International Review of Economics and Finance 89 (2024) 605–617

Contents lists available at ScienceDirect

International Review of Economics and Finance

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

Intelligent portfolio construction via news sentiment analysis

Ming-Chin Hung [a], Ping-Hung Hsia [b], Xian-Ji Kuang [c], Shih-Kuei Lin [c,*]

[a] Department of Financial Engineering and Actuarial Mathematics, Soochow University, Taipei, Taiwan
[b] Cruxover Company, LTD., New Taipei City, Taiwan
[c] Department of Money and Banking, National Chengchi University, Taipei, Taiwan

A R T I C L E I N F O

Keywords:
Portfolio theory
Sentiment analysis
Deep learning method
Natural language processing

A B S T R A C T

In this study, we apply deep learning and natural language processing methods to construct the view distribution in the Black–Litterman model. We implement this approach for portfolio allocation and perform statistical analysis to assess portfolio performance. The empirical analysis yields two main results. For the three deep learning models, we use mean square error to compare the model prediction results. The gated recurrent unit (GRU) model outperforms the other two models in the price prediction of seven stock assets. Moreover, it is more effective in capturing future trends and stock prices. The long short-term memory (LSTM) model outperforms the recurrent neural network (RNN) model. Moreover, in the comparison of the portfolio models, the Black–Litterman model, constructed by using Google's Bidirectional Encoder Representations from Transformers (BERT) to measure news sentiment and by using the GRU model to predict stock prices, yields the highest annualized return rate of 46.6%. In addition, it has the highest Sharpe and Sortino ratios of 13.0% and 17.9%, respectively, which means that under a certain degree of risk, the Black–Litterman model still outperforms other constructed portfolios.

1. Introduction

In the past few decades, numerous investors have developed methods for constructing a robust investment portfolio, but no perfect method has emerged. Markowitz (1952) proposes modern portfolio theory, which is also known as mean–variance portfolio optimization because it uses the mean and variance of assets to configure the weight of the investment. Investors can determine portfolio weights by maximizing returns under a given portfolio risk or by minimizing the risk of a given expected portfolio return. The set of optimal portfolios calculated through these procedures is called the efficient frontier. Since this method is easy to implement, it is suitable for many individual investors.

However, multiple studies address the shortcomings of mean–variance analysis. Michaud (1989) demonstrates that the error propagation problem arises when matrix multiplication is used in the theory. Best and Grauer (1991) discover that the model is sensitive to sample mean changes. Black and Litterman (1992), He and Litterman (2002), and Litterman (2004) demonstrate that the portfolio is overly focused on specific assets. Furthermore, Avramov and Zhou (2010) note that the weights are overly concentrated, resulting in an imbalanced portfolio lacking in feasibility and practical significance. Taken together, these findings indicate several challenges when this theory is used to configure portfolio assets in practice.

Almost four decades after the introduction of modern portfolio theory, Black and Litterman (1990) seek to overcome its drawbacks. On this basis, they develop the Black–Litterman model, which conducts allocation adjustment by considering investors'

* Corresponding author.
E-mail address: square@nccu.edu.tw (S.-K. Lin).

https://doi.org/10.1016/j.iref.2023.07.103
Received 12 October 2021; Received in revised form 4 May 2022; Accepted 27 July 2023
Available online 3 August 2023

views of expected returns, thereby achieving improved performance. The construction of the Black–Litterman model consists of two parts, the prior equilibrium distribution and the view distribution, which, through Bayesian estimation, can be combined into a new asset return distribution. To address the shortcomings of mean–variance portfolio optimization, the Black–Litterman model chooses an appropriate input variable as the expected return vector. Using market equilibrium points in the prior equilibrium distribution to solve this problem is considered superior to using mean–variance portfolio optimization. The Black–Litterman model is more practicable than is mean–variance analysis because it does not only focus on a few specific assets. Furthermore, as mentioned Lee (2000), the Black–Litterman model solves the estimation error problem of mean–variance optimization. For these reasons, we apply the Black–Litterman model to asset weight allocation and portfolio construction in this paper.

Nevertheless, the Black–Litterman model has its own problems. The model assumes that the view can be categorized as an absolute view or a relative view. Notably, the model does not limit the number of views that can be constructed for each asset. For example, if we invest a high number of assets, numerous views are constructed, resulting in high computational complexity. Conversely, if overly few views are constructed, some financial phenomena are unaccounted for. The view value should be manually determined by the investor through information such as the price and trading volume of the investment assets. However, this method is not only inefficient but also sometimes inaccurate because of the nature of subjective judgment and the uncertainty of financial markets. Xing et al. (2018) mathematically prove that both absolute and relative views can be expressed as absolute views. Therefore, the form of the view matrix becomes considerably simpler, and we need only generate the absolute views on each asset. Some fund managers use the expected return as a point of the asset view, but estimating the expected return in the view distribution is also relatively challenging.

Recent technological advancements and the development of the Internet allow for the complete storage of large quantities of information. Moreover, multiple deep learning models, previously hindered by technological constraints, have been developed. Numerous studies indicate that sentiment analysis is useful for estimating the absolute views on each asset and for applying deep learning models with stock sentiments to the Black–Litterman model. Sentiment analysis, which refers to using natural language processing (NLP), text mining, and computer linguistics to identify and extract subjective information from the source material, can facilitate the understanding of textually conveyed emotional states.

Sentiment analysis is applied in various fields to provide feature-related information. Ding et al. (2014) perform sentiment analysis on textual information to enhance the performance of stock price prediction. Sentiment analysis for stock predictions is applied to data from two main sources. Previous studies have mainly used news information as the basis for sentiment analysis. However, many recent studies have opted to use social media messages instead because of their immediacy. Of course, such messages also have their drawbacks, such as many of them being based on the subjective opinions of investors, being less credible, and being more difficult to preprocess.

Sentiment analysis approaches can be classified into various types. Knowledge-based methods are highly reliant on the expertise of linguists, who manually define the meaning of words and reorganize the relationships between them. Similar to how one constructs a dictionary, they build a thesaurus and place synonyms or related words in the same group. Next, interword relationships, such as hypernym relationships, are defined, and the relevance of these linkages is determined. Relationships between words can be visualized through knowledge graphs. Statistical techniques for detecting sentiment toward the target task include the bag-of-words model and the Latent Dirichlet Allocation (Blei et al., 2003).

With the increasing popularity of neural networks, numerous models are dedicated to solving problems involving sequential data, including data in stock price analysis and NLP. Recurrent neural network (RNN)-based models are representative in this context; they include long short-term memory (LSTM) models and gated recurrent unit (GRU). Wang et al. (2018) convert words collected from social media platforms into word embeddings that are then fed into an LSTM model for sentiment classification. His empirical results indicate that the LSTM model outperforms the vanilla RNN model. Ding et al. (2014) consider the influence of news events on trends of stock price movement. They use the open information extraction task to extract structured events from the news. Chang and Lin (2011) use a support vector machine as a linear model and a deep neural network model as a nonlinear model to determine whether news events have close relationships with stock movements. The experiments reveal that the news event-based representation outperforms the bag-of-words ones. Moreover, the deep neural model captures stock movement-related information better than do the other methods.

