Politecnico di Milano

Department of Management, Economics and Industrial Engineering

Ph.D. in Management, Economics and Industrial Engineering

XXXI Cycle

Investigating the Impact of Public Mood in the Stock Market

Lorenzo Malandri

Thesis supervisor:

Prof. Carlo Vercellis

Thesis Structure

The manuscript will be organized as a *Consolidated Thesis by papers*, a coherent piece of writing based on the presentation of 4 articles, that address a single main overarching goal, articulated in further sub-goals, instrumental to the achievement of the main goal.

The four papers are listed below:

- ***Public Mood Driven Asset Allocation: The Importance of Financial Sentiment in Portfolio Management***
  Lorenzo Malandri, Frank Z. Xing, Carlotta Orsenigo, Carlo Vercellis and Erik Cambria
  Cognitive Computation journal (October 2018)

- ***Discovering Bayesian Market Views for Intelligent Asset Allocation***
  Frank Z. Xing, Erik Cambria, Lorenzo Malandri, and Carlo Vercellis
  2018 ECML-PDKK Conference (10 - 14 September, 2018, Dublin, Ireland)

- ***Ensemble of Technical Analysis and Machine Learning for Market Trend Prediction***
  Andrea Picasso Ratto, Simone Merello, Lorenzo Malandri, Yukun Ma, Luca Oneto and Erik Cambria
  2018 IEEE SSCI Conference (18 - 21 November, 2018, Bengaluru, India)

- ***Sentiment-Conditional Generation of Synthetic Text***
  Working paper

Abstract

The overall purpose of this thesis is to study the correlation between scattered public mood and financial markets. The study of the impact of investors' sentiment on stock returns has gained increasing momentum in the past few years, and it has been widely accepted that public mood is correlated with financial markets. Nevertheless, only a very scam number of studies discuss in which way public financial sentiment affects the fundamental problems of computational finance. For this reason, the first three articles present a solution to three classical problems of applied finance: the portfolio allocation problem, the formalization of market views and the use of mixed methods for stock returns prediction. Those solutions are implemented through machine learning methods driven by public financial sentiment, lagged data and others technical indicators. The financial sentiment is collected from different online sources and analysed by means of sentiment analysis (SA) techniques.

One of the main issue observed in the conclusion of these three articles is the scarcity available data. Financial lexicon, financial sentiment time series and financial mood data in general are few and often incomplete. For this reason, the fourth paper focuses on the problem of data augmentation, proposing a novel approach to simultaneously train a sentiment classifier on sentiment classes and generate synthetic sentiment-conditional and class-conditional data. The aim of this research is to produce a sufficient data base for the training of Machine Learning (ML) models for sentiment classification.

List of attached papers:

PAPER I

Public Mood Driven Asset Allocation: The Importance of Financial Sentiment in Portfolio Management

Lorenzo Malandri[1], Frank Z. Xing[2], Carlotta Orsenigo[1], Carlo Vercellis[1] and Erik Cambria[2]

[1] *Department of Management, Economics and Industrial Engineering, Politecnico di Milano, Italy*

E-mail: {lorenzo.malandri,carlotta.orsenigo,carlo.vercellis}@polimi.it

[2] School of Computer Science and Engineering, Nanyang Technological University, Singapore

E-mail: {zxing001,cambria}@ntu.edu.sg

Abstract: *Background:* The study of the impact of investors' 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 public financial mood affects one of the fundamental problems of computational finance: portfolio management.

Methods: 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 is here represented by the best asset allocation strategy ex post.

Results: Experiments performed on five portfolios show that long short-term memory networks are superior to multilayer 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-weighed portfolio.

Conclusion: Results show that our all-in-one 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, Financial Sentiment, Capital Growth Theory, Recurrent Neural Networks.

PAPER II

Discovering Bayesian Market Views for Intelligent Asset Allocation

Frank Z. Xing[2], Erik Cambria[2], Lorenzo Malandri[1] and Carlo Vercellis[1]

[1] *Department of Management, Economics and Industrial Engineering, Politecnico di Milano, Italy*

E-mail: {lorenzo.malandri,carlo.vercellis}@polimi.it

[2] School of Computer Science and Engineering, Nanyang Technological University, Singapore

E-mail: {zxing001,cambria}@ntu.edu.sg

Abstract: Along with the advance of opinion mining techniques, public mood has been found to be a key element for stock market prediction. However, how market participants behaviour is affected by public mood has been rarely discussed. Consequently, there has been little progress in leveraging public mood for the asset allocation problem, which is preferred in a trusted and interpretable way. In order to address the issue of incorporating public mood analysed from social media, we propose to formalize public mood into market views, because market views can be integrated into the modern portfolio theory. In our framework, the optimal market views will maximize returns in each period with a Bayesian asset allocation model. We train two neural models to generate the market views, and benchmark the model performance on other popular asset allocation strategies. Our experimental results suggest that the formalization of market views significantly increases the profitability (5% to 10% annually) of the simulated portfolio at a given risk level.

Keywords: Market views, Public Mood, Bayesian Fusion, Asset Allocation

PAPER III

Ensemble of Technical Analysis and Machine Learning for Market Trend Prediction

Andrea Picasso Ratto[1] Simone Merello[1], Lorenzo Malandri[3], Yukun Ma[2], Luca Oneto[1], and Erik Cambria[2]

[1] *Department of Informatics, Bioengineering, Robotics and System Engineering, University of Genova, Italy*

E-mail: {simone.merello,andrea.picasso}@smartlab.ws, luca.oneto@unige.it

[2] School of Computer Science and Engineering, Nanyang Technological University, Singapore

E-mail: cambria@ntu.edu.sg, mayu0010@e.ntu.edu.sg

[3] *Department of Management, Economics and Industrial Engineering, Politecnico di Milano, Italy*

E-mail: lorenzo.malandri@polimi.it

Abstract: Over the last twenty years, researchers and practitioners have attempted in many ways to effectively predict market trends. Till date, however, no satisfactory solution has been found. Many approaches have been applied to predict market trends, from technical analysis to fundamental analysis passing through sentiment analysis. A promising research direction is to exploit market technical indicators together with market sentiments extracted from social media for predicting market directional movements. In this paper, we propose a new approach that leverages technical analysis to predict market directional movements. In particular, we aim to predict the directional movement of the NASDAQ's most capitalized

stocks by solving a classification problem. The results on real-world data show that our proposal achieves interesting performance when predicting the market directional movements. This work focuses on forecasting a portfolio of different stocks, instead of concentrating on a single stock which most of the works in this field do. Furthermore, the proposed model is able to solve the issue of skewed classes through the use of appropriate data balancing techniques. This project represents a step forward to improve the robustness of stock trend forecasting techniques and provides a starting point for technical analyst to better understand the market behavior

Keywords: Market Trend Prediction, Technical Analysis, Machine Learning

PAPER IV

Sentiment-Conditional Generation of Synthetic Text

Working paper

In this paper, we propose an extremely simple yet novel approach for data augmentation through a sequence to sequence generative model. In the first phase, an LSTM network is trained for learning sentiment and domain conditioned distribution of lemmas from the available labelled sentences. In a second instance new samples are generated through the previously trained model and a larger dataset is used for sentiment classification with different machine learning algorithms. Preliminary results show that sentiment classification can be improved up to 6% for the datasets tested and the improvement given by the data augmentation is consistent over two different datasets.

Table of Contents

1. Introduction

In this research we develop new algorithms for the investigation of public financial mood as it is expressed on social media and its effect on the stock market.

Sentiment Analysis (SA), or Opinion Mining (OM), is an ongoing research field in Text Mining. SA studies the recognition of opinions, feelings and emotions from text objects. World wide web has become a bottomless source of unstructured data, with quintillions of bytes of data generated daily and publicly accessible. Nevertheless, this impressive amount of unstructured data is noisy and hard to be structured. At the same time its potential is quite intriguing, since it could allow people's opinions to be mined in quantities never seen before, without the need for expensive surveys nor resorting to private and corporate data.

In the financial domain, sentiment analysis grabbed the attention of many scholars and practitioners. The problem of financial time series forecast has always been of paramount importance in finance and economics, and opinion mining presents a potentially useful tool for improving classical techniques with human reactions and social sentiment. However, this field is still in an embryonic phase and there is substantial room for improvement. For example, even if a number of research studies have faced the problem of stock returns prediction, only a few focused on real world applications. In addition, the availability and quality of experimental datasets is scarce (Nassirtoussi et al., 2014). Some scholars have tried to build their own dataset, ending with a number of sets of data, heterogeneous in content and scarce in information available. Our study tackles this two challenges.

On the on hand, we will employ financial sentiment data in order to tackle some important problems of financial engineering, like the portfolio allocation problem and the market views formalization problem, including in the predictors the mood and the opinions of the investors.

On the other hand, we will develop new techniques that will allow us to better mine sentiment information from public available text data. Those two parts belong to different fields, financial time series forecasting and sentiment analysis, but they find a common ground both in terms of applications and methodologies. In the financial domain is particularly difficult to find quality sentiment data. Even though the quantity of comments, posts and news in this field is quite impressive, current methodologies do not allow to extract opinions from this immense source of unstructured and unlabelled data, and this fact practically hinders the performance of the models for financial time series forecasting. Thus a unified methodological framework is essential to the advancement of science in this field.

2. Problem description and motivation

Understanding market movements is as difficult as important, since it could lead to huge gains or losses and help to observe in advance signals of a financial crisis. Moreover, primary and secondary financial markets are becoming increasingly more tied with real economies.

The rise of World Wide Web before and web 2.0 applications afterwards, together with the increasing popularity of social media, are providing a massive amount of publicly available information at an extraordinarily fast pace. Furthermore, this phenomenon is bringing attention to the subjective view of a large base of investors, stakeholders, experts and even common people, unveiling information that where hidden behind few years ago.

Among both scholars and practitioners, the impression is increasing that investors and markets are too complex to be analysed through a series of indexes, biases and trading frictions. Better results could rise through what is sometimes called a “top-down” approach, that analyses aggregate sentiment and its effect on stock markets (Baker and Wurgler, 2007).

The study of aggregated people opinions and public mood falls under the field of sentiment analysis. Sentiment analysis studies the recognition of opinions, feelings and emotions from unstructured objects, like text, images or audio. We will focus on textual data, which conveys the largest part of the public financial sentiment.

With 2.5 quintillion (2.5x10^18) bytes of data produced every day (IBM, 2016) and over 2.3 billion active social media users, World wide web has become a bottomless source of unstructured data. Every minute are generated approximatively 350.000 tweets (Twitter live statistics) and every day thousands of users release reviews on Amazon. This immense and public “data lake” is quite hard to be structured. At the same time its potential is quite

intriguing, since it could allow people's opinions to be mined in quantities never seen before, without the need for expensive surveys nor resorting to private and corporate data.

The past decade witnessed many steps forward in analysing and understanding investors' opinion based on social media content. Nevertheless, the existing literature does not acknowledge for a number of relevant factors. In this research we will take on two of them:

1- Investigate the effect of public mood on real world financial engineering problems and techniques
2- Develop a data augmentation algorithm in order to improve supervised learning techniques for sentiment classification

1- In the last years, public mood has been found to be a key element for stock market prediction (Bollen et al., 2011; Nassirtoussi et al., 2014; Chu et al., 2017). However, in what manner the market participants are affected by public mood has been rarely discussed. As a result, there has been little progress in leveraging public mood for financial engineering problems. Despite many researchers studied the direct correlation between public opinion and stock market returns, only a few investigated the effect of public mood on more complicated financial problems, like portfolio allocation or market views formalization. Moreover, in paper III, we use both investors' sentiment and indicators, like RSI, MACD and moving average. This is an attempt to blend opinion mining and technical analysis techniques for an integrated approach, that will give us a fuller picture of stock prices determinants.
2- A known problem in this field is the scarcity of labelled data. Many scholars have consistently shown that supervised learning algorithms perform well in the field of

sentiment analysis. Yet, they need a large amount of data to be trained. In the financial domain is quite hard to find labelled data, for a number of reasons. First, in other fields like customer or movie reviews, website reviews usually come with a free-text field and a score value. On amazon we find a five star Likert scale, on IMDB a one-to-ten response scale and so on. Those values can be used as a label for the free-text field, and used to train an algorithm for opinion mining on unlabelled text, like, for instance, social media posts and comments. The same process cannot be replicated on financial text data, given the lack of labelled data. The second reason lies in the time at which financial mood must be analysed. Investors continuously change their beliefs, thus the financial opinion must be analysed diachronically. Financial news and social media posts are released continuously. Some researchers even identify the optimal time window for sentiment based stock predictions in 20 minutes. This makes the manual labelling of sufficiently large datasets undoable. Last, scholars and practitioners are finding extremely difficult to apply Transfer Learning approaches in the financial domain. Transfer Learning (or Cross Domain sentiment analysis) aim at accruing knowledge while learning in one domain, and applying it to a different one. This is extremely difficult since the financial lexicon uses extensively technical terms, metaphors and other specific elements.

3. State of the art

3.1 Sentiment Analysis

The roots of Sentiment Analysis, belonging to the domain of computational linguistics, lie in the late eighties of the last century as a particular application of text data mining. Natural Language Processing (NLP) models were already offering highly efficient algorithms for classical text mining, like text clustering, text categorization and summarization, part of speech detection and entity relation modelling. Sentiment Analysis is not only limited to the semantic component of a speech, but must detect the sentiment similarities between the words and the overall sentiment of a sentence. This represents a further challenge, but still worth studying because it could allow exploitation of the huge amount of textual opinions from blogs, social networks and internet reviews and websites. A closely related field is the determination of the semantic orientation of words and sentences, often denoted as sentiment classification or opinion mining. Semantic orientation is sometimes expressed by its polarity (positive or negative) and/or strength (degree to which the word/sentence is positive/negative) toward an object or one of its features (Taboada et al., 2011). Most of the time the sentiment identified by means of sentiment analysis is used to determine the polarity of the word. For this reason, and because both of the fields employ text mining techniques, sentiment analysis and opinion mining are often used as synonymous (Cambria et al., 2013). Sentiment analysis can be lexicon or learning based:

Lexicon based SA:

Lexicon based sentiment analysis uses opinion words, phrases and idioms in order to assess words and sentences' polarity. Opinion words are words that are commonly used to express

a positive or negative opinion (Liu, 2012). A collection of opinion words is called Opinion Lexicon, and this forms the namesake of this methodology for SA.

In 2004 Hu et al. (Hu et al., 2004) achieve an average accuracy of sentence orientation prediction over 84% on five hi-tech products with a lexicon based algorithm. However, precision and recall are both under 70%. This is one of the main problems of lexicon based sentiment analysis. The reason is that the opinion lexicon cannot include all the existing opinion words. In addition, opinion words can change their meaning over time and often are domain dependent. Popescu and Etzioni (2007) refine the methodology from Hu et al. (2004) and test their proposed techniques on the same dataset. Two important new methods presented in their work are the use of Point-wise mutual information (PMI) statistics for extraction of high quality features and the adoption of unsupervised collective classification techniques for semantic orientation (SO) detection. Their results, mainly because of the PMI statistics, outperform Hu et al. (2004) precision (+2% for opinion extraction and +10% for polarity detection) but is still under 80% in opinion extraction. Furthermore, their results in recall are not clearly improving on the previous methodology (+4% for opinion extraction and -11% for polarity detection).

In general, for the above mentioned reasons, the overall performance of Lexicon-based algorithms is less than 80% and the recall even lower (Zhang et al., 2011; Sommar and Wielondek, 2015). The good points of these methods are the high scalability (they do not need manual labeling) and the fact that they capture the relational structure of the lexicon better than learning based methods.

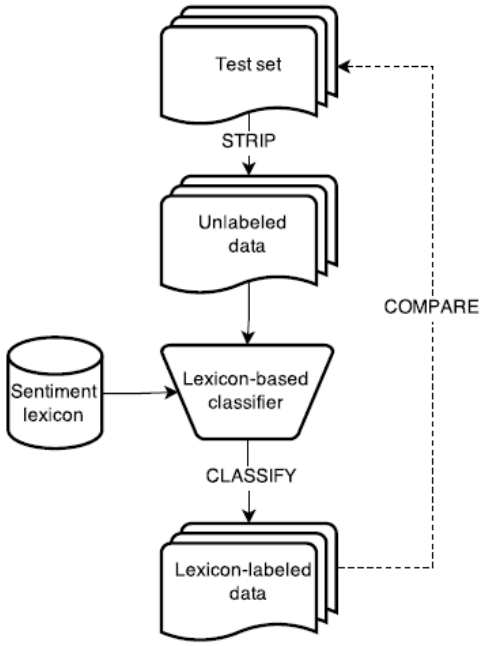


Figure 3.1.1: A flow diagram of the lexicon-based method (Sommar and Wielondek, 2015)

Learning Based SA:

The learning based methods use machine learning techniques to identify patterns in previously labelled data. They reach very high accuracy, precision and recall in many domains, but require a large set of manually labelled data. Since they are trained on a set of feature vectors with polarity as the target variable, some scholars refer to them as supervised learning methods. Pang et al. (2002) test Naive Bayes (NB), Maximum Entropy (ME) and Support Vector Machines (SVM) classifiers on a movie reviews database. Such cohort was rather convenient for two reasons. First, the authors found many collections of movie reviews on-line. Second, each review includes a rating indicator that has been converted into three categories (positive, negative and neutral) and used as the target variable. As a consequence, there has been no need for manual labeling. For these reasons the movie review database from Pang et al. (2002) has become a benchmark database for learning based sentiment

analysis. For translating text into features vectors the researchers adopt the Bag of Words (BoW) method. Each word is considered as a feature (unigram) and both the presence and the frequency of each word are contemplated. The relative position of words does not matter. They consider also a variation of BoW using multiple words, in particular bigrams, as features. However, Pang et al. find that this does not implicate an improvement over using only unigrams. This is in contrast with the work by Wang and Manning (2012) who find that the use of bigrams raises significantly the performances of machine learning algorithms for sentiment classification. BoW has been later improved through the term frequency-inverse document frequency (tf-idf) method (Manning et al., 2008; Aggarwal et al., 2012). Inverse document frequency is the generalization of the stop words removal. The words common to many documents (usually common words, like “if”, “that” or “not”) are given a smaller value since they could create noise. Term frequency is a normalization of word frequency within a document, since the excessive presence of a single word could compromise the similarity computation. Anyway, many researchers observed that considering the presence of the terms yield much better results than the frequency (Pang et al.,2002; Aggarwal et al., 2012; Sommar and Wielondek, 2015). Other researchers show how a preliminary detection of opinions (subjective sentences) leads to at least comparable levels of accuracy, given a much shorter text (Pang and Lee, 2004). After this preliminary phase Pang and Lee improve their previous work getting to an overall accuracy of 87,2% with the implementation of a SVM applied on BoW, improving previous results from Pang et al. (2002) of 82,9%. SVM are in general the most accurate classifier, but for short texts they can be outperformed by NB. For full reviews, like the ones in the movie review database introduced above, SVM perform better and, with NB log count rations as feature values, can be very accurate (over 90%) and robust (Wang and Manning, 2012). In conclusion, learning-based methods for Sentiment Analysis can reach very

high levels of accuracy and recall, but need manual labeling for the training dataset, thus are scarcely scalable.

