IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 24, 2019, accepted May 11, 2019, date of publication May 30, 2019, date of current version June 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2919993

Financial Latent Dirichlet Allocation (FinLDA): Feature Extraction in Text and Data Mining for Financial Time Series Prediction

NONT KANUNGSUKKASEM AND TEERAPONG LEELANUPAB
Faculty of Information Technology, King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand

Corresponding author: Teerapong Leelanupab (teerapong@it.kmitl.ac.th)

This work was partially supported by the Faculty of Information Technology, King Mongkut's Institute of Technology Ladkrabang, Thailand, (IT.KMITL funded project 2017) under the project number: 2560-06-003. The funding was given to T. Leelanupab for mainly sub-granting N. Kanungsukkasem for his Ph.D. study.

ABSTRACT News has been an important source for many financial time series predictions based on fundamental analysis. However, digesting a massive amount of news and data published on the Internet to predict a market can be burdensome. This paper introduces a topic model based on latent Dirichlet allocation (LDA) to discover features from a combination of text, especially news articles and financial time series, denoted as Financial LDA (FinLDA). The features from FinLDA are served as additional input features for any machine learning algorithm to improve the prediction of the financial time series. We provide posterior distributions used in Gibbs sampling for two variants of the FinLDA and propose a framework for applying the FinLDA in a text and data mining for financial time series prediction. The experimental results show that the features from the FinLDA empirically add value to the prediction and give better results than the comparative features including topic distributions from the common LDA.

INDEX TERMS Bayesian method, data mining, data preparation, data processing, feature extraction, financial time series, information processing, latent Dirichlet allocation, news, prediction, stock market, text mining, topic modeling.

I. INTRODUCTION

The efficient market hypothesis (EMH) formulated by Fama [1], [2] suggested that price changes instantly respond to new information, and they are unpredictable. Accordingly, historical data cannot be used to make profitable predictions. However, many approaches have been used to predict financial market movement, crashes or booms [3], and the prediction still remains the subject of active continuing research.

Basically, technical and fundamental analyses are used by investors to predict financial time evolution, such as stock prices. Technical analysts believe that historical market data, primarily price and volume, provide features for price prediction [4]. Also, price and volume can be extended to more complex indices, such as relative strength index (RSI), Accumulation/Distribution Oscillator (A/D), etc. Technical analysis focuses on using methods to extract other information from the historical price and volume. In contrast, various data sources can be used in fundamental analysis; they can be any information about a company or its sector, e.g., cash flow ratio, return on assets (ROA), etc., or macroeconomic, e.g., US gross national product, US consumer price index (CPI), etc. Furthermore, the fundamental data can be unstructured textual data, e.g., global news articles, messages in a webboard, public company disclosures, etc., from which are more difficult to extract information. Accordingly, financial models for stock prediction are usually based on numerical technical and fundamental data and focus on modeling to improve the results, e.g., ARCH models, GARCH models, machine learning algorithms, etc. [5]–[8]

However, after text mining had emerged and become practical to extract information from text, financial research took the unstructured textual information into account more often. Many relied on recognizing keywords: Wüthrich *et al.* [9], for example, extracted articles published on The Wall Street Journal website based on lists of keywords records, judged to

The associate editor coordinating the review of this manuscript and approving it for publication was Xin Luo.

2169-3536 © 2019 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

be influential by domain experts. Then, the keyword counts were weighted and used by their rules, applied to predict the 1-day trend for five major equity indices.

Although taking news into consideration, some studies set textual information aside and only used its numerical data, e.g., the number of news articles and their timestamps. Chan [10], for example, used the date of the news on which the stock was mentioned (stock name was used as a keyword) and found evidence of 'post-news drift'. Their data supported behavioral finance theory [11], [12] about both investor over- and under-reaction to new information coming from investor irrationality.

Some studies considered every word, rather than only some keywords. For instance, Fung *et al.* [13] investigated the immediate impact of news articles, archived by Reuters, on the price changes of Hong Kong Stock Exchange (SEHK) stocks and presented a system to predict rise and fall trends of stock price. They described guided clustering to filter out articles that were not useful in trend prediction. Every word in the article was extracted and assigned a numerical value by tf-idf [14] and then some of the articles were filtered out by an extension of incremental k-Means clustering [15]. Then, they were distinguished by a new differentiated weighting scheme to become features in a Support Vector Machine (SVM) [16] to predict the trends.

Some research in financial text mining extracted more related textual information so as to identify key topics in the underlying story. Jin *et al.* [17], for instance, employed the Latent Dirichlet Allocation (LDA) to reduce Bloomberg news articles into 30 topics and then manually aligned them with currency fluctuation so that top and related topics could be identified. They identified relevant sentences through predefined keywords and calculated movement by using customized sentiment dictionaries. Their system was used to forecast most of the studied events using simple linear regression. For another recent example that showed that financial news topics affect financial time series, Feuerriegel *et al.* [18] used the common LDA to extract 40 topics from German ad-hoc press releases and discovered that some topics significantly affected abnormal returns of stocks.

These examples show that text and data mining for stock prediction is an interdisciplinary research that requires linguistics, machine learning and behavioral finance. So, there remains room for improvement, depending on which area or aspect is used. Practically, the key components, for obtaining a final result, are data selection, data preparation and modeling. Additionally, the LDA model, which extracts new lower-dimensional features, is also considered to be a feature extraction in data preparation. Moreover, many researchers previously reported that news and financial market were related to each other and reducing news to a topic distribution gave value to the relationship. Accordingly, we introduce Financial LDA (FinLDA), a modified LDA, to take changes in financial time series into account to improve feature extraction from text for the prediction. The topic distributions from FinLDA are domain-specific features specialized to the financial domain. Such topic distributions are then considered as features to be employed in a machine learning model to take the advantages of our FinLDA.

Machine learning has become a main stream for financial time series prediction. For over a decade, some researchers have applied artificial neural networks (ANN) and support vector regression (SVR) to predict financial time series, e.g., [6], [19], [20]. Others compared the results from their SVR with the results from the earlier models, especially, Multi-Layer Perceptron (MLP), e.g., [21], [22].

As proposed in this article, FinLDA is a domain-specific type of topic modeling for feature extraction in a data preparation phase. We therefore focus on the explanation of its model and an experiment to study the potential benefits of the features, i.e., abstract topics, derived from it. To evaluate the performance of FinLDA, we need at least one machine learning algorithm to get the final results in a modeling phase. Accordingly, we mainly followed the recent research by Guo *et al.* [22] to use conventional SVR and MLP as our base algorithms. It is our intention by not using other more advanced algorithms like those based on gradient boosting, such as XGBoost [23], LightGBM [24] and CatBoost [25]. This is because we want to investigate the true benefits of FinLDA as an important aspect in feature extraction, leaving those for our soon future work to gain deeper insights in FinLDA when combining it with other algorithms. Even though using them might improve the prediction, the complexity of the other algorithms in a modeling phase could deviate our focus from the feature extraction in a data preparation phase to a variety of algorithms in a modeling phase. Additionally, the most recent gradient boosting algorithm from Yandex, CatBoost, more focuses on handling categorical features whereas the outputs of FinLDA are numerical features [26]. Note that decision trees, which are commonly used as base predictors in gradient boosting, require numerical features for training data.

Unlike Guo *et al.* [22] that compared the performance of SVR and MLP with that of their algorithms as they implemented a new algorithm in a modeling phase, we used SVR and MLP to validate the advantages of the features from FinLDA by comparing the final results from the same machine learning algorithm when using the different sets of features.

Accordingly, we summarize our major contributions as follows:

- We introduced a new domain-specific topic model, the **FinLDA** model, incorporating changes in financial time series into the common Latent Dirichlet Allocation to generate a new set of latent topics related to the changes in a time series.
- We described two variant of FinLDA:
    1) discrete FinLDA (d-FinLDA) uses, as input, the movements that are changes classified into a discrete set of values (e.g., no change, significantly up, minor up, etc.), whereas

2) continuous FinLDA (c-FinLDA) uses real numbers or actual differences.

- We provided posterior distributions used in Gibbs sampling for parameter estimation and inference in topic modeling with FinLDA.
- We considered FinLDA to be a feature extraction in data mining. As a result, this article focuses on the feature extraction in a data preparation phase, but we still need the other phases in data mining to get the final results. Accordingly, we provided the details of a framework for applying FinLDA in text and data mining for financial time series prediction.
- We used two approaches to prepare datasets for evaluation, i.e., train-test split and walk-forward testing routine [20], [27], also called walk-forward testing, walk-forward validation and a part of walk-forward analysis [28].
- We used two conventional machine learning algorithms, i.e., SVR and MLP, in modeling phase to validate the advantages of the features from FinLDA by comparing the final results when their input data were 4 different sets of features, which were combinations of features from LDA, d-FinLDA, c-FinLDA and S&P 500.