Word embedding methods such as Word2vec (Mikolov et al., 2013) can be employed to encode text data for integration. Other tools, such as Doc2vec (Le & Mikolov, 2014), directly encode the entire paragraph or document into a low-dimensional vector. Akita et al. (2016) utilize this method to convert news articles into vector representations. They measure stock price movements with the LSTM model. The distributed representation outperforms the numerical data-based and bag-of-words approaches. Furthermore, the researchers demonstrate that the LSTM model is effective in capturing information from time series data. Before the development of the Embeddings from Language Models (ELMo; Peters et al., 2018), deep learning models use word embeddings as their representation. Previous models using word embedding contain no contextual information. Contextual word representation is essential because it allows word tokens of the same word type to have different representations in different contexts. ELMo and the Google Bidirectional Encoder Representations from Transformers (BERT) model (Devlin et al., 2018) utilize the concept of contextualized word embedding to enhance performance. BERT has been employed in sentiment analysis in multiple studies.

The efficient market hypothesis (Malkiel & Fama, 1970) and the random walk hypothesis (Fama, 1995) state that current stock prices are completely reflected in past information. However, scholars have held differing opinions. As computational intelligence becomes more sophisticated, various types of algorithms have been employed to improve performance in numerous studies. Deep learning methods are used to achieve better performance in fields such as NLP and computer vision. RNN-based networks exhibit excellent performance in the NLP domain. As sequential data, stock prices possess similar characteristics to NLP data to some degree. Therefore, studies have begun employing RNN-based networks in predicting stock prices.

Fundamental information on assets can be used as model features. Moreover, news and social media messages exert strong influences on stock price movements. Ding et al. (2014) demonstrate that asset price movement is influenced by stock news. They also verify that the news information on the market is essential for predicting stock changes. Xing et al. (2018) take market sentiment, prices, and trading volume as the input features of the evolving clustering method and an LSTM network, which is used to construct the view toward assets in the Black–Litterman model. Their empirical results verify the supposition that placing market sentiment features in the model is a favorable choice. Therefore, we aver that sentiments from natural language provide more information for predicting stock trends. However, Xing et al. (2018)'s approach is so complex that it is not easy to implement. In one study, market sentiment is measured using SenticNet and user labels (Cambria et al., 2010). Although this knowledge-based approach performs well empirically, the model is highly reliant on linguists' expertise. Moreover, frequent updates of new language knowledge are challenging. Thus, our goal is to construct a feasible and straightforward model for individual investors.

Herein, we use the BERT model (Devlin et al., 2018), which has state-of-the-art performance on 11 NLP tasks, to measure news sentiment. The model is pretrained with the BookCorpus data set as well as English-language data from Wikipedia; therefore, we need only fine-tune our downstream task. This also means that the model is easy to implement. Rather than employ conventional word embedding, which applies only to a certain dimension, BERT uses the concept of contextualized word embedding; that is, it considers the contextual relationships between words. The model architecture uses the encoder in Transformer (Vaswani et al., 2017), which contains a self-attention layer, to enable the model to pay attention to other elements in its sequence and obtain contextual information therefrom.

After sentiment analysis with BERT, we employ the results as features for deep learning models. We use three deep learning methods, namely vanilla RNN (Rumelhart et al., 1985), LSTM (Hochreiter & Schmidhuber, 1997), and GRU (Cho et al., 2014), for predicting asset returns. These models all have the same layers and hyperparameters available from the Python package. In general, GRU outperforms the others in terms of mean square error. The Black–Litterman model constructed using the view vector of GRU has the best performance with regard to annualized return, Sharpe ratio, and Sortino ratio.

The remainder of the paper is organized as follows. In Section 2, we introduce our main financial portfolio model and discuss how we measure the stock sentiments and view vector. The empirical results are presented in Section 3. Section 4 concludes the paper.

2. Methodology

In this section, in addition to discussing the architecture of the two most popular models in modern portfolio theory, we lay out the problems mentioned in related works and detail the methods we use to resolve them.

2.1. Modern portfolio theory-based model

We selected and weighted n assets and assign weights to these assets. The returns of the mean–variance optimization on each asset can be measured by the weighted mean of the expected returns on each asset, and the portfolio risk is determined through the variance of the return vector. As mentioned, the portfolio points can be determined by maximizing the return under a given portfolio risk or by minimizing the risk of a given portfolio return. Therefore, we obtain the efficient frontier with the following condition:

$$\begin{aligned} & min\ \lambda \sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j \sigma_{i,j} - \sum_{i=1}^{n} w_i \mu_i \\ & \quad s.t.\ w_i \geq 0, \sum_{i=1}^{n} w_i = 1, \end{aligned} \tag{1}$$

where μ_i is the expected return on assets i, σ_{ij} is the covariance between assets i and j, and λ is a risk aversion coefficient. Through the first-order condition, the optimal asset allocation weights of the Markowitz portfolio can be obtained as follows:

$$w^* = (2\lambda\Sigma_n)^{-1}\mu, \tag{2}$$

where μ is an $n \times 1$ vector consisting of expected returns and Σ_n is a $n \times n$ covariance matrix of asset returns. With the efficient portfolio weights w^*, we can construct efficient portfolio points at the given risk aversion coefficient λ. When we connect all the efficient portfolio points, we obtain the boundary curve called the efficient frontier. No points below the efficiency frontier curve are better than those on the efficient frontier.

When we estimate the optimal allocation w^*, the traditional estimators of μ and Σ_n are the sample mean $\hat{\mu}$ and the sample covariance matrix $\hat{\Sigma}_n$, respectively. Historical data from the past T days are used to obtain the estimators as follows:

$$\hat{\mu} = \frac{1}{T}\sum_{t=1}^{T} \log \frac{p_t}{p_{t-1}} \tag{3}$$

$$\hat{\Sigma}_n = \frac{1}{T}\sum_{t=1}^{T} (\log \frac{p_t}{p_{t-1}} - \hat{\mu})(\log \frac{p_t}{p_{t-1}} - \hat{\mu})', \tag{4}$$

where p_t is the price vector of length n for time t and $\hat{\mu}$ is usually called the log return. However, Idzorek (2007) empirically demonstrates that the historical return vector has a high variation, resulting in an extreme portfolio. Avramov and Zhou (2010) state that this leads to an imbalanced portfolio that is difficult to implement in practice.

2.2. Black–Litterman model

With the purpose of solving the problem of asset weight concentration on certain assets, as well as that of input sensitivity, the equilibrium risk premium vector $\boldsymbol{\Pi}$ and the view vector of the investor assets $\boldsymbol{Q}$ are introduced.

