Comput Econ (2016) 47:589–622
DOI 10.1007/s10614-015-9492-9

Exploiting Financial News and Social Media Opinions for Stock Market Analysis using MCMC Bayesian Inference

Manolis Maragoudakis · Dimitrios Serpanos

Accepted: 26 January 2015 / Published online: 25 February 2015

Abstract Stock market analysis by using Information and Communication Technology methods is a dynamic and volatile domain. Over the past years, there has been an increasing focus on the development of modeling tools, especially when the expected outcomes appear to yield significant profits to the investors' portfolios. In alignment with modern globalized economy, the available resources are becoming gradually more plentiful, thus difficult to be analyzed by standard statistical tools. Thus far, there have been a number of research papers that emphasize solely in past data from stock bond prices and other technical indicators. Nevertheless, throughout recent studies, prediction is also based on textual information, based on the logical assumption that the course of a stock price can also be affected by news articles and perhaps by public opinions, as posted on various Web 2.0 platforms. Despite the recent advances in Natural Language Processing and Data Mining, when data tend to grow both in number of records and attributes, numerous mining algorithms face significant difficulties, resulting in poor forecast ability. The aim of this study is to propose a potential answer to the problem, by considering a Markov Chain Monte Carlo Bayesian Inference approach, which estimates conditional probability distributions in structures obtained from a Tree-Augmented Naïve Bayes algorithm. The novelty of this study is based on the fact that technical analysis contains the event and not the cause of the change, while textual data may interpret that cause. The paper takes into account a large number of technical indices, accompanied with features that

M. Maragoudakis (✉)
Department of Information and Communication Systems Engineering,
University of the Aegean, Samos, Greece
e-mail: mmarag@aegean.gr

D. Serpanos
Qatar Computing Research Institute (QCRI), Doha, Qatar
e-mail: dserpanos@qf.org.qa

are extracted by a text mining methodology, from financial news articles and opinions posted in different social media platforms. Previous research has demonstrated that due to the high-dimensionality and sparseness of such data, the majority of widespread Data Mining algorithms suffer from either convergence or accuracy problems. Results acquired from the experimental phase, including a virtual trading experiment, are promising. Certainly, as it is tedious for a human investor to read all daily news concerning a company and other financial information, a prediction system that could analyze such textual resources and find relations with price movement at future time frames is valuable.

Keywords Stock return forecasting · Data mining · Hierarchical Bayesian methods · Trading strategies

1 Introduction

Stock market prediction has always gained certain attention from researchers. There is a controversy as regards to whether there is a method for accurate prediction of stock market movement, mainly due to the fact that modeling market dynamics is a complex and volatile domain. Stock market research encapsulates two main philosophical attitudes, i.e. fundamental and technical approaches Technical-Analysis (2005). The former states that stock market movement of prices derives from a security's relative data. In a fundamentalist trading philosophy, the price of a security can be determined through the nuts and bolts of financial numbers. These numbers are derived from the overall economy, the particular industry's sector, or most typically, from the company itself. Figures such as inflation, joblessness, return on equity (ROE), debt levels, and individual price to earnings (PE) ratios can all play a part in determining the price of a stock.

In technical analysis, it is believed that market timing is the key concept. Technicians utilize charts and modeling techniques from past data to identify trends in price and volume. These strategists believe that market timing is critical and opportunities can be found through the careful averaging of historical price and volume movements and comparing them against current prices. Technicians also believe that there are certain high/low psychological price barriers such as support and resistance levels where opportunities may exist. They further reason that price movements are not totally random. Nevertheless, according to several researchers, the goal is not to question the predictability of financial time series but to discover a good model that is capable of describing the dynamics of stock market.

Towards the latter direction, stock market analysis by utilizing Information Technology methods is a dynamic and challenging domain. Over the past years, there has been an increasing focus on the development of modeling systems, especially when the expected outcomes appear to yield significant profits to the investors' portfolios. In alignment with modern globalized economy, the available resources are becoming gradually more plentiful, thus difficult to be analyzed by standard statistical tools. Therefore, technical analysis experts judge that stock market is an excellent representative field of application with strong dynamics in research through data mining,

mainly due to the quantity and the increase rate that data is being produced. Thus far, there have been a number of research papers that emphasize solely in past data from stock bond prices and other technical indicators. Nevertheless, throughout recent studies, prediction is also based on textual records, based on the logical assumption that the course of a stock price can be influenced by news articles, ranging from companies releases and local politics to news of superpower economy Ng and Fu (2003).

However, unrestricted access to news information was not possible until the early 1990's. Nowadays, news are easily accessible, access to important data such as inside company information is relatively cheap and estimations emerge from a vast pool of economists, statisticians, journalists, etc., through the World Wide Web. Despite the large amount of data, advances in Natural Language Processing and text mining allow for effective computerized representation of unstructured document collections, analysis for pattern extraction and discovery of relationships between document terms and time-stamped data streams of stock market quotes. Despite the fact that news play an important role towards influencing stock market trends, public mood states or sentiment, as expressed through various means that promote inter-connectivity, such as Web 2.0 platforms, may also play a similarly important role. Targeted research in the domain of psychology has proven that emotions in addition to information have a direct impact in human decision-making Liu et al. (2007). Therefore, a logical assumption would be for someone to consider public opinion as a factor that could also affect stock market values.

In the present study, the main goal is to study the impact of technical analysis, news articles and public opinions to the task of predicting stock market value. The importance of this study lies to the fact that technical analysis contains the event and not the cause of the change, while textual data may interpret that cause. Following recent trends in the task at hand, this paper takes into account a large number of technical indices, accompanied with features that are extracted by a textual analysis phase from financial news articles and public opinions from financial information portals and Twitter. We incorporate Machine Learning algorithms, that have been adjusted to match the characteristics of the collected dataset, which is characterized by a plethora of attributes. Our proposed methodology is based on a novel Markov Chain Monte Carlo (MCMC) Bayesian Inference approach, which estimates the conditional probability distributions of network structures that are obtained by a Tree-Augmented Naïve Bayes (TAN) algorithm. Experimental results, including a virtual trading experiment, are promising. Certainly, as it is tedious for a human investor to read the plethora of available daily news and public reactions concerning a company as well as other financial information, a prediction system that could analyze such textual resources and find relationships with price movement at future time windows is beneficial.

The paper is structured as follows: Section 2 provides an overview of literature concerning stock market prediction, in an attempt to link previous works with the article and also to provide a clear motivation for the proposed study. Section 3 presents the methodology overview, introducing Bayesian networks and their use towards modeling and reasoning under conditions of uncertainty. Since standard Bayesian networks face certain issues when dealing with classification problems in high-dimensional domains, in Section 4, we provide the theoretical framework of MCMC inference which suits our needs. Section 5 deals with the proposed method, which utilizes the

main characteristics of MCMC but also introduces a novel approach on estimating conditional probability distributions from networks that favor classification tasks. Section 6 presents the input data characteristics as well as the processing phase of textual information and Section 7 describes the experimental evaluation process. Concluding remarks and future directions are found in Section 8.

2 Previous Research

Towards linking the current paper with previous studies and presenting a clear motivation for the suggested methodology, the present section discusses previous research in stock market forecasting by following an event-based approach. More specifically, we intend to focus on the shift from traditional approaches and standard theories to machine learning approaches, in which a number of attempts in combining different types of explanatory variables into forecasting models is applied. Towards this latter direction, emphasis shall be given on works that study the influence of news articles and other forms of public opinions on stock markets since it is also the main target of our work.

The theory of *Random Walk* states that markets move in a random and unpredictable manner. This theory, reflecting the efficient market hypothesis, used to be very popular and widely accepted by academic financial economists in the first era of stock market modeling Fama (1970). The logic behind the random walk idea is that if the flow of information is unimpeded and information is immediately reflected in stock values, then a change in tomorrow's price will only reflect tomorrow's news and will be independent of the price changes today. But news is by definition unpredictable and, thus, resulting price changes must be unpredictable and random. Following the above hypothesis, one could expect no benefit when performing stock market forecasting. However, as shown by Lo and MacKinlay (1988), evidence of predictability exists to some extent. Macroeconomic factors were believed to provide some additional information for stock market forecasting. As illustrated by Chen (1991), various macroeconomic factors such as the default spread, the term spread, the one-month T-bill rate, the lagged industrial production growth rate, and the dividend-price ratio possess certain forecasting power. In a later work by Bilson et al. (2001), a set of common macroeconomic factors was examined on whether they affect returns of emerging markets. Their study showed that local macroeconomic factors such as money supply, inflation, industrial production, and exchange rates, as well as microeconomic factors such as price-to-earnings and dividend yield are most apparent in explaining the return variation for most emerging markets.

Nevertheless, not only fundamental factors were capable of providing information in predicting stock market movements. Technical indicators were also found to contain information that aids forecasting. For example, Yao et al. (1999) proposed a model that applies lagged index price and some well-known technical indicators such as Moving Average, Momentum, Relative Strength Index, Stochastics and Moving Average of Stochastics in predicting future trends. Recent studies have depicted a number of applications in combining different types of explanatory variables into forecasting models. In Bettman et al. (2009), the potential in combining fundamental and technical

variables into US equities forecasting is discussed. Results favor the complementary nature of fundamental and technical factors, which yields a better explanatory power in prediction.

With the advances of machine learning and data mining, a serious amount of work is performed towards utilizing large input features into more complex forecasting models than previously used (e.g. ARIMA, Linear Regression, Buy-and-hold, etc.). As thoroughly described in the survey of Atsalakis and Valavanis (2009), more intelligent techniques such as Neural Networks and Support Vector Machines have portrayed their forecasting capabilities in numerous applications.

As shown above, significant research work has been devoted to stock price prediction data mining techniques that rely only on structured data, like historical prices, traded volumes, and financial rates and figures. Approaches usually employ data mining and statistical analysis techniques to forecast the future price of a stock. On the other hand, work concerning the application of text mining to stock market prediction is limited, but has already proven that salient financial and political news affects stock price at least as strongly as the traditional financial attributes attributes usually selected to describe a stock. Thereby, a new area of research has emerged; the prediction of stock price movement based on financial news articles. Approaches regarding the correlation between the most recent financial articles and the future price of a stock are described in detail below.

Chung et al. (2002) were among the first to confirm the reaction of the market to news articles. They used salient political and economic news as proxy for public information. They have discovered that both types of news have impact on measures of trading activity including return volatility, price volatility, number of shares traded, and trading frequency.

Klibanoff et al. (1998) dealt with closed-end country funds prices and country specific salient news. They argued about the existence of a positive relationship between trading volume and news. They investigated the relationship between closed-end country funds' prices and country-specific salient news. The news that occupied at least two columns on The New York Times front-page were considered as salient news. They have discovered that there is a significant correlation between volume and news. Similar to the aforementioned approach, Chan and John-Wei (1996) discovered that news that is placed in the front page of the South China Post increase the return volatility in the Hong Kong stock market. Mitchell and Mulherin (2002) used the daily number of headlines of Dow Jones as a measure of public information. They mentioned the positive impact of news on absolute price changes.