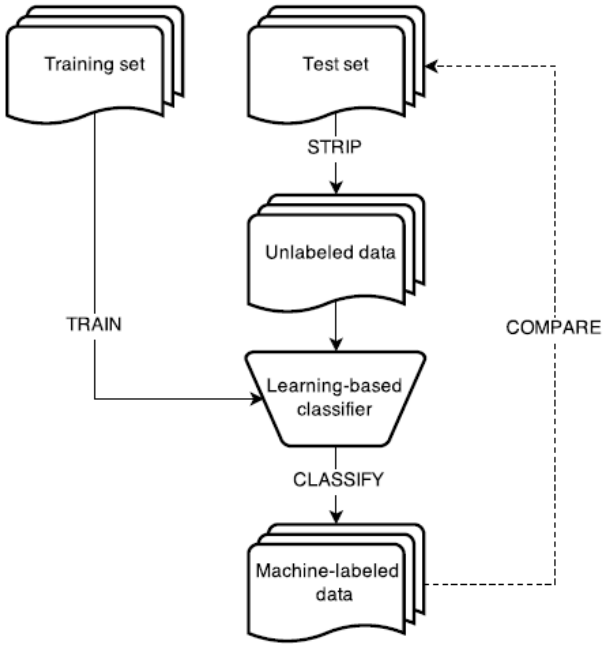


Figure 3.1.2: A flow diagram of the learning-based method (Sommar and Wielondek, 2015)

3.2 Sentiment Analysis and the Stock Market

A pillar of modern finance is the Efficient Market Hypothesis (EMH) by Fama (1965) which states that existing share prices always reflect all the relevant information. Therefore, stock prices follow a random walk and it is impossible to "beat the market". This theory always divided the field of finance. Fama himself (Malkiel and Fama, 1970) updated his thesis, indicating three levels of efficiency: strong, semi-strong and weak. From the mid-eighties onward there have been many attempts to discover imperfections in the market, showing how some patterns can be unveiled (Black, 1986; De Long et al., 1990; Kavussanos and

Dockery, 2001; Qian et al., 2007), questioning the EMH assumption. A few articles have also found out that, even if news is unpredictable, some indicators can be inferred from social media (Bollenand Huina, 2011). While the accuracy of most of the works is under 70%, in this particular field approaches over 55% are considered noteworthy, since even a small deviation from the chance case can generate huge gain and losses. For example, Brown and Cliff (2004) use sentiment surveys from companies and signal extraction techniques to extract investor sentiment from market indicators. They show that investors and employees sentiment has a consistent relation with large stocks. This is encouraging evidence for social media sentiment analysis. The techniques adopted are diverse. Das and Chen (2007) apply a combination of different classifiers and observe that, among the others, regression has the lowest explanatory power. Therefore, is not surprising that Tetlock et al., by means of OLS (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 (2010), who finds that the tone of forward-looking statements (FLS) "is positively correlated with future performance and has some explanatory power on other variables". The author uses both lexicon-based and Naive Bayes classifiers, but only the latter leads to significant results. Finally, some authors used SVM. For example, Schumaker and Chen's SVM (2009) perform with 57,1% of directional accuracy and Simulated Trading at 2.06% return. This is quite surprising because the simulation is made on the S&P 500 index, which represents a very stable and highly efficient stock market. Other studies conducted with SVM with "neutral zone" on tweets (Smailovic et al., 2013) can predict stock closing prices in cases when they have a big rise or fall.

Since the relation between sentiment data and the stock market is commonly known to be highly non-linear, in the last year the rise of neural networks in all their variants provided new opportunities for the research in this field. Ding et al. (2015) employ convolutional neural networks (CNN) and show that they can capture long time relationship better than Feed-Forward Neural Networks (FFNN). Li et al (2016) use Long Short-Term Memory networks (LSTM) outperforming the state of the art results on the Stock Dataset (Li et al., 2012). Unfortunately, very few scholars in this field state whether their datasets are imbalanced or not (Nassirtoussi et al., 2014). A dataset is imbalanced when there is a significant difference between the prior probabilities of different classes. Standard classification algorithms are usually biased toward the most numerous class. This often leads to misclassification of the class with lowest prior probability, which is also usually the more interesting one. In financial application the problem of balanced datasets is even more important, since one of the main tasks in this literature is the forecast of directional accuracy (whether the returns of a stock will be positive or negative in the following period). One could say a forecasting methodology is good if its directional accuracy is higher than 50%, but since stocks are increasing on average, a dummy model which always predicts a rise in the price of the stock, in the long run will achieve an accuracy greater than 50%.

To conclude, a few scholars investigated the impact of public opinion on portfolio allocation. Smales (2016), using sentiment data provided by Thomson Reuters News Analytics, finds evidence that the effect of news on the stock market varies over time and has and when the "fear" emotion is high, systemic factors overwhelm industry-specific factors as investment drivers. Koyano and Ikeda (2017) maximize cumulative portfolio returns using stock microblogs and a follow-the-loser approach, with their approach beating the NIKKEI 225 market and other existing methods.

The adoption of sentiment analysis for financial forecast attracted a large number of scholars and practitioners in the last years. Nevertheless, as far as we know, the impact of public mood on the main problems of financial engineering has been rarely discussed. If on the one hand we have a long list of researches investigating the impact of financial sentiment on stock prices, on the other only a few scholars tackled relevant issues like portfolio optimization or market views formalization.

3. Contribution

The thesis is a collection of articles. Each article will contribute in a different way to the achievement of an overall research goal. The aim of this research is to advance the state-of-the-art in the field of sentiment analysis for financial applications. To do this, we face problems of two different natures. On the one hand, we contribute to the employment of public opinions in financial applications: the first three papers present a solution for many common problems in financial engineering. In the problem settings, the sentiment data are obtained from different providers who have collected, analysed and pre-processed the raw online opinion data. On the other hand, we develop a new methodology for machine learning based sentiment classification. In fact, learning-based approaches to opinion mining will perform better and are less-biased than lexicon-based techniques. The issue with the formers is that these approaches are not automated, since they need manual labeling of the training set. In some domains, labelled data are available (like product reviews, where the customer numerical evaluation of the product can be used as label), but in the others, such as the finance domain, the labelling must be done manually (collecting data from sources like twits, newspaper articles and blogs). In addition, investors' opinion changes frequently, thus manual annotation of a large amount of data should be repeated frequently. In the fourth paper, we develop a new data augmentation technique for sentiment classification. In other words, we start from an initial seed of textual data and, starting from it, we create new synthetic text that will be used to train the classification algorithms. The purpose is to improve the classification performance of learning based techniques, without the need of manually labelling a huge amount of data.

The contribution of each paper is listed in the following table:

Paper	List of contributions
I-*Public Mood Driven Asset Allocation: The Importance of Financial Sentiment in Portfolio Management* *Lorenzo Malandri, Frank Z. Xing, Carlotta Orsenigo, Carlo Vercellis and Erik Cambria* *Cognitive Computation* journal (October 2018)	1- We propose a novel all-in one and end-to-end methodology for the problem of temporal maximisation of portfolio allocation. The algorithm for portfolio allocation automatically generates an online investment strategy. Therefore, no handcrafted expert knowledge is required. Moreover, the model can easily be adapted to account for transaction costs and short positions. 2- The model integrates public mood and lagged data in an online fashion. Sentiment data can be processed in real-time. In a fast-evolving environment like financial markets, it is essential to have online models with good adaptability and able to learn long time series. Especially through the implementation of different Neural Network architectures (LSTM and FNN), we can perform incremental learning every time a new batch comes in without the need of retraining the whole model. 3- Our framework accounts for 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. 4- Simulations show that including financial sentiment improves the performance of the optimized portfolio. This result is statistically significant and consistent over five portfolios.
II-*Discovering Bayesian Market Views for Intelligent Asset Allocation* *Frank Z. Xing, Erik Cambria, Lorenzo Malandri, and Carlo Vercellis* *2018 ECML-PDKK* Conference (10 - 14 September, 2018, Dublin, Ireland)	1- In this article we present a novel definition of market views based on a Bayesian asset allocation model. We prove that our definition has the equivalent expressiveness as the original Black and Litterman formulation, but is simpler and easier to compute. 2- We develop a new methodology for the extraction of market views as expressed through online opinions. The views generated in this way are an expression of public mood and do not need expert knowledge or assumptions. 3- We propose a novel online method for portfolio optimisation which starting from market views and lagged data combined estimates the optimal

	portfolio allocation solving the problem of temporal maximisation of portfolio returns. 4- Our experiments show that the portfolio performance with market views blending public mood data stream is better than directly training a neural trading model without views. This superiority is robust for different models selected with the right parameters to generate market views.
III- ***Ensemble of Technical Analysis and Machine Learning for Market Trend Prediction*** *Andrea Picasso Ratto, Simone Merello, Lorenzo Malandri, Yukun Ma, Luca Oneto and Erik Cambria* *2018 IEEE SSCI* Conference (18 - 21 November, 2018, Bengaluru, India)	1- We blend technical and sentiment data in a unified and end-to-end model for stock market prediction. The result is a robust model that can achieve a directional accuracy of 61.69% on the 20 most capitalized tickers of the NASDAQ market. 2- After the application of appropriate data balancing techniques, the model performs well on different metrics, specifically accuracy, precision, recall and specificity. 3- We propose a novel cross-validation technique – *increasing windows cross validation* – which can be applied to time series problems, where

	classical k-fold cross validation with data shuffling cannot be performed 4- We employ with success feature ranking techniques. The purpose is to avoid handcraft feature selection without hinder the performance of the model.
IV- Sentiment-Conditional Generation of Synthetic Text Working paper	1- We develop a framework for sentiment-conditional and domain conditional text generation. The model can generate fixed length sentences with affective characteristics. 2- The sentences generated are different each other and smooth to read. 3- We generate new sentences from real labelled data and we use both the synthetic and real samples to train different machine learning algorithms. We prove that this technique can improve the classification accuracy up to 6% and is better than using only real data over both the dataset tested.

4. Discussion

In the last years, with the boom of internet 2.0 and app market, we witnessed an exponential growth of the number of inputs and opinions that the World Wide Web has thrust on all of us. Online newspapers, social media and blogs are transforming our life in a real agora where every topic can be discussed by expert and non-expert users living in different parts of the world. This affluence of information has arisen the interest of many researchers and practitioners. Is it possible to transform this huge and access-free sea of unstructured and noisy data into meaningful knowledge? In this research we addressed four different problems which respond to the same research question: *is it possible to collect information from online people opinion and use it to improve our forecasts on the stock market*.

This research question presents many different aspects, ranging from text mining to financial time series forecasting. We addressed four different problems. The first three refers to three different problems of real financial engineering. Although sentiment analysis attracted increasing attention in the financial domain, most of the papers deal with the problem of stock returns prediction solely. Real life financial problems are more than directional accuracy prediction and they have different levels of complexity. We offer a tailored solution for three of them. The last article deals with a very well-known and prominent problem of learning based sentiment analysis: in many domains, and the financial one is belongs to them, is extremely difficult to obtain labelled data to train machine learning algorithms for sentiment classification.

In the first paper, we demonstrated that is possible to incorporate public mood into the problem of portfolio optimization and solve it as an all-in-one, end-to-end problem that, giving lagged prices and sentiment time series as an input, can automatically prescribe an

optimal portfolio allocation which yields to returns higher than the equal-weighted portfolio. Moreover, we observed that the use of sentiment data statistically improves the performance of the portfolio allocation models. This insight is of great interest because shows that, even if the pattern of a single stock is difficult to predict, with the aim of public mood data we can make meaningful decisions on the allocation of different assets in a portfolio. Eventually, our results show that recurrent networks have better performance than static networks on this problem. A possible explanation is that the relationship between opinions and stock prices is dynamic and must be analysed diachronically.

In the second article, we developed a novel method which can formalize financial views starting from the public financial sentiment. In the Black and Litterman model public views are generated by expert investors based on their knowledge and experience. In our model the expert knowledge is replaced by public knowledge. The intuition is that by means of opinion mining information can be extracted from a larger base of investors, stake-holder and expert can. Our model can generate meaningful knowledge, as it is proven by the results achieved in the trading simulation. In addition, the model can automatically write a “story” explaining the operation rationales.

In the third article we the directional trend of stocks with a mixture of sentiment and technical analysis. This approach shows a number of benefits over the use of the traditional measures of stock market returns and volatility: fundamental and technical analysis. While the former includes many macroeconomic factors and is slow in updating variables, the second relies on price patterns and cannot incorporate new information and irrational behaviours of investors. Our model allows to extract information from both historical price trends and the investors

opinions, which can be extracted real time through sentiment analysis and an effective methodology for automatic feature ranking.

In the last working paper, we are developing a text mining methodology that can be useful for all the three problems stated above and many others in this field. Given a small dataset of labelled sentences, we generate new synthetic text data which will be used for training data augmentation. We prove that we can generate sentences with a sentiment value and improve the performance of sentiment classification. This research represents an important advance in the field of opinion mining for financial applications. The scarcity of labelled data is a well-known problem in this field. Through augmented data machine learning classifiers can be trained to better classify opinions sentiment and affective states, without the need for extensive and expensive manual labelling. Preliminary results can be expanded in many ways, including the use of different sources of data and the setting of a specific loss function for sentiment conditional text generation.

Overall, this research presents numerous advancements in the field of sentiment analysis for financial application. Many specific and real problems are addressed and new ad hoc methodologies are developed for each one of them. In addition, some of the proposed methods provide new interpretations of stock market dynamics. The models and frameworks that we propose are interesting from a research and discovery perspective, but at the same they are suitable to be adopted by professional investors.

5. Future works

In this section we provide a number of possible extensions that our work does not acknowledge, but that can constitute the starting point for new research avenues:

1. In the search space of text mining techniques for financial data, another promising field is transfer learning (TL), which accumulates linguistic knowledge in a domain to use it in a different one. The financial language is quite different from many others, since is rich of metaphors, technical terms, jargon words and ironic statements.
2. As mentioned in the conclusions, the research on data augmentation can be improved in many ways. The generative model tries to minimize the error with respect to unconditioned distributions of words in the sentences of the training set. A new loss function that minimizes this distribution conditional to sentiment classes and domains may be highly beneficial to the classification phase.
3. In future we will develop a model which can account for market frictions, in order to evaluate the performances of the proposed models in a real-world alike scenario.
4. The problem of portfolio selection has not been considered in any of the articles. The main reason is the lack of sentiment time series for many assets in the NYSE, even though is one of the stock markets with the largest number of online comments and discussion. The results of paper IV and of the researches at point 1 and 2 in this section will help collection more data on a larger number of stocks. With this data a proper algorithm for portfolio selection will be studied.
5. This research opens the lines for a closer observation of different market phenomena. For instance, who are the users that express a sentiment which is more (less)

correlated with future market trends? Which are the words that more characterize the financial domain and which sentiment they express? To answer those and other question a deeper analysis of the results must be undertaken.

6. References

Baker, Malcolm, and Jeffrey Wurgler. "Investor sentiment in the stock market." (2007).

Black, Fischer. "Noise." *The journal of finance* 41.3 (1986): 528-543.

Blitzer, John, Ryan McDonald, and Fernando Pereira. "Domain adaptation with structural correspondence learning." Proceedings of the 2006 conference on empirical methods in natural language processing. Association for Computational Linguistics, 2006.

Bollen, Johan, Huina Mao, and Xiaojun Zeng. "Twitter mood predicts the stock market." *Journal of Computational Science* 2.1 (2011): 1-8.

Brown, Gregory W., and Michael T. Cliff. "Investor sentiment and the near-term stock market." *Journal of Empirical Finance* 11.1 (2004): 1-27.

Chu, Victor W., et al. "Enhancing portfolio return based on sentiment-of-topic." *Data & Knowledge Engineering* (2017).

Dai, Wenyuan, et al. "Eigentransfer: a unified framework for transfer learning." Proceedings of the 26th Annual International Conference on Machine Learning. ACM, 2009.

Das, Sanjiv R., and Mike Y. Chen. "Yahoo! for Amazon: Sentiment extraction from small talk on the web." *Management Science* 53.9 (2007): 1375-1388.

De Long, J. Bradford, et al. "Noise trader risk in financial markets." *Journal of political Economy* (1990): 703-738.

Ding, Xiao, et al. "Deep learning for event-driven stock prediction." *Ijcai*. 2015.

Fama, Eugene F. "The behavior of stock-market prices." *The journal of Business* 38.1 (1965): 34-105.

Hu, Minqing, and Bing Liu. "Mining and summarizing customer reviews." *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004.

Jacobson, Ralph "2.5 quintillion bytes of data created every day. How does CPG & Retail manage it?" *IBM Consumer Products Industry Blog, 2016*

Jiang, Jing, and ChengXiang Zhai. "Instance weighting for domain adaptation in NLP." ACL. Vol. 7. 2007.

Kavussanos, Manolis G., and Everton Dockery. "A multivariate test for stock market efficiency: the case of ASE." *Applied Financial Economics* 11.5 (2001): 573-579.

Koyano, Shinta, and Kazushi Ikeda. "Online portfolio selection based on the posts of winners and losers in stock microblogs." *Computational Intelligence (SSCI), 2017 IEEE Symposium Series on*. IEEE, 2017.

Li, Bin, and Steven CH Hoi. "On-line portfolio selection with moving average reversion." *arXiv preprint arXiv:1206.4626*(2012).

Li, Feng. "The information content of forward-looking statements in corporate filings—A naïve Bayesian machine learning approach." *Journal of Accounting Research* 48.5 (2010): 1049-1102.

Li, Luyang, et al. "Truth discovery with memory network." *Tsinghua Science and Technology* 22.6 (2017): 609-618.

Liu, Bing. "Sentiment analysis and opinion mining." *Synthesis lectures on human language technologies* 5.1 (2012): 1-167.

Malkiel, Burton G., and Eugene F. Fama. "Efficient capital markets: A review of theory and empirical work." *The journal of Finance* 25.2 (1970): 383-417.

Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Vol. 1. No. 1. Cambridge: Cambridge university press, 2008.

Nassirtoussi, Arman Khadjeh, et al. "Text mining for market prediction: A systematic review." *Expert Systems with Applications* 41.16 (2014): 7653-7670.