(1) **Prior equilibrium distribution:** First, we construct the prior equilibrium distribution, one of the distributions in the Black–Litterman model. This involves estimating the equilibrium risk premium vector $\boldsymbol{\Pi}$. Black and Litterman (1990) state that using the expected return in the market equilibrium points as the neutral starting point is favorable. The equilibrium risk premium is typically calculated by using the capital asset pricing model:

$$\Pi = \beta_m(E(r_m) - r_f), \tag{5}$$

where $\boldsymbol{\Pi}$ is the $n \times 1$ equilibrium risk premium vector, $E(r_m)$ is the expected market return, r_f is the risk-free rate, and β_{im} is a measure of volatility relative to the entire market,

$$\beta_{i,m} = \frac{Cov(r_i, r_m)}{Var(r_m)}. \tag{6}$$

Because measuring β_{im} is challenging, we turn to the reverse optimization method to estimate the equilibrium risk premium vector as follows:

$$\boldsymbol{\Pi} = \lambda \Sigma w_{mkt} \tag{7}$$

where λ is the risk aversion coefficient, $\boldsymbol{\Sigma}$ is the covariance matrix of assets, and w_{mkt} is the market capitalization weight of the assets.

The Black–Litterman model assumes that the equilibrium returns r_{eq} are normally distributed $N(\mathbf{\Pi}, \tau\boldsymbol{\Sigma})$ with the mean being that of the equilibrium risk premium vector $\mathbf{\Pi}$, and the variance of the distribution is $\tau\boldsymbol{\Sigma}$, where $\boldsymbol{\Sigma}$ is the covariance matrix (the same as the matrix in the mean–variance portfolio) and τ is a scalar used to measure the confidence toward the estimation of the equilibrium risk premium vector $\boldsymbol{\Pi}$. With reference to the study by Idzorek (2007), we set this scalar τ to 0.025.

(2) **View distribution:** Second, we build the view distribution, another distribution in the Black–Litterman model. The most integral part of model construction is the views of the assets from the investor. Black and Litterman (1990) define two different types of views: relative and absolute. Relative view compares the future performance of two assets. This is a subjective decision by the fund manager that one asset will outperform another asset at some point. By contrast, the absolute view only considers a target asset, comparing it with the market value. We interpret the two definitions of view distributions as follows.

(2 − 1) **Definition 1:** The model uses two matrices $\boldsymbol{P}$, $\boldsymbol{\Omega}$, and one vector $\boldsymbol{Q}$ to represent a distribution with k views. It does not require investors to specify their views on all assets. Assuming that our portfolio contains n assets, we can select a set of k views. First, vector $\boldsymbol{Q}$ represents the expected returns on each view. This is the most difficult part of this process. We introduce the construction of the expected return vector $\boldsymbol{Q}$ in the next section.

Second, the uncertainty of the views is a random, unknown, independent, normally-distributed error term vector $\boldsymbol{\varepsilon}$. However, the variance in the views is not presented as $\boldsymbol{\varepsilon}$ but rather as a matrix $\mathbf{\Omega}$. This confidence matrix measures the covariance between views. Moreover, it is a diagonal matrix because the Black–Litterman model assumes that the views are independent of each other. The diagonal element is measured by the variances of the error terms ω. The variances represent the uncertainty of the views. By calculating the variance of the individual error terms, we form matrix $\boldsymbol{\Omega}$:

$$\begin{pmatrix} \omega_1 & 0 & \dots & 0 \\ 0 & \omega_2 & \dots & \dots \\ \dots & \dots & \dots & 0 \\ 0 & \dots & 0 & \omega_k \end{pmatrix}_{(k,k)}. \tag{8}$$

Therefore, we can estimate the error term vector $\boldsymbol{\varepsilon}$, which has a mean of 0, and the covariance matrix $\boldsymbol{\Omega}$:

$$\varepsilon \sim N(0, \Omega), \tag{9}$$

The view of the investor can be expressed by combining vector $\boldsymbol{Q}$ and the uncertainty of the views—that is, as $\boldsymbol{Q} + \boldsymbol{\varepsilon}$. The view vector, denoted as r_{views}, is also normally distributed:

$$r_{views} \sim N(Q, \Omega). \tag{10}$$

Finally, matrix $\boldsymbol{P}$ indicates the assets mentioned in the views. The expected return vector $\boldsymbol{Q}$ is matched to matrix $\boldsymbol{P}$. For different views, we can distinguish matrix $\boldsymbol{P}$ into relative and absolute views. The row summation of the relative view is 0, whereas that of the absolute view is 1.

Matrix $\boldsymbol{P}$ can be defined through several routes. Some studies use an equal weight method, wherein the weight of each asset is proportional to the inverse of the number of outperforming or underperforming assets. The assets that are not considered are set to 0. This method leads to error estimation because of the imbalance in market capitalization. Therefore, some studies consider the market capitalization factors instead of simply dividing them equally. Idzorek (2007) employs the market capitalization weight method, wherein the weight is replaced through the division of the total market capitalization of either the outperforming or underperforming assets of that particular view.

$(2-2)$ **Definition 2:** As is the case in definition 1, the market views on n assets are represented by the matrices $\boldsymbol{P}$, $\boldsymbol{Q}$, and $\boldsymbol{\Omega}$. Xing et al. (2018) derive the two most crucial properties of view matrices. They verify that any set of independent views is compatible with another and that all relative or absolute views can be expressed with a nonsingular absolute view matrix. Notably, definition 2 can be mathematically proven from definition 1. This method reduces the computational complexity because the view matrix $\boldsymbol{P}$ is within an $n \times n$ identity matrix. Thus, we need not be concerned about the tracking error caused by neglecting market capitalization. As is the case in definition 1, the expected return vector $\boldsymbol{Q}$ is matched with matrix $\boldsymbol{P}$ with each view in a $1 \times n$ row vector and $\boldsymbol{\Omega}$ is an $n \times n$ nonnegative diagonal matrix. Therefore, the view distribution in definition 2 is also normal. Through the Bayesian approach, the new combined return is also a normal distribution $N(\mu_{BL}, \Sigma_{BL})$. According to the steps outlined in Satchell and Scowcroft (2007), the mean and the variance of the Black–Litterman model are:

$$\mu_{BL} = [(\tau\Sigma)^{-1} + P'\Omega^{-1}P]^{-1}[(\tau\Sigma)^{-1}\Pi + P'\Omega^{-1}Q] \tag{11}$$

$$\Sigma_{BL} = \Sigma + [(\tau\Sigma)^{-1} + P'\Omega^{-1}P]^{-1}. \tag{12}$$

$(2-3)$ **Summary of definitions:** In definition 1, we determine how Black and Litterman (1990) originally represent the view through matrices $\boldsymbol{P}$, $\boldsymbol{Q}$, and $\boldsymbol{\Omega}$. However, matrix $\boldsymbol{P}$ easily results in a tracking error because market capitalization is not considered. Thus, we employ the definition derived from Xing et al. (2018) to obtain the identity matrix $\boldsymbol{P}$, which reduces the computational complexity. In sum, the view of our method is represented by matrices $\boldsymbol{P}_{n,n}$, $\boldsymbol{Q}_{n,1}$, and $\boldsymbol{\Omega}_{n,n}$, where $\boldsymbol{P}_{n,n}$ is an identity matrix; $\boldsymbol{Q}_{n,1} \in \mathbb{R}^n$.