Cho (1999) used the number of news released by Reuter's News Service measured in per unit of time as a proxy for public information. In contrast to Mitchell and Mulherin, they examined the impact of news on the intraday market activity. Their results suggest that there is a noteworthy positive relationship between news arrivals and trading volume. Mittermayer (2004) proposed a prediction system called NEWSCATS, which provided an estimate of the price after the publication of press releases. Shumaker and Chen (2006) examined three different textual representation formalisms and studied their abilities to predict discrete stock prices twenty minutes upon an article release in press. The Arizona Financial Text System (AZFinText) proposed by them, extracts proper nouns and selects the proper nouns that occurs three or more times to be used

as features. AZFinText is a regression system that attempts to forecast feature prices and does not perform true sentiment analysis, in the sense that this system labels each news article with a price value instead of sentiment label. The AZFinText system does this by labeling each news article with the stock price 20 minutes after it is published. The AZFinText system was tested on S&P 500 and compared against the top quantitative funds. It had an 8.5 % return in the given period, while the S&P 500 had a lesser return of 5.62 %. It ranked as number four against all the other quant funds. However, those quant funds that ranked above it traded in different markets than the AZFinText system. Another system, developed by Pegah Falinouss in her master thesis Falinouss (2007) consists of finding price trends by time series segmentation, then each news document are sentiment labeled by aligning them up with the price trend. The document preprocessing part consists of the three standard methods; tokenizing, stop-word removal and stemming. Document representation is accomplished by using the standard method of using a vector space model with tf-idf as the term weighting method. The system is reported by Falinouss to achieve an accuracy of 83 % for correctly labeling a news article as rise or drop. The recall of rise predictions are stated to be 67 %, and for drop predictions it is 93 %. The precision for rise predictions are claimed to be 87 % and for drop 81 %. Falinouss did not include an evaluation part on how good this system is when used for trading stocks. Some years earlier, Bollen et al. (2010) reported an interesting research in which Twitter data was analyzed in terms of mood dimensions (*calm, alert, sure, vital, kind, and happy*) using available mood-tracking tools and then performed a causality analysis using Self-Organized Fuzzy Neural Networks in order to predict the daily up and down changes of DJIA closing values. Experimental results supported their arguments on the correlation of some specific mood types on the closing value.

The use of Machine Learning in stock market prediction as well as in financial issues has gained significant attention throughout recent years. A noticeable approach that incorporates Genetic Algorithms was suggested by Thomas and Sycara (2000). In their method, they attempted to classify stock prices using the number of postings and size of related articles on a daily basis (textual data were originated from discussion boards on a financial forum). It was found that positive share price movement was correlated to stocks with more than 10,000 posts. However, discussion board postings are quite susceptible to bias and noise. Another popular classifier, the Naïve Bayes approach was used by Wuthrich et al. (1998) in order to represent each article as a weighted vector of keywords. Phrase co-occurrence and price directionality was learned from example articles which lead to the formation of training data. One problem with this algorithm is that articles may focus their attention on some other event and superficially reference a particular stock security. These types of problems can cloud the results of training by unintentionally attaching weight to a casually-mentioned security.

One of the most interesting approaches incorporated Support Vector Machines (SVM). In the work of Fung et al. (2003), regression analysis of technical data was used to identify price trends while SVM analysis of textual news articles was used to perform a binary classification in two predefined categories; stock price rise and drop. In cases where conflicting SVM classification follows, such that both rise and drop classifiers are determined to be positive, the system returns a 'no recommendation' decision. From their research using 350,000 financial news articles and a simulated Buy-Hold

strategy based upon their SVM classifications, they showed that their technique was mildly profitable. Mittermayer (2004) also used SVM is his research to find an optimal profit trading engine. While relying on a three-tier classification system, his research focused on empirically establishing trading limits. It was found that profits can be maximized by buying or shorting stocks and taking profit on them at 1 % up movement or 3 % down movement. This method slightly beat random trading by yielding a 0.11 % average return.

The work of Xidonas et al. (2009) focuses on modeling the overall corporate performance using financial analysis techniques. In Chandra et al. (2010), a hybrid Support Vector Machines and Neural Network methodology is adopted towards dealing with bankruptcy prediction. Data mining processes for reducing dimensionality in banking and insurance data are followed by Vasu and Ravi (2011). Genetic algorithms for tuning a technical trading system for the Dow Jones are demonstrated in the work of Nunez-Letamendia et al. (2011). Predicting credit card customer churn in banks using data mining has been suggested by Kumar and Ravi (2008). The study of Oyatoye and Arilesere (2012) combined an 'expanded Lagrangian function with a modified trust region method to propose a method for solving investment portfolio management problems of insurance companies. In Jayech and Zina (2012), measuring for financial contagion in the stock markets using a copula approach was used, while in Bebarta et al. (2012), a comparative study of stock market forecasting using different functional link artificial neural networks is presented.

In the work of Preis et al. (2013), it is suggested that within a given time period, Google Trends data did not only reflect the current state of the stock markets but may have also been able to forecast certain future trends. Their findings are consistent with the intriguing proposal that notable drops in the financial market are preceded by periods of investor concern. In such periods, investors may search for more information about the market, before eventually deciding to buy or sell. By following this logic, during the period 2004 to 2011 Google Trends search query volumes for certain terms could have been used in the construction of profitable trading strategies. They compared their approach against the "buy and hold" strategy and a purely random investment strategy on the Dow Jones Industrial Average (DJIA). The proposed Google Trends strategy utilizing the search volume of the term debt has yielded a profit of 326 %.

To sum up, while many of the previous studies have focused mainly on classifications based on historical stock prices, none of them have clearly managed to harness data mining algorithms in order to accurately and effectively determine a discrete stock price prediction, based on both stock prices and financial news articles. For existing approaches that relied on a "bag-of-words" textual representation format, the plethora of input variables has obviously posed algorithms with problems and this could be a reason for the small improvements that are reported in literature, as regards to stock market forecasting. Undoubtedly, semantically-rich text representation of articles is more beneficial, since less noise and outlier parameters are inserted to the classifier. However, for the majority of languages such as Modern Greek, semantic annotation tools and resources such as WordNet Fellbaum (1998) are not available. Thus, we have focused our effort in improving the ability of existing data mining algorithms to handle the "bag-of-words" problem, augmented by numerous additional parameters,

obtained from technical indices. The proposed approach discusses both theoretical and experimental issues of the new, hybrid algorithm we propose, which is suitable for large datasets particularly when the number of non-informative input features is large.

3 Methodology Overview

In this section we provide an overview of the proposed methodology, which includes the main challenges and limitations posed in using Bayesian Networks (BN) Heckerman (1999) for stock market prediction, and in general for similar financial decision support problems, and also how we address them. BN are suitable for reasoning under conditions of uncertainty, because they are flexible models for representing relationships among different interacting features (i.e. from technical analysis or text mining) that can be interpreted and visualized. Such relationships could be exploited both for predicting stock market and simultaneously for obtaining an insight into which are the input features that the classification process is based on. Recently, a number of researches about incorporating BN to the financial analysis domain have been presented, presenting a series of benefits from shifting from numerical methods to stochastic ones. In our formulation, we use BN to represent two aspects of financial modeling: a qualitative and a quantitative one. The qualitative structure depicts direct relationships between features and represents their relations as a Directed Acyclic Graph (DAG), where features are denoted as nodes and arcs represent probabilistic relationships among them. The quantitative structure describes such relationships as conditional probability distributions, explaining the strength of such relationships.

In order to place some emphasis on the benefits of BN in financial and economics domain, consider the following example that illustrates some of the functionalities of BN (Fig. 1). Suppose that one attempts to model whether the Liquidity indicator of a

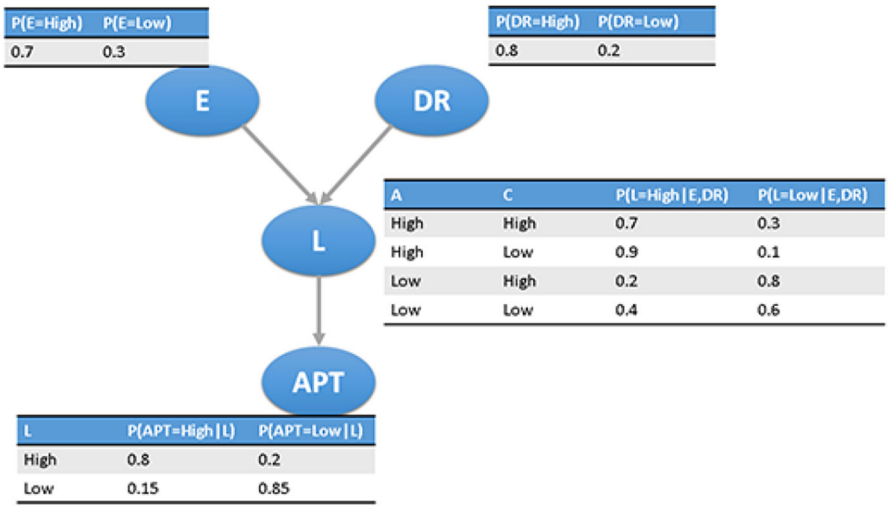


Fig. 1 A draft BN for financial modeling

company (denoted by L) would be *High* or *Low*. Such an event can affect the Accounts Payable Turnover (denoted by APT), which is the number of days in which the average amount of payables are settled. The outcome of Liquidity is influenced by the Equity (denoted by E) and/or from the Debt ratio (denoted by DR). For simplicity, let us assume that all variables are binary, i.e. *High* and *Low*. The Conditional Probability Distribution (CPD) of each variable is listed besides each node in the network. In this example, the parents of Liquidity are Equity and Debt Ratio. The child of Liquidity is Accounts Payable Turnover. By following the BN independence assumption, several statements can be observed:

(a) the variables Equity and Debt Ratio are marginally independent, but when Liquidity is given they are conditionally dependent. The type of this relation is often named as *explaining away*.
(b) when Liquidity is given, Accounts Payable Turnover is conditionally independent of its ancestors Equity and Debt Ratio.
(c) Instead of factorizing the joint distribution of all variables using the chain rule, i.e. $P(E, DR, L, APT) = P(E)P(DR|E)P(L|DR, E)P(APT|L, DR, E)$, the BN defines a compact CPD in a factored form, i.e. $P(E, DR, L, APT) = P(E)P(DR)P(L|E, DR)P(APT|L)$. Note that the BN form reduces the number of model parameters (i.e. the number of rows in the CPD table) from $2^4-1 = 15$ to only 8. This property is of utmost importance since it allows researchers to create a tractable model of domains with a plethora of features.

Such a reduction provides great benefits from inference, learning (parameter estimation), and computational perspective. When people use BNs, they behave similarly to expert systems, since it is able to represent beliefs and knowledge about a particular class of situations. The network represents the knowledge on a particular thematic area. Given evidence on the presence or absence of other situations, conclusions can be drawn on a particular instantiation of a situation. This important observation allows us to build information inference systems that are based on a straightforward probabilistic approach.

However, BN suffer from significant limitations when applied in particular applications of the financial domain such as the task at hand, mainly because prior knowledge is not available or extremely difficult to be defined, and the available data are characterized as "highly-dimensional" (i.e. having large numbers of features).

Another important limitation of applying generic BN to the task at hand is that our features have continuous ranges of values (which is quite usual in stock market analysis and in many other time series applications). Despite the fact that alternative solutions exist for dealing with continuous values in BN, the majority of them focus on the use of discrete valued features, since in the former case (continuous range of values) there are significant topology restrictions and only the Gaussian distribution is supported. Nevertheless, discretization is not a preferred approach in such cases, due to loss of information it causes to the original data. Additionally, BN learning consists of two separate processes, executed in a serial manner: the former is called 'structure learning' and the latter is called 'parameter estimation'. Structure learning is considered to be NP-hard Friedman and Koller (2003), since as the number of features grows the number of candidate network structures increases super-exponentially to

huge numbers. For example a dataset of only 10 features would result in the evaluation of more than 15.000 possible network structures throughout the learning phase. Also, upon evaluating the most probable network structure, estimation of parameters (Conditional Probability Distributions (CPDs)) of each BN is carried out. Estimating CPDs involves the calculation of $p(X_i|parents(X_i))$ for each of the features X_i where $parents(X_i)$ refers to the set of parent nodes of X_i in the network. This will cause a larger number of calculations to take place.

