

# Public Mood–Driven Asset Allocation: the Importance of Financial Sentiment in Portfolio Management

Lorenzo Malandri<sup>1</sup> · Frank Z. Xing<sup>2</sup> · Carlotta Orsenigo<sup>1</sup> · Carlo Vercellis<sup>1</sup> · Erik Cambria<sup>2</sup>

## Abstract

The study of the impact of investor sentiment on stock returns has gained increasing momentum in the past few years. It has been widely accepted that public mood is correlated with financial markets. However, only a few studies discussed how the public mood would affect one of the fundamental problems of computational finance: portfolio management. In this study, we use public financial sentiment and historical prices collected from the New York Stock Exchange (NYSE) to train multiple machine learning models for automatic wealth allocation across a set of assets. Unlike previous studies which set as target variable the asset prices in the portfolio, the variable to predict here is represented by the best asset allocation strategy ex post. Experiments performed on five portfolios show that long short-term memory networks are superior to multi-layer perceptron and random forests producing, in the period under analysis, an average increase in the revenue across the portfolios ranging between 5% (without financial mood) and 19% (with financial mood) compared to the equal-weighted portfolio. Results show that our all-in-one and end-to-end approach for automatic portfolio selection outperforms the equal-weighted portfolio. Moreover, when using long short-term memory networks, the employment of sentiment data in addition to lagged data leads to greater returns for all the five portfolios under evaluation. Finally, we find that among the employed machine learning algorithms, long short-term memory networks are better suited for learning the impact of public mood on financial time series.

**Keywords** Portfolio allocation · Sentiment analysis · Long short-term memory networks · Machine learning

## Introduction

Financial markets are becoming increasingly important as economies grow. However, in today's society, financial

✉ Erik Cambria  
cambria@ntu.edu.sg

Lorenzo Malandri  
lorenzo.malandri@polimi.it

Frank Z. Xing  
zxing001@ntu.edu.sg

Carlotta Orsenigo  
carlotta.orsenigo@polimi.it

Carlo Vercellis  
carlo.vercellis@polimi.it

<sup>1</sup> Department of Management, Economics and Industrial Engineering, Politecnico di Milano, Milan, Italy

<sup>2</sup> School of Computer Science and Engineering, Nanyang Technological University, Singapore, Singapore

markets are highly unpredictable and more correlated than decades ago. This is because market movements are influenced by a number of different factors, among which there is public mood. Like emotions have an impact on our personal behavior and decisions, in a similar way market sentiment could be correlated or even predictive of collective decision-making [1, 2]. World Wide Web and social media have become a bottomless source of text data, curating people's opinions on a wide range of topics. In this context, public mood provides a global and efficient representation of the inclination of investors [3].

A cornerstone of modern theory of finance is the Efficient-market hypothesis (EMH), proposed by Fama [4], states that current stock prices already reflect all the past information, and stock prices will only react to new information. As a consequence, future prices follow a random walk and it is impossible to “beat the market” on a risk-adjusted basis. This theory has originated a longstanding debate in the financial field and, from the mid-1980s onward, there have been many attempts to discover imperfections in the market, showing how some patterns can be unveiled [5–8] and disputing the EMH assumption.

Fama himself, in his later work [9], revised his statement indicating different levels of efficiency.

The last decade witnessed a massive boost in online content, like digital newspapers and social media, allowing people's opinions to be analyzed in such an unprecedented amount through text mining. Stock investors are continuously updating their beliefs. This massive amount of ever-changing information cannot be assimilated by traditional financial theories [10], even though it expresses the will of the investors and could possibly forerun their actions or influence other people. Based on the assumption that public sentiment is correlated or even a predictor of stock market behavior, it is imperative to develop effective techniques accounting for financial mood.

Investor sentiment has been a matter of interest even before the advent of text mining and the outburst of social media. Brown and Cliff [11] used sentiment surveys from companies and signal extraction techniques to derive investor sentiment from market indicators. They show that investor and employee sentiment has a consistent relation with large stocks. In the era of social media and Web 2.0, the interest in natural language-based financial forecasting [12] has grown fast. In 2008, Tetlock et al. [13], by means of ordinary least squares regression, find that "pessimism" weakly predicts market volatility and does not give clear information about market fundamentals in the short term. Slightly better results are achieved by Li [14], who finds that the tone of forward-looking statements is positively correlated with future performance. The author uses both lexicon-based and Naïve Bayes classifiers, but only the latter leads to significant results. Finally, some scholars adopted support vector machines (SVM) for stock direction classification (referred to the increase or decrease of the stock prices). For example, Schumaker and Chen trained an SVM [15] which performs with 57.1% of directional accuracy and simulated trading at 2.06% return. This is quite surprising because simulation is made on the S&P 500 index, which represents a very stable and highly efficient stock market. Other studies relying on SVM with "neutral zone" on tweets [16] can predict stock closing prices when they have a big rise or fall, while other scholars use dynamic evolving neuro-fuzzy inference systems (DENFIS) and long short-term memory (LSTM) networks in order to build a method which incorporates public mood to generate market views computationally [17]. Regarding the social media sentiment data, several studies used Twitter [1, 18–21] as source, given its standard format and the availability of APIs. Other scholars made use of aggregated news [22], message boards [23, 24], or a combination of those sources. After texts are collected, sentiment analysis tools [25–27] are adopted in order to extract mood from texts.