(3) **Estimating parameters:** After measuring the view distribution, we need to estimate the parameters for model import. First, the expected portfolio returns in the posterior distribution of the Black–Litterman model are based on three different estimators: the equilibrium volatility as a covariance matrix $\boldsymbol{\Sigma}$, the investor's confidence in their own views $\boldsymbol{\Omega}$, and the investor's expected returns in their view $\boldsymbol{Q}$.

$(3-1)$ **Estimating volatility:** All elements in the covariance matrix $\boldsymbol{\Sigma}$ are formed by the σ_{ij}, where i and j represent the i_{th} and j_{th} assets, respectively. Volatility is estimated as follows:

$$\hat{\sigma}_{i,j} = \frac{1}{k}\sum_{n=1}^{k} r_{i,n} \cdot r_{j,n} - \frac{1}{k^2}\sum_{n=1}^{k} r_{i,n}\sum_{n=1}^{k} r_{j,n} \tag{13}$$

$$r_{i,n} = \log\frac{p_{i,n}}{p_{i,n-1}}. \tag{14}$$

$(3-2)$ **Estimating confidence:** Regarding investor confidence, the method employed by the original Black–Litterman model to measure confidence involves manual setting on the basis of the investor's experience. He and Litterman (2002) present a numerical method for measuring the covariance matrix, in which the confidence matrix can be derived from the covariance matrix:

$$\hat{\Omega} = diag(P(\tau\hat{\Sigma})P'). \tag{15}$$

Because the Black–Litterman model assumes that views are independent of each other, the off-diagonal elements of the confidence matrix are 0. As mentioned, definition 2 states that matrix $\boldsymbol{P}$ is an identity matrix. Therefore, the computational complexity is greatly reduced, facilitating the understanding of the matrix multiplication of $\boldsymbol{P}(\tau\hat{\Sigma})\boldsymbol{P}'$. This can be conceived as the covariance matrix of the expected returns on the views.

The variance of the view with respect to asset $\boldsymbol{i}$ is related to the volatility of asset $\boldsymbol{i}$. As long as the asset is excessively risky, investor confidence is absent regardless of how the view is formed. The past price of the asset's volatility is often used to estimate the $\boldsymbol{\Omega}$.

$(3-3)$ **Estimating returns:** Finally, we must estimate the expected returns on the assets as the inputs of the Black–Litterman model. As mentioned, Xing et al. (2018) assume that there must be a strategy that can profit on the basis of market sentiment. Such a strategy would require the summary of a large quantity of textual data on a website for estimation. Moreover, the quality of the textual data would greatly affect the views of assets. We assume that the Black–Litterman model uses past stock price series $p_{t,k}$ and trading volumes $v_{t,k}$ to form the expected returns of different views. However, we also consider news sentiment s_t as a critical input for constructing the Black–Litterman model. The method for estimating expected returns will be introduced in the following section.

Finally, by summarizing the above, the portfolio weight w_t^* at each time point of the Black–Litterman model can be obtained by Eq. (2). At the same time, $\mu_{BL,t}$ and $\Sigma_{BL,t}$ can be substituted into Eqs. (12) and (13), respectively. Therefore, the w_t^* takes the following form:

$$\begin{aligned} w_t^* &= [\lambda(\Sigma_t + [(\tau\Sigma_t)^{-1} + P']\Omega_t^{-1}P)]^{-1}]^{-1} \\ &[(\tau\Sigma_t)^{-1} + P'\Omega_t^{-1}P]^{-1}[(\tau\Sigma_t)^{-1}\Pi + P'\Omega_t^{-1}Q]. \end{aligned} \tag{16}$$

After estimating the volatility and confidence by using Eqs. (14) and (16), we can substitute Σ_t, $\Omega_{t,}$ and vector $\boldsymbol{Q}$ to obtain the optimal weights of the Black–Litterman model:

$$\begin{aligned} \hat{w}_t^* &= [\lambda(\hat{\Sigma}_t + [(\tau\hat{\Sigma}_t)^{-1} + P']\hat{\Omega}_t^{-1}P)]^{-1}]^{-1} \\ &[(\tau\hat{\Sigma}_t)^{-1} + P'\hat{\Omega}_t^{-1}P]^{-1}[(\tau\hat{\Sigma}_t)^{-1}\Pi + P'\hat{\Omega}_t^{-1}\hat{Q}]. \end{aligned} \tag{17}$$

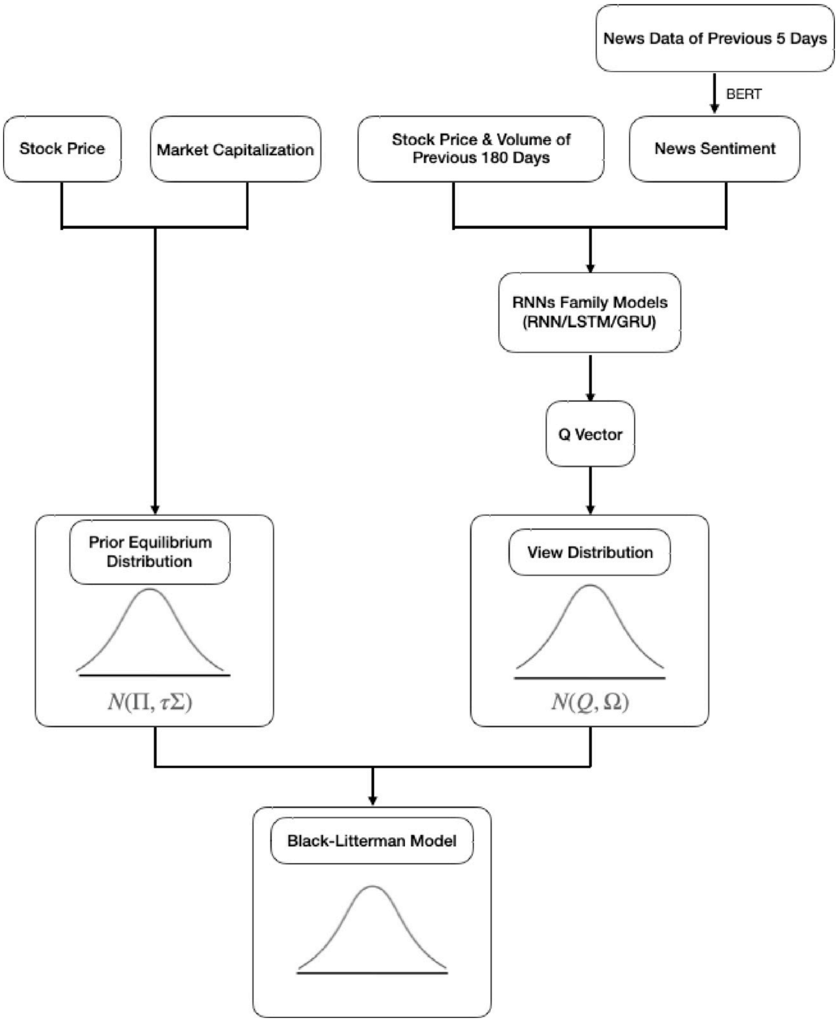

Fig. 1. Flow chart of construction of the Black–Litterman model.