Finally, a fourth obstacle in BN is the lack of orientation towards the class feature (which we want to be the root node in the estimated network, as stock course classification is the primary aim of this research), which could pose significant problems to the classification process. BN are, by principle, designed to allow for reasoning under conditions of uncertainty. This does not necessarily mean that they are suitable for classification. Since the class node is treated in the same way as all other nodes, a BN does not have special knowledge on the class feature and the topology is not oriented to allow for reasoning over the class label, given evidence of the values of the other features; therefore the class node will not be necessarily the root node in the estimated models. To summarize the main challenges and limitations of using BN for stock market classification are:

(a) BN cannot deal efficiently with high-dimensional datasets.
(b) BN do not operate optimally when dealing with continuous variables.
(c) BN learning of structure and CPD is prone to errors and ambiguity when dealing with high-dimensional datasets and limited training samples, and can necessitate large amounts of calculations.
(d) BN are not oriented towards classification, so the networks estimated do not necessarily have the class node as root (which is essential for classification).

This paper makes a contribution in this area by presenting a BN analysis framework for identifying causal as well as independent relationships among features of financial analysis, augmented by textual analysis features,which addresses the above challenges and limitations. In particular, similar to other machine learning approaches, but unlike most BN methods, we are handling features as continuous rather than discrete, addressing the above-mentioned challenge (b). Additionally, due to the high dimensionality nature of our dataset, exact computation of the CPDs is infeasible and computationally costly. Hence, the joint distribution is approximated by stochastic simulation commonly referred to as "sampling". Using Markov Chain Monte Carlo (MCMC) we can fit a distribution to the data that converges to the posterior distribution (i.e. the distribution of the class, treated as a random variable, conditional on the evidence obtained from the dataset) and retain the samples. MCMC can cope with domains where the state space is vast (i.e. large number of features) with large number of samples needed to approximate the probabilities reasonably well, by selecting each sample using the previous sample resulting in the well-known Monte Carlo Markov Chain (MCMC) methods and its variants Pearl (2000). In this way we address the aforementioned challenges (a), (c) and (d). In particular, we propose a new approach to approximate the conditional probability distributions of complex BN using a MCMC algorithm. Our work is principally based upon a novel idea, in which the CPD computation is based on the ordered ranking of a structure similar to traditional BNs, which is

oriented towards classification. This structure is called Tree-Augmented Naïve Bayes (TAN) and unlike general, unrestricted BNs, in TAN the class node is the root node, i.e. the parent of all other nodes, which can form a BN among them, addressing in this way challenge (d). This type of structure is proven to be more efficient than BN for classification purposes Pearl (1988), since in traditional BN the class node is not considered as a special type of node, and it is treated as an ordinary one, so it may even not appear in the network resulting in a lack of classification capabilities. Then we use it in order to perform forecasting of the course of a stock, studying the impact that textual features from financial news and from opinions in social media to standard technical indices features. Additionally, some widely used alternative classifiers have been applied to the same data for comparison purposes.

4 MCMC Bayesian Inference

In this section, the theoretical framework of MCMC sampling is outlined, with a focus on Gibbs sampling, a variation of MCMC, more suitable for DAG structures Liu (2001). Bayesian inference involves the mathematical integration of high-dimensional probability distributions, such as those for the task at hand. Nevertheless, this process is analytically intractable; therefore it is common to deploy Monte Carlo (MC) techniques. MC techniques are requiring sampling from the probability distributions which are to be integrated. However, in many cases, it is not possible to draw such samples directly from the distributions. MCMC methods provide a unified framework for coping with such problems. The way they operate is twofold: at first, a Markov Chain is generated that converges to the target probability distribution. Subsequently, the target sample values are obtained using Monte Carlo integration.

4.1 MCMC Methods

A probability distribution is specified through a DAG G (a set of interconnected nodes, each of which corresponds to one of the features) and a set of conditional probability distributions S (parameters), one for each feature X_i (node) in G. A BN is actually a DAG G where the topology refers to its structure and the CPD is encoded as a table, named as CPT (Conditional Probability Table). By definition, in G, every node is conditionally independent of all other nodes given the set of its parents. The CPD of a BN is encompassing the probabilities of observing all values of feature (node) X_i, given the values of its parent nodes. Large network models will introduce more parameters, so exact computation will be infeasible and thus approximation of the CPD is achieved through sampling techniques. The structure of G is essential for sampling and can be obtained by applying a greedy search over the entire space of all possible structures. However, the number of possible DAG structures increases super-exponentially as the number of features grows, so greedy search on the space of all possible structures is not efficient as it requires too much computation. Several methodologies for alleviating this problem have been proposed, such as the K2 algorithm Heckerman (1999) or the Bayesian Scoring Method Pearl (2000). In the following section, a description of the

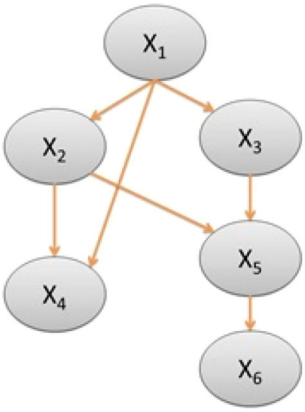


Fig. 2 An example BN consisting of 6 binary nodes with states *True* or *False* each

suggested technique for obtaining graph structures more straightforwardly is provided, thus allowing for constructing BNs that enable efficient classification process.

Regardless of the structure learning algorithm, given a structure G with nodes $X = X_1, X_2, \ldots, X_n$, the process of obtaining the CPD with sampling is described below. For reasons of comprehension, suppose that G is referring to the example BN depicted in Fig. 2.

Let us also assume that each node X_i is a binary node with values T or F. For each node X_i in G:

- Randomly select a value for all other nodes except for X_i.
  - e.g. <?,T,T,F,F,T>
- Compute the probability distribution over the states of X_i, i.e. $p(X_i|X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n)$. Note that since G is a Bayesian network, the above probability is simplified to include only the Markov Blanket of X_i, i.e.:

$$p(X_i|X_1, , X_{i-1}, X_{i+1}, X_n) = p\Big(X_i|parents(X_i)) \prod_{j=1}^{k}(Y_j|parents(Y_j)\Big), \tag{1}$$

where Y_j denotes the set of child nodes of X_i.
  - e.g.: $p(X_1 = T|X_2 = T, X_3 = T, X_4 = F, X_5 = F, X_6 = T) = p(X_1 = T)p(X_4 = F|X_2 = T)p(X_4 = F|X_1 = T)p(X_2 = T|X_1 = T)p(X_3 = T|X_1 = T)$ and
  - $p(X_1 = F|X_2 = T, X_3 = T, X_4 = F, X_5 = F, X_6 = T) = p(X_1 = F)p(X_4 = F|X_2 = T)p(X_4 = F|X_1 = F)p(X_2 = T|X_1 = F)p(X_3 = T|X_1 = F)$
- From the probability distribution, randomly select a value of X_i to complete the sample vector.
  - e.g. suppose that value T is selected for node X_1.

Monte Carlo sampling requires drawing of n samples from the BN with each instance of feature states forming its value as explained above. For our research, all attributes contain continuous values, therefore, we adopt the method of Ram and

Chetty (2008), meaning that we project all samples as a histogram and afterwards we smooth the histogram to obtain the probability density function of the features of the dataset. In most approaches, the selection of a feature value is performed using the distribution that best resembles the available data set. This approach is however not suitable for large feature sets, such as the task at hand, because they tend to be slow and cannot converge to the actual posterior distribution. Therefore, a Markov Chain Monte Carlo (MCMC) approach is more preferable for approximating the challenging high dimensional distributions. The Gibbs sampler was chosen as an MCMC utilization method, because it is more suitable to DAG structures Liu (2001). Furthermore, a Gibbs sampler can allow for convergence in reasonable computation time and its implementation code is widely available in the academic community (e.g. WinBUGS Lunn et al. 2000).

4.2 MCMC and Gibbs Sampling

Before describing the Gibbs sampler, a few introductory comments on Markov Chains are provided. Since Markov Chains by principle contain the concept of time, they used to be mostly associated with applications of data mining and pattern recognition which directly encompass this dimension, such as speech recognition and time series analysis Nummelin (2004). However, Markov Chains could also be applied to BN search process, where each time step denotes a candidate network structure that is evaluated. In our approach, we design a Markov chain where each state is a full joint instantiation of the distribution (i.e. values are assigned to all features of the network). Hence, a transition in time is a transfer from one joint instantiation to another. The target sampling distribution is the posterior joint distribution $P(x|e)$ where x is the class feature and e is the set of evidence features. It is typically the unknown that we want to evaluate. Let X_t^i denote the value of a random variable X_i at time (or step) t, and let the state space refer to the range of possible X_i values. This random variable is a Markov process if the transition probabilities between different values in the state space depend only on the random variable's current state, i.e.:

$$p(X_{t+1}^i = s_j X_0^i = s_l, \ldots, X_t^i = s_k) = p(X_{t+1}^i = s_j | X_t^i = s_k) \tag{2}$$

In other words, for a random variable to be considered a Markov process, the only information about the past needed in order to predict the future is the current state of it. Any knowledge about the values of earlier states does not affect the transition probability. A Markov chain refers to a sequence of random variables generated by a Markov process. A particular chain is defined most critically by its transition matrix $P(j \rightarrow k)$, which is the probability that a process at state space s_j moves to state s_k in a single step, i.e.:

$$P(j \rightarrow k) = p(X_{t+1}^i = s_k | X_t^i = s_j) \tag{3}$$

For reasons of readability, we shall simplify the notion of X_t^i into X_t to denote that a random variable X takes a specific value at time t. Let $\pi_j(t) = p(X_t = s_j)$ denote the probability that the chain is in state j at time t, and let $\pi(t)$ denote the row vector of the state space probabilities at step t. We start the chain by specifying a starting vector $\pi(0)$.

Often, all the elements of $\pi(0)$ are zero except for a single element of 1, corresponding to the process starting in that particular state. As the chain progresses, the probability values get spread out over the possible state space. Using matrix notation, we can define the probability transition matrix P as the one whose element (i, j) denotes the $P(i \rightarrow j)$ transition kernel. The probability that the chain has state value s_i at time (or step) $t + 1$ is given by:

$$\pi(t+1) = \pi(t)P = (\pi(t-1)P)P = \ldots = \pi(0)P^{(t+1)} \tag{4}$$

In other words, as the above equation implies, a Markov chain can reach a stationary (final) distribution $\pi*$, regardless of the selection for the initial distribution parameters. In order to explain this more systematically, consider a random process in which the state S_0 is initialized according to an initial distribution p_0. On each time step t, with probability γ we "stop" the chain and output the current state S_t. Moreover, with probability $1 - \gamma$, we will take a state transition step and sample S_{t+1} according to the transition probabilities $p(S_{t+1}|S_t)$. Since the number of steps T is distributed according to a geometric distribution with parameter $(1 - \gamma)$, the random state that is generated by this process will also be distributed according to π.

A straightforward method of approaching this distribution includes sampling. While there are numerous sampling strategies, the Gibbs sampler Liu (2001) is well-suited for DAGs, as we shall describe in the next paragraphs.