A well-known problem in this thread of research is the absence of a reliable benchmark dataset [28]. On one hand, the available datasets are in different format and lack of adequate information [3]. On the other hand, building a reference dataset in this field is complex. First, a long time series is required: this means that data should have been collected for a long time from many different sources and for all the stocks in a given market. Second, many companies are reluctant to disclose financial sentiment data they have collected and analyzed for their own purposes. Finally, performing natural language processing (NLP) on financial data is a non-trivial task due to the intense use of sarcasm, metaphors, common sense, and domain-specific terms, or the lack of labeled data [29].

Another known issue in this area of investigation is the evaluation of the results. Very few scholars have examined whether their datasets are imbalanced or not [3], and many of them aimed at forecasting the directional accuracy of the stocks. In this field, an accuracy value which significantly differs from 50% could be retained as a proof of effectiveness of the forecasting results [12], but, since on average there is a rising trend for stock prices, a dummy model which always predicts a rise in the price will achieve an accuracy higher than 50%. For this reason, we will compare our results against a Naïve benchmark in portfolio management, the so-called equal-weighted (EW) portfolio that will be presented in "Data and Methods Overview."

Despite the considerable interest raised in discovering financial sentiment in the past years, to the best of our knowledge, only a small number of researches focused on the problem of portfolio allocation. Koyano and Ikeda [30] propose a semi-supervised learning method using stock microblogs for the maximization of the cumulative return of the portfolio using a follow-the-loser approach. Another recent work [31] uses an ensemble of evolving clustering and LSTM to formalize sentiment information into market views that will be later integrated into mean-variance portfolio theory through a Bayesian approach. Online portfolio selection is one of the core problems in financial engineering and has always drawn a lot of attention from both scholars and practitioners. Two main schools investigated this problem: the mean-variance theory [32, 33] and the capital growth theory (CGT) [34, 35]. While the former focuses on the trade-off between expected return (mean) and risk (variance) of the portfolio in the single period, the latter aims at minimizing the expected growth rate of a portfolio over a temporal interval through asset allocation. Expected growth rate maximization is a problem tailored for the online scenario [36] and will be set as the optimization objective of this research.

In this paper, a new model for portfolio allocation is proposed. This model will account for both stock returns

and public mood for the automatic formalization of the asset reallocation strategy. In particular, the optimal allocation strategy will be generated simultaneously for all the stocks in the portfolio and no predictions on the single stocks will be made. Three different machine learning algorithms will be employed: LSTM, multi-layer perceptron (MLP), and random forest classifier (RFC). The portfolios generated by the three techniques will be compared against the EW portfolio. Moreover, the importance of sentiment data in addition to traditional lagged data will be assessed by means of a statistical test over five different portfolios.

The contribution of this work can be summarized as follows. First, we propose a new method for incorporating public mood in portfolio allocation. Second, the algorithm for portfolio allocation automatically generates an online investment strategy. As a consequence, no hand-crafted expert knowledge is required and the model can easily be adapted to account for transaction costs and holding positions. In addition, the proposed model can be updated in real time. In particular, with LSTM and MLP, everytime a new batch of data comes in, the model is updated *without being re-trained from scratch*. Also, sentiment data can be monitored and added to the model in real time. In a fast-evolving environment like financial markets, it is essential to have online models with good compatibility [12]. Furthermore, in our model, we account for the temporal structure of people’s opinions, which is of paramount importance together with the time correlation between opinions and returns. By means of LSTM networks, the model can learn long-time dependencies and process sentiment and lagged data in sequence (Fig. 1). Last, our simulations show that including financial sentiment improves the performance of the optimized portfolio. This result is consistent over five portfolio analyses in our experiments and is statistically significant.

The remainder of this article is organized as follows: “Data and Methods Overview” provides an overview of the data collection process, of the portfolio allocation strategy, and of the machine learning algorithms used in the present study; “Experiments” describes the experimental setting and the computational results achieved; finally, “Conclusion” concludes the paper and discusses some future research directions.

## Data and Methods Overview

### Data collection

We gathered financial and sentiment data for 15 different stocks for the time period from 24 January 2012 to 2 June 2017. For the entire period, data have been collected with daily granularity excluding weekends and holidays since

trading is suspended during those days. All the data used in this research are publicly available and there are no missing data. We obtained financial data through the Quandl API [37] and sentiment data through the StockFluence API [38]. Financial data include daily time series of lagged prices and trading volumes for 15 popular stocks. Both prices and volumes have been adjusted in order to account for stock splits. Sentiment data are composed by five values for each day and each stock, including the number of positive, negative, and neutral comments, a measure of change in positive and negative comments compared with the previous days (*change*) and a measure of positive and neutral versus negative reviews (*sentimentscore*). StockFluence collects and analyzes everyday about 1.5 million comments between Twitter and articles.

### Methodology

Consider  $N$  financial portfolios  $p_n, n = 1, \dots, N$ . Each portfolio is composed by  $M$  stocks in which we invest our wealth  $w$  for a sequence of  $T$  training periods.