To our knowledge, this is the first endeavor to formally define a domain-specific topic modeling based upon the combination of text and change of financial time series for financial prediction. Moreover, we showed that the features from our FinLDA give additional values to predictive models in stock market index prediction.

The rest of this article is organized as follows. In Section II, we briefly summarize related work. Section III introduces two types of FinLDA and details an approach to parameter estimation and inference, and we describe a framework for applying FinLDA in text and data mining for financial time series prediction at the end of the section. We set an experiment and show its comparative results with our discussion in Section IV, followed by our conclusion in Section V.

II. RELATED WORK

Here, we recap the common flow of data mining to show the phase, in which our model is implemented, and summarize machine leaning algorithms, used in our experiments, and LDA.

A. DATA MINING

Typically, a data mining project is managed through six phases based on CRISP-DM (CRoss Industry Standard Process for Data Mining) [29], [30] - see Fig. 1. The solid arrows denote the common flow, from one phase to another, without any revision for correction or improvement. The dashed arrows show flow between phases that could be repeated and the sequence is not rigorous. For example, after an evaluation, the current stage could also return to the data selection or the data preparation phase. We bold Phase 3 to highlight where our FinLDA is implemented in.

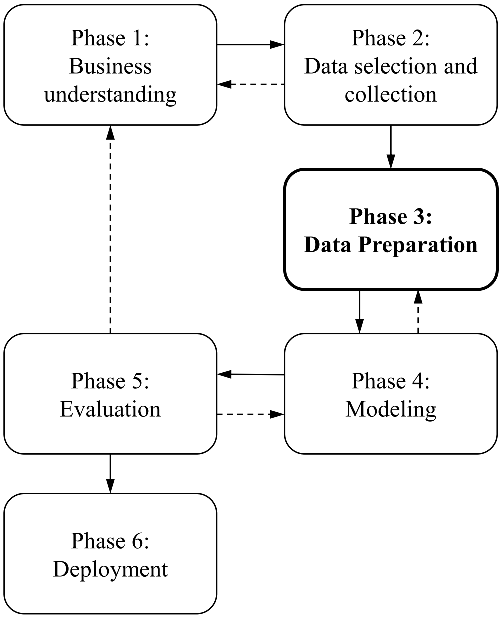


FIGURE 1. Typical life cycle of a data mining project based on CRISP-DM.

SVM and MLP have been commonly used as base models for comparison among different time series prediction models. For example, Tay and Cao [21] compared SVR and Backpropagation Neural Network (BPNN) and showed that SVR performed better. However, Ince and Trafalis [6] showed that MLP outperformed SVR. Recently, Guo *et al.* [22] showed that their improved adaptive SVR outperformed SVR and BPNN. However, they showed that BPNN outperformed SVR in three out of five datasets. Accordingly, as we focus our research on improving data preparation by using our FinLDA, we adopt only the widely used SVR and BPNN in our modeling phase whereas other models could be further investigated in the future.

SVMs [16] are a set of supervised machine learning models. They are discriminative models, many of which are used for classification and some are used for regression analysis. The common method used to solve regression problem is SVR, introduced by Drucker *et al.* [31]. The main idea of SVM is the hyperplane with its margin. The hyperplane in the traditional Support Vector Classification (SVC) divides data into classes, whereas the hyperplane in SVR is used to predict the target value. Moreover, the hinge loss function in SVC is calculated only from the training data in the hyperplane margin, but SVR does not penalize the training data in the margin of tolerance, denoted by epsilon, ϵ. The performance of SVR mostly depends on the selection of a kernel, the penalty coefficient, C, and the parameters of the selected kernel. The Radial Basis Function (RBF) kernel, also called Gaussian kernel, is commonly used [22]. So, we used SVR with the RBF kernel in our experiment. The RBF kernels can be formalized:

$$\text{RBF: } k(x, x') = \exp(-\gamma \|x - x'\|^2) \quad (1)$$

where k stands for kernel and $k(x, x')$ is a similarity function between training data, x, and unlabeled data, x'. γ is a parameter used to tune the similarity function.

Moreover, Guo *et al.* [22] indicated that cross validation was used to determine parameter values. Additionally, they stated that the SVR parameters were set beforehand and SVR with fixed parameters did not apply to constantly changing financial high frequency data. Accordingly, they did not use cross validation in their experiment.

The Multi-Layer Perceptron (MLP) is an earlier supervised machine learning algorithm, a class of feedforward ANNs. Perceptron, introduced by Rosenblatt [32], is a representation learning algorithm and one of the first ANNs. Then, Rumelhart *et al.* [33] presented a variant of the MLP, trained by using backpropagation. Backpropagation has become a common approach for training MLPs and when the backpropagation is used with MLP, such the MLP is commonly denoted by MLP-BP or BPNN. Basically, MLP is a mapping function from input layers, passing through hidden layers, to output layers. The value in each neuron in a hidden layer is calculated from its previous layer with a weighted linear summation followed by a nonlinear activation function. The backpropagation method is used to update the weight, while an MLP model is trained on a given dataset. Moreover, an MLP can perform either classification or regression. So, we used MLP to perform regression in our experiment.

B. LATENT DIRICHLET ALLOCATION (LDA)

Latent Dirichlet Allocation was introduced by Blei *et al.* [34] as a generative probabilistic model to infer multiple latent topics from collections of discrete data, esp. a set of text documents. Blei *et al.* [35] presented further details of LDA and the smoothed LDA model, which became the state-of-the-art LDA. LDA is an unsupervised topic model and, as its only input, requires only basic units of discrete data which are words in documents in text corpora. The input is represented by a gray shaded circle in Fig. 2, as shaded circle is basically the representation of observed node in probabilistic graphical model, where $w_{d,n}$ is the index of word n in document d. The final results from LDA are a predefined number, K, of latent topics, represented by vocabulary word distributions, $\boldsymbol{\beta}$, and a topic distribution, $\boldsymbol{\theta}_d$, per each document, d. $\boldsymbol{\theta}_d$ is calculated from $z_{d,n}$, which is the topic number of word n in document d. $\boldsymbol{\beta}_k$ is a word distribution of topic k, $k \in \{1, \ldots, K\}$, Additionally, α and η are the hyperparameters of the Dirichlet distribution of $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$, respectively.

An approximation of the posterior can be computed by many approximate inference algorithms, e.g., variational Bayes, expectation propagation, Laplace approximation and Markov Chain Monte Carlo (MCMC). Blei *et al.* [34], [35] chose the variational Bayes method for inference and parameter estimation in LDA. However, Griffiths [36] presented an alternative approach for the parameter estimation and the inference in LDA by using Gibbs sampling. Gibbs sampling is a Markov Chain Monte Carlo (MCMC), which is an algorithm for sampling from an intricate probability distribution. Gibbs sampling allows us to calculate approximate parameters and to infer the distribution by repetitive sampling [37]. Moreover, Griffiths and Steyvers [38] showed that Gibbs sampling was empirically the most efficient method for static topic models.

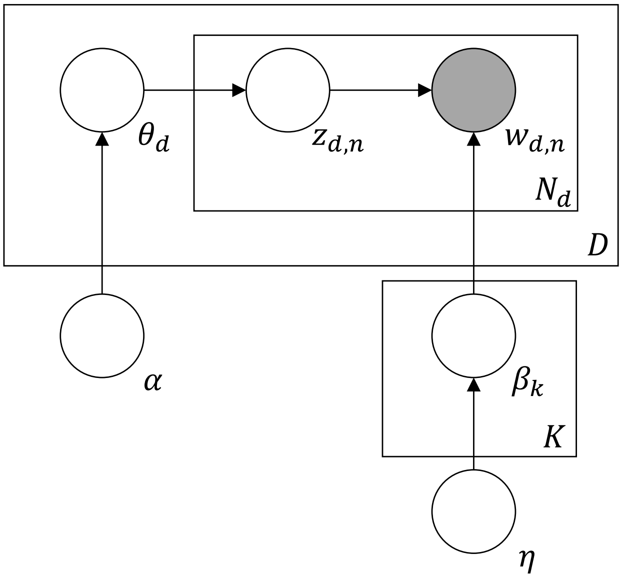


FIGURE 2. Probabilistic graphical model that represents the smoothed latent dirichlet allocation.

III. FinLDA

In this section, we describe parameters and generative processes of our discrete and continuous FinLDA, followed by their parameter estimation and inference. In the end of this section, we describe a framework for using our FinLDA in text and data mining for financial time series prediction.