4.3 Gibbs Sampler

The main notion of this methodology is that only univariate conditional distributions are taken into account, i.e. distributions where all of the random variables except for one are assigned fixed values. The reason for the above consideration lies to the fact that such conditional distributions are more straightforward to simulate than complex joint distributions and usually have simpler forms. To introduce the Gibbs sampler, consider a bivariate random variable (x, y) and suppose we request the computation of one or both probabilities, p(x) and p(y). The idea behind the sampler is that it is far easier to consider a sequence of conditional distributions, $p(x|y)$ and $p(y|x)$, than it is to obtain the probability by integration of the joint density $p(x, y)$, e.g. $p(x) = \int(p(x, y)dy)$. The sampler starts with some initial value y_0 for y and obtains x_0 by generating a random variable from the conditional distribution $p(x|y = y_0)$. Then, the sampler uses x_0 to generate a new value of y_1, drawing from the conditional distribution based on the value of x_0, $p(y|x = x_0)$ and so forth. It proceeds as follows:

$$\begin{aligned} x_i &\sim p(x|y = y_{i-1}) \\ y_i &\sim p(y|x = x_i) \end{aligned} \tag{5}$$

Repeating this process k times, generates a Gibbs sequence of length k, where a subset of points (x_j, y_j) for $i \leq j \leq m < k$ are taken as the simulated draws from the full joint distribution. To obtain the desired total of m sample points (here each "point" on the sampler is a vector of the two parameters), one samples the chain:

(i) after a sufficient burn-in process (i.e. a number of initial samples to be removed due to removal of the bad effects of the initial sampling values) and (ii) at set time points (say every n samples) following the burn-in phase. The Gibbs sequence converges to a stationary distribution that is independent of the starting values, and by the principle of MCMC, this stationary distribution is the target distribution we are trying to simulate Liu (2001).

5 Methodology Framework

As mentioned above, one critical challenge is that in the occurrence of high-dimensional input vectors, the set of plausible network models is large, thus a full comparison of all the posterior probabilities associated to the candidate models becomes infeasible. A solution to this can be grounded on the MCMC method and its variation, namely the Gibbs sampler. The proof is based on Markov chain theory, in particular on the fact that the probability of each individual state of a Markov process with positive transition probabilities approaches a certain limit determined by the stationarity condition. For each variable of the BN, this stationary distribution is identical with its posterior distribution Pearl (1988). Note, however, that a direct application of the above algorithm for BN estimation within the stock market analysis domain faces limitations, due to the high dimensionality of the data. This implies that the variance in the values taken by each variable is high and this phenomenon may obstruct the process of producing independent uniform samples. The suggested MCMC sampling framework, shown in Fig. 3, can overcome this limitation.

Initially inspired by the work of Ram and Chetty (2008), which proposed the use of an initial set of 10–20 dissimilar but high scoring BN (as regards to the probability of the network structure S given the input data D), the probability $p(S|D)$ could be used for calculating the Bayesian posterior probability distribution of all features. Our approach is different than the previous one in two points: the former deals with the fact that considering the top-k ranked networks would result in obtaining very similar network structures and therefore, would result in having a set of distributions with limited variation. The reason is that when using traditional scoring algorithms, such as K2 Heckerman (1999) or Bayesian Scoring Method Pearl (2000), each candidate network is produced from the previous, most-likely one by performing simple graph operation such as arc additions, removals or reversals. One possible solution would be to consider multiple and parallel search implementation, but this could create an extra computational overhead. The latter aspect that our framework differs is the orientation towards classification, which is not inherent in traditional BN approaches.

The approach of Ram and Chetty (2008) performs Bayesian inference on the class feature given the set of input variables having simulated generic network structures that do not consider the class node as a special one. Our suggestion focuses on creating simple and straightforward BN structures which are suitable for the classification process (nevertheless, the main goal is to predict the course of a stock). Such classification-oriented network structures are constructed using the Tree-Augmented Naïve Bayes (TAN) algorithm Pearl (1988). By definition, the TAN algorithm creates networks where the class node is a parent of all features nodes. The rest of the input

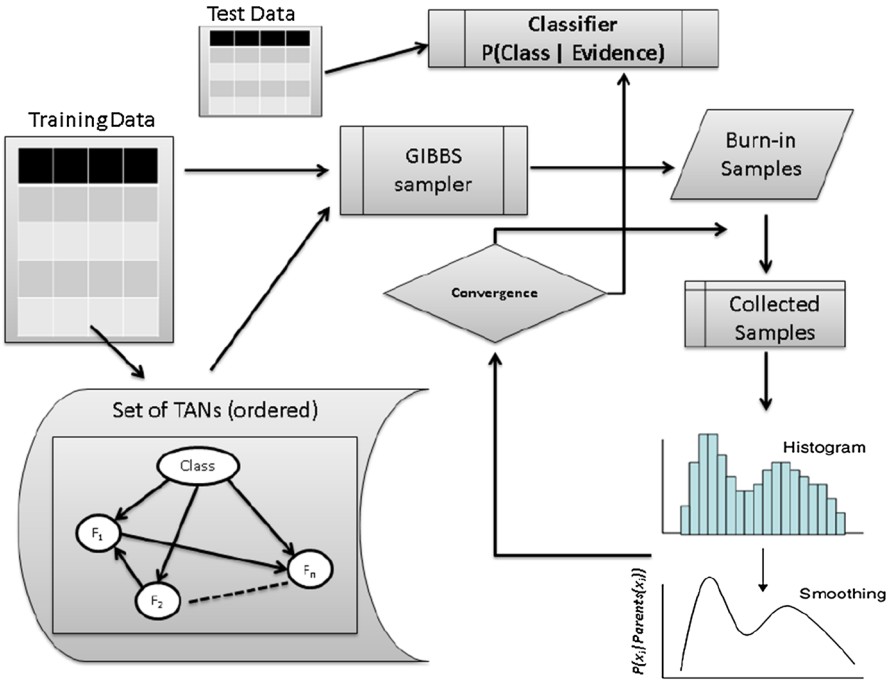


Fig. 3 The flowchart of the proposed methodology, showing its main components: TAN learning phase, Gibbs sampling phase and finally convergence phase

features form a traditional BN amongst them in which each node has one parent at most, in order to retain the structure and the CPD simple. Furthermore, compared to the traditional BN learning algorithms, the TAN methodology can produce networks approximately 50–100 times faster than the BN approach, depending on the number of input features and the number of states each feature has. Finally, TANs are considered more appropriate for classification than both BNs and the Naïve Bayes approach. This conclusion is attributed to the structural characteristics of the former, which considers the class node as a parent of all other nodes (as Naïve Bayes does) but also considers features as not to be conditional independent given the class node, a fact that is non-realistic in a plethora of domains.

From the samples drawn from a set of different TAN structures, we can obtain the posteriors after convergence, and then determine the probability estimates of the model in a straightforward manner. Despite the fact that the inferred high-scoring TAN structures are disjoint (i.e. cannot be combined into one network structure), they can all be combined independently to the underlying probability distribution. Hence, all of these network structures are sampled to estimate the probability distribution accurately. The important element of our methodology is the use of fast-learned TAN structures and a rank ordering amongst them. In the figure below, the flowchart of the proposed methodology can been inspected, depicting its above-mentioned main components: TAN learning phase, Gibbs sampling phase and, finally, the convergence phase.

5.1 TAN Phase

Based on the following process, a set of 10 TAN network structures was produced:

1. Built a Naïve Bayesian structure, where the class node C is a parent to all feature nodes X_i and all feature nodes are not connected with each other.
2. For each pair of different features X_i, X_j, compute the conditional mutual information given the class $I(X_i; X_j|C)$, using the formula provided below:
   (a) $I(X_i; X_j|C) = \sum_{X_i, X_j, C} \frac{p(X_i, X_j, C) log(p(X_i, X_j|C))}{p(X_i|C)p(X_j|C)}$
3. Build a complete, undirected graph to connect all features and use $I(X_i; X_j|C)$ to weight all arcs.
4. Build a maximum weighted spanning tree.
5. Transform the resulting undirected tree to a directed one by choosing a root feature and setting the direction of all edges to be outward from it.

For maximizing the performance of TAN, we applied a feature selection algorithm based on SVM Bi et al. (2003) and eliminated the features that scored below 0.1, thus achieving a mean value of 40–60 % reduction in the number of input features for the TAN learner. According to the authors of the aforementioned article, feature selection by SVM is more beneficial that other wrapper approaches such as Information Gain and Odds Ratio Brank et al. (2002) when being applied in high-dimensional datasets.

The different TAN structures were obtained by choosing different features as root, in the 5^{th} step of the previously mentioned TAN algorithm. As mentioned above, an ordinary Gibbs sampler chooses features at random and then samples a new value from the estimated posterior of the neighboring variables. Friedman and Koller (2003) argued that sampling from the space of total orders on variables rather than directly sampling DAGs was more efficient than application of ordinary MCMC directly in random manner. Since the Gibbs sampler also samples the new value of a feature based on the parent variables, an ordering of the rank of the TANs, based on their scores was applied. The score of each network S is calculated as the probability of S given dataset D, $p(S|D)$ and is given by the following formula Nummelin (2004):

$$p(S|D) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_i j + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}! \tag{6}$$

where n equals to the number of features, r_i denotes the number of values in the ith feature, q_i denotes the number of possible different value combinations the parent features can take, N_{ij} depicts the number of rows in data that have jth value combinations for parents of ith feature, N_{ijk} corresponds to the number of rows that have kth value for the ith variable and which also have jth value combinations for parents of ith variable.

Note that other graph scoring metrics could be used as well, such as the BIC-TAN measure, proposed by Chickering et al. (1995). The applied scoring metric was chosen because it is implemented in a variety of programming languages and is freely available.

5.2 Gibbs Sampling Phase

For the Gibbs sampling phase, uniform prior distributions for all the features in the domain needed to be defined. Instead of applying a random instantiation of the network, a multivariate Dirichlet distribution was chosen, inspired by Ram and Chetty (2008). The initial distribution of the values of nodes in the network was assigned by using the density function. It was estimated after smoothing of the histogram of normalized feature data. Since all nodes have parent(s), we sampled from the conditional distribution of their TAN. Similarly, n independent samples were drawn from the target distribution $P(x)$. The samples collected were plotted using a histogram with n bins as depicted in the figure of the flowchart above. The probability density function $P(x)$ of a continuous feature was approximated by smoothing the histogram.

5.3 Convergence Phase

Convergence is the process of reaching a stationary probability distribution. The initial phase of the convergence is called the *burn-in* phase. For the proposed approach, multiple TAN structures were fed to a parallelized series of Markov Chains, in order to obtain a large number of samples from the entire input space of the domain. Recall that each Markov Chain connects states of the network instantiation and sampling process. In other words, if S_0 represents the first instantiation of features $(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n)$ then we can sample a new value x_1 for feature X_1 using $p(X_1 = x_1 | X_2 = x_2, \ldots, X_n = x_n)$. In similar manner, we can sample the remaining new values for features $X_2, X_3, \ldots X_n$ until we have a new state S_1, instantiated as: $< X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n >$. In the above Fig. 4, a sample Markov Chain is depicted for a mock-up TAN structure, with two features, each being binary. The chain represents four states for each instantiation of features X_1 and X_2.

Throughout the process of multiple chain runs, samples are exchanged between the chains and the overall samples of a number of variables in the top of the specified order are monitored. When the sample values do not exceed a variation threshold (manually defined to 0.05) after a large number of iterations, convergence is assumed. Upon convergence on the stationary distribution, the process of classification of a previously unseen example is straightforward. We only compute the probability of the class c given evidence e (expressed as an input vector of the considered feature values), calculated as $p(c|e)$ and classify it to the most probable class.