Pan, Sinno Jialin, et al. "Cross-domain sentiment classification via spectral feature alignment." Proceedings of the 19th international conference on World wide web. ACM, 2010.

Pang, Bo, and Lillian Lee. "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts." *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2004.

Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. "Thumbs up?: sentiment classification using machine learning techniques." *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, 2002.

Ponomareva, Natalia, and Mike Thelwall. "Biographies or blenders: Which resource is best for cross-domain sentiment analysis?." International Conference on Intelligent Text Processing and Computational Linguistics. Springer Berlin Heidelberg, 2012.

Qian, Bo, and Khaled Rasheed. "Stock market prediction with multiple classifiers." *Applied Intelligence* 26.1 (2007): 25-33.

Schumaker, Robert P., and Hsinchun Chen. "Textual analysis of stock market prediction using breaking financial news: The AZFin text system." *ACM Transactions on Information Systems (TOIS)* 27.2 (2009): 12.

Smales, Lee A. "Time-varying relationship of news sentiment, implied volatility and stock returns." *Applied Economics* 48.51 (2016): 4942-4960.

Smailović, Jasmina, et al. "Predictive sentiment analysis of tweets: A stock market application." *Human-Computer Interaction and Knowledge Discovery in Complex, Unstructured, Big Data*. Springer Berlin Heidelberg, 2013. 77-88.

Sommar, Fredrik, and Milosz Wielondek. "Combining Lexicon-and Learning-based Approaches for Improved Performance and Convenience in Sentiment Classification." (2015).

Taboada, Maite, et al. "Lexicon-based methods for sentiment analysis." *Computational linguistics* 37.2 (2011): 267-307.

Tan, Songbo, and Yuefen Wang. "Weighted SCL model for adaptation of sentiment classification." Expert Systems with Applications 38.8 (2011): 10524-10531.

Wang, Sida, and Christopher D. Manning. "Baselines and bigrams: Simple, good sentiment and topic classification." *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*. Association for Computational Linguistics, 2012.

Wong, Pak Chung, Paul Whitney, and Jim Thomas. "Visualizing association rules for text mining." *Information Visualization, 1999.(Info Vis' 99) Proceedings. 1999 IEEE Symposium on*. IEEE, 1999.

Zhang, Dell, Riddhiman Ghosh, Mohamed Dekhil, Meichun Hsu, and Bing Liu. "Combining lexicon-based and learning-based methods for twitter sentiment analysis. " *HP Laboratories: Technical Report.* HPL*,* 2011. *89.*

7. Appended papers

In the following pages, we report the 3 published or accepted articles and the working paper that belongs to this collection of papers in the following order:

1- ***Public Mood Driven Asset Allocation: The Importance of Financial Sentiment in Portfolio Management***
Lorenzo Malandri, Frank Z. Xing, Carlotta Orsenigo, Carlo Vercellis and Erik Cambria
Cognitive Computation journal (October 2018)

2- ***Discovering Bayesian Market Views for Intelligent Asset Allocation***
Frank Z. Xing, Erik Cambria, Lorenzo Malandri, and Carlo Vercellis
2018 ECML-PDKK Conference (10 - 14 September, 2018, Dublin, Ireland)

3- ***Ensemble of Technical Analysis and Machine Learning for Market Trend Prediction***
Andrea Picasso Ratto, Simone Merello, Lorenzo Malandri, Yukun Ma, Luca Oneto and Erik Cambria
2018 IEEE SSCI Conference (18 - 21 November, 2018, Bengaluru, India)

4- ***Sentiment-Conditional Generation of Synthetic Text***
Working paper

Cognitive Computation manuscript No.
(will be inserted by the editor)

Public Mood Driven Asset Allocation: The Importance of Financial Sentiment in Portfolio Management

Lorenzo Malandri · Frank Z. Xing · Carlotta Orsenigo · Carlo Vercellis · Erik Cambria

Received: date / Accepted: date

Abstract
Background 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.
Methods 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.
Results Experiments performed on five portfolios show that long short-term memory networks are superior to multilayer 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-weighed portfolio.
Conclusion 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.

L. Malandri, C Orsenigo, and C. Vercellis
Department of Management, Economics and Industrial Engineering, Politecnico di Milano, Italy
E-mail:
{lorenzo.malandri,carlotta.orsenigo,carlo.vercellis}@polimi.it

F. Xing and E. Cambria
School of Computer Science and Engineering, Nanyang Technological University, Singapore
E-mail:
{zxing001,cambria}@ntu.edu.sg

1 Introduction

Financial markets are becoming increasingly important as economies grow. However, in todays society financial 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]. 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 [2].

A cornerstone of modern theory of finance is the Efficient Market Hypothesis (EMH) proposed by Fama [3], which 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 always divided the studies in finance and, from the mid-eighties onward, there have been many attempts to discover imperfections in the market, showing how some patterns can be unveiled [4–7] and disputing the EMH assumption.

Fama himself, in his later work [8], 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 unprecedent 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 [9], 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 [10] used sentiment surveys from companies and signal extraction techniques to derive investor sentiment from market indicators. They show that investors and employees 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 [11] has grown fast. In 2008, Tetlock et al. [12], 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 [13], 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 stocks prices). For example, Schumaker and Chen trained an SVM [14] 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 [15] 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 [16]. Regarding the social media sentiment data, several studies used Twitter [1,17–20] as source, given its standard format and the availability of APIs. Other scholars made use of aggregated news [21], message boards [22,23], or a combination of those sources. After texts are collected, sentiment analysis tools [24] 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 [25]. On one hand, the available datasets are in different format and lack of adequate information [2]. 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 [26].

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 [2], 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 [11] 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 Section 2.

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 [27] 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 [28] 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 [29,30] and the capital growth theory (CGT) [31,32]. 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 prob-

lem tailored for the online scenario [33] 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, every time a new batch of data comes in, the model is updated *without being retrained 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 [11]. 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. Last, our simulations show that including financial sentiment improves the performance of the optimized portfolio. This result is consistent over five portfolios analyses in our experiments and is statistically significant.

The remainder of this article is organized as follows: Section 2 provides an overview of the data collection process, of the portfolio allocation strategy and of the machine learning algorithms used in the present study; Section 3 describes the experimental setting and the computational results achieved; finally, Section 4 concludes the paper and discusses some future research directions.

2 Data and Methods Overview

2.1 Data collection

We gathered financial and sentiment data for 15 different stocks for the time period from January 24th 2012 to June 2nd 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 [34] and sentiment data through the StockFluence API [35]. 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 analyses every day about 1.5 Million comments between Twitter and articles.

2.2 Methodology

Consider N financial portfolios $p_n, n = 1, ..., N$. 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:

$$S_n = \{\mathbf{s}_n^1, ..., \mathbf{s}_n^T\},$$

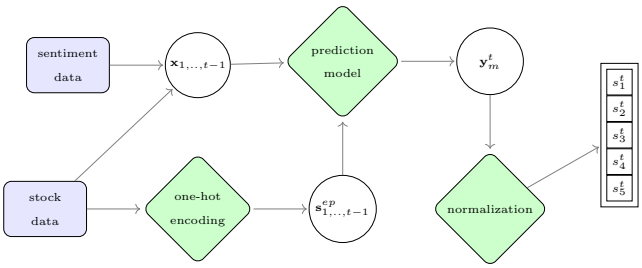
where $S_n \in \mathbb{R}^{M,T}$ and the generic term $\mathbf{s}_n^t$ is a M dimensional vector representing the weight to be allocated to each one of the M assets in period t for portfolio n. Our aim is to find, for each portfolio, the strategy S^* such that

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

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

Fig. 1: Model framework combining sentiment and lagged data



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 $\mathbf{s}^{ep}_t \in \mathbb{R}^M, t = 1, ..., T$ obtained through one-hot encoding in the following way

$$s^{ep}_{i,t} = \begin{cases} 1, & \text{if } r_{i,t+1} = \max\limits_m r_{m,t+1} \ , m = 1, ..., M \\ 0, & \text{o/w} \end{cases}$$

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

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 7 attributes for each stock. From Quandl, we obtained the daily adjusted closing price and volume, at the same time from StockFluence we collected the value positive, neutral and negative reviews, change and the sentiment score. Since each portfolio comprises 5 stocks, for each day we will have 35 predicting variables which, together with the 5 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 $\mathbf{s}^t_m$ 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 $\mathbf{s}^t = \{s^t_1, ..., s^t_M\}$ the condition $\sum_{m=1}^M s^t_m = 1$ must hold, the prediction vectors $\mathbf{y}_t$ will be normalized through the following formula:

$$z^t_m = \frac{y^t_m - \min\limits_m y^t_m}{\max\limits_m y^t_m - \min\limits_m y^t_m}, m = 1, ..., M,$$

$$s^t_m = \frac{z^t_m}{\sum_{m=1}^M z^t_m}, m = 1, ..., M.$$

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.

2.3 Prediction Models

Random Forest Classifier

Random forests [36] represent a powerful extension of decision trees [37], 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 Section 3.

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 [38]. Our implementation of the RFC is based on the Scikit-learn Python package [39].

Multi-Layer Perceptron

The financial stock market is well known to be highly non-linear, 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 [40].

Most of the research studies using neural networks for financial forecasting problems have successfully adopted a feed-forward MLP [41]. Consistently with some successful applications for financial time series prediction [42, 43], 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 Section 3.

MLPs have been implemented with Keras [44], a high-level neural networks 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 works very well in practice because they can learn long time dependencies, unlike traditional RNN which suffer from vanishing/exploding gradient when backpropagation 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 the 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.

3 Experiments

3.1 Model settings

The 15 selected stocks have been divided in 5 different portfolios. For the first three, we randomly selected 5 stocks for each one without repetition. The remaining two are composed by the 5 stocks which, in the selected period, performed best and worst respectively. 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 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 $t, t = 204, ..., T$ 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 5 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 3 values for nt (25, 50, 75) and 6 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 doesn't require to compute logarithmic functions and is therefore computationally less expensive.

For MLP and LSTM, we used a three layers 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 5 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 [45],

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

where N_s is the number of samples, N_i the number of input nodes, N_o the number of output nodes and α an arbitrary scaling factor usually ranging from 2 to 5 [45]. In our test, we selected the values 2 and 5.

3.2 The five portfolios

We constructed 5 virtual portfolios consisting of each one of 5 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 5 stocks with higher returns over the period considered (AAPL, BA, COST, MCO, SBUX), and the fifth one with the 5 titles with lowest returns (BP, INTC, MRVL, TGT, VMW). We constructed these two portfolios to evaluate the goodness of our algorithm in 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.

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	108280
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

3.3 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 3 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 return and Sharpe Ratio, although usually EW portfolios have higher risk and turnover [46, 47].

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 $w_T = w_0 S_n^\top R_n$ for each portfolio and each prediction model is reported in Table 2.

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

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 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 LSTM are RNNs which are able to capture time dependencies both in sentiment and financial time series.

For each prediction model the final value varies quite a lot across the 5 portfolios. This is due not only to the 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 (Figure 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.

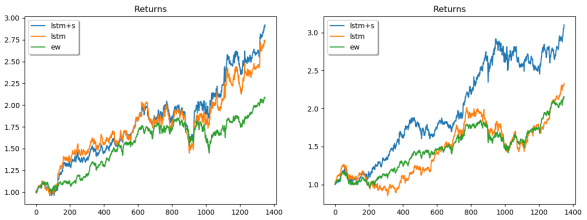
Table 3: Benchmark value

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

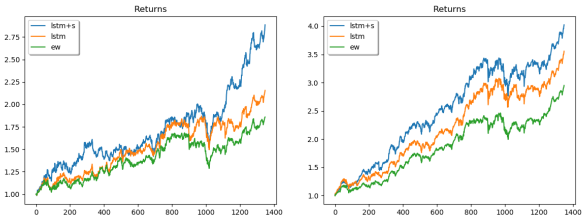
Among the selected prediction models, LSTM is the one which better captures the sentiment and gives better results in general. With LSTM, adding the sentiment scores as attributes increases the final weight of each of the 5 portfolios. In order to assess the statistical significance of this increment, we perform a paired t-test on the pairs w_T with and without sentiment for each portfolio. Results are presented in Table 4.

Fig. 2: Portfolio returns over the test period

(a) p1 (b) p2



(c) p3 (d) p4



(e) p5

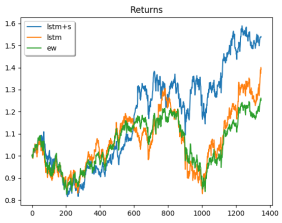


Table 4: Paired T-test: LSTM vs LSTM+s

	LSTM+s	LSTM
Mean	1.1865	1.0574
Variance	0.0057	0.0001
Observations	5	5
df	4	-
t Stat	3.6949	-
P(T¡=t)	0.0105	-

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 5 different portfolios and statistically significant and is captured by LSTM networks.

4 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 [48], 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 3 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 5 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 [49], despite 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. Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of computational science*, 2(1):1–8, 2011.
2. Arman Khadjeh Nassirtoussi, Saeed Aghabozorgi, Teh Ying Wah, and David Chek Ling Ngo. Text mining for market prediction: A systematic review. *Expert Systems with Applications*, 41(16):7653–7670, 2014.
3. Eugene F Fama. The behavior of stock-market prices. *The journal of Business*, 38(1):34–105, 1965.
4. Fischer Black. Noise. *The journal of finance*, 41(3):528–543, 1986.
5. J Bradford De Long, Andrei Shleifer, Lawrence H Summers, and Robert J Waldmann. Noise trader risk in financial markets. *Journal of political Economy*, 98(4):703–738, 1990.
6. Manolis G Kavussanos and Everton Dockery. A multivariate test for stock market efficiency: the case of ase. *Applied Financial Economics*, 11(5):573–579, 2001.
7. Bo Qian and Khaled Rasheed. Stock market prediction with multiple classifiers. *Applied Intelligence*, 26(1):25–33, 2007.
8. Eugene F Fama. Efficient capital markets: A review of theory and empirical work. *The journal of Finance*, 25(2):383–417, 1970.
9. Qing Li, LiLing Jiang, Ping Li, and Hsinchun Chen. Tensor-based learning for predicting stock movements. In *AAAI*, pages 1784–1790, 2015.
10. Gregory W Brown and Michael T Cliff. Investor sentiment and the near-term stock market. *Journal of empirical finance*, 11(1):1–27, 2004.
11. Frank Xing, Erik Cambria, and Roy Welsch. Natural language based financial forecasting: A survey. *Artificial Intelligence Review*, 50(1):49–73, 2018.
12. Paul C Tetlock, Maytal Saar-Tsechansky, and Sofus Macskassy. More than words: Quantifying language to measure firms' fundamentals. *The Journal of Finance*, 63(3):1437–1467, 2008.
13. Feng Li. The information content of forward-looking statements in corporate filings—a naïve bayesian machine learning approach. *Journal of Accounting Research*, 48(5):1049–1102, 2010.
14. Robert P Schumaker and Hsinchun Chen. Textual analysis of stock market prediction using breaking financial news: The azfin text system. *ACM Transactions on Information Systems (TOIS)*, 27(2):12, 2009.
15. Jasmina Smailović, Miha Grčar, Nada Lavrač, and Martin Žnidaršič. Predictive sentiment analysis of tweets: A stock market application. In *Human-Computer Interaction and Knowledge Discovery in Complex, Unstructured, Big Data*, pages 77–88. Springer, 2013.

16. Frank Xing, Erik Cambria, Lorenzo Malandri, and Carlo Vercellis. Discovering bayesian market views for intelligent asset allocation. In *ECML*, 2018.
17. Jianfeng Si, Arjun Mukherjee, Bing Liu, Qing Li, Huayi Li, and Xiaotie Deng. 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)*, volume 2, pages 24–29, 2013.
18. Gabriele Ranco, Darko Aleksovski, Guido Caldarelli, Miha Grčar, and Igor Mozetič. The effects of twitter sentiment on stock price returns. *PloS one*, 10(9):e0138441, 2015.
19. Angeliki Papana, Catherine Kyrtsou, Dimitris Kugiumtzis, and Cees Diks. Detecting causality in non-stationary time series using partial symbolic transfer entropy: evidence in financial data. *Computational Economics*, 47(3):341–365, 2016.
20. Ali Tafti, Ryan Zotti, and Wolfgang Jank. Real-time diffusion of information on twitter and the financial markets. *PloS one*, 11(8):e0159226, 2016.
21. Ryo Akita, Akira Yoshihara, Takashi Matsubara, and Kuniaki Uehara. Deep learning for stock prediction using numerical and textual information. In *Computer and Information Science (ICIS), 2016 IEEE/ACIS 15th International Conference on*, pages 1–6. IEEE, 2016.
22. Sanjiv R Das and Mike Y Chen. Yahoo! for amazon: Sentiment extraction from small talk on the web. *Management science*, 53(9):1375–1388, 2007.
23. Thien Hai Nguyen and Kiyoaki Shirai. 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)*, volume 1, pages 1354–1364, 2015.
24. Erik Cambria. Affective computing and sentiment analysis. *IEEE Intelligent Systems*, 31(2):102–107, 2016.
25. B Shravan Kumar and Vadlamani Ravi. A survey of the applications of text mining in financial domain. *Knowledge-Based Systems*, 114:128–147, 2016.
26. Erik Cambria, Soujanya Poria, Alexander Gelbukh, and Mike Thelwall. Sentiment analysis is a big suitcase. *IEEE Intelligent Systems*, 32(6):74–80, 2017.
27. Shinta Koyano and Kazushi Ikeda. Online portfolio selection based on the posts of winners and losers in stock microblogs. In *Computational Intelligence (SSCI), 2017 IEEE Symposium Series on*, pages 1–4. IEEE, 2017.
28. Frank Xing, Erik Cambria, and Roy Welsch. Intelligent asset allocation via market sentiment views. *IEEE Computational Intelligence Magazine*, 13(4), 2018.
29. Harry Markowitz. Portfolio selection. *The journal of finance*, 7(1):77–91, 1952.
30. Harry M Markowitz, G Peter Todd, and William F Sharpe. *Mean-variance analysis in portfolio choice and capital markets*, volume 66. John Wiley & Sons, 2000.
31. John L Kelly Jr. A new interpretation of information rate. In *The Kelly Capital Growth Investment Criterion: Theory and Practice*, pages 25–34. World Scientific, 2011.
32. Paul A Samuelson. Lifetime portfolio selection by dynamic stochastic programming. In *Stochastic Optimization Models in Finance*, pages 517–524. Elsevier, 1975.
33. Bin Li and Steven CH Hoi. Online portfolio selection: A survey. *ACM Computing Surveys (CSUR)*, 46(3):35, 2014.
34. Quandl API. Various end-of-day data. `https://www.quandl.com/data`, 2017.
35. StockFluence API. Financial sentiment data series. `https://www.stockfluence.com/`, 2017.
36. Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
37. Breiman Leo, Jerome H Friedman, Richard A Olshen, and Charles J Stone. Classification and regression trees. *Wadsworth International Group*, 1984.
38. Jigar Patel, Sahil Shah, Priyank Thakkar, and K Kotecha. Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications*, 42(1):259–268, 2015.
39. Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.
40. Chi-Jie Lu, Tian-Shyug Lee, and Chih-Chou Chiu. Financial time series forecasting using independent component analysis and support vector regression. *Decision Support Systems*, 47(2):115–125, 2009.
41. Rodolfo C Cavalcante, Rodrigo C Brasileiro, Victor LF Souza, Jarley P Nobrega, and Adriano LI Oliveira. Computational intelligence and financial markets: A survey and future directions. *Expert Systems with Applications*, 55:194–211, 2016.
42. Yakup Kara, Melek Acar Boyacioglu, and Ömer Kaan Baykan. Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the istanbul stock exchange. *Expert systems with Applications*, 38(5):5311–5319, 2011.
43. Fagner A de Oliveira, Cristiane N Nobre, and Luis E Zárate. Applying artificial neural networks to prediction of stock price and improvement of the directional prediction index–case study of petr4, petrobras, brazil. *Expert Systems with Applications*, 40(18):7596–7606, 2013.
44. François Chollet et al. Keras. `https://keras.io`, 2015.
45. Martin T Hagan, Howard B Demuth, Mark H Beale, and Orlando De Jesús. *Neural network design*, volume 20. Pws Pub. Boston, 1996.
46. Yuliya Plyakha, Raman Uppal, and Grigory Vilkov. Why does an equal-weighted portfolio outperform value-and price-weighted portfolios? *Available at SSRN 1787045*, 2012.
47. Robert Ferguson and David Schofield. Equal weighted portfolios perform better. *Financial Times*, 17 Oct 2010.
48. Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.
49. Saliha Minhas and Amir Hussain. From spin to swindle: Identifying falsification in financial text. *Cognitive computation*, 8(4):729–745, 2016.