Let us indicate our daily reallocation strategy for portfolio  $n$  as:

$$S_n = \{s_n^1, \dots, s_n^T\},$$

where  $S_n \in \mathbb{R}^{M,T}$  and the generic term  $s_n^t$  is a  $M$  dimensional vector representing the weight to be allocated to each one of the  $M$  assets in period  $t$  for portfolio  $n$ . Our aim is to find, for each portfolio, the strategy  $S^*$  such that

$$S_n^* = \arg \max_{S_n} S_n^\top R_n,$$

where  $R_n \in \mathbb{R}^{M,T}$  is the daily returns matrix for the assets in portfolio  $n$ . Since we optimize each portfolio separately and independently, from now onward the subscript  $n$  will be omitted.

The optimal strategy will be automatically generated by the algorithm after an appropriate training. In particular, the best ex post allocation will be used to train the algorithm. Knowing the returns of portfolio’s assets in the following period, the best apportioning strategy is trivial: allocate all the wealth to the asset that will generate the greatest return in the next period. For this reason, the best allocation strategy ex post for each period will be represented by a 5-dimensional vector of ex post best allocation strategy  $s_t^{ep} \in \mathbb{R}^M, t = 1, \dots, T$  obtained through one-hot encoding in the following way:

$$s_{i,t}^{ep} = \begin{cases} 1, & \text{if } r_{i,t+1} = \max_m r_{m,t+1}, m = 1, \dots, M \\ 0, & \text{o/w} \end{cases}$$

where  $\mathbf{r}_{t+1} \in \mathbb{R}^M$  is the returns vector for period  $t + 1$ .

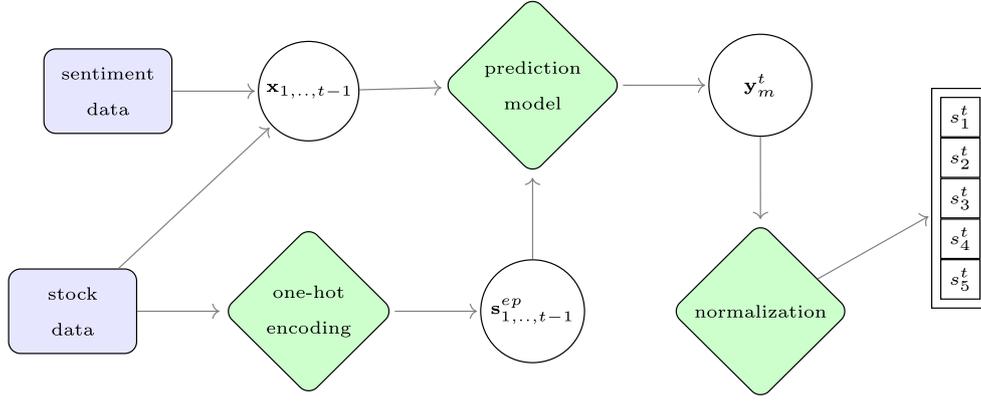


Fig. 1 Model framework combining sentiment and lagged data

The rows of our dataset will be composed by the 1350 days under examination. For each row, the input vector of predictors  $\mathbf{x}_t$  will include seven attributes for each stock. From Quandl we obtained the daily adjusted closing price and volume, and from StockFluence the number positive, neutral and negative reviews, the change value, and the sentiment score. Since each portfolio comprises five stocks, for each day we will have 35 predicting variables which, together with the five target variables, will form 40 columns. Rows are time-ordered and will be processed day by day. In order to use all the available data, like in a real-world situation, each day the optimal allocation  $\mathbf{s}_m^t$  will be automatically generated by the predictive model using all the previous data as input.

After being normalized, the output vector  $\mathbf{y}_t$  of predictions will represent the automatically generated strategy. Notice that it will not be a one-hot vector, since for each entry the prediction will represent the score function of that asset to be the one with the greatest return. In a supervised classification task, the score function may be associated with the likelihood that a label comes from a particular class. Since for each reallocation vector  $\mathbf{s}^t = \{s_1^t, \dots, s_M^t\}$  the condition  $\sum_{m=1}^M s_m^t = 1$  must hold, the prediction vectors  $\mathbf{y}_t$  will be normalized through the following formula:

$$z_m^t = \frac{y_m^t - \min_m y_m^t}{\max_m y_m^t - \min_m y_m^t}, m = 1, \dots, M,$$

$$s_m^t = \frac{z_m^t}{\sum_{m=1}^M z_m^t}, m = 1, \dots, M.$$

Since the algorithm will predict the optimal weight of  $M$  different stocks together, a multi-target prediction model must be generated, in which multiple target variables are predicted simultaneously from the same set of explanatory features. To address the multi-target prediction task, an extension of the basic algorithm of the aforementioned machine learning techniques, described in the following subsection, must be employed. Specifically, multi-target RFCs will be obtained by storing  $n$  output values in the leaves of the trees

instead of one, where  $n$  is the number of variables to be predicted. In this case, the splitting criterium will compute the average in the impurity reduction across the  $n$  different outputs. Classical MLP and LSTM networks, instead, can be easily extended to multi-target purposes by simply using a neuron in the output layer for each of the target variables. Thus, in our setting, the output layer will be composed by five different binary variables, each one predicting the optimal weight to be assigned to a different stock.