6 Data Management

As mentioned earlier, articles containing financial news were combined with a plethora of technical indices in order to search for direct influence patterns of the former to the latter. More specifically, we focused on three heterogeneous stock securities from the Greek stock market (Athens Stock Exchange, .ATG), two large Greek banks (Alpha Bank, .ALPHA and Eurobank Ergasias Bank, .EUROB), the principal telecommunication provider of Greece (OTE, .OTE) and one of the biggest Greek airline companies (Aegean, .AEGN). We included past data from the major European, Asian and Ameri-

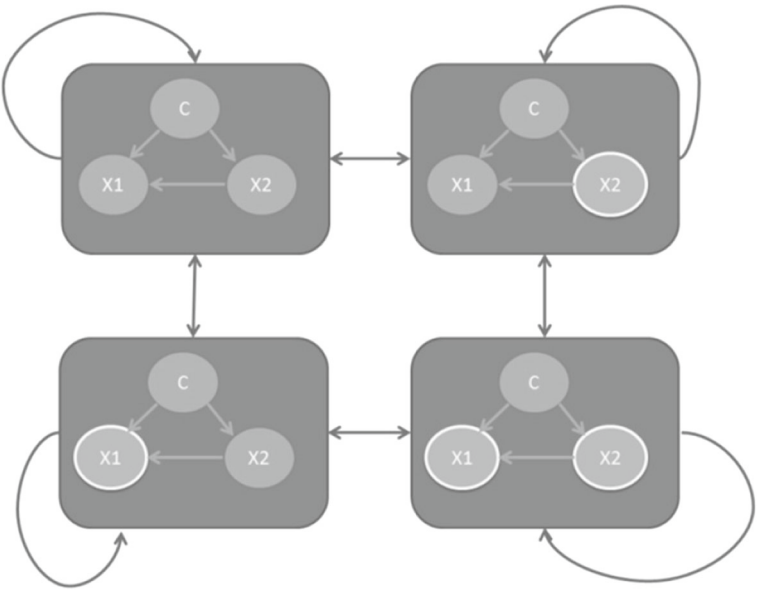


Fig. 4 An example Markov Chain for a mock-up TAN structure—Each feature is instantiated to either true (highlighted circle) or false (non-highlighted circle) at each state

Table 1 The benchmark tickers

Name	Category	#Articles	#Opinions	Symbol
O.T.E.	Telephony	2623	4250	.OTE
Alpha Bank	Bank	1879	3014	.ALPHA
Eurobank Ergasias	Bank	1988	3443	.EUROB
Aegean Airlines	Airline	1013	1505	.AEGN

can stock markets, as well as data from energy and metal commodities. Finally, for each of the aforementioned three stock securities, a variety of major technical indices was employed. News articles were automatically extracted from the electronic versions of the leading Greek financial newspapers, i.e. "Naftemporiki" (www.naftemporiki.gr) and "Capital" (www.capital.gr). Opinions were also crawled from the afore-mentioned financial portals, as well as from Twitter, using the corresponding API and selecting posts that contained either the hashtag of each company or the name of it. The time period for all collected data was from January 2013 to January 2014. The technical indices were specified using the AnalyzerXL (www.analyzerxl.com) software tool. Table 1 tabulates data regarding the three benchmark stocks and their corresponding articles that were collected, while Table 2 presents historical data of other main markets and commodities.

Tables 3 and 4 refer to metal and energy fuel data considered respectively and Table 5 depicts a categorized list of the technical indices and liquidity proxies that were also taken into consideration. As regards to the latter, given the inclusion of data

Table 2 Market and commodities data

Category	Description	#Days	Symbol
European markets	FCHI-CAC 40 index	212	.FCHI
	FTSE-FTSE 100 index	215	.FTSE
	GDAXI-Xetra dax index	207	.GDAXI
	ATG-Athens stock exchange	211	.ATG
Asia/Pasific markets	HIS-Hang Seng index	216	.HSI
	AORD-All ordinaries index	169	.AORD
	N225-Nikkei 225 average index	156	.N225
United States markets	GSPC-S&P 500 index	158	.GSPC
	IXIC-Nasdaq composite index	131	.IXIC
	DJI-Dow Jones industrial average index	139	.DJI
Energy	Brent DTD	166	BRT-
	WTI CUSHING	121	WTC-
Metals	Silver	187	XAG-HH
	Gold bullion	187	XAU-B-HH

Table 3 Metals data

Name	Number of quotes
Silver	178
Gold bullion	178

Table 4 Energy fuel data

Name	Number of quotes
BRENT DTD	207
WTI CUSHING	202

series such as metals and commodities we considered useful to additionally include measures of liquidity, based on a rapidly growing literature that uses liquidity proxies constructed from low-frequency (daily) stock data Fong et al. (2011).

Stock quotes were gathered on a per day basis and articles were aligned according to their release date. In case an article was published on a Friday evening (after the closing of the Athens stock market) or during the weekend, it was considered as published on Monday. Figure 5 and its sub-figures, depict the closing values of each of the above mentioned stocks that were selected as evaluation data, along with volume data. The left axis symbolizes volume and the right axis represents the closing value in Euros. The impressive increase in the values of Eurobank (Fig. 5c) was caused due to the cancellation of a potential merging (announced on April 8, 2013) with another major bank of Greece, named as National Bank of Greece (NBG). At the same period, a burst was also observed in terms of available news and comments for this company. While the mean value of the distributions for news was about 4.4 per day and about

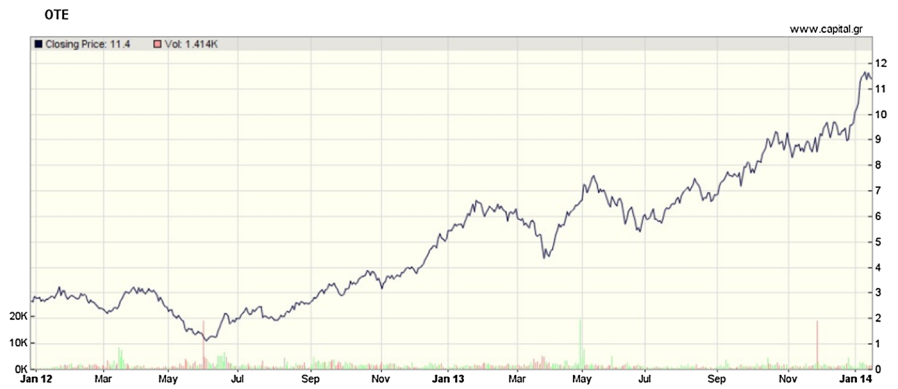
Table 5 Technical Indices categorized list

Group name	Function or indicator name
Basic functions	Median price (AKA Typical Price Indicator)
Statistical functions	Standard deviation
Trend indicators	MACD indicator
	Simple moving average
	Exponential moving average
	Line weighted moving average
Volatility indicators	Average true range
	Bollinger band width
Liquidity proxies	Amihud liquidity measure
	LOT Mixed impact
Momentum indicators	Williams %R
	TRIX indicator
	Wilder RSI indicator
	Chande momentum oscillator
	Price Rate-of-charge indicator
	Cutler's relative strength index
	DX (Directional movement indicator)
	Stochastic oscillator
	Price oscillator percentage difference
Market strength indicators	Chaikin A/D oscillator
	Average of volume ROC
	Market facilitation index (MFI)
Support and resistance indicators	Envelope

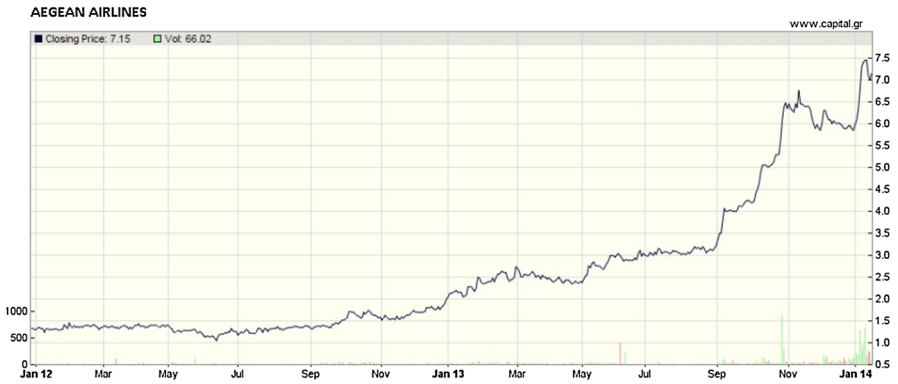
14.2 for opinions, after the announcement of the cancellation of the merge and for a period of more than 20 days, the distributions were boosted to 25.5 and 123.5 for news and opinions respectively.

6.1 Text Mining and Sentiment Annotation

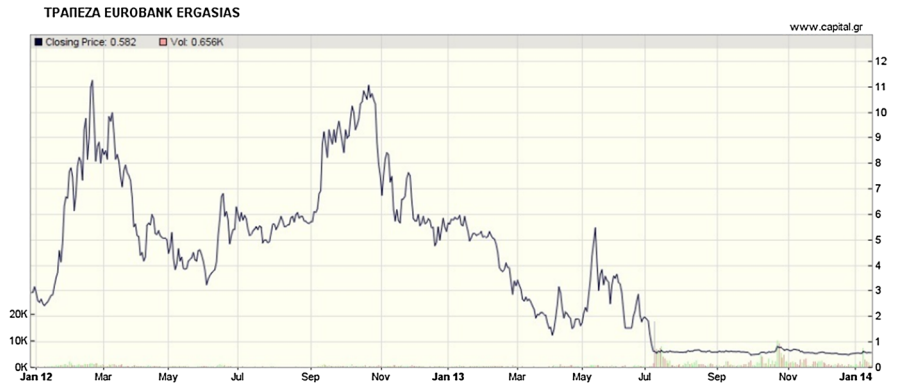
The plethora of available text in general and news articles in particular nowadays has shifted the focus of researchers to mining information from unstructured and semi-structured information sources. The text content of the latest news articles and financial reports is taken into account when trying to automatically predict stock behavior. The web sentiment, i.e. "positive", "neutral" or "negative" content of web articles regarding a stock, has been exploited previously using a "bag-of-words" model or a more sophisticated language model Sehgal and Song (2007). Automated approaches to sentiment analysis also include the use of Word Net in order to estimate the sentiment of words appearing in the text by measuring their semantic similarity with prototype



(a)

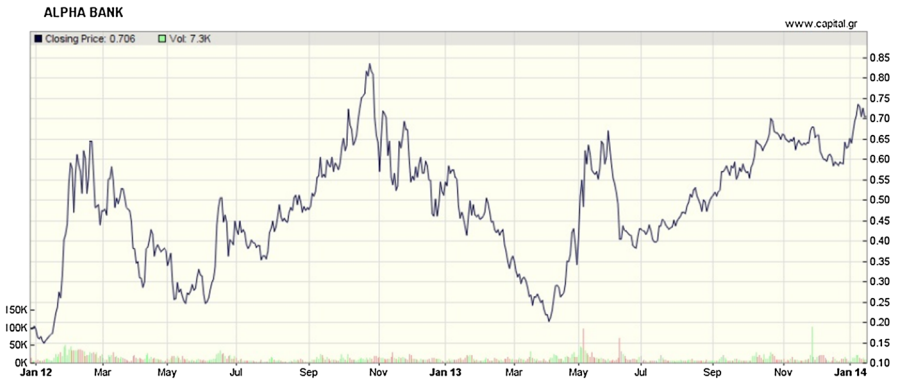


(b)



(c)

Fig. 5 Daily Close values for the 4 benchmark stocks. **a** .OTE (Telecommunications), **b** .AEGN (Airlines), **c** .EUROB (Bank), **d** .ALPHA (Bank)



(d)

Fig. 5 continued

positive and negative words (e.g. "good" or "bad"). Due to the lack of such sophisticated resources for Modern Greek , in the present work the sentiment of a word is determined manually, by a financial analysis expert, and the cumulative weight of word sentiments is used to estimate the sentiment of the entire text. The words in the acquired lexicon with an occurrence frequency of at least five times have been semantically annotated by domain experts, according to their positive or negative meaning for stock value prediction. One of five discrete weights, i.e. -2 (clearly negative), -1 (relatively negative), 0 (neutral), $+1$ (relatively positive), $+2$ (clearly positive) was assigned to each word. The aforementioned weighting scheme was previously applied by Klibanoff et al. (1998) and provided satisfactory results. The reason for selecting five as a threshold of term frequency was mainly attributed to the fact that it balances the number of extracted words with the painstaking process of manual annotation by the domain expert. Table 6 displays the ten most frequent words for each semantic label. The textual analysis phase consisted of three activities:

1. Removal of stop words (i.e. words that are filtered out since they do not provide any special meaning to the text mining concept. Usually they contain articles, pronouns, special characters, etc.)
2. Lemmatization (i.e. the process of grouping together the various inflected forms of a word so they can be analyzed as a single item) of words using a Levenshtein distance based Greek lemmatizer Lyras et al. (2007).
3. Removal of terms appearing less than 30 times within the complete article corpus and taking the 150 most frequent of them.