Discovering Bayesian Market Views for Intelligent Asset Allocation

Frank Z. Xing[1], Erik Cambria[1], Lorenzo Malandri[2], and Carlo Vercellis[2]

[1] School of Computer Science and Engineering, Nanyang Technological University
[2] Data Mining and Optimization Research Group, Politecnico di Milano
cambria@ntu.edu.sg, carlo.vercellis@polimi.it

Abstract. Along with the advance of opinion mining techniques, public mood has been found to be a key element for stock market prediction. However, how market participants behavior is affected by public mood has been rarely discussed. Consequently, there has been little progress in leveraging public mood for the asset allocation problem, which is preferred in a trusted and interpretable way. In order to address the issue of incorporating public mood analyzed from social media, we propose to formalize public mood into market views, because market views can be integrated into the modern portfolio theory. In our framework, the optimal market views will maximize returns in each period with a Bayesian asset allocation model. We train two neural models to generate the market views, and benchmark the model performance on other popular asset allocation strategies. Our experimental results suggest that the formalization of market views significantly increases the profitability (5% to 10% annually) of the simulated portfolio at a given risk level.

Keywords: Market views • Public mood
Bayesian fusion • Asset allocation

1 Introduction

Sales and macroeconomic factors are some of the driving forces behind stock movements but there are many others. For example, the subjective views of market participants also have important effects. Along with the growing popularity of social media in the past decades, people tend to rapidly express and exchange their thoughts and opinions [21]. As a result, the importance of their views has dramatically risen [6]. Currently, stock movements are considered to be essentially affected by new information and the beliefs of investors [17].

Meanwhile, sentiment analysis has emerged as a new tool for analyzing the opinions shared on social media [7]. It is a branch of affective computing research that aims to classify natural language utterances as either positive or negative, but sometimes also neutral [9]. In the financial domain, sentiment analysis is frequently used to obtain a data stream of public mood toward a company, stock, or the economy. Public mood is the aggregation of individual sentiments which can be obtained and estimated from various sources, such as stock message boards [2,19], blogs, newspapers, and RSS (Really Simple Syndication) feeds [34].

Recently, Twitter has become a dominant microblogging platform on which many works rely for their investigations, such as [27,23,20]. Many previous studies support the claim that public mood helps to predict the stock market. For instance, the fuzzy neural network model considering public mood achieves high directional accuracy in predicting the market index. The mood time series is also proved a Granger cause of the market index [4]. Si et al. build a topic-based sentiment time series and predict the market index better with a vector autoregression model to interactively link the two series [26]. The Hurst exponents also suggest a long-term dependency for time series of mood extracted form financial news, similar to many market indices [8].

Despite the important role in stock market prediction, we assume that public mood does not directly effect the market: it does *indirectly* through market participants' views. The actions taken by market participants as agents, are dependent on their own views, and their knowledge about other agents' views. The changes of asset prices are the consequences of such actions. These assumptions are very different from econometric research using productivity, equilibrium, and business cycle models, e.g. [1], but closer to agent-based models, e.g. [14]. However, the mechanism of how market views are formed from public mood is heavily overlooked even in the latter case. An intuitive hypothesis could be: the happier the public mood, the higher the stock price. In the real-world market, however, this relationship is far more complicated. Therefore, existing superficial financial applications of AI do not appear convincing to professionals.

In this paper, we attempt to fill this gap by proposing a method for incorporating public mood to form market views computationally. To validate the quality of our views, we simulate the trading performance with a constructed portfolio. The key *contributions* of this paper can be summarized as follows:

1. We introduce a stricter and easier-to-compute definition of the market views based on a Bayesian asset allocation model. We prove that our definition is compatible, and has the equivalent expressiveness as the original form.

2. We propose a novel online optimization method to estimate the expected returns by solving temporal maximization problem of portfolio returns.

3. Our experiments show that the portfolio performance with market views blending public mood data stream is *better* than directly training a neural trading model without views. This superiority is robust for different models selected with the right parameters to generate market views.

The remainder of the paper is organized as follows: Sect. 2 explains the concept of Bayesian asset allocation; following, we describe the methodologies developed for modeling market views in Sect. 3; we evaluate such methodologies by running trading simulations with various experimental settings in Sect. 4 and show the interpretability of our model with an example in Sect. 5; finally, Sect. 6 concludes the paper and describes future work.

2 Bayesian Asset Allocation

The portfolio construction framework [18] has been a prevalent model for investment for more than half a century. Given the an amount of initial capital, the investor will need to allocate it to different assets. Based on the idea of trading-off between asset returns and the risk taken by the investor, the mean-variance method proposes the condition of an efficient portfolio as follows [18,29]:

$$\begin{aligned} &\text{maximize} && \overbrace{\sum_{i=1}^{N} \mu_i w_i}^{\text{return item}} \; \overbrace{-\frac{\delta}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} w_i \sigma_{ij} w_j}^{\text{risk item}} \\ &\text{subject to} && \sum_{i=1}^{N} w_i = 1, \; i = 1, 2, ..., N. \quad w_i \geq 0. \end{aligned} \tag{1}$$

where δ is an indicator of risk aversion, w_i denotes the weight of the corresponding asset in the portfolio, μ_i denotes the expected return of asset i, σ_{ij} is the covariance between returns of asset i and j. The optimized weights of an efficient portfolio is therefore given by the first order condition of Eq. 1:

$$w^* = (\delta \Sigma)^{-1} \mu \tag{2}$$

where Σ is the covariance matrix of asset returns and μ is a vector of expected returns μ_i. At the risk level of holding w^*, the efficient portfolio achieves the maximum combinational expected return.

However, when applying this mean-variance approach in real-world cases, many problems are faced. For example, the two moments of asset returns are difficult to estimate accurately [25], as they are non-stationary time series. The situation is worsened by the fact that, the Markowitz model is very sensitive to the estimated returns and volatility as inputs. The optimized weights can be very different because of a small error in μ or Σ. To address the limitation of the Markowitz model, a Bayesian approach that integrates the additional information of investor's judgment and the market fundamentals was proposed by Black and Litterman [3]. In the Black-Litterman model, the expected returns μ_{BL} of a portfolio is inferred by two antecedents: the equilibrium risk premiums Π of the market as calculated by the capital asset pricing model (CAPM), and a set of views on the expected returns of the investor.

The Black-Litterman model assumes that the equilibrium returns are normally distributed as $r_{eq} \sim \mathcal{N}(\Pi, \tau\Sigma)$, where Σ is the covariance matrix of asset returns, τ is an indicator of the confidence level of the CAPM estimation of Π. The market views on the expected returns held by an investor agent are also normally distributed as $r_{views} \sim \mathcal{N}(Q, \Omega)$.

Subsequently, the posterior distribution of the portfolio returns providing the views is also Gaussian. If we denote this distribution by $r_{BL} \sim \mathcal{N}(\bar{\mu}, \bar{\Sigma})$, then $\bar{\mu}$ and $\bar{\Sigma}$ will be a function of the aforementioned variables (see Fig. 1).

$$\left[\bar{\mu}, \bar{\Sigma}\right] = f(\tau, \Sigma, \Omega, \Pi, Q) \tag{3}$$

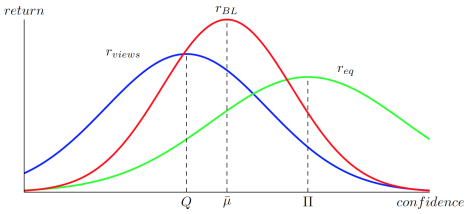


Fig. 1. The posterior distribution of the expected returns as in the Black-Litterman model, which has a mean between two prior distributions and a variance less than both of them.

The function can be induced from applying Bayes' theorem on the probability density function of the posterior expected returns:

$$pdf(\bar{\mu}) = \frac{pdf(\bar{\mu}|\Pi)\ pdf(\Pi)}{pdf(\Pi|\bar{\mu})} \tag{4}$$

Then, the optimized Bayesian portfolio weights have a similar form to Eq. 2, only substituting Σ and μ by $\bar{\Sigma}$ and $\bar{\mu}$:

$$w^*_{BL} = (\delta\bar{\Sigma})^{-1}\bar{\mu}. \tag{5}$$

The most common criticism of the Black-Litterman model is the subjectivity of investor's views. In other words, the model resorts to the good quality of the market views, while it leaves the question of how to actually form these views unanswered. In Sect. 3, we will investigate the possibility of automatically formalizing the market views from public mood distilled from the Web and the maximization of portfolio returns for each time period.

3 Methodologies

3.1 Modeling Market Views

The Black-Litterman model defines a view as a statement that the expected return of a portfolio has a normal distribution with mean equal to q and a standard deviation given by ω. This hypothetical portfolio is called a *view portfolio* [13]. In practice, there are two intuitive types of views on the market, termed *relative views* and *absolute views*, that we are especially interested in. Next, we introduce the formalization of these two types of views.

Because the standard deviation ω can be interpreted as the confidence of expected return of the view portfolio, a *relative view* takes the form of "I have ω_1 confidence that asset x will outperform asset y by a% (in terms of expected return)"; an *absolute view* takes the form of "I have ω_2 confidence that asset z will outperform the (whole) market by b%". Consequently, for a portfolio consisting of n assets, a set of k views can be represented by three matrices $P_{k,n}$, $Q_{k,1}$, and $\Omega_{k,k}$.

$P_{k,n}$ indicates the assets mentioned in views. The sum of each row of $P_{k,n}$ should either be 0 (for relative views) or 1 (for absolute views); $Q_{k,1}$ is a vector comprises expected returns for each view. Mathematically, the confidence matrix $\Omega_{k,k}$ is a measure of covariance between the views. The Black-Litterman model assumes that the views are independent of each other, so the confidence matrix can be written as $\Omega = diag(\omega_1, \omega_2, ..., \omega_n)$. In fact, this assumption will not affect the expressiveness of the views as long as the k views are compatible (not self-contradictory). Because when $\Omega_{k,k}$ is not diagonal, we can always do spectral decomposition: $\Omega = V\Omega^{\Lambda}V^{-1}$. Then we write the new mentioning and new expected return matrices as $P^{\Lambda} = V^{-1}P$, $Q^{\Lambda} = V^{-1}Q$, where Ω^{Λ} is diagonal. Under these constructions, we introduce two important properties of the view matrices in Theorem 1 and Theorem 2.

Theorem 1 (Compatibility of Independent Views). *Any set of independent views are compatible.*

Proof. Compatible views refer to views that can hold at the same time. For example, {asset x will outperform asset y by 3%, asset y will outperform asset z by 5%, asset x will outperform asset z by 8%} is compatible. However, if we change the third piece of view to "asset z will outperform asset x by 8%", the view set becomes self-contradictory. Because the third piece of view is actually a deduction from the former two, the view set is called "not independent".

Assume there is a pair of incompatible views $\{p, q\}$ and $\{p, q'\}$, $q \neq q'$. Both views are either explicitly stated or can be derived from a set of k views. Hence, there exist two different linear combinations, such that:

$$\sum_{i=1}^{k} a_i p_i = p \qquad \sum_{i=1}^{k} a_i q_i = q$$

$$\sum_{i=1}^{k} b_i p_i = p \qquad \sum_{i=1}^{k} b_i q_i = q'$$

where $(a_i - b_i)$ are not all zeros.

Thus, we have $\sum_{i=1}^{k}(a_i - b_i)p_i = \mathbf{0}$, which means that matrix P is rank deficient and the k views are not independent. According to the law of contrapositive, the statement "all independent view sets are compatible" is true. □

Theorem 2 (Universality of Absolute View Matrix). *Any set of independent relative and absolute views can be expressed with a non-singular absolute view matrix.*

Proof. Assume a matrix P with r relative views and $(k - r)$ absolute views.

$$P_{k,n} = \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{r,1} & p_{r,2} & \cdots & p_{r,n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{k,1} & p_{k,2} & \cdots & p_{k,n} \end{bmatrix}$$

The corresponding return vector is $Q = (q_1, q_2, \ldots, q_k)$, the capital weight vector for assets is $w = (w_1, w_2, \ldots, w_k)$. Hence, we can write $(r+1)$ equations with regard to r new variables $\{q'_1, q'_2, ..., q'_r\}$, where $j = 1, 2, ..., r$:

$$1 + q'_j = \sum_{i \neq j}^{r} (1 + q'_i) \frac{w_i}{\sum\limits_{s \neq j} w_s} (1 + q_j)$$

$$\sum_{i=1}^{r} q'_i w_i + \sum_{i=r+1}^{k} q_i w_i = Q w^{\intercal}$$

If we consider $\{asset_{r+1}, \ldots, asset_k\}$ to be one asset, return of this asset is decided by $P_{r,n}$. Hence, r out of the $(r+1)$ equations above are independent.

According to Cramer's rule, there exists a unique solution $Q' = (q'_1, q'_2, \ldots, q'_r, q_{r+1}, \ldots, q_k)$ to the aforementioned $(r+1)$ equations, such that view matrices $\{P', Q'\}$ is equivalent to view matrices $\{P, Q\}$ for all the assets considered, where

$$P'_{k,n} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & p_{r,r} = 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ p_{k,1} & p_{k,2} & \cdots & p_{k,n} \end{bmatrix}.$$

Now, $P'_{k,n}$ only consists of absolute views. By deleting those dependent views, we can have a non-singular matrix that only consists of absolute views and is compatible. □

Given Theorem 1 and Theorem 2, without loss of generality, we can use the following equivalent yet stricter definition of market views to reduce computational complexity.

Definition 1. *Market views on n assets can be represented by three matrices $P_{n,n}$, $Q_{n,1}$, and $\Omega_{n,n}$, where $P_{n,n}$ is an identity matrix; $Q_{n,1} \in \mathbb{R}^n$; $\Omega_{n,n}$ is a nonnegative diagonal matrix.*

3.2 The Confidence Matrix

In the most original form of the Black-Litterman model, the confidence matrix Ω is set manually according to investors' experience. Whereas in the numerical example given by [13], the confidence matrix is derived from the equilibrium covariance matrix:

$$\hat{\Omega}_0 = diag(P(\tau\Sigma)P') \tag{8}$$

This is because $P(\tau\Sigma)P'$ can be understood as a covariance matrix of the expected returns in the views as well. Using our definition, it is easier to understand this estimation, because P is an identity matrix, $P(\tau\Sigma)P'$ is already diagonal. The underlying assumption is that the variance of an absolute view on asset i is proportional to the volatility of asset i. In this case, the estimation of Ω utilizes past information of asset price volatilities.

3.3 Optimal Market Views

We obtain the optimal market views $\{P, Q, \Omega\}$ in a hybrid way, first we adopt the confidence matrix $\hat{\Omega}_0$, then Q can be derived from the inverse optimization problem using the Black-Litterman model.

We start from the optimal portfolio weights that maximize the portfolio returns for each period t. Obviously, without short selling and transaction fees, one should re-invest his whole capital daily to the fastest-growing asset in the next time period.

The optimal holding weights for each time period t thus take the form of a one-hot vector, where $\oslash$ and $\odot$ denote element-wise division and product:

$$w_t^* = \text{argmax} \;\; w_t \oslash price_t \odot price_{t+1} \tag{9}$$

Let this w_t^* be the solution to Eq. 1, we will have:

$$w_t^* = (\delta \bar{\Sigma}_t)^{-1} \bar{\mu}_t \tag{10}$$

where the Black-Litterman model gives[3]:

$$\bar{\Sigma}_t = \Sigma_t + [(\tau\Sigma_t)^{-1} + P'\hat{\Omega}_t^{-1}P]^{-1} \tag{11}$$

$$\bar{\mu}_t = [(\tau\Sigma_t)^{-1} + P'\hat{\Omega}_t^{-1}P]^{-1}[(\tau\Sigma_t)^{-1}\Pi_t + P'\hat{\Omega}_t^{-1}Q_t] \tag{12}$$

According to Eq. 10, 11, and 12, the optimal expected returns for our market views for each period t is:

$$\begin{aligned} Q_t^* &= \hat{\Omega}_{0,t}\{[\,(\tau\Sigma_t)^{-1} + P'\hat{\Omega}_{0,t}^{-1}P\,]\,\bar{\mu}_t - (\tau\Sigma_t)^{-1}\Pi_t\} \\ &= \delta[\,\hat{\Omega}_{0,t}(\tau\Sigma_t)^{-1} + \mathbb{I}\,]\,\bar{\Sigma}_t w_t^* - \hat{\Omega}_{0,t}(\tau\Sigma_t)^{-1}\Pi_t \\ &= \delta[\,\hat{\Omega}_{0,t}(\tau\Sigma_t)^{-1} + \mathbb{I}\,]\,[\,\Sigma_t + [(\tau\Sigma_t)^{-1} + \hat{\Omega}_t^{-1}]^{-1}\,]w_t^* \\ &\quad - \hat{\Omega}_{0,t}(\tau\Sigma_t)^{-1}\Pi_t \end{aligned} \tag{13}$$

3.4 Generating Market Views with Neural Models

Eq. 13 provides a theoretical perspective on determining the expected return of optimal market views. However, computing w_t^* requires future asset prices, which is not accessible. Therefore, the feasible approach is to learn approximating Q_t^* with historical data and other priors as input. We use the time series of asset prices, trading volumes, and public mood data stream to train neural models (nn) for this approximation problem of optimal market views:

$$\hat{Q}_t = nn(prices, volumes, sentiments;\, Q_t^*) \tag{14}$$

We denote the time series of asset prices $price_{t-k}, price_{t-k+1}, ..., price_t$ by a lag operator $\mathcal{L}^{0\sim k}price_t$. The notation of trading volumes follows a similar form. Then the model input at each time point: $[\mathcal{L}^{0\sim k}price_t, \mathcal{L}^{0\sim k}volume_t, sentiment_t, capital_t]$ can be denoted by $[p, v, s, c]_t$ in short.