3. Empirical analysis

We used experimental results to construct a deep learning sequence model that considers news sentiment factors. We mainly discuss how we measure the sentiment and vector $\boldsymbol{Q}$ in the view distribution. Subsequently, we can construct the Black–Litterman model. Fig. 1 displays the flow chart of model construction.

3.1. Data sources

In this study, Thomson Reuters and CNBC news on stocks were used as data. Specifically, we obtain the news data through web crawlers. Our data set covered seven companies listed on the S&P 500, namely Apple, Amazon, Johnson & Johnson, JP Morgan, Microsoft, Visa, and Google. The purpose of asset selection is to ensure a sufficient amount of news for model training. The news search period is from January 7, 2015, to November 20, 2019. In total, 8251 news articles are returned for Apple, 6698 for Amazon, 4882 for Johnson & Johnson, 10,618 for JP Morgan, 8021 for Microsoft, 8745 for Visa, and 8928 for Google.

Data on stock prices, trading volume, and market capitalization are obtained from the *pandas-datareader* package in Python; it retrieves relevant stock information by being synced with Yahoo! Finance. Specifically, this information encompasses the opening price, highest price, lowest price, and closing price of the stock. The time interval of selection is the same as that applied for the stock news.

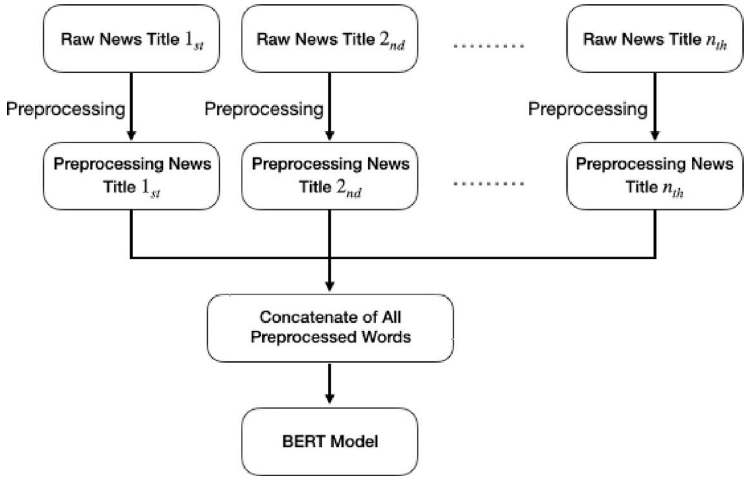

Fig. 2. Preprocessing of the stock news data.

3.2. Data preprocessing

When constructing the Black–Litterman model, we must build vector $\boldsymbol{Q}$ according to the investor's view, where each element in the vector corresponds to the stock returns. To obtain the returns, we must predict the stock prices using deep learning models. We select deep learning-based sequence models, namely the vanilla RNN, LSTM, and GRU models. Model inputs contain news sentiment, stock prices, and trading volume as features.

Data preprocessing is divided into two parts. First, the stock news data are preprocessed such that the textual information can be used as the input of the BERT model to predict the rise or fall of the stock every trading day. Fig. 2 presents the procedure for text preprocessing. Next, we reconstruct the stock price and trading volume data to predict the stock prices and thereby calculate the returns.

3.2.1. Stock news

From every stock news article, we remove the stop words and punctuation. The stop words include keywords related to Reuters and CNBC, such as "CNBC news update"and "Thomson Reuters". Because the stock news is collected from two financial news platforms, overlaps (of news from the same or similar sources) are present. Therefore, we calculate the similarity between each news article using the *SequenceMatcher* package in Python. When the similarity index between sentences in two articles is higher than 95%, one of the articles is deleted and excluded from further analysis.

The data set is divided into the training set and the testing set. The time interval of the training set is from January 7, 2015, to December 31, 2018, and that of the testing set is from January 2 to November 20, 2019. The training and testing sets contain approximately 80% and 20% of the data, respectively. On a daily basis, we collect news about each asset published over the past 5 days. Then, we concatenate the news and use it as a feature to measure news sentiment.

3.2.2. Stock prices and trading volume

We must also predict the adjusted closing price of the following day on a daily basis. For the features, we employ stock price and trading volume data from 180 days before the trading day. The time intervals of the training and testing sets are same as those applied in news sentiment measurement. Moreover, the price and trading volume of each asset passes through the MinMaxScaler function in the *scikit-learn* package in Python.

$$x_{minmax} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{18}$$

The scaling algorithm translates and scales each feature to be between 0 and 1. The main purpose of using the MinMaxScaler function is to reduce the unit-induced difference between different features, which compromises the predictive ability of the model.

3.3. News sentiment

BERT, developed by Google (Devlin et al., 2018), is a state-of-the-art model that yields the best performance on 11 NLP tasks. This model is effective in obtaining more favorable results in various fields. In numerous studies on sentiment analysis, an NLP task, BERT performs incredibly well. Therefore, we apply BERT to our model to measure new sentiment.

We categorize news sentiment through binary classification. When the category is one, it means that we are optimistic about this asset; when the category is zero, we are not optimistic about it. Because we do not have the true value of news sentiment, we

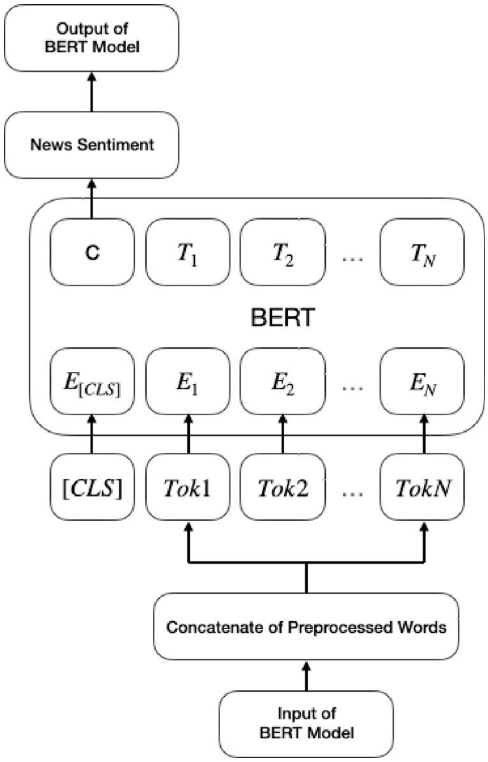

Fig. 3. Sentiment analysis, performed using the BERT model.

Table 1
BERT classification accuracy.

Dataset	Accuracy
Training set on 12th epochs	0.584
Testing set	0.568

use the rise and fall of stock prices as the ground truth (Birbeck & Cliff, 2018). Fig. 3 demonstrates the parameter inputs and news sentiment score outputs of the BERT model.

Specifically, the preprocessed stock news data are the inputs of the BERT model. The subword segmentation algorithm WordPiece tokenizes the input sentences (Wu et al., 2016). BERT limits the length of its input tokens to no more than 512 tokens. Therefore, when the number of tokens exceeds 512, the extra parts are discarded.