Manual consideration of a sentiment lexicon has some drawbacks. Lexical ambiguity is a common problem even for Greek, despite the fact that it is highly inflectional. Furthermore, the sentiment of a given term could change over time. For example, the term *Κύπρου (Cyprus)* could intuitively associated with a negative sentiment, mainly due to the latest economic events. In our case the same term was considered as neutral because at the time of creation, the domain expert was not aware of the future problems. Finally, the context within a word also influences the sentiment, particularly

Table 6 A sample of frequent terms and their annotated weight

+2	+1	0	-1	-2
Κέρδος (Profit)	Έσοδο (Income)	Ευρώ (Euro)	Κόστος (Cost)	Μείωση (Decrease)
Αύξηση (Increase)	Χορήγηση (Sponsorship)	Πειραιώς (Piraeus)	Χαμηλός (Low)	Κρίση (Crisis)
Ανάπτυξη (Development)	Κατάθεση (Deposit)	Τράπεζα (Bank)	Κρίσιμος (Critical)	Απώλεια (Loss)
Κερδοφορία (Profit)	Υψηλός (High)	Εκατ (Million)	Πρόβλημα (Problem)	Πτώση (Drop)
Άνοδος (Ascension)	Ισχυρός (Strong)	Μετοχές (Stock)	Πίεση (Pressure)	Επιβράδυνση (Deceleration)
Βελτίωση (Improvement)	Επέκταση (Expansion)	Δις (Billion)	Υποχώρηση (Retreat)	Υποβάθμιση (Degradation)
Θετικός (Positive)	Απόδοση (Yield)	Τιμή (Value)	Έκτακτος (Unscheduled)	Ζημιά (Damage)
Αναβάθμιση (Upgrade)	Ενδιαφέρον (Interest)	Κύπρου (Cyprus)	Καθυστέρηση (Delay)	Κίνδυνος (Danger)
Επιτυχία (Success)	Προσφορά (Offer)	Ανακοίνωση (Announcement)	Επίπτωση (Consequence)	Αρνητικός (Negative)
Ενίσχυση (Strengthening)	Συμφωνία (Agreement)	Αγορά (Buy)	Δύσκολος (Difficult)	Επιδείνωση (Deterioration)

when dealing with social media content. For example, in ordinary situations the term *Ανάπτυξη (Development)* is definitely positive. However, in many social media posts this is often written together with emoticons (i.e. pictorial representation of a facial expression) that denote irony or frustration, as "Development" was a main political slogan by the Greek government during the austerity period that faced significant controversy by citizens. Certainly, the constant update of such domain-specific lexicons is of major importance, particularly for language with limited linguistic resources.

The class attribute to be predicted was set to the closing value of the following day. Since the proposed method is suited for nominal classification, we discretized the class attribute from numerical into nominal according to the following rule: A value from a set of three discrete labels, namely *UP*, *STEADY* and *DOWN* is chosen, if the stock quote closed at a price more than 1 %, between 1 and -1 % and less than -1% in the following day respectively. A window of five days was chosen empirically in order to predict the class, resulting in a high-dimensional dataset of more than 720 features.

Table 7 summarizes the properties of the system described above. It is organized in four parts: The first part provides a rough idea about our prototype, the second parts details the parameter settings for the techniques used, the third part summarizes the data used for training, and the final part gives an overview of the major performance figures reported, explained in more details in the following section.

7 Experimental Results

Textual as well as stock quotes data was processed by our proposed methodology (MCMC-TAN) and evaluated against several well known classifiers that are either used in previous researches or are well-known for their robustness when dealing with high-dimensional data. More specifically, we considered the traditional Naïve Bayesian classifier (NB), an ensemble algorithm named as Random Forests (RF), proposed by Breiman (2001), which is proved to perform feature selection, Radial Basis Functions neural network (RBF) that has two layers and is a special class of multi-layered feed forward networks. Finally, the last benchmarking algorithm used in our evaluation is a well-known machine learning classifier that has previously applied in a plethora of financial forecasting applications Huang et al. (2005), namely Support Vector Machines (SVM). Following the suggestion of Huang et al. (2005), a radial kernel was utilized in the case of SVM.

Table 7 Summarized properties of our prototype

Prototype idea	
Aims to forecast...	Price trends
Underlying	Technical indices
Forecast horizon	24 h
Text mining	
Feature definition	Manually
Number of features	150
Feature granularity	Words
Primary classifier	MCMC-TAN
Number of class labels	3
Stock Market Parameters	
Number of Features	42
Input data	
Information age	0–24 h
Text analyzed	Head/body
Labeling	Manually
Price frequency	Daily close
Test set	
Period	January 2013 to January 2014
Training/Test ratio	10-fold cross validation
Market	4 stocks (Athens Stock Exchange—Greece)
Prototype accuracy versus Regression model	20.6 % gain

Note that some of the aforementioned algorithms, including the proposed method of MCMC-TAN work well in discrete valued labels and some other operate solely with numerical labels. More specifically, MCMC-TAN, NB and RF were tested on the discretized data set, where the class attribute contained one of the three nominal values. RBF and SVM require a numerical label, therefore, in this case, the predicted value was compared to the actual value of the previous day and according to their difference, the forecast was discretized based on the approach described earlier (i.e. *UP*, *STEADY* and *DOWN*).

Regarding the experimental design, two different approaches were followed:

Experiment 1: In order to evaluate the impact of articles on the predictability of a stock closing trend (i.e. *UP*, *STEADY*, *DOWN*), the former approach dealt with standard, 10-fold cross validation, classification in terms of stock quotes closing price, using:

(a) a dataset that contained only technical analysis indices (TA),
(b) a dataset that contained technical analysis indices and news articles (TA-NA),
(c) a dataset that consisted of (b) plus public opinions from users (TA-NA-PO) and, finally,
(d) a dataset that contained technical analysis indices plus opinions (TA-PO).

Table 8 Confusion matrix and Recall and Precision metrics for each class (T and F)

	Predicted as...			Precision T	Recall T	Precision F	Recall F
		T	F	$\frac{a}{a+c}$	$\frac{a}{a+b}$	$\frac{d}{b+d}$	$\frac{d}{c+d}$
Actual class	T	a	b	f-measure:	$\frac{2 \times (precision \times recall)}{precision+recall}$		
	F	c	d				

In order to evaluate the performance of the aforementioned algorithms, the F-measure measurement was used, which is the harmonic mean of precision and recall, two fundamental metrics of data mining algorithms as regards to classification evaluation. In particular, the precision metric of a classifier about a class label is defined as the percentage of correctly classified instances among those that the algorithm classifies to belong to this class. The recall of a class is then defined as the fraction of correctly classified in it instances among all instances that actually belong to this class. These definitions are illustrated in the following Table 8 showing a "confusion matrix" which tabulates the classification performance of an algorithm (the columns of the table) against the actual class distribution (the rows of the table) for a binary classification problem (classes T and F). Lower case letters a,b,c and d represent the number of times (an integer value) an instance belonging to an actual class (either T or F) was predicted to belong to the same or a different class.

Two additional model performance measures that are common in forecasting applications (e.g. Clark and McCracken (2013) and West (2006)) have been taken into consideration, namely Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). The former is defined by:

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |P_i - T_i| \tag{7}$$

where P_i is the value that the model predicted for the i^th sample (from a set of n examples) and T_i is the target value for that same example. Similarly, the latter measure is calculated by:

$$RSME = \sqrt{\frac{\sum_{i=1}^{n} (P_i - T_i)^2}{n}} \tag{8}$$

Tables 9, 10, 11 and 12 tabularize the F-measure score of all machine learning algorithms for each of the four experimental stocks. Correspondingly, Figs. 6 and 7 illustrate the MAE and RMSE of each algorithm against each stock and each dataset respectively.

From the outcomes of the F-measure as well as the MAE and RMSE, we could initially observe that combining information from both time series and textual data leads to a substantial improvement of forecast for all adopted methodologies. Furthermore, by using only technical analysis data, SVM perform similar to MCMC-TAN

Table 9 Classification performance in terms of F-measure for .OTE

Dataset	MCMC-TAN	RF	SVM	NB	RBF
(a) Technical analysis (TA)	65.6	56.5	65.5	56.2	53.5
(b) TA including news articles (TA-NA)	73.6	60.8	68.3	58.3	56.7
(c) TA-NA including public opinions(TA-NA-PO)	73.5	60.7	66.9	58.1	56.4
(d) TA including public opinions (TA-PO)	64.6	55.2	64.6	54.4	53.2

Table 10 Classification performance in terms of F-measure for .ALPHA

Dataset	MCMC-TAN	RF	SVM	NB	RBF
(a) Technical analysis (TA)	66.3	61.5	64.3	59.6	62.4
(b) TA including news articles (TA-NA)	68.8	63.5	67.0	61.0	64.8
(c) TA-NA including public opinions(TA-NA-PO)	67.4	64.0	66.7	63.1	62.8
(d) TA including public opinions (TA-PO)	65.8	60.7	65.0	59.3	61.5

Table 11 Classification performance in terms of F-measure for .EUROB

Dataset	MCMC-TAN	RF	SVM	NB	RBF
(a) Technical analysis (TA)	71.5	58.7	70.7	58.3	58.6
(b) TA including news articles (TA-NA)	77.5	62.3	73.0	61.6	62.3
(c) TA-NA including public opinions(TA-NA-PO)	76.9	62.2	73.4	59.4	59.6
(d) TA including public opinions (TA-PO)	72.4	58.3	70.4	58.1	58.7