Two types of neural models, including a neural-fuzzy approach and a deep learning approach are trained for comparison. Fig. 2 provides an illustration of the online training process using LSTM, where $\hat{Q}$ is the output.

[3] The proof of Eq. 11 and 12 can be found from the appendix of [24].

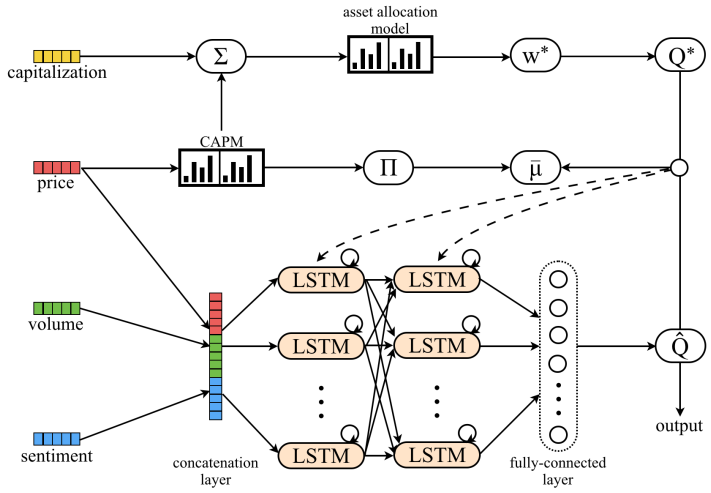


Fig. 2. Model training process (LSTM) with/without sentiment information.

Dynamic evolving neural-fuzzy inference system (DENFIS) is a neural network model with fuzzy rule nodes [16]. The partitioning of which rule nodes to be activated is dynamically updated with the new distribution of incoming data. This evolving clustering method (ECM) features the model with stability and fast adaptability. Comparing to many other fuzzy neural networks, DENFIS performs better in modeling nonlinear complex systems [32].

Considering the financial market as a real-world complex system, we learn the first-order Takagi-Sugeno-Kang type rules online. Each rule node has the form of:

$$IF\ \mathcal{L}^{0\sim k} attribute_{t,i} = pattern_i,\ i = 1, 2, ..., N$$
$$THEN\ \hat{Q}_t = f_{1,2,...,N}([p, v, s]_t)$$

where we have 3 attributes and $(2^N - 1)$ candidate functions to activate. In our implementation of the DENFIS model, all the membership functions are symmetrical and triangular, which can be defined by two parameters $b \pm d/2$. b is where the membership degree equals to 1; d is the activation range of the fuzzy rule. In our implementation, b is iteratively updated by linear least-square estimator of existing consequent function coefficients.

Long short-term memory (LSTM) is a type of recurrent neural network with gated units. This unit architecture is claimed to be well-suited for learning to predict time series with an unknown size of lags and long-term event dependencies. Early attempts, though not very successful [11], have been made to apply LSTM to time series prediction. It is now recognized that though LSTM cells can have many variants, their performance across different tasks are similar [12].

Therefore, we use a vanilla LSTM unit structure. Our implementation of LSTM cells follows the update rules of the input gate, forget gate, and output gate as in Eq. 15:

$$\begin{aligned} i_t &= \sigma(W_i \cdot [\, h_{t-1}, [p, v, s]_t \,] + b_i) \\ f_t &= \sigma(W_f \cdot [\, h_{t-1}, [p, v, s]_t \,] + b_f) \\ o_t &= \sigma(W_o \cdot [\, h_{t-1}, [p, v, s]_t \,] + b_o) \end{aligned} \tag{15}$$

where σ denotes the sigmoid function, h_{t-1} is the output of the previous state, W is a state transfer matrix, and b is the bias.

The state of each LSTM cell c_t is updated by:

$$\begin{aligned} c_t &= f_t \odot c_{t-1} + i_t \odot (W_c \cdot [\, h_{t-1}, [p, v, s]_t \,] + b_c) \\ h_{t-1} &= o_t \odot \tanh(c_{t-1}) \end{aligned} \tag{16}$$

We make the training process online as well, in a sense that each time a new input is received, we use the previous states and parameters of LSTM cells $[c_{t-1}, \mathbf{W}, \mathbf{b}]$ to initialize the LSTM cells for period t.

4 Experiments

To evaluate the quality and effectiveness of our formalization of market views, we run trading simulations with various experimental settings.

4.1 Data

The data used in this study are publicly available on the Web[4]. We obtain the historical closing price of stocks and daily trading volumes from the Quandl API[5]; the market capitalization data from Yahoo! Finance; the daily count and intensity of company-level sentiment time series from PsychSignal[6]. The sentiment intensity scores are computed from multiple social media platforms using NLP techniques. Fig. 3 depicts a segment example of the public mood data stream. The market is closed on weekends, so a corresponding weekly cycle of message volume can be observed.

We investigate a window of around 8 years (2800 days). All the time series are trimmed from 2009-10-05 to 2017-06-04. For missing values such as the closing prices on weekends and public holidays, we fill them with the nearest historical data to train the neural models. The lagged values we use for both price and trading volume consist of 4 previous days and a moving average of the past 30 days, that is, the input of our neural models takes the form of Eq. 17 and 18:

[4] http://github.com/fxing79/ibaa
[5] http://www.quandl.com/tools/api
[6] http://psychsignal.com/

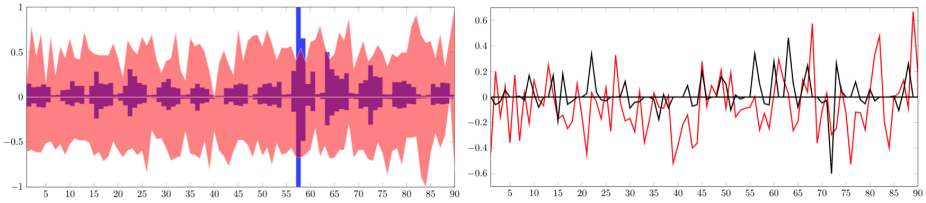


Fig. 3. The volume of daily tweets filtered by cashtag `AAPL` (blue, left); average sentiment intensity (red, left); net sentiment polarity (red, right); daily returns (black, right) in a time period of 90 days (2017-03-04 to 2017-06-04). All the series are normalized.

$$\mathcal{L}^{0\sim k} price_t = (p_t, p_{t-1}, p_{t-2}, p_{t-3}, \frac{\sum_{i=1}^{30} p_i}{30}) \tag{17}$$

$$\mathcal{L}^{0\sim k} volume_t = (v_t, v_{t-1}, v_{t-2}, v_{t-3}, \frac{\sum_{i=1}^{30} v_i}{30}) \tag{18}$$

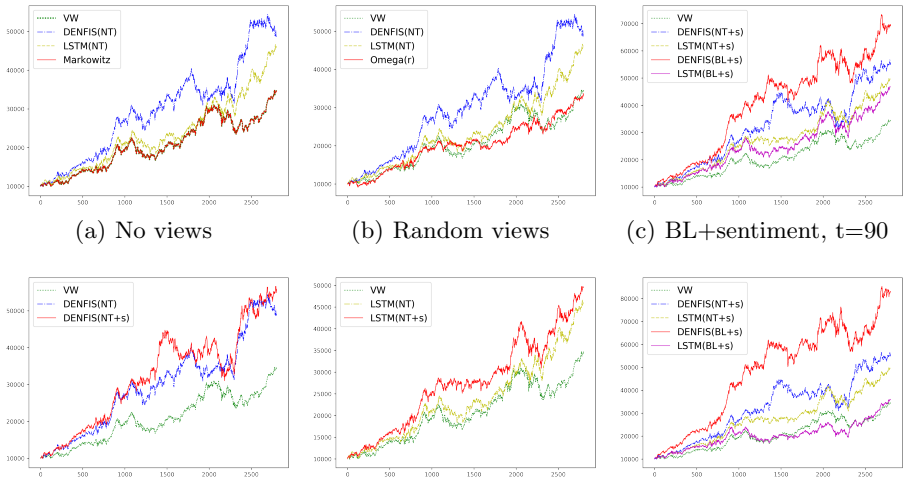
4.2 Trading Simulation

We construct a virtual portfolio consisting of 5 big-cap stocks: Apple Inc (`AAPL`), Goldman Sachs Group Inc (`GS`), Pfizer Inc (`PFE`), Newmont Mining Corp (`NEM`), and Starbucks Corp (`SBUX`). This random selection covers both the NYSE and NASDAQ markets and diversified industries, such as technology, financial services, health care, consumer discretionary etc. During the period investigated, there were two splits: a 7-for-1 split for `AAPL` on June 9th 2014, and a 2-for-1 split for `SBUX` on April 9th 2015. The prices per share are adjusted according to the current share size for computing all related variables, however, dividends are not taken into account. We benchmark our results with two portfolio construction strategies:

1) The value-weighted portfolio (VW): we re-invest daily according to the percentage share of each stock's market capitalization. In this case, the portfolio performance will be the weighted average of each stock's performance. This strategy is fundamental, yet empirical study [10] shows that beating the market even before netting out fees is difficult.

2) The neural trading portfolio (NT): we remove the construction of market views and directly train the optimal weights of daily position with the same input. For this black-box strategy, we can not get any insight on how this output portfolio weight comes about.

In the simulations, we assume no short selling, taxes, or transaction fees, and we assume the portfolio investments are infinitely divisible, starting from $10,000$ dollars. We construct portfolios with no views ($\Omega_\varnothing$, in this case the degenerate portfolio is equivalent to Markowitz's mean-variance portfolio using historical return series to estimate covariance matrix as a measure of risk), random views (Ω_r), the standard views using the construction of Black-Litterman model (Ω_0), with and without our sentiment-induced expected returns (s). The trading performances are demonstrated in Fig. 4.



(a) No views (b) Random views (c) BL+sentiment, t=90

(d) DENFIS+sentiment (e) LSTM+sentiment (f) BL+sentiment, t=180

Fig. 4. Trading simulation performance with different experimental settings: (x-axis: number of trading days; y-axis: cumulative returns). In particular, we use a timespan of 90 and 180 days for our approach. The performance of neural trading is independent from timespan, accordingly the two neural models are compared in 4(d) and 4(e) respectively for better presentation.

Following the previous research [13], we set the risk aversion coefficient $\delta = 0.25$ and confidence level of CAPM, $\tau = 0.05$. Let the activation range of fuzzy membership function $d = 0.21$, we obtain 21 fuzzy rule nodes from the whole online training process of DENFIS. This parameter minimizes the global portfolio weight error. For the second neural model using deep learning, we stack two layers of LSTMs followed by a densely connected layer. Each LSTM layer has 3 units; the densely connected layer has 50 neurons, which is set times larger than the number of LSTM units. We use the mean squared error of vector Q as the loss function and the rmsprop optimizer [30] to train this architecture. We observe fast training error convergence in our experiments.

4.3 Performance Metrics

Diversified metrics have been proposed to evaluate the performance of a given portfolio [5,15,31]. We report four metrics in our experiments.

Root mean square error (RMSE) is a universal metric for approximation problems. It is widely used for engineering and data with normal distribution and few outliers. We calculate the RMSE of our realized portfolio weights to the optimal weights:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \|w_i - \hat{w}_i\|^2} \tag{19}$$

Annualized return (AR) measures the profitability of a given portfolio. We calculate the geometric mean growth rate per year, which is also referred to as compound annual growth rate (CAGR) for these 2800 days.

Sharpe ratio (SR) is a risk-adjusted return measure. We choose the value-weighted portfolio as a base, consequently the Sharpe ratio of VW will be 1:

$$\mathrm{SR} = \frac{\mathbb{E}(R_{portfolio}/R_{VW})}{\sigma(R_{portfolio})/\sigma(R_{VW})} \tag{20}$$

SR uses the standard deviation of daily returns as the measure of risk. Note that to distinguish between good and bad risk, we can also use the standard deviation of downside returns only [28]. Our results suggest that the Sortino ratios, which are not reported due to page limit, are very close to SRs and lead to the same conclusion.

The maximum drawdown (MDD) measures the maximum possible percentage loss of an investor:

$$\mathrm{MDD} = \max_{0<t<\tau} \left\{ \frac{Value_t - Value_\tau}{Value_t} \right\} \tag{21}$$

Asset allocation strategies with large MDD are exposed to the risk of withdrawal. Table 1 presents the metrics.

Table 1. Performance metrics for various portfolio construction strategies, timespan=90 and 180 days. Top three metrics are in bold.

	RMSE	SR	MDD(%)	AR(%)
VW	0.8908	1.00	25.81	17.49
Markowitz90($\Omega_\varnothing$)	0.9062	1.00	25.81	17.51
Markowitz180($\Omega_\varnothing$)	0.8957	1.00	25.82	17.45
BL90(Ω_r)	0.9932	0.90	**23.47**	17.17
BL180(Ω_r)	0.9717	1.06	**20.59**	22.31
DENFIS(NT)	0.9140	**2.94**	29.84	23.09
DENFIS(NT+s)	0.9237	**4.35**	**23.07**	**25.16**
DENFIS(BL90+s)	0.9424	1.52	24.44	**28.69**
DENFIS(BL180+s)	0.9490	**1.58**	24.19	**29.49**
LSTM(NT)	**0.8726**	1.38	25.68	22.10
LSTM(NT+s)	0.8818	1.42	25.96	23.21
LSTM(BL90+s)	**0.8710**	1.34	25.90	22.33
LSTM(BL180+s)	**0.8719**	1.07	24.88	17.68

4.4 Findings

We have some interesting observations from Fig. 4 and Table 1. SR and AR are usually considered as the most important, and besides, RMSE and MDD

are all very close in our experiments. The correlation between RMSE and the other three metrics is weak, though it is intuitive that if the realized weights are close to the optimal weights, the portfolio performance should be better. On the contrary, the LSTM models seem to overfit as they are trained on the mean squared error of weights or expected return of views [22]. However, as mentioned in Sect. 1, the relationship between weights and daily returns is non-linear. Therefore, *holding portfolio weights that are close to the optimal weights does not necessarily means that the AR must be higher*. In fact, it is dangerous to use any seemingly reasonable metrics outside the study of asset allocation, such as directional accuracy of price change prediction [4,33], to evaluate the expected portfolio performance.

The Markowitz portfolio ($\Omega_\varnothing$) displays a very similar behavior to the market-following strategy. This is consistent with the inefficacy of the mean-variance approach in practice mentioned by previous studies: holding the Markowitz portfolio is holding the market portfolio. In fact, if the CAPM holds, the market portfolio already reflects the adjustments to risk premiums, that is, fewer market participants will invest on highly risky assets, for this reason their market capitalization will be smaller as well.

However, the Black-Litterman model does not always guarantee better performance over the Markowitz portfolio. "Garbage in, garbage out" still holds for this circumstance. Given random views (Ω_r), it can be worse than market-following in terms of both SR and AR. The lesson learned is that *if the investor knows nothing, it is better to hold no views and follow the market than pretending to know something.*

In our experiments, DENFIS generally performs better than LSTM models, achieving higher SRs and ARs. The reason may be LSTM models adapt faster to the incoming data, whereas financial time series are usually very noisy. The ECM mechanism provides DENFIS models with converging learning rates, which may be beneficial to the stability of memorized rules. However, it is important to note that *the ARs for both neural models improve with the blending of sentiments.* The timespan used to estimate correlation and volatility of assets seems not that critical. DENFIS models perform better with longer timespan, while LSTM models perform better with shorter timespan. The Markowitz portfolio is less affected by timespan.

5 A Story

One of the main advantages of our formalization and computing of market views is that some *transparency* is brought to the daily asset reallocation decisions. In most cases, a stock price prediction system based on machine learning algorithms cannot justify "why he thinks that price will reach that predicted point". Unlike these systems, our method can tell a story of the portfolio to professional investors and advice seekers. Take June 1st 2017 as an example:

"On June 1st 2017, we observe 164 positive opinions of polarity +1.90, 58 negative opinions of polarity −1.77 on AAPL stock; 54 positive opinions of polarity +1.77, 37 negative opinions of polarity −1.53 on GS stock; 5 positive opinions of polarity +2.46, 1 negative opinion of polarity −1.33 on PFE stock; no opinion on NEM stock; and 9 positive opinions of polarity +1.76, 5 negative opinions of polarity −2.00 on SBUX stock. Given the historical prices and trading volumes of the stocks, we have 6.29% confidence that AAPL will outperform the market by −70.11%; 23.50% confidence that GS will outperform the market by 263.28%; 0.11% confidence that PFE will outperform the market by −0.50%; 1.21% confidence that SBUX will outperform the market by 4.57%. Since our current portfolio invests 21.56% on AAPL, 25.97% on GS, 29.43% on PFE, and 23.04% on SBUX, by June 2nd 2017, we should withdraw all the investment on AAPL, 2.76% of the investment on GS, 81.58% of the investment on PFE, and 30.77% of the investment on SBUX, and re-invest them onto NEM."

6 Conclusion and Future Work

In previous studies which have considered sentiment information for financial forecasting, the role of the investor as a market participant is often absent. In this paper, we present a novel approach to incorporate market sentiment by fusing public mood data stream into the Bayesian asset allocation framework.

This work is pioneering in formalizing sentiment-induced market views. Our experiments show that the market views provide a powerful method to asset management. We also confirm the efficacy of public mood data stream based on social media for developing asset allocation strategies.