A. MODEL DESCRIPTION

It was previously reported that changes in financial time series had association with topic generation. Inspired by this, our FinLDA is therefore developed as a modification of the common LDA to accommodate such the changes for feature extraction. Fig. 3 presents the probabilistic graphical model of our FinLDA, where a change in financial time series, after document d is published, is represented by the observed node, f_d, in the blue shaded circle. Furthermore, we added a distribution of the changes per topic, δ_k, in the figure, as a hidden node to link with the other distributions through f_d. The historical data of a selected financial time series is processed to find its price change, f_d, after the document d is published. The time-lag that is used to consider the price change can be any interval, e.g., 5 minutes, 1 hour, 2 days, etc. In the training process, when the financial time series are

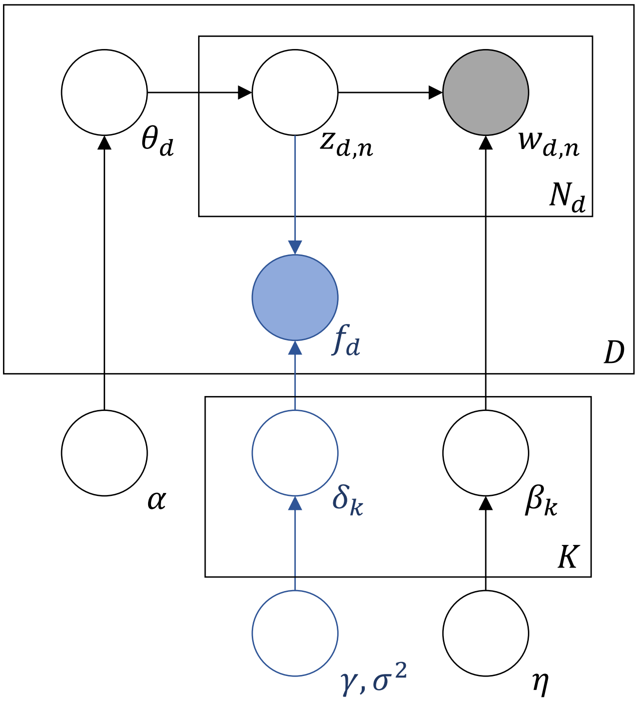


FIGURE 3. Probabilistic graphical model that represents FinLDA. Note that σ^2 is used in continuous FinLDA only.

available, the distribution of changes affects the distribution of words in every topic k, $\boldsymbol{\beta}_k$, and also the distribution of topics in any document d, $\boldsymbol{\theta}_d$. After the parameter estimation, the estimated word distributions in FinLDA are used in the inference method on a new document to get its latent topic distribution which we consider as input features for any machine learning algorithm. In summary, FinLDA uses a combination of words and changes of financial time series to improve the estimated parameters for inference from a new document and then get better input features from the inference for any machine learning algorithm to predict the time series.

Assume a corpus of D documents contains V vocabulary words, and document d consists of N_d word tokens, $(w_{d,1}, \ldots, w_{d,N_d})$. Each word token n in document d, $w_{d,n}$, is assigned as a numeric token, essentially an index in the vocabulary, $w_{d,n} \in \{1, \ldots, V\}$. The number of topics, K, is assumed to be known and fixed.

Derived from LDA, FinLDA is also a probabilistic generative model. Accordingly, any word n in document d, $w_{d,n}$, is assumed to be generated from a list of probabilities of V words when the word is topic k, $\beta_{k,v}$, where the topic of the word, $z_{d,n}$, is generated from a list of probabilities of K topics when the word is generated in the document d, $\theta_{d,k}$. However, we consider the price changes, f_d, in two different ways, i.e., discrete and continuous. Note that our work can be applied either to stock price or index value changes: although we used the S&P 500 index in our experiments, we follow others and use 'price' as a generic term for both.

1) DISCRETE FINLDA (D-FINLDA)

We categorize continuous changes of financial time series into M movements: f_d in d-FinLDA is a movement of price after document d is published. The number of categories, M, depends on thresholds, e.g., there are only two movements if the threshold is 0.5% change in both increase and decrease. But, if we set the threshold to be -1% change and $+1\%$ change, there will be three movements, $M = 3$, i.e., price change, $\Delta > 1\%$, $\Delta < -1\%$ or $-1\% \leqslant \Delta \leqslant 1\%$.

Accordingly, in the generative process, the movement after the document d is published, f_d, is generated from a list of probabilities of M movements, when the probability of the movement m is calculated by weighted average of the probabilities of movement of each topic, $\delta_{k,m}$, by the number of word tokens in each topic k in the document d. Furthermore, we use symmetric Dirichlet priors for all Dirichlet distributions which are conjugate to multinomial distributions in the model. Accordingly, each of the hyperparameters, η, γ and α, is a single value.

The assumption of d-FinLDA, following its probabilistic graphical model, is described by a generative process as follows:

1) For each topic $k \in \{1, \ldots, K\}$:
   a) Generate $\boldsymbol{\beta}_k|\eta = (\beta_{k,1}, \ldots, \beta_{k,V}) \sim Dir(\eta)$
   b) Generate $\boldsymbol{\delta}_k|\gamma = (\delta_{k,1}, \ldots, \delta_{k,M}) \sim Dir(\gamma)$
2) For each document $d \in \{1, \ldots, D\}$:
   a) Generate $\boldsymbol{\theta}_d|\alpha = (\theta_{d,1}, \ldots, \theta_{d,K}) \sim Dir(\alpha)$
   b) For each word token $n \in \{1, \ldots, N_d\}$:
      i) Generate $z_{d,n}|\boldsymbol{\theta}_d \in \{1, \ldots, K\} \sim Mult(\boldsymbol{\theta}_d)$
      ii) Generate $w_{d,n}|\boldsymbol{\beta}, z_{d,n} \in \{1, \ldots, V\} \sim Mult(\boldsymbol{\beta}_{z_{d,n}})$
   c) Generate $f_d|\boldsymbol{\delta}, \mathbf{z}_d \in \{1, \ldots, M\} \sim Mult(\boldsymbol{\mu}_d)$ where $\boldsymbol{\mu}_d = ((\sum_n \delta_{z_{d,n},1})/N_d, \ldots, (\sum_n \delta_{z_{d,n},M})/N_d)$

The joint distribution of the hidden and observed variables for d-FinLDA is:

$$P(\boldsymbol{\beta}, \boldsymbol{\delta}, \boldsymbol{\theta}, \mathbf{z}, \mathbf{w}, \boldsymbol{f}|\alpha, \eta, \gamma) = \prod_{k=1}^{K} P(\boldsymbol{\beta}_k|\eta) \prod_{k=1}^{K} P(\boldsymbol{\delta}_k|\gamma) \prod_{d=1}^{D} \Bigg(P(\boldsymbol{\theta}_d|\alpha) \times \Big(\prod_{n=1}^{N_d} P(z_{d,n}|\boldsymbol{\theta}_d) P(w_{d,n}|z_{d,n}, \boldsymbol{\beta}) \Big) P(f_d|\mathbf{z}_d, \boldsymbol{\delta}) \Bigg) \quad (2)$$

where $\boldsymbol{\beta}$ is a $K \times V$ matrix of the probabilities of V words for all K topics and $\beta_{k,v}$ is a probability of word v when it occurs in topic k.

2) CONTINUOUS FINLDA (C-FINLDA)

In the continuous model, we used the price changes directly, $f_d = price_{t+timelag} - price_t$. Accordingly, in the generative process, the price change after the document d is published, f_d, is generated from a normal distribution with the weighted average of means, μ_d, and a variance, σ^2, instead of a multinomial distribution. We use a single variance, σ^2, because it is the inherited variance from their parents, $\boldsymbol{\delta}$, which are distributed normally with the expectation of the distribution, γ, and their variance, σ^2.