## Prediction Models

### Random Forest Classifier

Random forests [39] represent a powerful extension of decision trees [40], which are among the most popular techniques for classification and regression. It belongs to the family of ensemble algorithms since it grows a collection of trees from  $nt$  bootstrap samples drawn from the original data. Furthermore, the recursive partitioning of the nodes in a tree is based on a random subset of candidate predictors for which the best split is determined according to a suitable quality measure, such as the Gini impurity index or the Entropy. Once the forest of random trees is built, the final classification is performed based on two alternative schemes. By means of hard majority voting, the most popular class, i.e., the class which the majority of the trees come up with, is selected. Through soft voting, instead, the probability of belonging to a class is given by the average of the score (probability) for that class predicted by each of the  $nt$  trees. In this paper, the latter approach has been adopted.

Random forests depend mainly on three parameters: the number of trees in the forest ( $nt$ ), the maximum number of predictors to consider in individual trees ( $p$ ) for splitting each node, and the maximum depth of the tree ( $md$ ). In our computational setting, these parameters were tuned in order to obtain the most accurate predictions, as described in “Experiments.”

Random forests have shown great potential by achieving comparable performances compared to more complex classification algorithms. With respect to traditional decision trees, it has proven to be more robust and less prone to overfitting. Moreover, even though MLPs and SVM are by far the most common used techniques for predicting stock market returns, in this field some scholars reported outperforming results obtained by random forests for specific tasks [41]. Our implementation of the RFC is based on the Scikit-learn Python package [42].

## Multi-Layer Perceptron

The financial stock market is well known to be highly non-linear and highly complex and chaotic, owing to the interplay of complex factors influencing its behavior. For this reason, in the last years MLPs have become very popular in this field. MLPs are data-driven models, composed by an arbitrary number of layers of interconnected neurons activated by a linear function. They are universal approximators, capable to capture non-linear behaviors of time series without any statistical assumption about the data [43].

Most of the research studies using neural networks for financial forecasting problems have successfully adopted a feed-forward MLP [44]. Consistently with some successful applications for financial time series prediction [45, 46], in this research, we will adopt a three-layer network trained with back-propagation.

The main parameters that will be tuned for both MLP and LSTM networks are the number  $n$  of neurons for each layer of the network, the activation function, the loss function, and the number of epochs, as described in “Experiments.”

MLPs have been implemented with Keras [47], a high-level neural network API written in Python.

## Long Short-Term Memory Network

LSTMs, initially proposed by Hochreiter and Schmidhuber (1997), belong to the family of recurrent neural networks (RNNs), a family of neural networks with loops in them, allowing information to persist from a loop to another. LSTMs work very well in practice because they can learn long-time dependencies, unlike traditional RNN which suffer from vanishing/exploding gradient when back-propagation is through many time layers. In particular, we will use a stateful LSTM model. When a model is stateful, it means that the last state for a sample of index  $j$  in a batch will be the initial state for the sample of index  $j$  in the following batch. If we select a unitary sample size and no shuffle (we process data day by day from the first day to day  $T$ ), the state of the model will be propagated from the first to the last day of the period under analysis. Like the MLP, the LSTM has been implemented through Keras.

# Experiments

## Model Settings

The 15 selected stocks have been divided in five different portfolios. For the first three, we randomly selected five stocks for each one without repetition. The remaining two are composed by the five stocks which, in the selected period, performed best and worst. For each portfolio, we start the simulation with a unitary portfolio. The portfolio’s wealth will be re-apportioned every day through the automatically generated strategy.

Data from the 24th of January 2012 until the 9th of November of the same year (15% of the dataset) are only used to train the model and tune the parameters. For the following days, we perform a trading simulation. For each of the three algorithms, optimal parameters are obtained by grid search maximizing the return of the portfolio at the 24th of January 2012. Then hyper-parameters are fixed and for each period  $t$ ,  $t = 204, \dots, T$  all the data available from day 1 to day  $t$  are utilized for the generation of the optimal allocation strategy for period  $t + 1$  and the weight’s update. Therefore, all the features and real returns (after binary maximization) for period  $t + 1$  will be added to the predicting data to generate the optimal strategy for period  $t + 2$ , and so on, until period  $T$ . In this way, a quasi-realistic online trading simulation is reproduced. In reality, parameters can be tuned at each iteration, but in this paper we did it once and for all since tuning hyper-parameters 1350 times for five portfolios and three algorithms would have taken an unworkable amount of computational time. For this reason, results will be sub-optimal with respect to a real online trading situation.

For the RFC, we tuned two parameters, represented by the overall number  $nt$  of trees generated and the maximum depth  $md$  of each tree, in order to control the growth of the trees and avoid overfitting. The maximum number of predictors  $p$  to select for splitting the nodes was instead fixed to the Scikit-learn default value, defined as the total number of explanatory features comprised in the dataset. For each portfolio, a total of 18 combinations were considered, obtained by testing three values for  $nt$  (25, 50, 75) and six values for  $md$  (from 5 to 10 with step 1). In Scikit-learn, two impurity measures are implemented: the Gini index and the Entropy. Between the two, the Gini index was finally selected since it does not require to compute logarithmic functions and is therefore computationally less expensive.