A limitation of this work is that we fixed a portfolio with five assets, though in practice the portfolio selection problem is of equal importance. How to assess the quality of sentiment data is not discussed in this paper as well. We are not at the stage to distinguish or detect opinion manipulation though concern like the open networks are rife with bots does exist. Another limitation is that survivor bias is not taken into account: the risk that assets selected in the portfolio may quit the market or suffer from a lack of liquidity. This problem can be alleviated by only including high quality assets. In the future, we will study examining the quality of sentiment data obtained using different content analysis approaches. We also plan to develop a Bayesian asset allocation model that can deal with market frictions.

References

1. Angeletos, G., La'O, J.: Sentiments. Econometrica **81**(2), 739–779 (2013)
2. Antweiler, W., Frank, M.Z.: Is all that talk just noise? the information content of internet stock message boards. The Journal of Finance **59**(3), 1259–94 (2004)
3. Black, F., Litterman, R.: Asset allocation: Combining investor view with market equilibrium. The Journal of Fixed Income **1**, 7–18 (1991)
4. Bollen, J., Mao, H., Zeng, X.: Twitter mood predicts the stock market. Journal of Computational Science **2**(1), 1–8 (2011)

5. Brandt, M.W.: Portfolio choice problems, In Handbook of Financial Econometrics, vol. 1, chap. 5, pp. 269–336. Elsevier B.V., Oxford, UK (2009)
6. Cambria, E.: Affective computing and sentiment analysis. IEEE Intelligent Systems **31**(2), 102–107 (2016)
7. Cambria, E., Das, D., Bandyopadhyay, S., Feraco, A. (eds.): A Practical Guide to Sentiment Analysis. Springer International Publishing, Switzerland (2017)
8. Chan, S.W., Chong, M.W.: Sentiment analysis in financial texts. Decision Support Systems **94**, 53–64 (2017)
9. Chaturvedi, I., Ragusa, E., Gastaldo, P., Zunino, R., Cambria, E.: Bayesian network based extreme learning machine for subjectivity detection. Journal of the Franklin Institute **355**(4), 1780–97 (2018)
10. Fama, E.F., French, K.R.: Luck versus skill in the cross-section of mutual fund returns. The Journal of Finance **65**(5), 1915–47 (2010)
11. Gers, F.A., Eck, D., Schmidhuber, J.: Applying lstm to time series predictable through time-window approaches. In: ICANN, LNCS, vol. 2130. pp. 669–676 (2001)
12. Greff, K., Srivastava, R.K., Koutnik, J., Steunebrink, B.R., Schmidhuber, J.: Lstm: A search space odyssey. IEEE TNNLS **28**(10), 2222–32 (2017)
13. He, G., Litterman, R.: The intuition behind black-litterman model portfolios. Goldman Sachs working paper (1999). https://doi.org/10.2139/ssrn.334304
14. Hommes, C.: The New Palgrave Dictionary of Economics, chap. Interacting agents in finance. Basingstoke: Palgrave Macmillan, 2 edn. (2008)
15. Hyndman, R.J., Koehler, A.B.: Another look at measures of forecast accuracy. International Journal of Forecasting **22**(4), 679–688 (2006)
16. Kasabov, N.K., Song, Q.: Denfis: Dynamic evolving neural-fuzzy inference system and its application for time-series prediction. IEEE Transactions on Fuzzy Systems **10**, 144–154 (2002)
17. Li, Q., Jiang, L., Li, P., Chen, H.: Tensor-based learning for predicting stock movements. In: AAAI. pp. 1784–90 (2015)
18. Markowitz, H.: Portfolio selection. The Journal of Finance **7**, 77–91 (1952)
19. Nguyen, T.H., Shirai, K.: Topic modeling based sentiment analysis on social media for stock market prediction. In: ACL. pp. 1354–64 (2015)
20. Nofer, M., Hinz, O.: Using twitter to predict the stock market: Where is the mood effect? Business & Information Systems Engineering **57**(4), 229–242 (2015)
21. O'Connor, B., Balasubramanyan, R., Routledge, B.R., Smith, N.A.: From tweets to polls: Linking text sentiment to public opinion time series. In: ICWSM. pp. 122–129 (2010)
22. Pant, P.N., Starbuck, W.H.: Innocents in the forest: Forecasting and research methods. Journal of Management **16**(2), 433–460 (1990)
23. Ranco, G., Aleksovski, D., Caldarelli, G., Grčar, M., Mozetič, I.: The effects of twitter sentiment on stock price returns. PLoS ONE **10**(9), 1–21 (2015)
24. Satchell, S., Scowcroft, A.: A demystification of the black-litterman model: Managing quantitative and traditional portfolio construction. Journal of Asset Management **1**(2), 138–150 (2000)
25. Shen, W., Wang, J.: Portfolio selection via subset resampling. In: AAAI. pp. 1517–23 (2017)
26. Si, J., Mukherjee, A., Liu, B., Li, Q., Li, H., Deng, X.: Exploiting topic based twitter sentiment for stock prediction. In: ACL. pp. 24–29 (2013)
27. Smailović, J., Grčar, M., Lavrač, N., Žnidaršič, M.: Predictive sentiment analysis of tweets: A stock market application. In: LNCS. vol. 7947, pp. 77–88. Springer, Berlin (2013)

28. Sortino, F.A., Price, L.N.: Performance measurement in a downside risk framework. The Journal of Investing **3**, 59–64 (1994)
29. Steinbach, M.C.: Markowitz revisited: Mean-varian-ce models in financial portfolio analysis. SIAM Review **43**(1), 31–85 (2001)
30. Tieleman, T., Hinton, G.E.: Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning (2012)
31. Xing, F.Z., Cambria, E., Welsch, R.E.: Natural language based financial forecasting: A survey. Artificial Intelligence Review **50**(1), 49–73 (2018)
32. Xing, F.Z., Cambria, E., Zou, X.: Predicting evolving chaotic time series with fuzzy neural networks. In: IJCNN. pp. 3176–83 (2017)
33. Yoshihara, A., Seki, K., Uehara, K.: Leveraging temporal properties of news events for stock market prediction. Artificial Intelligence Research **5**(1), 103–110 (2016)
34. Zhang, W., Skiena, S.: Trading strategies to exploit blog and news sentiment. In: ICWSM. pp. 375–378 (2010)

Ensemble of Technical Analysis and Machine Learning for Market Trend Prediction

1st Andrea Picasso Ratto
DIBRIS
University of Genova
Via Opera Pia 11A, I-16145 Genova, Italy
andrea.picasso@smartlab.ws

2th Simone Merello
DIBRIS
University of Genova
Via Opera Pia 11A, I-16145 Genova, Italy
simone.merello@smartlab.ws

3rd Yukun Ma
Rolls-Royce@NTU Corporate Lab
Nanyang Technological University
50 Nanyang Ave, Singapore
mayu0010@e.ntu.edu.sg

4rd Lorenzo Malandri
Department of Management, Economics and Engineering, Politecnico di Milano
Via raffaele lambruschini 4b, 20155, Milano
lorenzo.malandri@polimi.it

5th Oneto Luca
DIBRIS
University of Genova
Via Opera Pia 11A, I-16145 Genova, Italy
luca.oneto@unige.it

6rd Erik Cambria
SenticNet Lab
Nanyang Technological University
50 Nanyang Ave, Singapore
cambria@ntu.edu.sg

***Abstract*—Over the last twenty years, researchers and practitioners have attempted in many ways to effectively predict market trends. Till date, however, no satisfactory solution has been found. Many approaches have been applied to predict market trends, from technical analysis to fundamental analysis passing through sentiment analysis. A promising research direction is to exploit market technical indicators together with market sentiments extracted from social media for predicting market directional movements. In this paper, we propose a new approach that leverages technical analysis to predict market directional movements. In particular, we aim to predict the directional movement of the NASDAQ's most capitalized stocks by solving a classification problem. The results on real-world data show that our proposal achieves interesting performance when predicting the market directional movements. This work focuses on forecasting a portfolio of different stocks, instead of concentrating on a single stock which most of the works in this field do. Furthermore, the proposed model is able to solve the issue of skewed classes through the use of appropriate data balancing techniques. This project represents a step forward to improve the robustness of stock trend forecasting techniques and provides a starting point for technical analyst to better understand the market behavior.**

***Index Terms*—Market Trend Prediction, Technical Analysis, Machine Learning**

I. Introduction

Stock market prediction is a very interesting and challenging problem. In the past years, researchers and financial analysts have attempted to find an explanation for the market behavior by building theoretical hypotheses. The efficient market hypothesis (EMH) [1] 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 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 adaptive market hypothesis (AMH) [2] tries to connect the rational EMH principles, 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.

Within financial analysis, we can outline two different schools of thought regarding stock market prediction: fundamental analysis and technical analysis. According to fundamental analysis [3], 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/less than the market price; even if, the proponents of EMH argue that the intrinsic value of a stock is always equal to its current price. On the other hand, technical analysts believe market price movements tell everything; hence, their strategies are based on the stock prices and technical indicators like RSI, MACD, and moving average.

Researchers have strongly worked on financial forecasting using artificial intelligence tools [4]. They have applied different kinds of approaches from simpler models like Naïve Bayes to much more complex ones like artificial neural network [5], [6]. However, small data sets used in most of these works limit the generalization of models. Huang et al. have exploited a support vector machine (SVM) to forecast the stock market direction by using a small data set made up of 676 pairs of observation, achieving a hit ratio of nearly 70% [7]. We believe that increasing its dimension could have lead to more trustworthy performances as a small data set limits the generalization of the model. On the other hand, some works have fed a bigger data set inside a neural network architecture but with the goal of predicting only a specific index of the market [5], [8]. Yao et al. in their work, have developed a

model to forecast only a single index, the Kuala Lumpur Stock Exchange using a data set of around 2000 samples.

In this paper, all the results reported are calculated as the average between twenty different stocks, so that our performance are more statistically relevant than the previous ones. The aim of our work is to make the most of technical indicators through the development of a robust model based on an accurate feature ranking and a correct preprocessing of the available data. During this research, we have come across issues in implementation of data balancing and cross-validation technique, which has not been take into consideration by various works on financial forecasting [5], [9], [10].

To overcome these limitations, we have chosen two approaches. In the first approach, we balance the classes in each train and validation set separately, so that the best classifier chosen from the cross-validation is not biased. Conversely, we leave the test set unchanged because real use case data could be unbalanced. In the second approach, we propose the 'increasing-window cross-validation', a different cross-validation method for time series forecasting. The target of this process is to build a powerful tool to tackle the problem of trend classification on a portfolio of stocks.

The remainder of the paper is organized as follows: Section II introduces the research question; Section III underlines our novelties and approach to such a question; Section IV describes available datasets; Section V reports the experimental setup; Section VI list results; finally, Section VII points out our conclusion and future works.

II. Problem Formalization

In order to predict the direction of the market trend, an input X, a sequence of vectors, is considered where:

$$X = [x(0), x(1), .., x(N-1), x(N)]$$

with N=number of samples. By selecting a generic sample $x(t) \in R^{\mathrm{F}}$ with F=number of features and t the time stamp of the sample, it can be decomposed in:

$$x(t) = [x(t)_0, x(t)_1, .., x(t)_F]$$

The target of the problem is a sequence

$$Y = [y(0), y(1), .., y(N)]$$

of the same length of X such that every element of Y, called $y(t)$ is computed:

$$y(t) = \frac{sgn(pc(t+w) - pc(t)) + 1}{2} \tag{1}$$

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

The sign function (sgn) was applied to the price delta and then the results were moved to the discrete interval [0,1]. At this point the label $y(t)$ is:

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

III. Proposed Approach

The approach proposed for solving this task was an auto-recursive classification problem between uptrend and downtrend. The information in $x(t)$ was exploited to predict the related $y(t)$ value. To achieve this target, an SVM was used [8].

SVM is a classifier method that performs classification tasks by constructing hyperplanes in a multidimensional space in order to separate the classes. The training of SVM model is equivalent to determine a linearly constrained quadratic programming problem that could be solved in the dual formulation as:

$$\begin{aligned} \max_{\boldsymbol{\alpha} \in \mathbb{R}^N} \quad & \frac{1}{2}\boldsymbol{\alpha}^T K \boldsymbol{\alpha} - \boldsymbol{\alpha}^T \boldsymbol{y} \\ \text{s.t.} \quad & \sum_{i=1}^{N} \alpha_i = 0 \\ & 0 \leq y_i x_i \leq C, \ i \in \{1, \cdots, N\} \end{aligned} \tag{2}$$

where N is, as underlined in Section II, the number of samples. Thus, the solution of SVMs is unique, optimal, and absent from local minima, unlike other networks training which requires non-linear optimization thus running the danger of getting stuck in a local minimum. Furthermore, it has been reported that, in this field, SVM has the highest forecasting accuracy among the individual forecasting methods [7], [11]. 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.

The techniques being exploited to achieve the task of predicting the market trend have been discussed in the subsections below :

A. Data Balancing

When performing data mining on financial time series, it is often possible to have very skewed classes because of a strong up or down trend of the stock under study. This issue could affect the predictions performance leading to a biased classifier. The most of the previous works haven't focused their attention on this problem [7], [12], [13].

Recently [6], [14] have tried to balance the classes using a threshold but they have balanced the whole dataset together and it is not enough. In fact, even if the complete dataset is balanced, the training set or validation set themselves could be unbalanced causing a bias in the trained model. Moreover, the test set should not be balanced because in the real use case, data to be predicted will not be necessary balanced.

To solve this problem, we have used different techniques. At first, we applied under-sampling inside the training and validation test separately. The issue of this method was that we were wasting some important samples of our dataset decreasing its generalization power. Hence, we moved to apply

over-sampling techniques following the suggestion of [15]. In particular, we have started with a random over-sampling and at the end, we have chosen for SMOTE [16] and ADASYN [17] because they are the two most used algorithm according to the literature.

Synthetic Minority Over-sampling Technique (SMOTE) is an over-sampling approach in which the minority class is over-sampled by creating synthetic examples. This oversampling technique starts by taking each minority class sample and introducing synthetic examples along the line segments joining any of the K minority class nearest neighbors. Synthetic samples are generated in the following way: take the difference between the sample under consideration and its nearest neighbor. Multiply this difference by a random number between 0 and 1, and add it to the samples under consideration. Instead, the key idea of Adaptive Synthetic Sampling (ADASYN) algorithm is to use a density distribution obtained by the following process. For each sample $x_i \in minorityclass$, find K nearest neighbors based on the Euclidean distance in F dimensional space, with F number of features of each sample, see Section II. Afterwards, the ratio r_i is computed, where r_i is defined as:

$$r_i = \Delta_i / K, \quad i = 1, ..., m$$

where m is the number of samples in the minority class and Δ_i is the number of examples in the K nearest neighbors of x_i that belong to the majority class, therefore $r_i \in [0, 1]$. Normalize r_i according to :

$$R_i = \frac{r_i}{\sum_{i=1}^{m} r_i}$$

so that R_i is a density distribution: $\sum_i^m R_i = 1$

Physically, R_i is a measurement of the distribution of weights for different minority class examples according to their level of difficulty in learning. The resulting dataset post ADASYN, will not only provide a balanced representation of the data distribution but it will also force the learning algorithm to focus on those examples difficult to learn. This is the major difference compared to the SMOTE [16] algorithm, in which equal numbers of synthetic samples are generated for each minority data example.

B. Cross-Validation method

In time series forecasting it is not always possible to use a K-fold cross-validation technique because the samples are not independent, especially if indicators like Simple Moving Average (SMA) and Exponential Moving Average (EMA) are used. In fact, when the SMA is computed on the price with a window of D elements at the time t, two neighbour inputs, $x(t)$ and $x(t+1)$, are no longer independent. During the computation of the SMA feature, defined as $x(t)_i$, the set of closing price values, $pc(t)$, which are taken in account, differ only in one element. To underline this concept the computation for $x(t)_i$ and $x(t+1)_i$ is given below:

$$x(t)_i = \frac{\sum_{k=1}^{D} pc_(t-k)}{D} \tag{3}$$

$$x(t+1)_i = \frac{\sum_{k=1}^{D} pc_(t+1-k)}{D} \tag{4}$$

The overlapping sets prevent the use of k-fold cross validation, thus it is not possible to shuffle the data and pick up randomly the train and validation portions. The samples from the validation set will be strongly dependent on the training ones. To walk around this issue, few researchers have divided the data set into three parts (train, validation, and test) without doing cross-validation [5], [9]. This choice is not a real solution because it leads to a poorly generalizing model, especially with the small data sets used in financial forecasting field.

Our proposal is a cross-validation technique that takes some intuitions from the 'walk forward testing' method [18], [19]. In this project, a method, called 'increasing-window cross-validation' 'Fig. 1', was designed to run cross-validation.

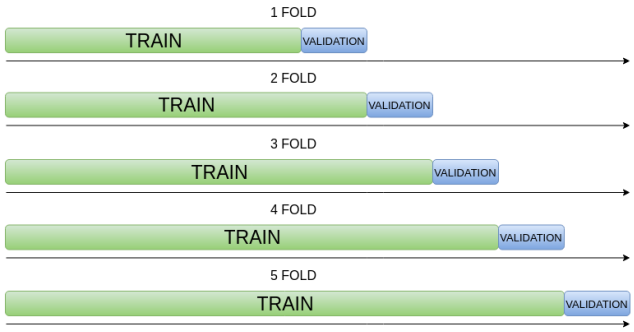


Fig. 1. Increasing-window CrossValidation.

'Increasing-window' technique attempts to make the most of the available data. It includes in each training fold all the previous samples without having overlapping section between one validation fold and the other. Specifically, in this technique the training window is increased in each fold and the validation set is shifted ahead in time. The train and validation set were balanced using the previously mentioned techniques.

C. Feature Ranking

Technical analysts have developed an enormous quantity of indicators to better understand the stock price. It is very difficult to manually dig into them to select the most useful group. Up to now, researchers in this field rely mainly on a handmade feature engineering which leads to the use of indicators like RSI, Williams R and Moving Average [11], [20], [21]. Huang et al. [13] select the most meaningful features using a wrapper approach. Inspired by their work, in this project, the most meaningful features were selected, however, using ranking techniques. The indicators previously exploited in the literature were grouped and others indicators such as Average True Range, Bollinger Bands, and MACD which usually are not considered, were added. The list of indicators employed with relative formulas has been showed in 'Table. I'.

Additionally, different versions of the same indicator but with sundry parameters settings were included into the features

TABLE I
INDICATORS

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

With pc= close price, po= open price, pl= low price, ph= high price

set to make the best of those having time-invariant parameters like RSI or Moving Average. For example, Moving Average was computed with more than one window for the mean as showed in the following vector [2,4,6,8,12,14,16,18]. The same strategy is adopted with RSI, following the same time parameters commonly used in trading. The final result is a vector of 111 features.

Chandrashekar et al. suggest increasing the number of features in machine learning applications is not always useful [22]. Therefore, from the 111 dimensional feature vector, 8 different smaller sets were selected. To fine tune the feature set, two ranking method were applied, precisely Pearson Coefficient and Mutual Information.