The assumption of c-FinLDA is described by a generative process as follows:

1) For each topic $k \in \{1, \ldots, K\}$:
   a) Generate $\boldsymbol{\beta}_k|\eta = (\beta_{k,1}, \ldots, \beta_{k,V}) \sim Dir(\eta)$
   b) Generate $\delta_k|\gamma, \sigma^2 \sim \mathcal{N}(\gamma, \sigma^2)$
2) For each document $d \in \{1, \ldots, D\}$:
   a) Generate $\boldsymbol{\theta}_d|\alpha = (\theta_{d,1}, \ldots, \theta_{d,K}) \sim Dir(\alpha)$
   b) For each word token $n \in \{1, \ldots, N_d\}$:
      i) Generate $z_{d,n}|\boldsymbol{\theta}_d \in \{1, \ldots, K\} \sim Mult(\boldsymbol{\theta}_d)$
      ii) Generate $w_{d,n}|\boldsymbol{\beta}, z_{d,n} \in \{1, \ldots, V\} \sim Mult(\boldsymbol{\beta}_{z_{d,n}})$
   c) Generate $f_d|\boldsymbol{\delta}, z_d \sim \mathcal{N}(\mu_d, \sigma^2)$ where $\mu_d = (\sum_n \delta_{z_{d,n}})/N_d$,

The joint distribution of the hidden and observed variables for c-FinLDA is:

$$\begin{aligned}&P(\boldsymbol{\beta}, \boldsymbol{\delta}, \boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{w}, \boldsymbol{f}|\alpha, \eta, \gamma, \sigma^2)\\ &\quad= \prod_{k=1}^{K} P(\boldsymbol{\beta}_k|\eta) \prod_{k=1}^{K} P(\delta_k|\gamma, \sigma^2) \prod_{d=1}^{D} \Bigg(P(\boldsymbol{\theta}_d|\alpha)\\ &\qquad \times \Big(\prod_{n=1}^{N_d} P(z_{d,n}|\boldsymbol{\theta}_d) P(w_{d,n}|z_{d,n}, \boldsymbol{\beta})\Big) P(f_d|z_d, \boldsymbol{\delta})\Bigg)\end{aligned} \tag{3}$$

B. PARAMETER ESTIMATION

The essential task to make FinLDA ready for the inference is to get the estimated values of all hidden variables, i.e., $\boldsymbol{\theta}$, $\boldsymbol{\beta}$, $\boldsymbol{\delta}$ and z. The values of hidden variables can be estimated by computing the posterior distribution of each of them given the words in documents, $\boldsymbol{w}$, and the price changes, after the documents were published, $\boldsymbol{f}$. The computation is based on the assumption in the previous subsection.

1) PARAMETER ESTIMATION FOR D-FINLDA

The posterior distribution of the hidden variables given the observed variables for d-FinLDA is

$$P(\boldsymbol{\beta}, \boldsymbol{\delta}, \boldsymbol{\theta}, z|\boldsymbol{w}, \boldsymbol{f}, \alpha, \eta, \gamma) = \frac{P(\boldsymbol{\beta}, \boldsymbol{\delta}, \boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{w}, \boldsymbol{f}, \alpha, \eta, \gamma)}{P(\boldsymbol{w}, \boldsymbol{f}, \alpha, \eta, \gamma)} \tag{4}$$

Theoretically, this can be calculated as the sum of the joint distributions from every possible hidden variable value. However, it is intractable to compute. Therefore, we applied the Gibbs sampling algorithm to find the inference. Each state of the Markov chain samples a value for each parameter, given the current values of the other parameters.

After the random variables, β, θ and δ, are analytically marginalized out, the posterior distribution for sampling in parameter estimation for discrete FinLDA is

$$\begin{aligned}&P(z_{d,n} = k|\boldsymbol{w}, \boldsymbol{f}, \boldsymbol{z} \setminus z_{d,n}, \alpha, \eta, \gamma) \propto\\ &\quad \times \left\{(N_{d,k} + \alpha) \times \frac{N_{k,w_{d,n}} + \eta}{N_k + V\eta} \times \frac{N_{k,f_d} + \gamma}{N_k + M\gamma}\right\}_{-(d,n)}\end{aligned} \tag{5}$$

where $N_{d,k}$ is the number of word tokens in document d that are assigned to topic k. $N_{k,w_{d,n}}$ is the number of word tokens that are the same vocabulary word as $w_{d,n}$ in topic k. N_{k,f_d} is the number of word tokens that are assigned to topic k and in any document with movement f_d. N_k is the number of word tokens in topic k. α, η, γ are the predefined hyperparameters. The value of each parameter within curly braces with the subscript $-(d, n)$ is the value of each parameter when $w_{d,n}$ is excluded.

After a sufficient number of sampling iteration, the samples become the estimated values of the hidden parameters, i.e., $\boldsymbol{\beta}$, $\boldsymbol{\theta}$, z and $\boldsymbol{\delta}$. The estimated value of a single parameter can be calculated by

$$\theta_{d,k} = \frac{N_{d,k} + \alpha}{N_d + K\alpha} \tag{6}$$

$$\beta_{k,w_{d,n}} = \frac{N_{k,w_{d,n}} + \eta}{N_k + V\eta} \tag{7}$$

$$\delta_{k,f_d} = \frac{N_{k,f_d} + \gamma}{N_k + M\gamma} \tag{8}$$

Parameter estimation by Gibbs sampling for the discrete FinLDA is set out as pseudo-code in Algorithm 1.

Algorithm 1 Gibbs Sampling Algorithm for d-FinLDA

```
Data: All word tokens in $D$ Documents, $\boldsymbol{w}$, and their
      price movement, $\boldsymbol{f}$
Result: $\boldsymbol{z}, \boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\delta}$
initialize $\alpha, \eta, \gamma, K, N_{iter}$
initialize $z_{d,n}$ randomly for all $N_d$ words in all $D$ documents
foreach iteration do
    for $d = 1$ to $D$ do
        for $n = 1$ to $N_d$ do
            sample $z_{d,n}$ from $P(z_{d,n}|\boldsymbol{w}, \boldsymbol{f}, \boldsymbol{z} \setminus z_{d,n}, \alpha, \eta, \gamma)$
            update $\boldsymbol{\theta}_d$, $\boldsymbol{\beta}$ and $\boldsymbol{\delta}$
        end
    end
end
```

2) PARAMETER ESTIMATION FOR C-FINLDA

The posterior distribution of the hidden variables, given the observed variables for c-FinLDA, is:

$$P(\boldsymbol{\beta}, \boldsymbol{\delta}, \boldsymbol{\theta}, z|\boldsymbol{w}, \boldsymbol{f}, \alpha, \eta, \gamma, \sigma^2) = \frac{P(\boldsymbol{\beta}, \boldsymbol{\delta}, \boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{w}, \boldsymbol{f}, \alpha, \eta, \gamma, \sigma^2)}{P(\boldsymbol{w}, \boldsymbol{f}, \alpha, \eta, \gamma, \sigma^2)} \tag{9}$$

The posterior distribution for sampling in parameter estimation for continuous FinLDA is:

$$\begin{aligned}&P(z_{d,n} = k|\boldsymbol{w}, \boldsymbol{f}, \boldsymbol{z} \setminus z_{d,n}, \alpha, \eta, \gamma, \sigma^2) \propto\\ &\quad \times \left\{(N_{d,k} + \alpha) \times \frac{N_{k,w_{d,n}} + \eta}{N_k + V\eta} \times \exp\left\{-\frac{(f_d - \delta_k)^2}{2\sigma^2}\right\}\right\}_{-(d,n)}\end{aligned} \tag{10}$$

where $\delta_k = \sum_d (f_d N_{k,f_d}) / \sum_d N_{k,f_d}$

The estimation of each θ and β in c-FinLDA after sufficient sampling iteration is the same as that in d-FinLDA. The only difference is the estimation of δ. It has only one value per topic because of its Gaussian distribution. The probability for a single sample can be calculated:

$$p(z_{d,n} = k|f_d) = \frac{\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{(f_d-\delta_k)^2}{2\sigma^2} \right\}}{\sum_k \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{(f_d-\delta_k)^2}{2\sigma^2} \right\}} \tag{11}$$

Parameter estimation by Gibbs sampling for the continuous FinLDA is set out in Algorithm 2: it follows the Algorithm 1 with the additional calculation of the mean and variance, γ and σ^2, in line 2, 3 and 7.