We use the PyTorch machine learning framework to build the BERT model. The hyper-parameters of the BERT model are as follows: epochs 12, batch size 16, and a stochastic gradient descent optimizer with a learning rate of 0.002. The accuracy of the training and testing sets is displayed in Table 1. Using the BERT model to predict the future trend of individual stocks from news articles yields an average accuracy of 58% in the training set and 56% in the testing set. In other words, our model is effective in predicting whether a company's stock price will rise or fall the following day more than half of the time. Given that the accuracy of the testing set is very close to that of the training set, overfitting is not a concern.

3.4. Stock price prediction

To construct vector $\boldsymbol{Q}$, we must obtain the return of each investment asset. Rather than directly predict the asset returns using the model, we predict the stock prices before calculating the returns and then choosing the log returns instead of simple returns. The calculated returns are used as an element of the vector $\boldsymbol{Q}$.

$$return_t = log(p_t) - log(p_{t-1}) = log(\frac{p_t}{p_{t-1}}) \tag{19}$$

We use the vanilla RNN, LSTM, and GRU models to predict vector $\boldsymbol{Q}$. The vanilla RNN model leverages the concept of memory, storing information about past points in time (Rumelhart et al., 1985). However, the model often encounters the vanishing gradient problem and the exploding gradient problem. Therefore, managing long-term dependency is challenging. The LSTM model uses a

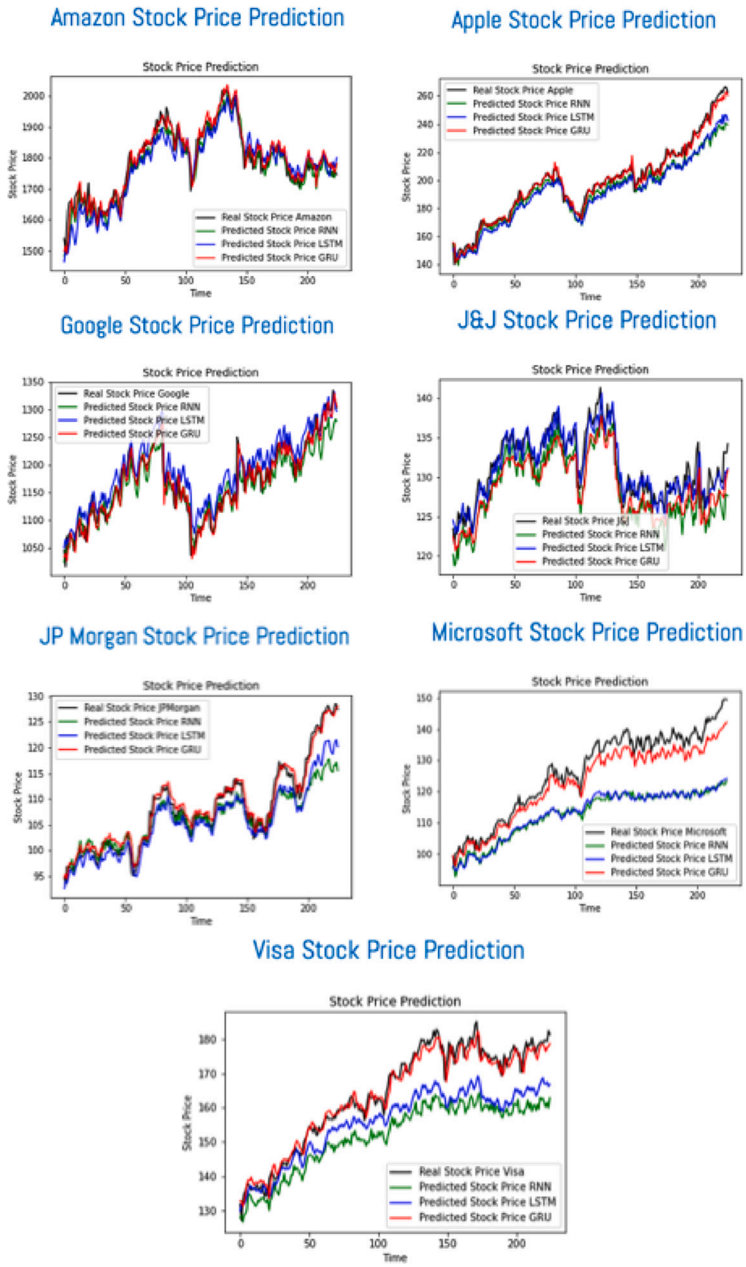

Fig. 4. Price prediction of seven assets.

gating mechanism to resolve this problem (Hochreiter & Schmidhuber, 1997). Furthermore, the GRU model improves on this gating mechanism (Cho et al., 2014).

The input features and the hyperparameters of the model are exactly the same. For each time point, the input features are the opening, highest, lowest, and closing prices; the trading volumes of the previous 180 days after scaling; and the news sentiments from the BERT model. MinMaxScaler is employed as the scaling function. We set our deep learning model as comprising three multilayers and 50 units. In the final step, we add a dense layer to predict stock prices. To avoid overfitting, we add a 20% dropout layer. The hyperparameters are epochs = 64, batch size = 16, and learning rate = 0.001 for an Adam optimizer. Fig. 4 displays the prediction plot of the seven assets with real stock price trends and three stock price prediction models.

In Fig. 4, the black solid line represents the real stock price trends, and the other three lines represent the predicted stock price trends of the three models. The green line represents the vanilla RNN model, the blue line represents the LSTM model, and the red line represents the GRU model. We observe that in the stock price forecasts for Amazon, Google, and Johnson & Johnson, the prediction results of our three models are very close to the real stock price trends, and differentiating the quality of these three

Table 2
Mean square errors obtained in asset predictions by the vanilla RNN, LSTM, and GRU models.

MSE	Apple	Amazon	Google	Microsoft	J&J	JP Morgan	Visa
RNN prediction	122.25	2326.19	657.69	175.31	19.63	20.72	393.95
LSTM prediction	93.40	1146.13	506.36	132.39	3.37	22.80	100.16
GRU prediction	30.35	1003.97	507.37	81.92	2.19	11.78	21.89

models is difficult. GRU clearly outperforms the other two models in predicting the stock prices of Apple, JP Morgan, Microsoft, and Visa, and the predictions are close to the real stock price trends. These results are not unexpected; some studies have verified that the GRU model has a favorable predictive ability when applied to time series data. Table 2 presents the mean square errors of these three models in the stock price predictions. Except for the predictions for Google, the predictive ability of the GRU model surpasses those of the other two models. Moreover, the LSTM model outperforms the vanilla RNN model in the predictions.

3.5. Black–Litterman model performance

We have now executed the most challenging part of implementing the Black–Litterman model upon obtaining the predictions of news sentiment and future stock prices—using BERT and various sequence methods, respectively. We can use the vector $\boldsymbol{Q}$ generated through model predictions to construct the view distribution and combine each of them with the prior equilibrium distribution. Subsequently, we can obtain the weights of the Black–Litterman model constructed under different models and use those weights to allocate assets in each period.