For MLP and LSTM, we used a three-layer network, with one input layer, one hidden layer, and one dense output layer. Four parameters are tuned: the number of neurons  $n$ , the activation function, the loss function, and the number of epochs. In particular, we used *tanh* and *linear* activations,

while for the loss we considered the *hinge* and the *logcosh* functions. Regarding the number of epochs, we tested five different levels for MLP (from 20 to 100 with step 20) and 14 for LSTM (from 2 to 15 with step 1). The number of neurons for the hidden layers has been calculated through the following formula, derived from neural network design guidelines [48],

$$n = \frac{N_s}{(\alpha) * (N_i + N_o)},$$

where  $N_s$  is the number of samples,  $N_i$  is the number of input nodes,  $N_o$  is the number of output nodes, and  $\alpha$  is an arbitrary scaling factor usually ranging from 2 to 5 [48]. In our test, we selected the values 2 and 5.

### The five portfolios

We constructed five virtual portfolios consisting of each one of five stocks from the NYSE. The first portfolio includes Alliance Data System Corporation (ADS), British Petroleum plc (BP), Intel Corporation (INTC), Moody’s Corporation (MCO), and Philip Morris International Inc. (PM). In the second one, we have Apple Inc. (AAPL), Goldman Sachs Group Inc. (GS), Marvell Technology Group, Ltd. (MRVL), Pfizer Inc. (PFE), and Starbucks Corporation (SBUX). In the third one, we can find The Boeing Company (BA), Costco Wholesale Corporation (COST), Red Hat, Inc. (RHT), Target Corporation (TGT), and VMware, Inc. (VMW). The fourth portfolio is composed by the five stocks with higher returns over the period considered (AAPL, BA, COST, MCO, SBUX), and the fifth one with the five titles with lowest returns (BP, INTC, MRVL, TGT, VMW). We constructed these two portfolios to evaluate the goodness of our algorithm in the presence of performing and not performing titles. In Table 1 are reported the returns and number of comments for each stock over the entire period under examination.

### Results

The aim of the experiments is twofold. In a first stage, the different algorithms adopted will be compared, while in a second phase will be assessed the significance of using sentiment data in addition to lagged data. In the first phase, the returns generated by the three algorithms will be compared against a widely adopted benchmark portfolio, called EW portfolio, which gives the same importance to each stock. Each of the  $M$  stocks in the portfolio will have a fixed weight of  $1/M$  for the entire time horizon. This strategy is widely used and has been shown to outperform value- and price-weighted portfolios in terms of total mean

**Table 1** Stock returns and number of comments for the period in exam

Stock	Return	Pos	Neg	Neutral
INTC	1.65	18,645	4,861	129,836
PM	2.05	12,312	3,947	93,112
MCO	3.52	10,073	5,449	55,396
BP	1.12	16,589	5,919	111,522
ADS	2.21	12,690	3,504	108,280
AAPL	2.87	26,135	5,018	128,520
GS	2.12	10,728	4,618	119,097
MRVL	1.25	10,901	3,065	82,107
PFE	1.81	10,096	4,168	102,999
SBUX	2.90	24,863	7,834	120,378
RHT	1.98	19,387	4,242	104,032
COST	2.62	18,613	6,550	104,290
BA	2.90	14,557	4,446	128,303
TGT	1.30	23,025	7,359	112,404
VMW	1.08	14,752	3,647	98,875

return and Sharpe Ratio, although usually EW portfolios have higher risk and turnover [49, 50].

We performed an online trading simulation with daily reallocation for 5 years (1259 days in total). Initially every portfolio has unitary wealth. After each period, the wealth of the portfolio is updated through the following equation:

$$w_t = w_{t-1} \sum_{m=1}^M r_m^t s_m^t,$$

where  $w_t$  is the wealth of the portfolio at time  $t$ , with  $w_0 = 1$ . The final wealth  $w_T = w_0 \prod_{n=1}^T R_n$  for each portfolio and each prediction model is reported in Table 2.

Table 2 reports the final value of the portfolios with initial wealth of 1. Six models are presented: three with lagged data only and with the supplement of sentiment data. The presence of sentiment data will be denoted by adding the letter *s* to the name of the algorithm. All the six models work well and outperform the EW portfolio. The best results are reached by the LSTM + *s* for portfolios 2, 3, 4, and 5 and from the RFC + *s* model for portfolio 1. Anyway, for portfolio 1, the difference with the final

**Table 2** Final wealth

Portfolio	EW	LSTM + s	LSTM	MLP + s	MLP	RFC + s	RFC
1	1.93	2.23	2.03	2.22	2.06	2.25	2.17
2	2.21	2.72	2.30	2.39	2.43	2.43	2.40
3	1.78	2.30	1.90	2.12	1.83	1.98	1.80
4	2.52	2.81	2.71	2.69	2.60	2.60	2.66
5	1.45	1.65	1.53	1.62	1.59	1.50	1.48

value of the LSTM + s portfolio is slight. In addition to that, the LSTM portfolio is the only one where the use of sentiment data consistently improves the prediction model. This was expected since LSTMs are RNNs which are able to

capture time dependencies both in sentiment and in financial time series.