Algorithm 2 Gibbs Sampling Algorithm for c-FinLDA

```
Data: All word tokens in $D$ Documents, $\boldsymbol{w}$, and their
      price changes, $\boldsymbol{f}$
Result: $\boldsymbol{z}, \boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\delta}$
1  initialize $\alpha, \eta, K, N_{iter}$
2  initialize $z_{d,n}$ randomly for all $N_d$ words in all $D$
   documents
3  calculate $\gamma$ and $\sigma^2$ by using $f_d$ and $N_d$ from all $D$
   documents
4  foreach iteration do
5      for $d = 1$ to $D$ do
6          for $n = 1$ to $N_d$ do
7              sample $z_{d,n}$ from
               $P(z_{d,n}|\boldsymbol{w}, \boldsymbol{f}, \boldsymbol{z} \setminus z_{d,n}, \alpha, \eta, \gamma, \sigma^2)$
8              update $\boldsymbol{\theta}_d$, $\boldsymbol{\beta}$ and $\boldsymbol{\delta}$
9          end
10     end
11 end
```

However, the number of iterations and the number of topics need to be defined before the sampling start. Griffiths and Steyvers [38] suggested the method to determine the number of iterations and the number of topics in LDA. They showed that the energy of assignments of all z was measured by The Hamiltonian, $H(z)$, and was directly proportional to $-\log P(\boldsymbol{w}, z)$:

$$H(z) \propto -logP(\boldsymbol{w}, z) \tag{12}$$

$$-logP(\boldsymbol{w}, z) = -logP(\boldsymbol{w}|z) - logP(z) \tag{13}$$

With Dirichlet distributions, $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$, as conjugate priors to $\boldsymbol{w}$ and z, $P(\boldsymbol{w}|z)$ and $P(z)$ are probability mass functions of the Dirichlet-multinomial distribution, also called the Dirichlet Compound Multinomial (DCM). After the marginal joint distribution is obtained,

$$P(\boldsymbol{w}|z) = \left(\frac{\Gamma(V\eta)}{\Gamma(\eta)^V}\right)^K \prod_{k=1}^{K} \frac{\prod_v \Gamma\left(N_{k,v} + \eta\right)}{\Gamma\left(N_k + V\eta\right)} \tag{14}$$

$$P(z) = \left(\frac{\Gamma(K\alpha)}{\Gamma(\alpha)^K}\right)^D \prod_{d=1}^{D} \frac{\prod_k \Gamma\left(N_{d,k} + \alpha\right)}{\Gamma\left(N_d + K\alpha\right)} \tag{15}$$

where $v \in \{1, \ldots, V\}$ and $N_{k,v}$ is the number of word tokens that are the same as vocabulary word v and assigned to topic k.

Accordingly, the log-likelihood, $-\log P(w, z)$, can be used to find the proper number of iterations and the proper number of topics, K, for our FinLDA. They indicated that appropriate assignments of words to topics should emerge after a number of iterations, when the log-likelihood has stabilized, and running several experiments with different numbers of topics should get a suitable number of topics when the log-likelihood reached a minimum.

C. INFERENCE FROM A NEW DOCUMENT

After fitting the model, we can infer the posterior distribution for a new document by using the value of z as well as the approximated $\boldsymbol{\beta}$, $\boldsymbol{\theta}$ and $\boldsymbol{\delta}$. However, the probabilistic graphical model of FinLDA in Fig. 3 shows that if the price change, f_d, is not known, $\boldsymbol{\delta}$ and γ (and σ^2 for c-FinLDA) will not affect the other parameters in the model. Accordingly, the inference from a new document, with unknown price change, is the same as the inference for the common LDA. The posterior distribution for sampling in the inference from a new document is:

$$\begin{aligned} &P(z_{new,n} = k|\boldsymbol{w}_{new}, \boldsymbol{z} \setminus z_{new,n}, \boldsymbol{\beta}, \alpha, \eta) \\ &\quad \propto \left\{ (N_{new,k} + \alpha) \times \frac{N_{k,w_{new,n}} + \eta}{N_k + V\eta} \right\}_{-(new,n)} \end{aligned} \tag{16}$$

The inference from a new document is shown as pseudocode in Algorithm 3, following the Algorithms 1 and 2 without $\boldsymbol{f}$, $\boldsymbol{\delta}$, γ and σ^2.

Algorithm 3 Inference From a New Document in Topic Modeling With FinLDA

```
Data: All word tokens in a new document, $\boldsymbol{w}_{new}$ and $\boldsymbol{\beta}$
Result: $z_{new}$ and $\boldsymbol{\theta}_{new}$
initialize $\alpha, \eta, N_{iter}$
initialize $z_{new,n}$ randomly for all $N_{new}$ words in the new
document
foreach iteration do
    for $n = 1$ to $N_{new}$ do
        sample $z_{new,n}$ from
        $P(z_{new,n}|\boldsymbol{w}_{new}, \boldsymbol{z} \setminus z_{new,n}, \boldsymbol{\beta}, \alpha, \eta)$
        update $\boldsymbol{\theta}_{new}$
    end
end
```

D. FINLDA IN TEXT AND DATA MINING FOR FINANCIAL TIME SERIES PREDICTION

In this subsection, we describe the calculation flow when FinLDA is applied in text and data mining for market prediction following the typical phases in a data mining project, see Fig. 1, as well as data and methods that are alternatives.

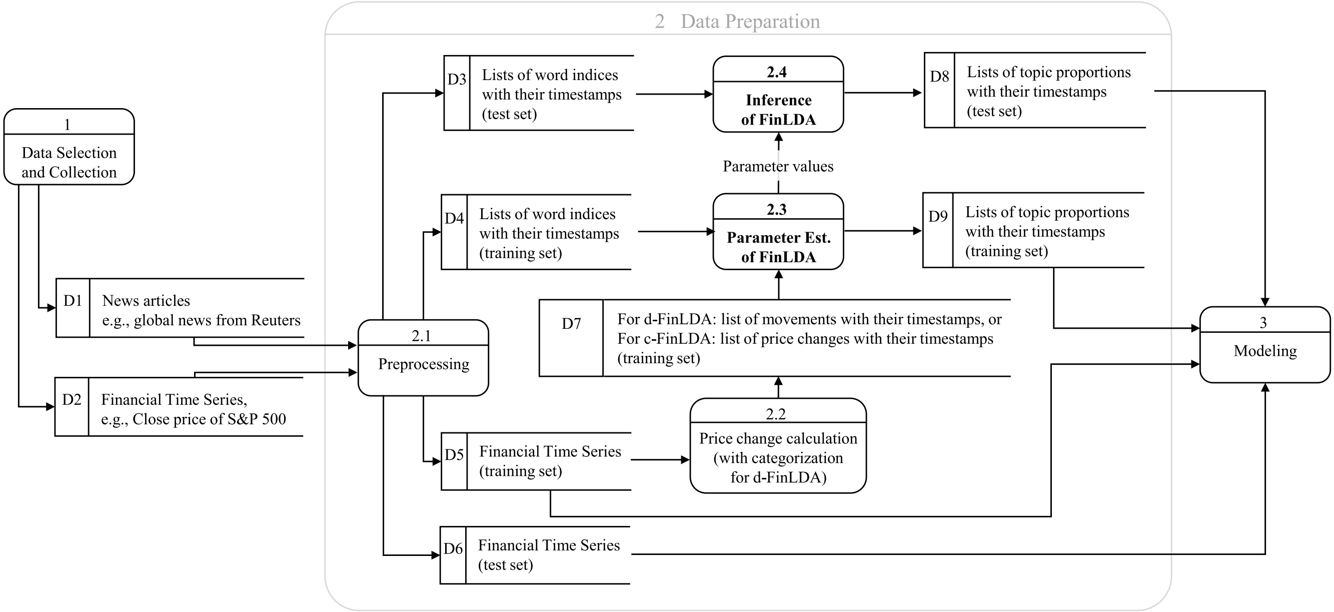

2 Data Preparation
1
Data Selection and Collection
D1 News articles e.g., global news from Reuters
D2 Financial Time Series, e.g., Close price of S&P 500
2.1
Preprocessing
D3 Lists of word indices with their timestamps (test set)
D4 Lists of word indices with their timestamps (training set)
D5 Financial Time Series (training set)
D6 Financial Time Series (test set)
2.4
Inference of FinLDA
Parameter values
2.3
Parameter Est. of FinLDA
D7 For d-FinLDA: list of movements with their timestamps, or For c-FinLDA: list of price changes with their timestamps (training set)
2.2
Price change calculation (with categorization for d-FinLDA)
D8 Lists of topic proportions with their timestamps (test set)
D9 Lists of topic proportions with their timestamps (training set)
3
Modeling

FIGURE 4. Data-flow diagram showing the place of FinLDA in a general data mining for financial time series prediction.

Here, we used FinLDA in text and data mining to predict financial time series and chose the S&P 500 index in our experiment. Alternatively, we could have chosen several other market indices or single stock prices. Also, the other large volumes of news in text form and possibly related to the S&P 500 index, stamped with publication time, down to the minute, could be used. Additionally, macro-economic data, e.g., unemployment rate, import or export trade amount, etc., or, for single stock price prediction, information about the company or its sector, e.g., net income, liabilities ratio, etc., could enhance the prediction.

In the data preparation phase of a general data mining exercise, there are many methods to process the data to get better features for the next phase and thus a better final result. Choosing a feature extraction model also depends on what kind of descriptive and target features are relevant to our goal. In the overall data mining context, a topic modeling with FinLDA is used for feature extraction in the data preparation phase. A usual pre-requisite is that some of the raw data need to be preprocessed. In our experiments, natural language text needs to be preprocessed by tokenization, stemming, stop word and noise removal, etc. A simple model often used to extract features from a text is the 'bag-of-words' model: this represents words in a document as vectors.