Pearson Coefficient (R) and Mutual Information (I) formulas are reported below:

$$R = \frac{N(\sum_{k=1}^{N} x_k y_k) - (\sum_{k=1}^{N} x_k) * (\sum_{k=1}^{N} y_k)}{\sqrt{[N \sum_{k=1}^{N} {x_k}^2 - (\sum_{k=1}^{N} x_k)^2][N \sum_{k=1}^{N} {y_k}^2 - (\sum_{k=1}^{N} y_k)^2]}}$$

where N is the number of samples, as previously defined in Section II.

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}$$

Where, $p(x,y)$ is the joint probability density function and $p(x), p(y)$ are the marginal probability density functions of X and Y respectively.

The two ranking methods were chosen because of their simplicity and successful implementation in practical applications [22]. First of all, the features were ranked in decreasing order of Pearson Coefficient values and 4 different sets were selected, using the first 20th, 40th, 60th, 80th feature. The same procedure was repeated for the Mutual Information ranking technique. A total of 8 sets were obtained, 4 sets for each ranking method.

IV. Available Data

The dataset selected for the stock market forecasting task is composed of the price of the 20th NASDAQ's most capitalized stocks, reported in 'Table. II'.

TABLE II
Stocks

Stock	Ticker	Capitalization
Apple Inc.	AAPL	$926.9B
Amazon.com Inc	AMZN	$781.29B
Microsoft Corporation	MSFT	$755.72B
Google	GOOGL	$752.95B
FaceBook Inc	FB	$535.27B
Intel Corporation	INTC	$258.35B
Cisco Syestems Inc.	CSCO	$203.4B
Netflix Inc	NFLX	$152.7B
NVIDIA Corporation	NVDA	$151.31B
Comcast Corporation	CMCSA	$146.1B
Pepsico Inc	PEP	$142.22B
Adobe Systems Incorporated	ADBE	$119.95B
Amgen Inc	AMGN	$117.94B
Texas Instrument Incorporated	TXN	$109.21B
Broadcom Inc	AVGO	$102.7B
Booking Holdings Inc	BKNG	$101.69B
PayPal Holdings Inc	PYPL	$96.13B
QUALCOMM Incorporated	QCOM	$88.91
Gilead Sciences Inc	GILD	$87.57B
Costco Wholesale Corporation	COST	$87.04B

The time span is from 17/07/2017 to 25/05/2018 with a frequency of 15 minutes between each sample, giving 6,000 samples for each stock, and reaching a total of 120,000 for all the stocks. The data has been collected with Google finance API[1]. The dimension of the dataset is larger than the most of available datasets in the literature [7], [9], [12]. A longer time span cannot be considered because of a limitation of the API. At first, in the raw dataset, each sample is a vector composed only of ['Open', 'Close', 'Low', 'High', 'Volume']. A pre-processing phase was executed on the dataset to obtain features related to the technical analysis. For each sample of the dataset, 106 different technical indicators were computed using StockStats library[2]. The final results was a dataset of 120,000 samples of 111 features each.

[1]https://pypi.org/project/googlefinance.client/

[2]https://github.com/jealous/stockstats

V. Experiment settings

In this section the settings and the evaluation techniques used in the experiments have been discussed. The previously collected data was fed to an SVM model, in particular a Linear SVM. To train, validate, and test the model the dataset was divided into two parts:

- the train and validation set consists of 80% from the total amount of data. Specifically, the first part of the time span available was used. The idea is to use information from the past to train the model.
- the test set consists of the last slice, representing the 20% of the entire time span available.

'Fig. 2' reports our split.
Furthermore, a safety margin was left between the two partitions to avoid recency problems. The reason of the recency margin has to be found in the work of Yao et al., according to which using side-by-side sample leads to a classification bias on the first part of the test set [23].

The aim of this project is to forecast the direction of the market, so a crucial point is to chose the length of the trend to be predicted. Xu et al. and Tay et al. used a time window of 5 days for their trend [9], [14], following the suggestion coming from previous research of Thomason at al. [24]. In our work, the model was evaluated in various windows configuration to have more insight into its prediction power. Thus, the prediction increasing the dimension of the trend window was plotted. In particular, a range between 15 minutes and 3 weeks was experimented.

Once trained the model, different metrics were used to evaluate the performance of our model.

A Gaussian Naïve Bayes, which is commonly used in the literature, was taken as baseline. In order to have a comparison with the previous works, the Directional Accuracy was considered as most of the other papers in this field [7], [12], [13].

We believe that, accuracy is not enough to evaluate carefully a classification problem with imbalanced classes. This is why the confusion matrix on the test set was plotted to figure out the real behavior of the model. The precision, specificy, and recall metrics were computed on the out-of-sample data to understand if the model was working fair with the two classes or it was biased due to the skewed training set. For a faster and complete evaluation of both precision and recall the F1-score metric was adopted:

$$F1 - score = 2 * \frac{precision + recall}{precision * recall}$$

VI. Results

The model was trained and tested on the dataset reported in Section IV with the experimental settings previously defined.

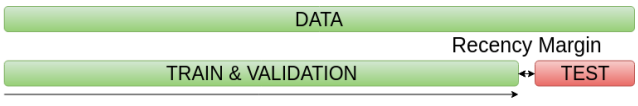


Fig. 2. Split between train and validation set and test set

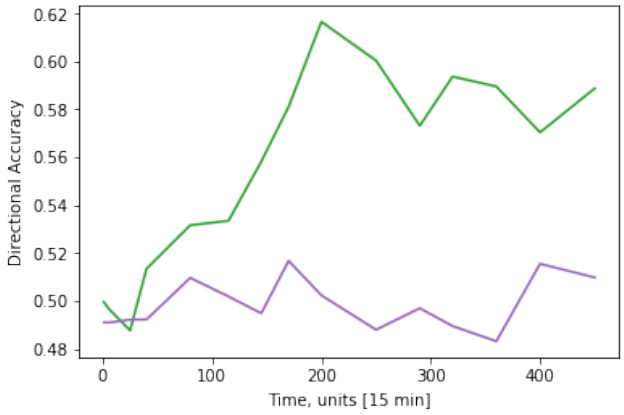


Fig. 3. Green: accuracy Model0 increasing the trend window Purple: accuracy baseline Gaussian Naïve Bayes. Unit 1=15min, 28=1day, 140=1week, 280=2weeks, 420=3weeks

Its performance is presented in 'Figure. 3'. The reported results were obtained with a model called 'Model0', whose features are reported below:

- SMOTE algorithm was applied separately on each train and validation set for balancing the skewed classes, with number of neighbors equals 5, as commonly used in the literature [16], [25].
- 'increasing-window' cross-validation technique was applied ('Figure. 1')
- the set of features employed is reported in 'Table. III'. They were evaluated as the 40 most meaningful ones within the whole set using Pearson Coefficient ranking
- trend length from 15 min up to 3 weeks was used

'Figure. 3' illustrate a clear uptrend of the accuracy up to $window = 200$ (around 1 week and a half in the future) and a subsequent decreasing trend. The baseline was realized with a Gaussian Naïve Bayes trained on the same data balanced with SMOTE technique and tested on the same out-of-sample data. In 'Figure. 4', the confusion matrix, precision, specificy, and recall values have been displayed to better understand if the model is not biased and mis-predicts in equal measure both negative and positive sample.

The trend pointed out can be explained with:

- strong volatility and related unpredictability of the market for short trend windows.

TABLE III
Best Features

Name	Time Parameter
Simple Moving Average SMA	2,4,6,8,10,12,14,16,18,20
Exponential Moving Average EMA	10,12,14,24,26,28
Bollinger Bands	Low, Mid, Up
Price	open, close, middle, high, low
Average True Range	1,7,14,21
CR	1,2,3
MACD	14,16,26,28
RSI	10,12,16,14,16

	True trend			Precision 64,46
Predicted trend		Down	Up	
	Down	310	219	Recall 63,5
	Up	210	381	
Directional Accuracy 61,69				Specificy 59,62

Fig. 4. Balancing with SMOTE

- lack of predictive power of indicators further in the future. In fact, when we increase the size of the window further in the future, indicators are no longer useful in predicting the market trend.
- the performance is reasonable in comparison with the works of Huang et al. and Tay at al., which have achieved comparable results with a time window of a week [7], [9].

In the next subsections, our achievement are reported:

A. Balancing Technique

In the experiments different balancing techniques were adopted and the comparison between them are reported in 'Figure. 4', 'Figure. 5', and 'Figure. 6': with SMOTE balance, with ADASYN technique, and without class balancing, respectively.

	True trend			Precision 61,56
Predicted trend		Down	Up	
	Down	186	65	Recall 89,16
	Up	334	535	
Directional Accuracy 64,37				Specificy 35,77

Fig. 5. Non balanced classes

	True trend			Precision 61,54
Predicted trend		Down	Up	
	Down	217	115	Recall 80,83
	Up	303	485	
Directional Accuracy 62,67				Specificy 41,73

Fig. 6. Balancing with ADASYN

The reported results illustrate the importance to have balanced classes in train and validation set. In fact, without balancing, higher accuracy was reached even if the confusion matrix was totally biased. Therefore, the accuracy reported in 'Figure. 5' is not trustworthy. Conversely, passing through the application of a proper balancing technique, the directional accuracy decreases but the quality of the confusion matrix and related metrics strongly increases ('Figure. 4'). In particular, SMOTE algorithm is working better than ADASYN.

B. Increasing window cross-validation

This work points out the importance of using an appropriate cross-validation method even in time series forecasting. With the use of 'increasing window cross-validation', the model was fit multiple times on different sets of in-sample data leading to an improvement in the prediction of the out-of-sample set. As a proof of effectiveness, the same model was trained without using increasing window cross-validation and the accuracy with a $window = 200$ decreased from 61.69% to 59.20%. The drop in performances was caused by the lower generalization power of the model trained without cross-validation.

C. Feature selection

After the pre-processing phase, each sample was a vector $x \in R^{\mathrm{F}}$ where $F = 111$ is the number of indicators. As specified in Section V, different experiments were executed using different sets of features obtained with Pearson Coefficient and Mutual Information ranking. In total, 9 experiments were performed on the Model0 with a fixed $window = 200$ changing only the feature set and the results have been reported in 'Table. IV'

TABLE IV
FEATURES SELECTION

Set	Directional Accuracy	F1-score
All the feature set	56.79%	57.96%
Pearson Coeff best 20	59.41%	60.61%
Pearson Coeff best 40	**61.69%**	**63.97%**
Pearson Coeff best 60	60.30%	60.69%
Pearson Coeff best 80	58.15%	57.07%
Mutual Info best 20	58.19%	59.53%
Mutual Info best 40	59.70%	58.57%
Mutual Info best 60	58.61%	56.92%
Mutual Info best 80	58.59%	58.25%

The directional accuracy obtained with the 40$^{\text{th}}$ most meaningful features according Pearson Coefficient ranking, 'Pearson Coeff best 40' in 'Table. IV', confirms that more features do not always lead to an increase in the predictive power, as stated by Lin et al. and Huang et al. [13], [26]. In fact, decreasing the amount of features reduces the noise due to meaningless features and it improves the results. In this experiment, the performance of the whole features vector are overcome from a smaller set of features carefully selected with ranking techniques. In particular, as reported in 'Table. IV', better results were achieved with Pearson Coefficient ranking than with Mutual information.

VII. Conclusion and future directions

In this paper, machine learning techniques have been applied to exploit the information extracted from the price time series, with the target of using a classification problem to predict the

market trend. The outcome is a robust model for trend prediction which is able to work on a portfolio of stocks, specifically, the 20th NASDAQ's most capitalized tickers. A directional accuracy of 61.69% was achieved when predicting the market with a trend window of one week and a half. However, a high accuracy is not sufficient to test completely a classifier model. Thus, confusion matrix, precision, specificy, and recall metrics were computed, which are more suitable to understand deeply the real performance of a classifier. Additionally, to complete this task, interesting insights regarding balancing technique, cross-validation method, and features selection were achieved which represent an improvement to the previous works. In particular, an appropriate data balancing technique was fundamental to avoid developing a biased classifier. In fact, a model trained on unbalanced data could bring high accuracy values but leads to a poor generalization and low robustness of the model itself. Another important achievement regards the cross-validation method applied during the model training. When using time series as input to machine learning models, shuffling the data inside the train and validation set is not allowed because of a strong dependency between different samples. Hence, in this work, a different cross-validation technique, 'increasing window cross-validation', was proposed, which overcomes the limitations imposed by data dependency to the most commonly used K-fold cross-validation. Furthermore, the impact of using feature ranking techniques was evaluated on our features set of indicators. Our aim was to avoid handmade features engineering, in favor of an effectiveness ranking between each indicator. This work is a fundamental chunk of a wider project that attempts to make the best of both technical analysis and sentiment analysis to build a machine learning tool able to predict the direction of the market trend. We believe that could be very useful to integrate information coming from the world of finance (technical analysis) and the world of natural language processing (sentiment analysis). A strong collaboration between machine learning, technical analysis, and sentiment analysis could lead to a big step of improvement toward solving a problem which has been distressing researchers for more than half a century.

REFERENCES

[1] E. F. Fama, "Efficient capital markets: Ii," *The journal of finance*, vol. 46, no. 5, pp. 1575–1617, 1991.

[2] A. W. Lo, "The adaptive markets hypothesis: Market efficiency from an evolutionary perspective," *Journal of Portfolio Management, Forthcoming*, 2004.

[3] J. S. Abarbanell and B. J. Bushee, "Abnormal returns to a fundamental analysis strategy," *Accounting Review*, pp. 19–45, 1998.

[4] F. Xing, E. Cambria, and R. Welsch, "Natural language based financial forecasting: A survey," *Artificial Intelligence Review*, vol. 50, no. 1, pp. 49–73, 2018.

[5] J. Yao, C. L. Tan, and H.-L. Poh, "Neural networks for technical analysis: a study on klci," *International journal of theoretical and applied finance*, vol. 2, no. 02, pp. 221–241, 1999.

[6] Z. Hu, W. Liu, J. Bian, X. Liu, and T.-Y. Liu, "Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, 2018, pp. 261–269.

[7] W. Huang, Y. Nakamori, and S.-Y. Wang, "Forecasting stock market movement direction with support vector machine," *Computers & Operations Research*, vol. 32, no. 10, pp. 2513–2522, 2005.

[8] A. M. Andrew, "An introduction to support vector machines and other kernel-based learning methods by nello christianini and john shawe-taylor, cambridge university press, cambridge, 2000, xiii+ 189 pp., isbn 0-521-78019-5 (hbk,£ 27.50).-," *Robotica*, vol. 18, no. 6, pp. 687–689, 2000.

[9] F. E. Tay and L. Cao, "Application of support vector machines in financial time series forecasting," *Omega*, vol. 29, no. 4, pp. 309–317, 2001.

[10] F. Xing, E. Cambria, and R. Welsch, "Intelligent asset allocation via market sentiment views," *IEEE Computational Intelligence Magazine*, 2018.

[11] R. Choudhry and K. Garg, "A hybrid machine learning system for stock market forecasting," *World Academy of Science, Engineering and Technology*, vol. 39, no. 3, pp. 315–318, 2008.

[12] K.-j. Kim, "Financial time series forecasting using support vector machines," *Neurocomputing*, vol. 55, no. 1-2, pp. 307–319, 2003.

[13] C.-J. Huang, D.-X. Yang, and Y.-T. Chuang, "Application of wrapper approach and composite classifier to the stock trend prediction," *Expert Systems with Applications*, vol. 34, no. 4, pp. 2870–2878, 2008.

[14] Y. Xu and S. B. Cohen, "Stock movement prediction from tweets and historical prices," *ACL*, 2018.

[15] A. Amin, S. Anwar, A. Adnan, M. Nawaz, N. Howard, J. Qadir, A. Hawalah, and A. Hussain, "Comparing oversampling techniques to handle the class imbalance problem: a customer churn prediction case study," *IEEE Access*, vol. 4, pp. 7940–7957, 2016.

[16] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[17] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*. IEEE, 2008, pp. 1322–1328.

[18] L.-J. Cao and F. E. H. Tay, "Support vector machine with adaptive parameters in financial time series forecasting," *IEEE Transactions on neural networks*, vol. 14, no. 6, pp. 1506–1518, 2003.

[19] I. Kaastra and M. Boyd, "Designing a neural network for forecasting financial and economic time series," *Neurocomputing*, vol. 10, no. 3, pp. 215–236, 1996.

[20] K.-j. Kim and I. Han, "Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index," *Expert systems with Applications*, vol. 19, no. 2, pp. 125–132, 2000.

[21] H. Mizuno, M. Kosaka, H. Yajima, and N. Komoda, "Application of neural network to technical analysis of stock market prediction," *Studies in Informatic and control*, vol. 7, no. 3, pp. 111–120, 1998.

[22] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014.

[23] J. Yao and H.-L. Poh, "Forecasting the klse index using neural networks," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 2. IEEE, 1995, pp. 1012–1017.

[24] M. Thomason, "The practitioner methods and tool," *Journal of Computational Intelligence in Finance*, vol. 7, no. 3, pp. 36–45, 1999.

[25] L. Lusa *et al.*, "Class prediction for high-dimensional class-imbalanced data," *BMC bioinformatics*, vol. 11, no. 1, p. 523, 2010.

[26] Y. Lin, H. Guo, and J. Hu, "An svm-based approach for stock market trend prediction," in *Neural Networks (IJCNN), The 2013 International Joint Conference on*. IEEE, 2013, pp. 1–7.

xxxxxx yyyyy manuscript No.
(will be inserted by the editor)

Sentiment-Conditional Generation of Synthetic Text

the date of receipt and acceptance should be inserted later

Abstract In this paper, we propose an extremely simple yet novel approach for data augmentation through a sequence to sequence generative model. In the first phase, an LSTM network is trained for learning sentiment and domain conditioned distribution of lemmas from the available labelled sentences. In a second instance new samples are generated through the previously trained model and a larger dataset is used for sentiment classification with different machine learning algorithms. Preliminary results show that sentiment classification can be improved up to 6% for the datasets tested and the improvement given by the data augmentation is consistent over two different datasets.