For each prediction model, the final value varies quite a lot across the five portfolios. This is due not only to the

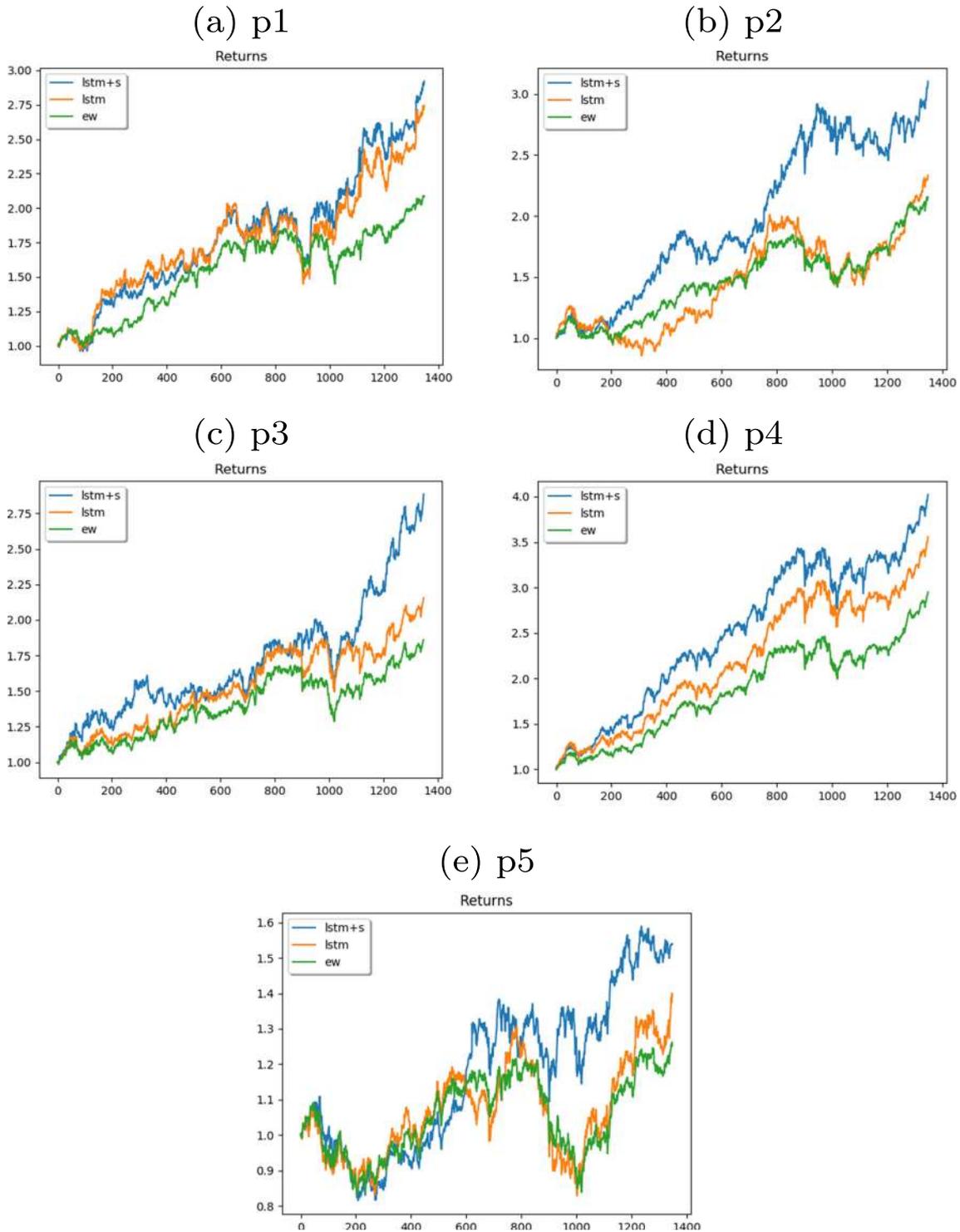


Fig. 2 Portfolio returns over the test period. a p1. b p2. c p3. d p4. e p5

**Table 3** Benchmark value

Portfolio	EW	LSTM + s	LSTM
1	1	1.16	1.05
2	1	1.23	1.04
3	1	1.29	1.06
4	1	1.11	1.07
5	1	1.14	1.05

goodness of the automatically generated strategy, but also to the different returns of the 15 selected stocks over the period under examination. Whatever the allocation strategy, in most of the cases, the returns trend will follow the average return of the stocks in the portfolio (Fig. 2). In order to provide a fairer comparison, we will compute the extra-returns with respect to the benchmark method. This is simply done by dividing the final value of each portfolio by the final value of the corresponding benchmark portfolio (EW) and is reported in Table 3. The return of the EW portfolio represents the average return of the different stocks. Thus, it constitutes a good comparison basis and will remove the effect of different stock returns.

Among the selected prediction models, LSTM is the one which better captures the sentiment and gives better results in general. With LSTM, adding the sentiment scores as attributes increases the final weight of each of the five portfolios. In order to assess the statistical significance of this increment, we perform a paired  $t$  test on the pairs  $w_T$  with and without sentiment for each portfolio. Results are presented in Table 4.

The paired  $t$  test highlights a statistically significant mean difference between the LSTM portfolio returns with and without sentiment. The  $p$  value of around 1% shows that sentiment data is informative and has a predictive value that is captured by the LSTM network. The contribution of public mood to portfolio allocation is thus robust over five different portfolios and statistically significant and is captured by LSTM networks.

**Table 4** Paired  $t$ -test. LSTM vs LSTM + s

	LSTM + s	LSTM
Mean	1.1865	1.0574
Variance	0.0057	0.0001
Observations	5	5
df	4	–
$t$ Stat	3.6949	–
$P(T_i = t)$	0.0105	–

## Conclusion

In this research, we investigate whether public mood collected from social media and online news is correlated or predictive of portfolio returns, and we introduce the framework of sentiment-driven portfolio allocation. We compare three different learning algorithms for the problem of portfolio allocation: LSTM, MLP, and RFC. We do not dwell on the problem of stock returns prediction, which has been extensively studied. Instead, we propose a novel approach which automatically produces an optimal online portfolio allocation strategy.