FinLDA requires only vectors of words, which 'bag-of-words' provides, and the historical market data as raw descriptive features in the preparation phase. However, historical data needs to be calculated to get price changes for the continuous FinLDA and categorized to movements for the discrete FinLDA - see Fig. 4. The outputs from parameter estimation in FinLDA, after trained by a training set, are three probability lists, i.e., topic proportions of all documents in the training set, word proportions for all topics and price movement/change distributions for all topics. The topic proportions become additional features for training a machine learning algorithm in the next phase. Furthermore, the word proportions become the mandatory parameter values for topic inference on a new document, which, in turn, generates a topic proportion of a new document and that proportion becomes additional features for a machine learning algorithm to predict a time series in the next phase. Moreover, simple historical market data, used as input features, can be converted to other derivative indices, e.g., moving average convergence divergence (MACD), stochastic oscillator (STO), etc., to form additional descriptive features.

Finally, a machine learning algorithm that can compute regressions, e.g., BPNN and SVR, used in this experiment, needs to be chosen to extract the advantage from our additional features. The chosen model needs to be trained by the topic distributions from the training data, which are outputs from parameter estimation in FinLDA, with other features. Eventually, the model can be used to predict the chosen index (or price or other series).

IV. EXPERIMENTAL RESULT AND DISCUSSION

This section shows an experiment setup and its results in each step, including discussion alongside the comparative results. Our experiments are based on the data flow shown in Fig. 5, which is similar to Fig. 1, but trimmed into four phases. We compared our system's predictions for close prices of Standard & Poor's 500 Index (S&P 500) in the five minutes after news articles appeared on the Reuters website [39]. This experiment was conducted to show the benefits of including

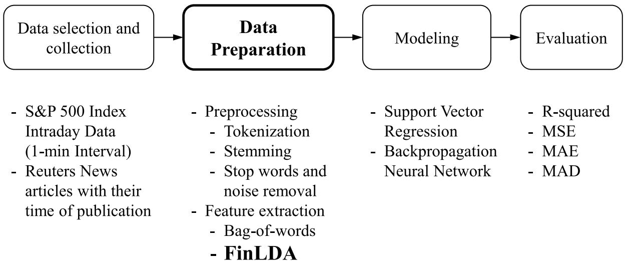

Data selection and collection
Data Preparation
Modeling
Evaluation
- S&P 500 Index Intraday Data (1-min Interval)
- Reuters News articles with their time of publication
- Preprocessing
- Tokenization
- Stemming
- Stop words and noise removal
- Feature extraction
- Bag-of-words
- FinLDA
- Support Vector Regression
- Backpropagation Neural Network
- R-squared
- MSE
- MAE
- MAD

FIGURE 5. **Data flow in the four phases of our experimental system.**

the features generated by FinLDA by comparing the final results when using four different sets of features.

- Open, Close, Low and High prices from S&P 500 alone (SP500), as base features,
- SP500 with topic distributions from LDA (LDA&SP500),
- SP500 with topic distributions from d-FinLDA (d-FinLDA&SP500), or
- SP500 with topic distributions from c-FinLDA (c-FinLDA&SP500).

As discussed in Section I, the focus of this experiment is on the evaluation of FinLDA in data preparation. We thus considered to use conventional BPNN and SVR as the tools to get the final results to validate the benefit of the features from FinLDA.

In addition, the experiment needs test sets for backtesting. Basically, for non-time series datasets, Hold-out sampling is a simple approach to prepare the dataset for an evaluation of machine learning algorithm, and k-fold cross validation is commonly used as a standard approach [29]. The latter simply partitions samples into k sub-samples and uses a partition as a test set and the rest as a training set. However, in financial time series prediction, past data are required to predict value in the future. Consequently, the simple k-fold cross validation that mixes between past and future data in a training set is inappropriate due to the temporal component. Accordingly, we used the other two approaches in backtesting for time series forecasting:

- The simple train-test split, also called the out-of-time validation, is useful and reliable when the dataset is *large* enough to train an accurate model. It is often used to fully evaluate the performance of the model [40] (see Fig. 6).
- Walk-forward testing routine [20], [27], which is a variation of cross-validation, divides dataset into k overlapping training-test sets (see Fig. 7).

To measure the performance of BPNN and SVR when using four different sets of features, we used four metrics, i.e., the coefficient of determination, R^2, Mean Squared Error (MSE), Mean Absolute Error (MAE) and Mean Absolute Deviation (MAD). Each is computed as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(observed_i - predicted_i)^2}{\sum_{i=1}^{N}(predicted_i - \overline{predicted})^2} \quad (17)$$

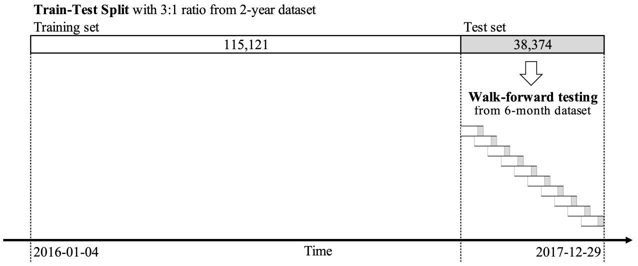


FIGURE 6. **Train-test split from 2-year dataset and walk-forward testing from the last 6 months of the dataset.**

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(observed_i - predicted_i)^2 \quad (18)$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|observed_i - predicted_i| \quad (19)$$

$$MAD = \frac{1}{N}\sum_{i=1}^{N}|observed_i - \overline{predicted}| \quad (20)$$

where N is the number of samples in the test set.

A. DATA SELECTION AND COLLECTION

S&P 500 is a capitalization-weighted index of common stocks based on the 500 largest companies listed on the NYSE or NASDAQ ranked by their market capitalization. The index covers about 80% of the available American equity market capitalization: the stocks in the index come from all 11 stock market sectors and most of the 157 sub industries in the U.S. Thus, it should be tested with global news which could affect many sectors. We collected 1 minute-level intraday S&P 500 market historical data from January 4, 2016 at 9.31 A.M. to December 29, 2017 at 4.07 P.M. from [41] - a total of 196,757 records, each of which has open, close, low and high prices in 1-minute interval. However, 79 records of 1 minute-level intraday market historical data in that period were missing.

Reuters has archived and provided past news articles with their time of publication on its public website [42]. Totally, 824,424 articles of global news were published on the website from January 1, 2016 to December 31, 2017. However, some were videos and slide shows from which we could not extract text, and a few were irretrievable. Thus, there are 820,731 usable text news articles.

B. DATA PREPARATION

To avoid some possibly abnormal price changes when the market is opening and closing, we set the experiment period on each trading day to start at 15 minutes after the market open and end at 15 minute before closing time, 9.45 a.m. to 3.45 p.m. EST (GMT-5). However, as we experiment to predict the price change in the next 5 minute after the news article is published on the website, we used only articles published between 9.45 a.m. to 3.40 p.m. EST (GMT-5)

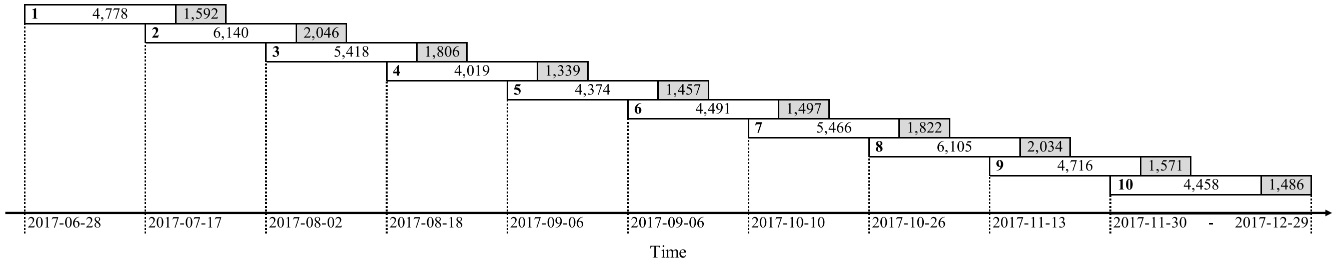


FIGURE 7. Each step in the walk-forward testing with the window of 21 working days and 3 to 1 split for training and test sets.

on trading days. Accordingly, the number of usable articles is reduced to 153,581. Additionally, due to 79 missing records of the historical data, 86 articles, which were published at that time or 5 minutes before that time, cannot be used. Consequently, 153,495 out of 153,581 documents are the text data for this experiment. After tokenization and stemming, by Python's Natural Language Toolkit [43], and removal of stop words and noise, 24,378,279 word tokens and 170,875 vocabulary words, V, remained. Each word in each document was converted to a numerical representation by using 'bag-of-words' function in class Dictionary [44] in Gensim [45] and matched with the price change in the 5-minute interval after the documents were published.