1 Introduction

Machine Learning (ML) techniques have recently been applied in text mining and natural language processing (NLP) applications [1] [2] [3]. One important task in text mining problems is sentiment analysis (SA). Also called opinion mining (OM), it studies the recognition of opinions, feelings and emotions from text objects. Understanding people's opinions from social media provides fascinating opportunities for both academic research and industrial applications in several domains, and we will list few relevant examples. Some researchers focused on the extraction of users opinions from textual customer reviews [4] [5] [6].Popescu et al. exploited unsupervised information extraction system together with a novel relaxation-labeling technique to determine the semantic orientation of potential opinion words. On the other hand, Sompraser et al. carried out an automatic process to summarize reviewers opinions with the use of dependency relations and ontological knowledge inside a probabilistic based model. The extracted opinions can be useful for both the seller of the product/service and other users that are making a purchasing decision. Other scholars [7] [8] found out that the reputation of companies and products is highly correlated with public opinions and hearsays. In the field of politics, has been demonstrated that public mood often offers a good political thermometer [9] [10]. Tumasjan et al. showed that Twitter is used extensively for political deliberation and the sentiment extracted from party mentions accurately reflects the election result. Moreover, Twitter, it has been shown that Twitter is not only used as a mean for the diffusion of political statement, but also as a platform for political discussion with other users. In the domain of behavioral finance, new discoveries indicate public mood as a key element for stock market prediction [11] [12] [13], and the opinion of investors is relevant for classical problems of financial engineering such as the portfolio allocation and market views formalization [14]. Recently, Peng et al. have demonstrated the over-performance of the sentiment analysis in comparison to the conventional technical analysis when a classification procedure is executed on the market trend [15]. Although, SA and OM have achieved interesting results in different fields the retrieval of textual datasets is still a difficult task. In fact, one of the main challenges in sentiment analysis is that the application of supervised learning techniques requires a considerable amount of labeled training data. In some domain, like movie or customer's reviews, the numerical customer evaluation (usually expressed through a likert scale or a 1-10 response scale) can be used as a proxy for the sentiment label. In other domains, like finance or public reputation, there is not such possibility. As a consequence, data must be labeled manually,

Address(es) of author(s) should be given

facing the issues of speed and scalability. Scholars in sentiment analysis proposed different solutions to this problem. Researchers in the field of transfer learning (TL) try to store knowledge in one domain and adapt it to different domains [16] [17] [18]. Specifically, Lu et al. proposed a transfer learning framework, Source Free Transfer Learning, that effectively selects helpful auxiliary data from an open knowledge space.

In this paper we tackle the problem of scarcity of labeled data from another perspective. Through the aim of generative model, we create sentiment conditional synthetic text. In this way, starting from a small seed of labeled data, we can build a dataset large enough to train a machine learning model. The model will be sentiment conditional since different models will be trained positive and negative corpora, in order to learn from a distribution belonging to different affective states. The key idea is that since the synthetic data will be used for sentiment classification, it should rather incorporates sentiment properties rather than semantic and syntactical characteristics. As stated before, it will be create through a generative sequence-to-sequence model, depicted in Fig. 1. Sequence-to-sequence models [19] [20] are used in a variety of tasks such as machine translation (NMT), speech recognition, text summarization, question/answering, document classification, spell checking and they can be seen as bi-modular encoder-decoder architectures. A seq2seq model first reads the source sequence and, by using an encoder, it builds an hidden representation; then a decoder processes the hidden representation to emit the target sequence. Usually, the seq2seq modules consist of two Recurrent Neural Networks (RNNs): the first (encoder) consumes the encoder inputs and the encoder lengths without making any prediction, while the second RNN (decoder) processes either the decoder inputs and the decoder lengths to generate new sentences.

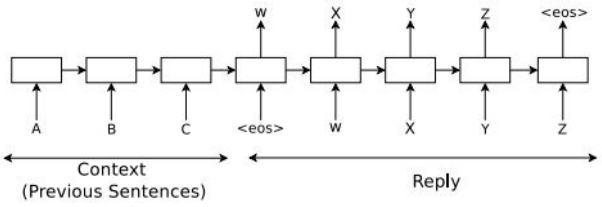


Fig. 1: Diagram of a generative model

2 Model

In this paper, complex word sequences are generated through a sequence to sequence generative model and are later used as a text augmentation method. A RNN is trained for sequence generation by processing sentences one step at a time. An input sequence $\mathbf{x} = (x_1, ..., x_T)$, representing a sentence, is passed through the recurrent connections of K hidden layers. In the RNN model each state $h_t^k(x_1, \cdots, x_t)$, $\forall k \in [1, K]$ acts as a summarization of the previous inputs in the sequence. The hidden state ${h^K}_T$ is computed with respect to every input sequence $\mathbf{x}$ and it is used for the prediction of the next word. On a first stage, a Dense layer characterized by the ReLu activation function is applied: $ReLu(w \cdot {h^K}_T + b)$. Then the prediction of the next word $\hat{y}_{T+1}$ is computed though the softmax activation function. In our generative model we adopt a particular kind of RNN called Long Short-term memory network (LSTM) [21], which can learn long time dependencies. Fig. 2 depicts the model. Specifically, we use two hidden LSTM layers.

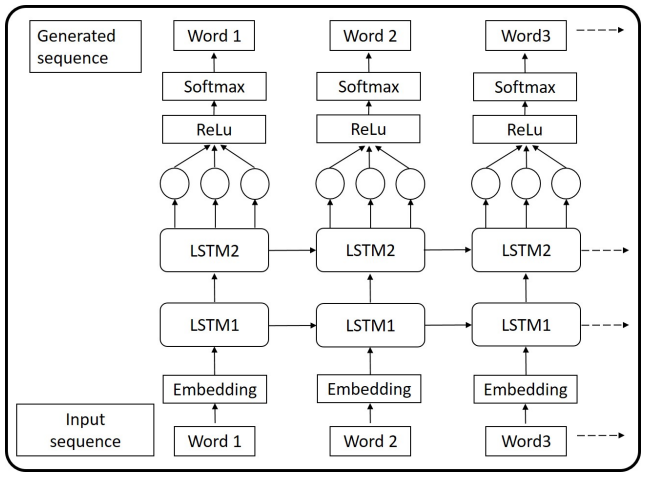


Fig. 2: Sentiment-Conditional Generation of Synthetic Text

3 Experiments

3.1 Data overview

The model will be tested on two different datasets. The first is the IMDB dataset [22] is a dataset of movie reviews for binary sentiment classification, collected from the Internet Movie Database[1] (IMDb). IMDb is an online database of information related to world movies, where users can express their opinion on movies through a textual comment and a 1 to 10 stars scale. The number of stars assigned to the movie can be used as a proxy for the sentiment polarity of the opinion expressed by the text, so that 1 to 5 stars mean a negative opinion,

[1] `https://www.imdb.com/`

and 6 to 10 a positive one. The second one is a collection of labeled financial tweets collected through the StockTwits[2], a social media platform designed for sharing ideas between investors, traders, and entrepreneurs, and filtered by the stock which is the target of the Twit. In particular, we selected only tweets commenting about the APPLE ticker (APPL) belonging to the New York Stock Exchange (NYSE) index. Both the datasets are constituted of 1400 samples, of which 700 belong to the class *positive* and 700 to the class *negative*. We will perform experiments in two settings: without and with synthetic data. In the first case, the training sample will be composed composed by 800 samples and stratified, thus the dataset is balanced. In the second one, we will add to the training data 1000 samples per class generated with the sequence-to-sequence model. For both the settings, the test will be performed on the remaining 600 instances. Since the dataset is balanced, all the models will be evaluated on the prediction accuracy.

3.2 Preprocessing

After the acquisition of the training data, some NLP preprocessing techniques are applied. Tokenization is the task of dividing a text into tokens, the basic elements that will be used as features. In this research we operate at word level. Then punctuation, special characters and words that are not alphabetic are removed and all the words are lowercased. Finally, we decide to not apply lemmatization. Lemmatization is the task of bringing each word to its root form, so that different inflections of a same root can be analyzed as a single word. Since we are analyzing words in sequence and in the context of the sentence, in order to not lose information on the use of different inflection of the same root word, we decide to keep the inflected words as features. All the different words present in the text will form our vocabulary and the number of features is equal to the dimension of the vocabulary *vocab_length*.

Now we have each sentence represented by a set of meaningful words. Before we feed our networks with the training samples, we apply two further transformations. First of all, we transform each word in a integer value, from 1 to *vocab_length*. Afterwards, we embed each feature in a vector. Word embedding is the process of mapping a word in a vector, moving from a space with one-dimension per word to a lower dimensional space. Word embedding have many advantages. For instance the dimensionality reduction alleviates the data sparsity problem which is typical of text classification problems and at the same time captures relationship between the occurrences of the different words. We use the standard embeddings provided by Keras.

[2] `https://stocktwits.com/`

3.3 Synthetic text generation

In the generative model each sentence constitutes a sample. The length of the lstm network is given by the number of features in the sentence. Since each sentence in constituted by a different number of sentences, we uniform all the samples to the same length by the addition of n padding characters at the beginning of each sentence, where the padding character used is the number 0 and n is equal to $max\_length - sentence\_lenght$, where max_length is the number of words of the longest sentence (in term of number of words) and $sentence\_lenght$ is the number of words of the padded sentence. In this way, the number of input features of the network is max_length. We train the model with 100 neurons for each LSTM layer and the Adam optimizer. We start training with a batch size of 128 and 10 epochs and we progressively reduce the batch size and increase the number of epochs in order to improve the quality of the results. The final model that we use for text generation is trained on 100 epochs and a batch size of 64.

4 Sentiment classification

In this research we employed three of the most used machine learning algorithms for text classification: Support Vector Machines (SVM), the Naive Bayes Classifier (NB) and LSTM network.

– **SVM**
Support Vector Machines are supervised learning models for binary classification (SVC) and regression (SVR), belonging to the family of separation methods. Originally proposed in 1995 [23], the Support Vector Classifier builds a separation hyperplane between the two classes and trough a maxmin optimization problem maximize the distance between the nearest points of the two classes:

$$\min_{\mathbf{w},b} \frac{1}{2}\|w\|^2 \tag{1}$$

$$\text{s.t. } y_i\,(\mathbf{w'x}_i - b) \geq 1 \tag{2}$$

where $\mathbf{w}$ is the reciprocal of the separation margin and constraint 5 forces each instance to the class value $y_i \in \{-1, 1\}$ with $i = 1...m$ observations.
The separation margin between this two (or more) points is defined as the distance between the pair of parallel canonical supporting hyperplanes, as shown

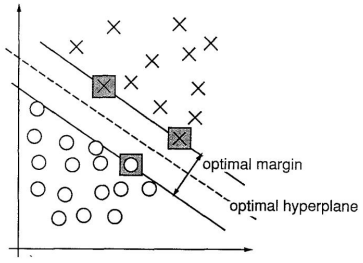


Fig. 3: An example of separable problem. The support vectors are marked with a grey square. Picture from [23]

in fig. 3. The problem in fig. 3 is linearly separable. Non linearly separable problems could be solved with the aim of kernel functions, which map the original observation into a different feature space. The most widely used kernel functions are the Radial Basis Kernel Function (RBF) and the polynomial kernel. In the non separable case, we can formulate the following optimization problem:

$$\min_{\mathbf{w},b} \frac{1}{2}\|w\|^2 + \lambda \sum_{i=1}^{m} d_i \tag{3}$$

$$\text{s.t. } y_i\,(\mathbf{w'x}_i - b) \geq 1 - d_i \tag{4}$$

$$d_i \geq 0 \tag{5}$$

where the sum of the slack variables d_i represents the empirical error- The term λ is introduced in order to regulate the trade-off between the generalization capability, represented by the reciprocal of the margin, and the accuracy on the training set, evaluated as the sum of the slack variables.

– **NB**
The Naive Bayes Classifier is the simplest and most commonly used classifier. NB classification model computes the posterior probability of a class, based on the distribution of the words in the document. It is called "Naive" since relies on the assumption the explanatory variables are conditionally independent. It uses Bayes Theorem to predict the probability that a given feature set belongs to a particular label.

$$P\,(y|\mathbf{x}) = \frac{P\,(\mathbf{x}|y)\,P\,(y)}{P\,(\mathbf{x})} \tag{6}$$

Where, $P(y)$ is the prior probability of the class y, $P(\mathbf{x}|y)$ is the likelihood that the example $\mathbf{x}$ is being classi

ed as a y, and $P(\mathbf{x})$, sometimes called evidence, is the prior probability that a given instance is occurred.

– **LSTM**
Long short-term memory networks have been proposed by Hochreiter and Schmidhuber (1997). They belong to the family of Recurrent Neural Networks (RNNs) a family of NN with loops in them, allowing information to persist from a loop to another. LSTMs are claimed to work very well in practice because they can learn long time dependencies, unlike traditional RNN which suffer from vanishing/exploding gradient when backpropagation is through many time layers. The reason is that information persists from a cell to the following one through a linear activation function. The state of an LSTM cell c_t can is updated by the following equation:

$$\begin{aligned} c_t &= f_t \odot c_{t-1} + i_t \odot (W_c \cdot [\,h_{t-1}, [p, v, s]_t\,] + b_c) \\ h_{t-1} &= o_t \odot \tanh(c_{t-1}) \end{aligned} \tag{7}$$

In addition, LSTM network are composed by and input gate, a forget gate and an output gate:

$$\begin{aligned} i_t &= \sigma(W_i \cdot [\,h_{t-1}, [p, v, s]_t\,] + b_i) \\ f_t &= \sigma(W_f \cdot [\,h_{t-1}, [p, v, s]_t\,] + b_f) \\ o_t &= \sigma(W_o \cdot [\,h_{t-1}, [p, v, s]_t\,] + b_o) \end{aligned} \tag{8}$$

New information enter into the cell whenever its sigmoid input gate is activated. The forget gate decides which information to discard and the output layer regulates the output of the cell.

5 Experimental results

We report below a few examples of sentences generated with different hyperparameters setting from the two datasets IMDB and FinTwits:

Example of positive sentences generated with a batch size of 128 and 10 epochs from the IMDB dataset:

"the film is a great film and the film is a great..."
"is a film and the film is a great film..."

Example of negative sentences generated with a batch size of 128 and 10 epochs from the FinTwits dataset:

"aapl the phone aapl aapl aapl..."
"aapl aapl aapl aapl aapl aapl..."

Example of positive sentences generated with a batch size of 64 and 100 epochs from the IMDB dataset:

"see him in the rest of the film is a great job ..." "the music of the best films i think..."

Example of negative sentences generated with a batch size of 64 and 100 epochs from the FinTwits dataset:

"bearish them they require direct service prevents ..."
"here i expect here aapl sliding down..."

With 10 training epochs and a batch size of 128 the model does not learn the distribution of the text in the sentences, but is only repeating the most common words in the text. For the StockTwits data there is a overwhelming prevalence of the word *aapl*, since every Twit begins with the ticker of the target stock, in our case Apple Inc. Increasing the number of epochs to 100 and halving the batch size, the generated sentences are more smooth to read and include some affective words. In most of the cases, a human reader can easily classify the sentiment of the generated sentences. In the table below we report the accuracy reached by machine learning classifiers.

Table 1: Experimental accuracy on the test datasets

Model	IMDB	FinTwits
SVM	63.23%	58.91%
SVM + SynData	66.05%	62.56%
NB	61.80%	60.75%
NB + SynData	62.58%	56.91%
LSTM	64.75%	61.25%
LSTM + SynData	**70.13%**	**65.11%**

Experimental results align with our hypothesis: the best results are achieved with augmented test. In addition we notice that the best performing algorithm for both the datasets are LSTM, which is not surprising since text data is sequential: one can understand the semantic value of a word based on the understanding of the previous words. Finally, we notice that the results on the IMDB dataset are better. Also this finding is in line with our expectations since financial text are harder to classify because of the presence of many jargons, metaphors, technical terms and ironic constructs.

6 Discussion

In this research we developed a novel methodology for text data augmentation through a sequence-to-sequence generative model. Synthetic data is domain and sentiment conditional, and will be used for sentiment classification. The synthetic sentences are smooth to read and present affective significance. Classification results show that data augmentation improves the classification accuracy for both the datasets.

The article is still in a provisional condition. In future and larger datasets will be tested, and we will assess the effect of using longer text resources, like newspaper articles, to generate manufactured sentences. In addition we will develop an ad hoc loss function for sentiment conditional training.

References

1. Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM, 2004.
2. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.
3. Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
4. Ana-Maria Popescu and Orena Etzioni. Extracting product features and opinions from reviews. In *Natural language processing and text mining*, pages 9–28. Springer, 2007.
5. Minqing Hu and Bing Liu. Opinion extraction and summarization on the web. In *AAAI*, volume 7, pages 1621–1624, 2006.
6. Gamgarn Somprasertsri and Pattarachai Lalitrojwong. Mining feature-opinion in online customer reviews for opinion summarization. *J. UCS*, 16(6):938–955, 2010.
7. Eunsang Yoon, Hugh J Guffey, and Valerie Kijewski. The effects of information and company reputation on intentions to buy a business service. *Journal of Business research*, 27(3):215–228, 1993.
8. James C Ward and Amy L Ostrom. The internet as information minefield: an analysis of the source and content of brand information yielded by net searches. *Journal of Business research*, 56(11):907–914, 2003.
9. Anders Olof Larsson and Hallvard Moe. Studying political microblogging: Twitter users in the 2010 swedish election campaign. *New Media & Society*, 14(5):729–747, 2012.
10. Andranik Tumasjan, Timm O Sprenger, Philipp G Sandner, and Isabell M Welpe. Election forecasts with twitter: How 140 characters reflect the political landscape. *Social science computer review*, 29(4):402–418, 2011.
11. Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of computational science*, 2(1):1–8, 2011.
12. Jasmina Smailović, Miha Grčar, Nada Lavrač, and Martin Žnidaršič. Predictive sentiment analysis of tweets: A stock market application. In *Human-computer interaction and knowledge discovery in complex, unstructured, Big Data*, pages 77–88. Springer, 2013.
13. Gabriele Ranco, Darko Aleksovski, Guido Caldarelli, Miha Grčar, and Igor Mozetič. The effects of twitter sentiment on stock price returns. *PloS one*, 10(9):e0138441, 2015.
14. Frank Z Xing, Erik Cambria, Lorenzo Malandri, and Carlo Vercellis. Discovering bayesian market views for intelligent asset allocation. *arXiv preprint arXiv:1802.09911*, 2018.
15. Yangtuo Peng and Hui Jiang. Leverage financial news to predict stock price movements using word embeddings and deep neural networks. *arXiv preprint arXiv:1506.07220*, 2015.

16. Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th international conference on World wide web*, pages 751–760. ACM, 2010.
17. Vincent Van Asch and Walter Daelemans. Using domain similarity for performance estimation. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, pages 31–36. Association for Computational Linguistics, 2010.
18. Zhongqi Lu, Yin Zhu, Sinno Jialin Pan, Evan Wei Xiang, Yujing Wang, and Qiang Yang. Source free transfer learning for text classification. In *AAAI*, pages 122–128, 2014.
19. Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.
20. Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.
21. Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
22. Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.
23. Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.