Our results reveal that the portfolio allocation problem can be tackled all-in-one in the context of end-to-end learning [51], with an algorithm which gets as input the historical series of lagged data and public mood and automatically returns the optimal portfolio allocation. We show that this methodology consistently outperforms the equal-weighted portfolio, and that the inclusion of financial sentiment is always beneficial. Among the three methods compared, LSTM is the one that provides better results. This aligns with our intuition since LSTM belongs to the family of RNN, which is designed to learn in sequence, with information persisting for long periods. Public opinion expressed at one day will probably be correlated with stock returns in the following days, and LSTMs can learn time dependencies of this kind. Finally, simulation results show that by using LSTM networks, the inclusion of collective mood consistently improves the results reached resorting solely to lagged data. This empirical finding is consistent over five different portfolios and is statistically significant. Although it has already been proved in the literature that public sentiment is correlated to stock prices, it has been seldom discussed how it affects fundamental problems of computational finance.

Our paper does not contemplate some aspects that will be addressed in future research. Most importantly, more sophisticated NLP tools should be adapted to the financial domain, in order to extract more complex and informative sentiment data. The use of mere polarity (positive, negative, neutral) subtracts depth to the analysis. The employment of a broader range of affective states, as suggested by [1], could be beneficial for the forecasting process. Moreover, more complete sentiment data on a larger number of stocks will allow adding the problem of portfolio selection into the model. Last, market frictions and transaction costs are not considered, as well as short positions and and credibility of text data [52], despite that they could be relevant to the problem of portfolio allocation.

### Compliance with Ethical Standards

**Conflict of Interest** The authors declare that they have no conflict of interest.

**Informed Consent** Informed consent was not required as no human or animals were involved.

**Human and Animal Rights** This article does not contain any studies with human or animal subjects performed by any of the authors.

