on Information Systems (TOIS), 27*(2), 12.

Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference, 90*(2), 227–244.

Si, J., Mukherjee, A., Liu, B., Li, Q., Li, H., & Deng, X. (2013). Exploiting topic based twitter sentiment for stock prediction. *Annual meeting of the association for computational linguistics*.

Srebro, N., Rennie, J., & Jaakkola, T. S. (2005). Maximum-margin matrix factorization. *Advances in neural information processing systems*.

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

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

Thomason, M. (1999). The practitioner methods and tool. *Journal of Computational Intelligence in Finance, 7*(3), 36–45.

Wu, J. L., Su, C. C., Yu, L. C., & Chang, P. C. (2012). Stock price predication using combinational features from sentimental analysis of stock news and technical analysis of trading information. *International Proceedings of Economics Development and Research*.

Xing, F., Cambria, E., & Welsch, R. E. (2018). Natural language based financial forecasting: A survey. *Artificial Intelligence Review, 50*(1), 49–73.

Xu, Y., & Cohen, S. B. (2018). Stock movement prediction from tweets and historical prices. *Annual Meeting of the Association for Computational Linguistics*.

Yao, J., & Poh, H. (1995). Forecasting the klse index using neural networks. *IEEE international conference on neural networks*.

Yao, J., Tan, C. L., & Poh, H. (1999). Neural networks for technical analysis: a study on klci. *International journal of theoretical and applied finance, 2*(02), 221–241.

You, Q., Luo, J., Jin, H., & Yang, J. (2015). Robust image sentiment analysis using progressively trained and domain transferred deep networks.. *AAAI conference on artificial intelligence*.

Zhang, G. P. (2003). Time series forecasting using a hybrid arima and neural network model. *Neurocomputing, 50*, 159–175.