3.5.1. Benchmark models

Three approaches are selected as our benchmark model as follows.

(1) **S&P 500:** This stock market index tracks 500 large companies listed on various US stock exchanges. Investors often use the S&P 500 as a benchmark, expecting their investments to outperform it. In this study, we buy the S&P 500 and hold it until the final trading session.

(2) **Equally weighted portfolio:** This is a relatively simple asset allocation method. Unlike other investment strategies wherein asset allocation must be considered under varying conditions, this method allocates weights equally to each asset. Because we invest in seven stock assets, each asset is assigned a weight of 1/7. In our case, the equally weighted portfolio performs substantially better than do other sophisticated portfolio models, including the mean–variance portfolio.

(3) **Markowitz portfolio:** In the preceding section, we describe the disadvantages of constructing a mean–variance portfolio. In our experiment, the weights configured by the mean–variance portfolio, as verified by other studies, are concentrated on specific assets. Moreover, the weight allocation is sensitive to the portfolio mean. We set the value of the risk parameter λ to be one.

3.5.2. Black–Litterman model-based portfolios

We use the vanilla RNN, LSTM, and GRU models to generate vector $\boldsymbol{Q}$ and construct Black–Litterman model–based portfolios. The vector is then fed into the view distribution of the Black–Litterman model; Black–Litterman models based on three different architectures are constructed. As with the Markowitz portfolio, we set the value of the risk parameter λ to be one.

3.5.3. Evaluation

In our experiment, we assume that the agent is holding $1 at the beginning of the trading period. Every trading day, the investors will reallocate all their assets. The transaction duration of the testing set is 224 days.

Fig. 5 presents the performance trend chart under different investment portfolios. We use the Black–Litterman model to predict stock prices. Specifically, the blue, orange, and green lines represent the GRU, LSTM, and vanilla RNN models, respectively. Furthermore, we compare three benchmark models, where the red line denotes the S&P 500 returns, the purple line indicates the mean–variance model, and the brown line denotes the equal-weight portfolio investment strategy. The Black–Litterman model performs better with the GRU or LSTM models than do other benchmark models, and even with the GRU model, the returns are almost 40%. However, we cannot evaluate the portfolios' performance solely by their total value. Therefore, we consider five financial metrics.

(1) **Annualized return:** This metric measures how much an investment has increased on average during a specific time period each year. We set $T = 250$, the approximate average number of stock exchange opening days in a year. Variable t represents the number of days we hold the assets—that is, 224.

$$AR = (log_{p_t} - log_{p_{t-1}})^{\frac{T}{t}} = log(\frac{p_t}{p_{t-1}})^{\frac{T}{t}} \tag{20}$$

(2) **Sharpe ratio:** Proposed by Sharpe (1966), this ratio considers not only asset returns but also the variation in assets held. It set the market's risk-free interest rate as the benchmark, where R_{rf} represents the risk-free rate.

$$Sh.R = \frac{R_{portfolio} - R_{rf}}{\sigma_{portfolio}} \tag{21}$$

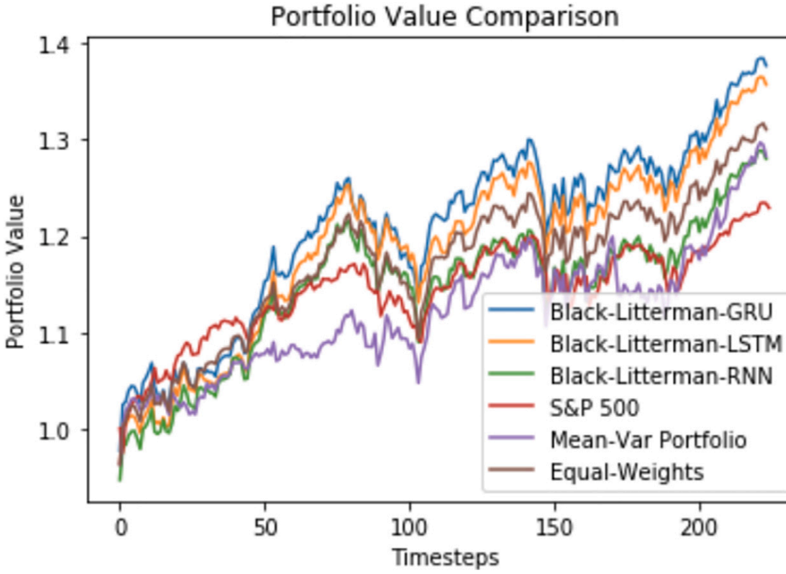

Fig. 5. Portfolio value comparison.

(3) **Sortino ratio:** Proposed by Sortino and Price (1994), this metric is similar to the Sharpe ratio to some extent. However, rather than the standard deviation, it uses the downside deviation. The implicit condition is that the rise of the investment portfolio meets the needs of investors and should not be included in the risk adjustment, where $\sigma_{downside}$ represents the downside deviation.

$$Sr.R = \frac{R_{portfolio} - R_{rf}}{\sigma_{downward}} \tag{22}$$

(4) **Maximum drawdown (MDD):** Drawdown is the loss of the strategy's gradual profit, the distance from a peak to a trough on the profit and loss curve. Maximum drawdown is the largest observed continuous loss from a peak to a trough in a portfolio.

$$MDD = \max_{0<t<\tau} \left\{ \frac{Value_t - Value_\tau}{Value_t} \right\} \tag{23}$$

(5) **Turnover :** This metric is often used to measure the transaction fees of a portfolio strategy. A higher value means a higher transaction fee is required. According to DeMiguel et al. (2009), we define the portfolio turnover as follows:

$$Turnover = \frac{1}{t} \sum_{i=1}^{t} \sum_{j=1}^{N} \left(\left| \hat{w}_{j,i+1} - \frac{\hat{w}_{j,i}(1 + return_{j,i+1})}{\sum_k \hat{w}_{k,i}(1 + return_{k,i+1})} \right| \right) \tag{24}$$

where N is the available assets, $\hat{w}_{j,i}$ is the portfolio weight in asset j at time i and $return_{j,i+1}$ is the log return of asset j from time point i to $i+1$.

As presented in Table 3, implementing the Black–Litterman model with the GRU or LSTM models leads to better performance than implementing the benchmark models, regardless of the annualized return, the Sharpe ratio, or the Sortino ratio. The Black–Litterman model with the GRU model has the highest Sharpe and Sortino ratios of 13.0% and 17.9%, respectively. The MDD indicates that our strategy yields results that are not inferior to those of the benchmark models, which means that from the perspective of risk, sudden huge losses will not be incurred under our strategy. Given that the Sharpe and Sortino ratios measure the performance of returns under relative risk, we have high confidence in the proposed strategy and believe that stable returns can be obtained with it. Nevertheless, the portfolio turnover reflects the more frequently changing asset behavior of our strategy. The Black–Litterman model with RNN has the highest turnover rate, implying that RNN is not a very practical model due to the problem of high transaction costs generated by a high turnover rate. Compared to the Markowitz portfolio, our strategies have a higher turnover rate due to news information, which allows investors to have more information than just price to adjust their holdings. Not surprisingly, the equal-weighted approach has a lower portfolio turnover, and combined with other metrics, it remains a trustworthy investment approach, as DeMiguel et al. (2009) concluded. Therefore, in the future, it is worthwhile to investigate how more information, such as financial fundamentals and market-wide news analysis,[1] can be used when constructing the views to reduce the turnover rate and make the model more feasible.