We set the time-lag to be 5 minutes and used percent change instead of price change. So, f_d for c-FinLDA is

$$f_d = \frac{price_{t+5min} - price_t}{price_t} \times 100 \tag{21}$$

For d-FinLDA, the changes must be classified into movements, and we considered the threshold based on the average of the 5-min changes in our training data collection period, which we used 3 to 1 ratio to split the data into training and test sets. The average is 0.04% change in both direction. We experimented with 0.05% change in both direction (rise or fall), which is a bit higher than the average, as our threshold. Accordingly, there are two movements, i.e., movement 1 if the price changes over the threshold, and movement 0 if the price does not change over the threshold.

$$f_d = \begin{cases} 1, & \text{if } \frac{|price_{t+5min} - price_t|}{price_t} \times 100 > 0.05 \\ 0, & \text{otherwise} \end{cases} \tag{22}$$

We found 30,853 articles connected with the movement 1 ($f_d = 1$) and 122,642 articles with the movement 0 ($f_d = 0$).

After all the data were preprocessed, we need to split the data into training and test sets by our two approaches for backtesting. For the first approach, we performed the out-of-time validation to validate FinLDA as a whole in the entire collection. We believe that our data are sufficiently large for the simple validation, and as such, it massively reduced the processing time of our experiment. The whole dataset was split into training and test sets, with 3 to 1 ratio, without shuffle because of their temporal component. Accordingly, there were 115,121 samples of data in training set and 38,374 samples of data in test set, as shown in Fig. 6.

For the second approach, as our dataset is pretty large, we selected approximately the last 6 months of the 2-year dataset, which had been the test set and never been used in the training set in the out-of-time validation. We divided it into 10 overlapping datasets with 21-working-day window (21 is the approximate number of working days per month) for each step. Each dataset was split into training and test sets with 3 to 1 ratio, as shown in Fig. 7. The results from the two different validations were separately evaluated in the evaluation phase. Accordingly, we mainly discuss the results in this data preparation phase based on the whole 2-year dataset.

The data that were split to be the inputs for training our FinLDA model are 115,121 documents, which are lists of word indices, $w_{d,n}$, a list of 115,121 change values, f_d, for c-FinLDA and a list of 115,121 movement values, f_d, for d-FinLDA. The other parameters, i.e., the hyperparameters of the Dirichlet distributions, the number of topics and the number of iterations, need to be set. The hyperparameters, α, γ and η, are basically less than 1, the value of each of which is specified, depending on how much sparse of the distribution we want. Without any particular principle, each research in the past chose a different choice of the value, e.g., Griffiths and Steyvers [38] used $\eta = 0.1$, $\alpha = 50/K$, Asuncion *et al.* [46] used $\eta = 0.1$, $\alpha = 0.1$ and Řehůřek and Sojka, used $\eta = 1/K$, $\alpha = 1/K$ in Gensim [45]. In our experiment, we set $\eta = 0.01$ for all topic models, so that we had a few words with high probability per topics. Similarly, we set $\gamma = 0.01$ for d-FinLDA but $\alpha = 0.1$, for all topic models, to get many latent topics with high probability per document. However, some experimentation was needed to find the optimum number of topics and iterations.

We set the number of iterations in LDA, $n_{iter} = 120$, with different numbers of topics, K, to show the effect of iteration count and topic numbers on log-likehoods. Fig. 8a shows that the log-likelihoods changed slowly, approximately, after $n_{iter} > 40$ and stabilized, approximately, when $n_{iter} > 100$. However, the experiment with LDA did not help us to decide the appropriate number of topics, K. We, then, attempted to find the optimum K from d-FinLDA which led to results in Fig.8b, which also shows stable results when $n_{iter} > 100$. However, the results are better viewed in Fig. 9 which shows log-likehoods against the number of topics. The plots suggest

TABLE 1. Example of topics extracted by LDA, d-FinLDA and c-FinLDA.

LDA					d-FinLDA				c-FinLDA			
Topic 1 Baseball	Topic 2 Health	Topic 3 Olympics	Topic 4 China	Topic 5 MidEast	Topic 1 Baseball	Topic 2 Health	Topic 3 China	Topic 4 MidEast	Topic 1 Baseball	Topic 2 Health	Topic 3 Olympics	Topic 4 MidEast
run	**health**	world	**china**	forc	game	**health**	**china**	**attack**	game	**health**	**olymp**	**islam**
game	**drug**	time	north	**attack**	team	**drug**	reuter	**govern**	**run**	studi	**rio**	unit
start	**research**	team	unit	offici	**run**	studi	chines	forc	hit	**drug**	world	**govern**
hit	studi	**olymp**	servic	**russia**	win	**olymp**	water	reuter	season	**research**	team	forc
lead	women	**rio**	media	**govern**	play	patient	world	report	**inning**	patient	game	report
season	peopl	final	south	al	hit	**rio**	industri	**islam**	play	peopl	time	reuter
home	risk	de	report	**islam**	time	**research**	countri	**millitari**	start	**zika**	reuter	al
five	patient	win	chines	**millitari**	season	women	develop	peopl	score	women	sport	**attack**
left	medic	won	time	unit	start	reuter	south	unit	lead	percent	edit	**millitari**
time	**zika**	sport	**korea**	kill	final	medic	million	al	win	medic	win	**syria**
play	univers	race	launch	**war**	**inning**	**zika**	power	secur	**home**	reuter	de	**russia**
third	found	play	ad	**iran**	score	report	plant	**syria**	team	risk	final	**iran**
score	caus	top	network	**syria**	player	percent	climat	offici	time	report	play	countri
win	hospit	event	technolog	peopl	lead	risk	**govern**	kill	left	cancer	race	**saudi**
sunday	care	game	includ	foreign	day	peopl	project	**russia**	**pitch**	diseas	report	**china**

Note that many words are truncated to their roots by stemming in the pre-processing stage: this ensures that closely related words, e.g., 'olympic', 'olympics' and 'olympiad', are assigned to the same token value. A nominal label, under each topic index, has been assigned to each topic to suggest similarities.

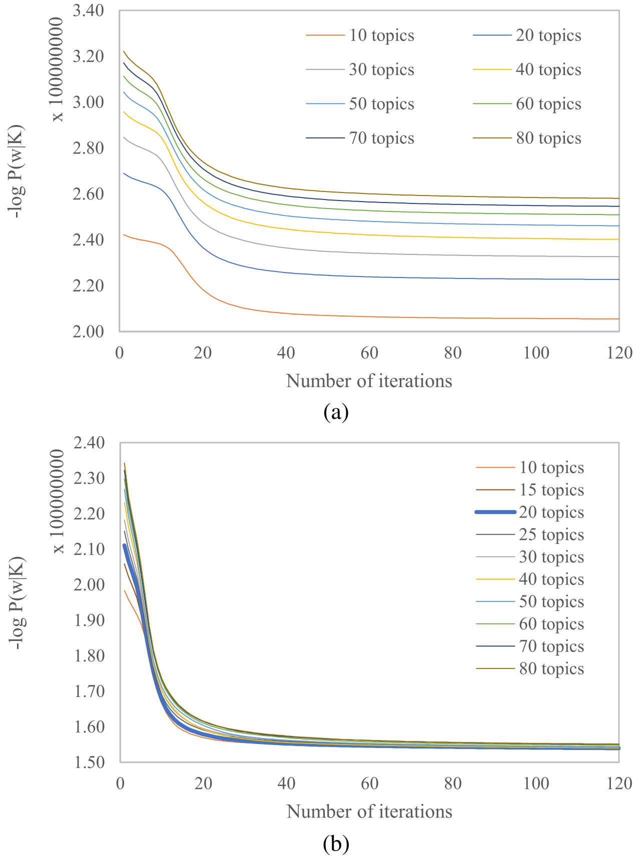


FIGURE 8. The log-likelihood after running the Gibbs Sampling algorithm for 1 to 120 iterations. (a) LDA. (b) d-FinLDA.