## References

1. Bollen J, Mao H, Zeng X. Twitter mood predicts the stock market. *J Comput Sci.* 2011;2(1):1–8.
2. Howard N, Cambria E. Intention awareness: improving upon situation awareness in human-centric environments. *Human-centric Computing and Information Sciences.* 2013;3(1):9.
3. Nassiroussi AK, Aghabozorgi S, Wah TY, Chek D, Ngo L. Text mining for market prediction A systematic review. *Expert Syst Appl.* 2014;41(16):7653–7670.
4. Fama EF. The behavior of stock-market prices. *J Bus.* 1965;38(1):34–105.
5. Black F. Noise. *J Finance.* 1986;41(3):528–543.
6. Bradford De Long J, Shleifer A, Summers LH, Waldmann RJ. Noise trader risk in financial markets. *J Political Econ.* 1990;98(4):703–738.
7. Kavussanos MG, Dockery E. A multivariate test for stock market efficiency: the case of ase. *Appl Financ Econ.* 2001;11(5):573–579.
8. Bo Q, Rasheed K. Stock market prediction with multiple classifiers. *Appl Intell.* 2007;26(1):25–33.
9. Fama EF. Efficient capital markets: a review of theory and empirical work. *J Finance.* 1970;25(2):383–417.
10. Li Q, Jiang L, Li P, Chen H. Tensor-based learning for predicting stock movements. In: *AAAI*, pp 1784–1790. 2015.
11. Gregory W. Brown and Michael T Cliff. Investor sentiment and the near-term stock market. *J Rmpir Finance.* 2004;11(1):1–27.
12. Xing F, Cambria E, Welsch R. Natural language based financial forecasting: a survey. *Artifx Intell Rev.* 2018;50(1):49–73.
13. Tetlock PC, Saar-Tsechansky M, Macskassy S. More than words: quantifying language to measure firms' fundamentals. *J Finance.* 2008;63(3):1437–1467.
14. Li F. The information content of forward-looking statements in corporate filings—a naïve Bayesian machine learning approach. *J Account Res.* 2010;48(5):1049–1102.
15. Schumaker RP, Chen H. Textual analysis of stock market prediction using breaking financial news The azfin text system. *ACM Trans Inf Syst (TOIS).* 2009;27(2):12.
16. Smailović J, Grčar M, Lavrač N, Žnidaršič M. Predictive sentiment analysis of tweets: a stock market application. In: *Human-Computer Interaction and Knowledge Discovery in Complex, Unstructured, Big Data.* Springer; 2013. p. 77–88.
17. Xing F, Cambria E, Malandri L, Vercellis C. Discovering Bayesian market views for intelligent asset allocation. In: *ECML*; 2018.
18. Si J, Mukherjee A, Liu B, Li Q, Li H, Deng X. Exploiting topic based twitter sentiment for stock prediction. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*; 2013. p. 24–29.
19. Ranco G, Aleksovski D, Caldarelli G, Grčar M, Mozetič I. The effects of twitter sentiment on stock price returns. 2015, Vol. 10.
20. Papan A, Kyrtsov C, Kugiumtzis D, Diks C. Detecting causality in non-stationary time series using partial symbolic transfer entropy: evidence in financial data. *Comput Econ.* 2016;47(3):341–365.
21. Tafti A, Zotti R, Jank W. Real-time diffusion of information on twitter and the financial markets. 2016, Vol. 11.
22. Akita R, Yoshihara A, Matsubara T, Uehara K. Deep learning for stock prediction using numerical and textual information. In: *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS).* IEEE; 2016. p. 1–6.
23. Das SR, Chen MY. Yahoo! for amazon: sentiment extraction from small talk on the web. *Manag Sci.* 2007;53(9):1375–1388.
24. Nguyen TH, Shirai K. Topic modeling based sentiment analysis on social media for stock market prediction. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*; 2015. p. 1354–1364.
25. Cambria E, Olsher D, Kwok K. Sentic activation: a two-level affective common sense reasoning framework. In: *AAAI.* Toronto; 2012. p. 186–192.
26. Poria S, Gelbukh A, Cambria E, Yang P, Hussain A, Durrani T. Merging senticnet and Wordnet-affect emotion lists for sentiment analysis. In: *2012 IEEE 11th International Conference on Signal Processing (ICSP).* IEEE; 2012. p. 2012.
27. Cambria E, Hussain A, Havasi C, Eckl C. SenticSpace: visualizing opinions and sentiments in a multi-dimensional vector space. *Knowledge-Based and Intelligent Information and Engineering Systems*, volume 6279 of *Lecture Notes in Artificial Intelligence*. In: Setchi R, Jordanov I, Howlett R, and Jain L, editors. Berlin: Springer; 2010. p. 385–393.
28. Shraavan Kumar B, Ravi V. A survey of the applications of text mining in financial domain. *Knowl-Based Syst.* 2016;114:128–147.
29. Cambria E, Poria S, Gelbukh A, Thelwall M. Sentiment analysis is a big suitcase. *IEEE Intell Syst.* 2017;32(6):74–80.
30. Koyano S, Ikeda K. Online portfolio selection based on the posts of winners and losers in stock microblogs. In: *2017 IEEE Symposium Series on Computational Intelligence (SSCI).* IEEE; 2017. p. 2017.
31. Xing F, Cambria E, Welsch R. Intelligent asset allocation via market sentiment views. *IEEE Comput Intell Mag.* 2018;13(4):25–34.
32. Markowitz H. Portfolio selection. *J Finance.* 1952;7(1):77–91.
33. Markowitz HM, Peter Todd G, Sharpe William F, Vol. 66. Mean-variance analysis in portfolio choice and capital markets. New York: Wiley; 2000.
34. Kelly Jr. JL. A new interpretation of information rate. In: *The Kelly Capital Growth Investment Criterion: Theory and Practice.* World Scientific; 2011. p. 25–34.
35. Samuelson PA. Lifetime portfolio selection by dynamic stochastic programming. In: *Stochastic Optimization Models in Finance.* Elsevier; 1975. p. 517–524.
36. Li B, Hoi SCH. Online portfolio selection: a survey. *ACM Comput Surv (CSUR).* 2014;46(3):35.
37. Quandl A. P. I. Various end-of-day data. <https://www.quandl.com/data>. 2017.
38. StockFluence API. Financial sentiment data series. <https://www.stockfluence.com/>. 2017.
39. Breiman L. Random forests. *Mach Learn.* 2001;45(1):5–32.
40. Leo B, Friedman JH, Olshen RA, Stone CJ. Classification and regression trees. Wadsworth International Group. 1984.
41. Patel J, Shah S, Thakkar P, Kotecha K. Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Syst Appl.* 2015;42(1):259–268.
42. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, et al. Scikit-learn: m learning in python. *J Mach Learn Res.* 2011;12(Oct):2825–2830.

43. Lu C-J, Lee T-S, Chiu C-C. Financial time series forecasting using independent component analysis and support vector regression. *Decis Support Syst.* 2009;47(2):115–125.
44. Cavalcante RC, Brasileiro RC, Souza VLF, Nobrega JP, Oliveira ALI. Computational intelligence and financial markets: a survey and future directions. *Expert Syst Appl.* 2016;55:194–211.
45. Kara Y, Boyacioglu MA, Baykan ÖK. Predicting direction of stock price index movement using artificial neural networks and support vector machines: the sample of the Istanbul stock exchange. *Expert Syst Appl.* 2011;38(5):5311–5319.
46. de Oliveira FA, Nobre CN, Zárata LE. Applying artificial neural networks to prediction of stock price and improvement of the directional prediction index—case study of petr4, petrobras, Brazil. *Expert Syst Appl.* 2013;40(18):7596–7606.
47. Chollet F, et al. Keras. <https://keras.io>. 2015.
48. Hagan MT, Demuth HB, Beale MH, De Jesús O, Vol. 20. *Neural network design*. Boston: Pws Pub; 1996.
49. Plyakha Y, Uppal R, Vilkov G. Why does an equal-weighted portfolio outperform value-and price-weighted portfolios? Available at SSRN 1787045. 2012.
50. Ferguson R, Schofield D. Equal weighted portfolios perform better. *Financial Times*. 2010.
51. Sutskever I, Vinyals O, Le Quoc V. Sequence to sequence learning with neural networks. In: *Advances in neural information processing systems*; 2014. p. 3104–3112.
52. Minhas S, Hussain A. From spin to swindle: identifying falsification in financial text. *Cogn Comput.* 2016;8(4):729–745.