[1] We thank an anonymous referee for giving this opinion.

Table 3
Portfolio performance according to four metrics.

Portfolio models	AR(%)	Sh.R(%)	Sr.R(%)	MDD(%)	Turnover(%)
Black–Litterman (GRU)	46.6262	13.02	17.96	4.45	68.35
Black–Litterman (LSTM)	46.5229	12.67	16.99	4.50	63.89
Black–Litterman (RNN)	40.0389	9.97	13.07	4.42	84.08
S&P 500	29.4953	10.93	14.48	3.51	0
Equal-Weights	41.1657	11.72	15.65	4.18	0.63
Markowitz-Portfolio	37.6600	9.93	13.07	5.09	25.07

4. Conclusion

We aim to build an approach for individual investors through the Black–Litterman model. However, the construction of the view distribution of the Black–Litterman model has long been challenging. On account of price predictions, the construction of vector $\boldsymbol{Q}$ is difficult. Moreover, measuring news sentiment, which affects stock trends, is also challenging. In this paper, we implement a novel and simple approach to solve these problems. Our first contribution to the literature is that we utilize the state-of-the-art BERT model to predict the ups and downs of stock trends as sentiments from stock-related news articles. Regarding their prediction performance, the training and testing sets had average accuracies of 58% and 56%, respectively. We find that using the BERT model as a prediction tool prevents the problem of overfitting on the training set. On this basis, we suggest that stock news can be used as an indicator of investor sentiment to assist in predicting and evaluating companies' future trends. Our second contribution is the integration of three RNN family models with news sentiment in predicting asset prices. The GRU model, combined with news sentiment, is highly suitable for predicting stock prices and has superior predictive ability in terms of the mean square error results. In addition, we employ the Black–Litterman with the GRU model to construct our portfolio. Its performance surpasses that of the three baseline models with regard to four key financial metrics: annualized return, Sharpe ratio, Sortino ratio, and MDD. The MDD results demonstrate our strategy is stable and that no sudden huge losses will be incurred under its use. Finally, we want to research the construction of views by adding more information in the future.

CRediT authorship contribution statement

Ming-Chin Hung: Formal analysis, Supervision, Project administration. **Ping-Hung Hsia:** Software, Data curation, Writing – original draft. **Xian-Ji Kuang:** Methodology, Validation, Writing – review & editing. **Shih-Kuei Lin:** Conceptualization, Investigation, Visualization, Funding acquisition.

Acknowledgments

We gratefully acknowledge research support from the Ministry of Science and Technology of R.O.C. Taiwan (108-2745-8-004-003-).

References

Akita, R., Yoshihara, A., Matsubara, T., & Uehara, K. (2016). Deep learning for stock prediction using numerical and textual information. In *2016 IEEE/ACIS 15th international conference on computer and information science* (pp. 1–6). IEEE.
Avramov, D., & Zhou, G. (2010). Bayesian portfolio analysis. *Annual Review Financial Economics*, *2*(1), 25–47.
Best, M. J., & Grauer, R. R. (1991). On the sensitivity of mean–variance-efficient portfolios to changes in asset means: Some analytical and computational results. *The Review of Financial Studies*, *4*(2), 315–342.
Birbeck, E., & Cliff, D. (2018). Using stock prices as ground truth in sentiment analysis to generate profitable trading signals. In *2018 IEEE symposium series on computational intelligence* (pp. 1868–1875). IEEE.
Black, F., & Litterman, R. (1990). *Asset allocation: combining investor views with market equilibrium* (p. 115). Goldman Sachs Fixed Income Research.
Black, F., & Litterman, R. (1992). Global portfolio optimization. *Financial Analysts Journal*, *48*(5), 28–43.
Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, *3*, 993–1022.
Cambria, E., Speer, R., Havasi, C., & Hussain, A. (2010). Senticnet: A publicly available semantic resource for opinion mining. In *2010 AAAI fall symposium series*.
Chang, C. C., & Lin, C. J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, *2*(3), 1–27.
Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.
DeMiguel, V., Garlappi, L., & Uppal, R. (2009). Optimal versus naive diversification: How inefficient is the 1/N portfolio strategy? *The Review of Financial Studies*, *22*(5), 1915–1953.
Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
Ding, X., Zhang, Y., Liu, T., & Duan, J. (2014). Using structured events to predict stock price movement: An empirical investigation. In *Proceedings of the 2014 conference on empirical methods in natural language processing* (pp. 1415–1425).
Fama, E. F. (1995). Random walks in stock market prices. *Financial Analysts Journal*, *51*(1), 75–80.
He, G., & Litterman, R. (2002). The intuition behind black–litterman model portfolios. *Social Science Research Network*, Article 334304.
Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.
Idzorek, T. (2007). A step-by-step guide to the black–litterman model: Incorporating user-specified confidence levels. In *Forecasting expected returns in the financial markets* (pp. 17–38). Academic Press.

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In *International conference on machine learning* (pp. 1188–1196). PMLR.
Lee, W. (2000). *Advanced theory and methodology of tactical asset allocation*. New York: John Wiley & Sons.
Litterman, B. (2004). *Modern investment management: an equilibrium approach, Vol. 246*. John Wiley & Sons.
Malkiel, B. G., & Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, *25*(2), 383–417.
Markowitz, H. M. (1952). Portfolio selection. *The Journal of Finance*, *7*(1), 77–91.
Michaud, R. O. (1989). The Markowitz optimization enigma: Is 'optimized'optimal? *Financial Analysts Journal*, *45*(1), 31–42.
Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 3111–3119.
Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. arXiv preprint arXiv:1802.05365.
Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). *Learning internal representations by error propagation*. California Univ San Diego La Jolla Inst for Cognitive Science.
Satchell, S., & Scowcroft, A. (2007). A demystification of the black–litterman model: Managing quantitative and traditional portfolio construction. In *Forecasting expected returns in the financial markets* (pp. 39–53). Academic Press.
Sharpe, W. F. (1966). Mutual fund performance. *Journal of Business*, *39*(1), 119–138.
Sortino, F. A., & Price, L. N. (1994). Performance measurement in a downside risk framework. *The Journal of Investing*, *3*(3), 59–64.
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Å., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *599*, 8–6008.
Wang, J. H., Liu, T. W., Luo, X., & Wang, L. (2018). An lstm approach to short text sentiment classification with word embeddings. In *Proceedings of the 30th conference on computational linguistics and speech processing* (pp. 214–223).
Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., & Klingner, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144.
Xing, F. Z., Cambria, E., & Welsch, R. E. (2018). Intelligent asset allocation via market sentiment views. *IEEE Computational Intelligence Magazine*, *13*(4), 25–34.