Table 12 Classification performance in terms of F-measure for .AEGN

Dataset	MCMC-TAN	RF	SVM	NB	RBF
(a) Technical analysis (TA)	66.9	55.3	63.7	57.1	52.6
(b) TA including news articles (TA-NA)	75.5	58.9	66.2	59.8	55.2
(c) TA-NA including public opinions(TA-NA-PO)	76.5	58.3	64.2	60.4	54.2
(d) TA including public opinions (TA-PO)	67.7	52.4	61.04	57.0	50.6

and significantly outperform all other approaches in most of the cases, while when incorporating textual information from news, MCMC-TAN is noticeably the best classification approach. This claim is also supported by the MAE and RMSE metrics, in which MCMC-TAN is constantly outperforming all other models, except for some cases which SVM seem to perform similarly to our method. Especially for the case of .EUROB, the improvement when incorporating textual features reached 6 % for MCMC-TAN method and also about 2 to 5 % for the other algorithms. This is mainly attributed to the large number of news that were generated at a specific time of period where a potential merge with another Greek bank was cancelled. Additionally, by

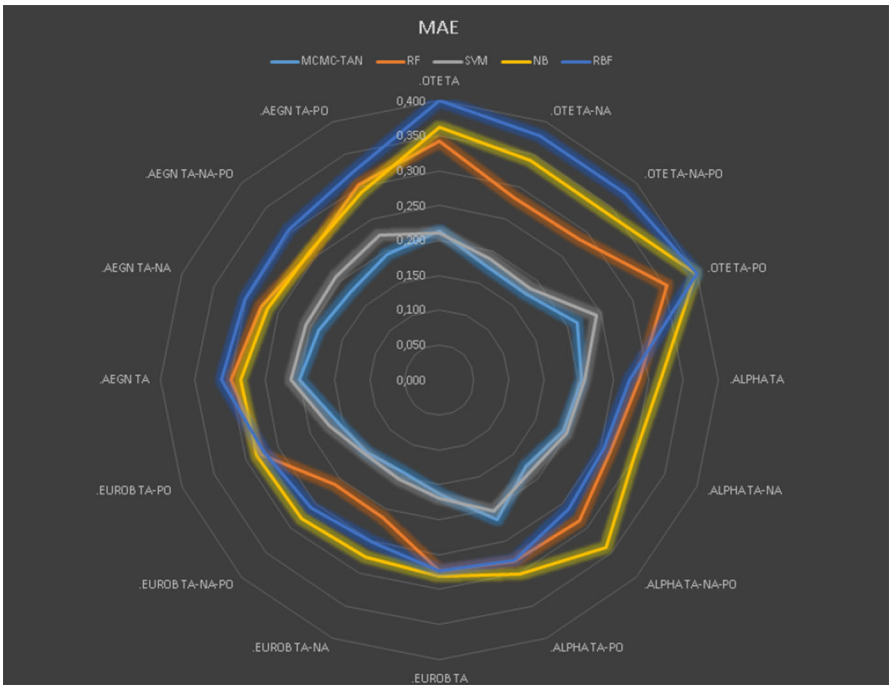


Fig. 6 Radar Plot of MAE for all models against all datasets per stock quote

observing the outcomes of Tables 9, 10, 11 and 12, one could observe that the performance of MCMC-TAN are one of the highest ever reported, with the drawback of a time and resource consuming training phase. Training times for each algorithm are reported in the following sub-section. Despite of the initial assumption that public opinion may play an important role to the task of classifying stock market trends, results do not support this claim. In the majority of cases, comments do not seem to affect accuracy in a clear positive or negative manner, meaning that in some cases there is a gain from 1 to 2 % but there are also cases where one could observe loss by the same range. The same applies when public opinions are combined with technical analysis indices without the news, i.e. performance is more or less the same or even worse. A closer look on opinions crawled from Twitter in problematic cases revealed two observations. First, numerous posts were simply re-transmitting previous posts (i.e. re-tweet). Second, the majority of tweets used abbreviations, idioms, emoticons and other peculiar linguistic expressions which were not captured by our limited lexicon-based text mining approach. Finally, a potential logical explanation could be the argument that most of public opinions are mainly triggered by the events that are already described in news, therefore, the latter are the main form of expressing an event and opinions may act supplementary. Nevertheless, in order to fully support this claim, advanced text mining techniques should be applied to social media opinions, which is not a straightforward approach, especially for languages with limited lexical and syntactical resources, such as Modern Greek.

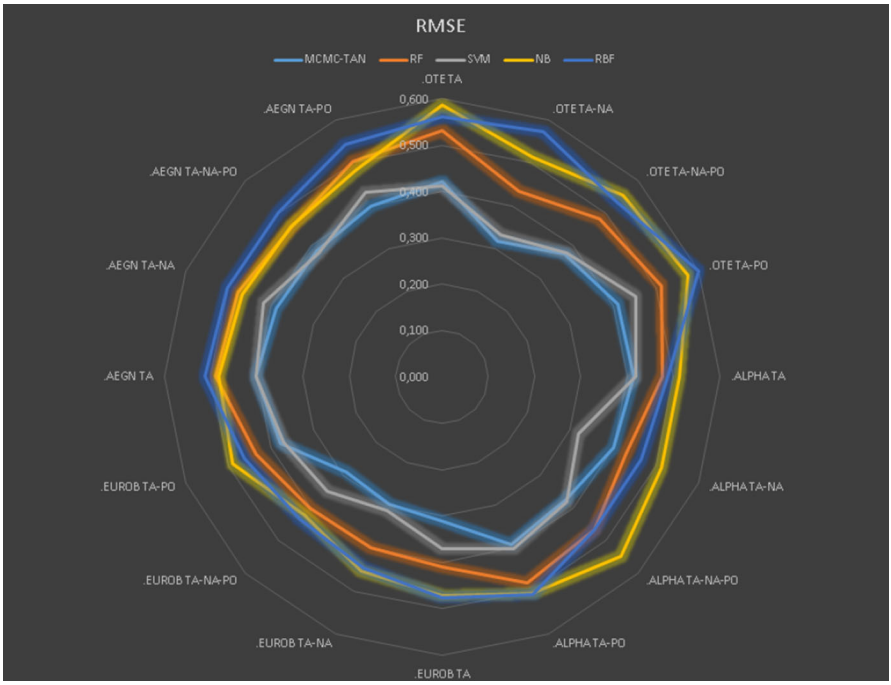


Fig. 7 Radar Plot of RMSE for all models against all datasets per stock quote

Experiment 2: The latter experimental design focused on a simulated trading strategy, in an attempt to further study whether the MCMC-TAN model could practically be applied to generate higher profits than those earned by employing the traditional regression model by simply following a buy-and-hold (passive) investment strategy. The operational details of the trading simulation are explained as follows: The trading simulation assumes that the investor has 1,00,000€ to create a portfolio by selecting a balanced percentage of each of the four Greek stock quotes mentioned earlier. Each day, the investor could buy, sell or wait, according to the class prediction of the MCMC-TAN model, trained for case (b) where all technical indices were combined with news articles (i.e. the TA-NA dataset), since that was the model with the highest rankings in prediction accuracy. This strategy is most suitable for the assumption of buying long at open rather than selling short. Despite the fact that the methodology that would be employed in the latter case is analogous to the former, short selling using machine learning approaches is a process that requires a larger number of training instances and a broader forecasting horizon (e.g. weeks instead of days). We assumed that transactional costs apply when buying or selling, according to Table 13. As one could observe, no short-selling costs have been taken into account. More specifically, the transactional costs of buying are estimated according to formula:

$$\begin{aligned} \textit{Transaction costs of buying} &= \textit{Commission} + \textit{Expenses ASE} \\ &+ \textit{Costs of transfers} = 0.335\,\% \end{aligned} \tag{9}$$

Table 13 Transaction costs

Parameter	Value (%)
Commission	0.250
ASE expenses	0.025
Sales Tax	0.015
Costs of transfers	0.060

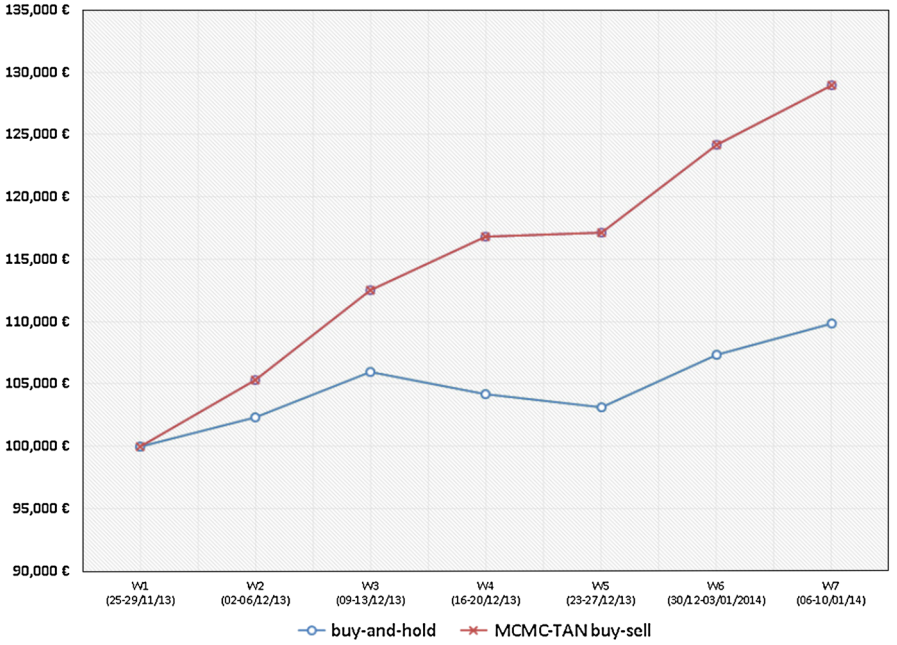


Fig. 8 Plot of portfolio outcomes using the two different trading strategies

while he transactional costs of selling are given by:

$$\begin{aligned} \textit{Transaction costs of selling} = \textit{Commission} + \textit{Expenses ASE} + \textit{Sales Tax} \\ + \textit{Costs of transfers} = 0.35\,\% \end{aligned} \tag{10}$$

Furthermore, a random choice between 5 and 10 % of the current portfolio volume was allowed to be traded each day. The time period was set to the last 35 weekdays of the aforementioned dataset, i.e. from 25/11/2013 to 10/01/2014. As Fig. 8 depicts, the red line, which represents the portfolio budget for the MCMC-TAN investing strategy is clearly outperforming the blue line of the *buy-and-hold* investment strategy by an average factor of 4.5 to 10 % for the first 4 weeks and from 12 to 17 % for the remaining ones, resulting in a profit of approximately 29,000€ rather than 9000€ from the *buy-and-hold* approach.

Table 14 Training time for each algorithm

Methodology	Training time (min)
Markov Chain Monte Carlo TAN (MCMC-TAN)	25.1
Random forests (RF)	1.7
Support vector machines (SVM)	2.5
Naïve Bayesian classifier (NB)	0.6
Radial basis functions (RBF)	2.4

7.1 Training Time

The experimental section concludes with a comparison of training time of all methodologies for each evaluation approach. Time is measured in seconds. The hardware configuration of the setting includes a modest PC of an Intel-I5® quad-core processor at 2.4 GHz with 6GB of RAM (5GB usable by the algorithms). MCMC-TAN is obviously the slowest of the selected algorithms as regards to training time. This is attributed to the time consuming process of finding the most probable Bayesian network structure, from which the MCMC step is initiated at each run. Note however, that this process is executed in an off-line manner, while the improvement of accuracy is of major importance, especially in financial domains, and finally, this process could straightforwardly be parallelized as in each run, different instances are selected for Bayesian network training. Table 14 tabularizes the elapsed time during the training stage (evaluating the class label of a previously unseen instance was performed instantaneously for each algorithm, therefore, it is not considered).

8 Concluding Remarks

The present article considered a novel Markov Chain Monte Carlo (MCMC) Bayesian Inference approach, which estimates conditional probability distributions in structures obtained from a Tree-Augmented Naïve Bayes (TAN) algorithm. The proposed methodology can handle datasets characterized by numerous continuous input features; it addresses the inherent limitations and challenges of using BN for such stock market forecasting problems. The proposed technique has been used for the stock market forecasting, a dynamic and volatile domain of high dimensionality, due to the fact that recent trends in stock market analysis pose the use of textual information apart from technical indices. Additionally, the influence of opinionated texts as expressed in various Web 2.0 platforms has been also taken into consideration. For this purpose, technical analysis indices have been expanded to include market and commodities data, financial news articles and opinions of users have been analysed using simple sentiment tagging approaches. Studies on four different Greek companies have resulted in various datasets, which are combinations of the above features, mirroring the course of each company as regards to its closing stock value from January 2013 to January 2014.

This first prototype gave encouraging results, taking into account the high levels of noise such as the heterogeneity of our dataset (in terms of acquisition method and conditions) adds and the difficulties this creates for forecasting. In particular, the proposed ensemble classification technique exhibited a higher classification performance than the traditional machine learning algorithms. Results support the claim that incorporation of financial news as a source of information may play a significant role to the improvement of forecasting, while this did not seem to validate for the case of public opinions. However, despite the fact that a plethora of sentiment-related features were manually annotated, we believe that by utilizing sophisticated Natural Language Processing tools for Modern Greek could automate the laborious tasks of hand-labeling features and also improve results.Taking into account the increasing trend for the electronic/digital acquisition of various data from financial resources, the development of such advanced and high performing classification techniques contributes to the emergence of the 'intelligent' investing tools. Further research is required for validation of this potential in other financial problems using various types of features will probably lead to improvement of the proposed technique.

An important drawback is obviously the time consuming process of the MCMC inference stage, at each TAN construction phase. Even though this process can be parallelized and improved, and even though testing an unknown instance is a rapid process, we believe that when the learning phase is improved our technique could be applied to datasets of massive volume of features. In addition to supervised classification, the idea of using semantic random sampling of features instead of simple random sampling has the potential to be effective in the unsupervised classification problem as well. It is our future plan to continue developing this work in that direction.

References

Atsalakis, G., & Valavanis, K. (2009). Surveying stock market forecasting techniques - Part II: Soft computing methods. *Expert Systems with Applications*, *36*, 5932–5941.

Bebarta, D. K., Biswal, B., & Dash, P. K. (2012). Comparative study of stock market forecasting using different functional link artificial neural networks. *International Journal of Data Analysis Techniques and Strategies*, *4*(4), 398–427.

Bettman, J. L., Sault, S. J., & Schultz, E. L. (2009). Fundamental and technical analysis: Substitutes or complements. *Accounting and Finance: ACCOUNT FINANC*, *49*(1), 21–36.

Bi, J., Bennett, K., Embrechts, M., Breneman, C., & Song, M. (2003). Dimensionality reduction via sparse support vector machines. *Journal of Machine Learning Resources*, *3*(Mar), 1229–1243.

Bilson, C. M., Brailsford, T. J., & Hooper, V. J. (2001). Selecting macroeconomic variables as explanatory factors of emerging stock market returns. *Pacific-Basin Finance Journal*, *9*(4), 401–426.

Bollen, J., Mao, H., & Pepe, A. (2010). Determining the public mood state by analysis of microblogging posts. In Proceedings of the Alife XII Conference, Odense, Denmark. MIT Press.

Brank, J., Grobelnik, M., Milic-Frayling, N., & Mladenic, D. (2002). Feature selection using support vector machines. *Proceedings of the 3rd international conference on data mining methods and databases for engineering, finance, and other fields*. September 2002, Bologna, Italy.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*, 5–32.

Chan, Y., & John-Wei, K. C. (1996). Political risk and stock price volatility: The case of Hong-Kong. *Pacific-Basin Finance Journal*, *4*(2–3), 259–275.

Chandra, D. K., Ravi, V., & Ravisankar, P. (2010). Support vector machine and wavelet neural network hybrid: Application to bankruptcy prediction in banks. *International Journal of Data Mining, Modelling and Management*, *2*(1), 1–21.

Chen, N. (1991). Financial investment opportunities and the macroeconomy. *The Journal of Finance*, *46*(2), 529–554.
Chickering, D., Geiger, D., & Heckerman, D. (1995). Learning Bayesian networks: Search methods and experimental results. In *Proceedings of 5th conference on artificial intelligence and statistics* (pp. 112–128). Fort Lauderdale, FL.
Cho, V. (1999). *Knowledge discovery from distributed and textual data*. Hong Kong: Dissertation Hong Kong University of Science and Technology.
Chung, F., Fu, T. Luk, R. & Ng, V. (2002). Evolutionary time series segmentation for stock data mining, In *Proceedings of IEEE international conference on data mining*, pp. 83–91. Larnaca.
Clark, T. E., & McCracken, M. W. (2013). Testing for unconditional predictive ability. In G. Elliott & A. Timmermann (Eds.), *Handbook-of-economic-forecasting* (Vol. 2). North-Holland: Elsevier.
Falinouss, P. (2007). Stock trend prediction using news articles: A text mining approach, Master's Thesis, Lulea University of Technology.
Fama, Eugene. (1970). Efficient capital markets: A review of theory and empirical work. *Journal of Finance*, *25*, 383–417.
Fellbaum, Christiane (Ed.). (1998). *WordNet: An electronic lexical database*. Cambridge, MA: MIT Press.
Fong, K., Holden, C., & Trzcinka, C. (2011). What are the best liquidity proxies for global research?. Available at SSRN: http://ssrn.com/abstract=1558447
Friedman, N., & Koller, D. (2003). Being Bayesian about network structure: A Bayesian approach to structure discovery in Bayesian networks. *Machine Learning*, *50*, 95–126.
Fung, G.P.C., Yu, J.X., & Lam, W. (2003). Stock prediction: Integrating text mining approach using real-time news. In *Proceedings IEEE international conference on computational intelligence for financial engineering* (pp. 395–402). Hong Kong.
Heckerman, D. (1999). A tutorial on learning with Bayesian networks. In M. Jordan (Ed.), *Learning in graphical models*. Cambridge: MIT Press.
Huang, W., Nakamori, Y., & Wang, S.-Y. (2005). Forecasting stock market movement direction with support vector machine. *Computer and Operations Research*, *32*, 2513–2522.
Jayech, S., & Zina, N. B. (2012). Measuring Financial contagion in the stock markets using a copula approach. *International Journal of Data Analysis Techniques and Strategies*, *4*(2), 154–180.
Klibanoff, P., Laymont, O., & Wizman, T. A. (1998). Investor reaction to Salient news in closed-end country funds. *Journal of Finance*, *53*(2), 673–699.
Kumar, D. A., & Ravi, V. (2008). Predicting credit card customer churn in banks using data mining. *International Journal of Data Analysis Techniques and Strategies*, *1*(1), 4–28.
Liu, J. S. (2001). *Monte Carlo strategies in scientific computing*. Heidelberg: Springer.
Liu, Y., Huang, X., An, A., & Yu, X. (2007). *ARSA: a sentiment-aware model for predicting sales performance using blogs*. New York, NY: ACM.
Lo, A. W., & MacKinlay, A. C. (1988). Stock market prices do not follow random walks: Evidence from a simple specification test. *The Review of Financial Studies*, *1*(1), 41–66.
Lunn, A., Thomas, G., Best, H., & Spiegelhalter, D. (2000). WinBUGS - A Bayesian modeling framework: Concepts, structure, and extensibility. *Statistics and Computing*, *10*, 325–337.
Lyras, D., Sgarbas, K., & Fakotakis, D. (2007). Using the Levenshtein edit distance for automatic lemmatization: A case study for modern greek and english, 19th IEEE international conference on tools with artificial intelligence (Vol. 2, pp. 428–435). Patras.
Mitchell, M. L., & Mulherin, J. H. (2002). The impact of public information on the stock market. *Journal of Finance*, *49*(3), 923–950.
Mittermayer, M.A. (2004). Forecasting intraday stock price trends with text mining techniques. In *Proceedings of the 37th annual Hawaii international conference on system sciences (HICS)*. IEEE Computer Society (vol. 3(3), 30064.2.) Washington, DC.
Ng, A., & Fu, A.W. (2003). Mining frequent episodes for relating financial events and stock trends. In *Proceedings of the 7th Pacific-Asia conference on advances in knowledge discovery and data mining, lectures notes in computer science* (vol. 2637, pp. 27–39). Seoul.
Nummelin, E. (2004). *General irreducible Markov chains and non-negative operators*. Cambridge: Cambridge University Press. 1984.
Nunez-Letamendia, L., Pacheco, J., & Casado, S. (2011). Applying genetic algorithms to wall street. *International Journal of Data Mining, Modelling and Management*, *3*(4), 319–340.

Oyatoye, E. O., & Arilesere, W. O. (2012). A non-linear programming model for insurance company investment portfolio management in nigeria. *International Journal of Data Analysis Techniques and Strategies*, *4*(1), 83–100.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks for plausible inference*. San Mateo: Morgan Kaufmann.

Pearl, J. (2000). *Causality: Models, reasoning and inference*. Cambridge: Cambridge University Press.

Preis, T., Moat, H.S. & Stanley, H.E. (2013). Quantifying trading behavior in financial markets using google trends, Scientific Reports, *3*.

Ram, R., Chetty, M. (2008). Constraint minimization for efficient modeling of gene regulatory network. In: M. Chetty, A. Ngom, S. Ahmad (Eds.) PRIB 2008. LNCS (LNBI) (vol. 5265, pp. 201–213) Heidelberg:Springer.

Sehgal, V., & Song, C. (2007). SOPS: Stock prediction using web sentiment. *Proceedings of the 7th IEEE international conference on data mining workshops*. Los Alamitos, CA.

Shumaker, R.P., & Chen, H. (2006). Textual analysis of stock market prediction using financial news articles, On the 12th American conference on information systems (AMCIS).

Technical-analysis. The trader' s glossary of technical terms and topics. (2005). http://www.traders.com

Thomas, J.D., & Sycara, K. (2000). Integrating genetic algorithms and text learning for financial prediction. In: *Proceedings GECCO-2000 workshop on data mining with evolutionary algorithms* (pp. 72–75). Las Vegas.

Vasu, M., & Ravi, V. (2011). A hybrid under-sampling approach for mining unbalanced datasets: Applications to banking and insurance. *International Journal of Data Mining, Modelling and Management*, *3*(1), 75–105.

West, K. D. (2006). Forecast evaluation. In G. Elliott, C. W. J. Granger, & A. Timmermann (Eds.), *Handbook of-economic-forecasting* (Vol. 1). North-Holland: Elsevier.

Wuthrich, B., Cho, V., Leung, S., Peramunetilleke, D., & Sankaran, K. (1998). Daily prediction of major stock indices from textual WWW data. In J. Zhang, W. Lam (Eds.) *Proceedings 4th ACM SIGKDD international conference on knowledge discovery and data mining* (pp 364–368). New York.

Xidonas, P., Ergazakis, E., Ergazakis, K., Metaxiotis, K., & Psarras, J. (2009). Evaluating corporate performance within the frame of the expert systems technology. *International Journal of Data Mining, Modelling and Management*, *1*(3), 261–290.

Yao, J., Tan, C. L., & Poh, H. (1999). Neural networks for technical analysis: A study on KLCI. *International Journal of Theoretical and Applied Finance*, *2*(2), 221–241.