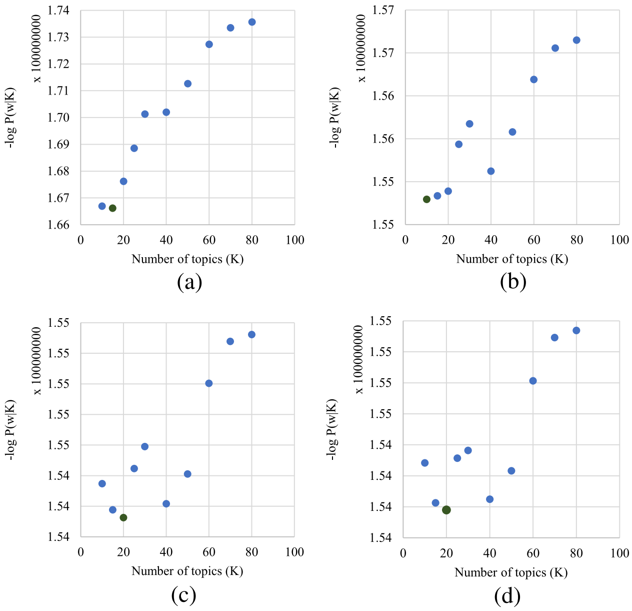


FIGURE 9. The log-likelihood of the data for different numbers of topics at different iterations when using d-FinLDA. (a) 10 iterations. (b) 50 iterations. (c) 100 iterations. (d) 120 iterations.

that the data are best matched when $K \sim 20$ and $n_{iter} \gtrsim 100$. Accordingly, we set $K = 20$ and $n_{iter} = 120$ for LDA, d-FinLDA and c-FinLDA.

After 120 iterations with the training dataset, some topics extracted from three different models are almost the same in their implicit meaning, i.e., the set of top words, ranked by their probabilities, are similar, but the orders of the ranking, implying their importance, are different. Furthermore, some topics from d-FinLDA and c-FinLDA seems to be a combination of 2 topics from LDA, some topics from d-FinLDA and c-FinLDA seems to be split from only one topics from LDA, and some topics are totally different.

We show an example of topics with their top 15 vocabulary words ranked by their probabilities from LDA, d-FinLDA and c-FinLDA in Table 1 to show the differences of topics from the three models. There are only four instead of five topics from both d-FinLDA and c-FinLDA in the table to show an example that a topic in d-FinLDA and a topic in c-FinLDA look like a combination of two topics from LDA.

Additionally, we bold the words that might be helpful to interpret the meaning of topic, leading to the nominal labels that we assigned to each topic, i.e., 'Baseball', 'Health', 'Olympics', 'China-Korea' and 'Mid-East'. We set the same color to the same word to show their duplicates with a top word in the topic extracted by LDA, and left it black if there is no duplicate word. Topic 1, 'Baseball', has similar sets of words in each case as colored in cyan, although an important word, i.e., "home", was not ranked in the top 15 words in any topic from d-FinLDA. However, d-FinLDA and c-FinLDA ranked "inning" and "pitch" more highly which clarifies that the topic is about baseball. In topic 2, 'Health', including drug research and the Zika virus, has similar, but not identical, lists in the three methods. However, topic 3, the Olympic Games in Rio de Janeiro, from LDA looks like it were combined with 'Health' topic and became topic 2 from d-FinLDA which seems to be a topic about Zika concern at the Olympic Games in Rio. Furthermore, topic 4 and 5 from LDA show a bit vague meaning about many countries and seem to be combined into topic 4 from c-FinLDA as colored in blue and magenta. For another example, not shown in the table, a topic about Trump from LDA was spread to two topics from c-FinLDA.

Practically, the interpretation and meaning of topics in this experiment is not necessary because the topics, which are word distributions, were then used to infer topic distribution from text. The topic distribution was considered as input features for a machine learning algorithm without any requirement to understand the meaning of those features. However, the different topics from three different models in the example show that changes from financial time series successfully affected word distribution of each topic in the parameter estimation. The benefit from the effect to the prediction is evaluated in the next phase.

In addition to the estimated word distributions for inference, we also got topic distributions of all documents in the training set from the parameter estimation process. The topic distributions were then used as input features to train SVR and BPNN in the next phase. We used estimated $\boldsymbol{\beta}$ values, affected by 5-min changes of S&P 500 in the parameter estimation process for FinLDA, and all words in the documents in the test set, 38,460 lists of word indices, $w_{d,n}$, to get the topic distributions of the documents in the test set. The topic distributions were then used as input features for SVR and BPNN for testing in the next phase. However, some topic distributions, from both training and test sets, needed another processing because they were the topic distributions of the documents that were published at the same time (on a minute scale). Accordingly, we averaged the topic distributions of the documents, published at the same time, and derived 67,553 topic distributions from the training set and 22,518 topic distributions from the test set.

After getting all the prepared data, we arranged them into 4 sets of features for both training and test sets, i.e., SP500, LDA&SP500, d-FinLDA&SP500 and c-FinLDA&SP500.

TABLE 2. R-squared from SVR-RBF with base features as input, with $C \in [10^{-1}, 10^6]$ and $g \in [10^{-7}, 10^{-2}]$ in the out-of-time validation.

g C	10^{-7}	10^{-6}	10^{-5}	10^{-4}	10^{-3}	10^{-2}
10^{-1}	-23.393	-21.944	-9.866	0.824	0.842	0.373
1	-21.946	-9.865	0.828	0.874	0.863	0.394
10^1	-9.859	0.828	0.878	0.895	0.875	0.385
10^2	0.826	0.878	0.899	0.913	0.899	0.385
10^3	0.877	0.897	0.915	0.935	0.899	0.385
10^4	0.505	0.917	**0.938**	0.935	0.899	0.385
10^5	0.564	0.937	**0.938**	0.935	0.899	0.385
10^6	-231.253	0.937	**0.938**	0.935	0.899	0.385

C. MODELING AND EVALUATION

As we concentrated on feature extraction in a data preparation phase, two conventional algorithms, i.e., SVR and BPNN, were applied to validating the benefits of FinLDA. To comply with our goal, we compared the results when using the four different sets of features in SVR (class SVR [47] based on LIBSVM [48]) with RBF kernel and BPNN (MLPRegressor [49]) in the scikit-learn library [50], [51] by using four measurements, i.e., R^2, MSE, MAE and MAD. Both classes in scikit-learn compute a prediction scoring metric, i.e., the coefficient of determination, R^2. We used scikit-learn to calculate MSE [52] and MAE [53] and used the mad function [54], in class DataFrame in pandas [55], to calculate MAD. We also show predicted/actual scatter plot with an ideal fit line and Regression Error Characteristics curves [56]–[58].

1) SVR-RBF

Initially, as we intended to compare the benefit derived from the different sets of features when using the same algorithm in a modeling phase, we used the default parameter values of SVR with default kernel, RBF, (SVR-RBF) in scikit-learn [47] for fair comparison. However, the results from all four sets of features were extremely poor. So, we gave the most advantage to the base features by tuning a suitable value of C and g (gamma) in SVR-RBF, based on trials with features from 2 years of SP500 only, which are the base features for this experiment.

We tested the combination of C and g in the range of 10^{-10} to 10^{10} for both parameters in SVR-RBF, but we show only the results from the small range of C and g that are not far from the best C and g in Table 2. The results show that $g \approx 10^{-5}$ and $C \geq 10^4$ are optimum for SVR-RBF when using our base features, SP500 only. Accordingly, for SVR-RBF in this experiment, we set $C = 10^4$, $g = 10^{-5}$, with default values for the remaining parameters. Then, we used SVR-RBF with the datasets that were split by the two approaches of backtesting in the data preparation phase, as described in IV-B.

a: RESULTS OF SVR-RBF IN THE OUT-OF-TIME VALIDATION

The results when using the simple train-test split are shown in Fig. 10 to compare between the actual price of S&P 500 and

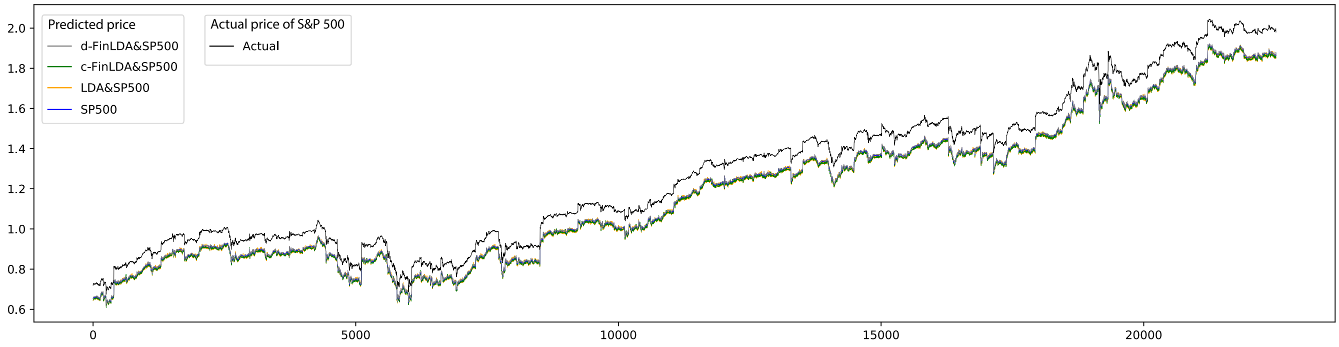
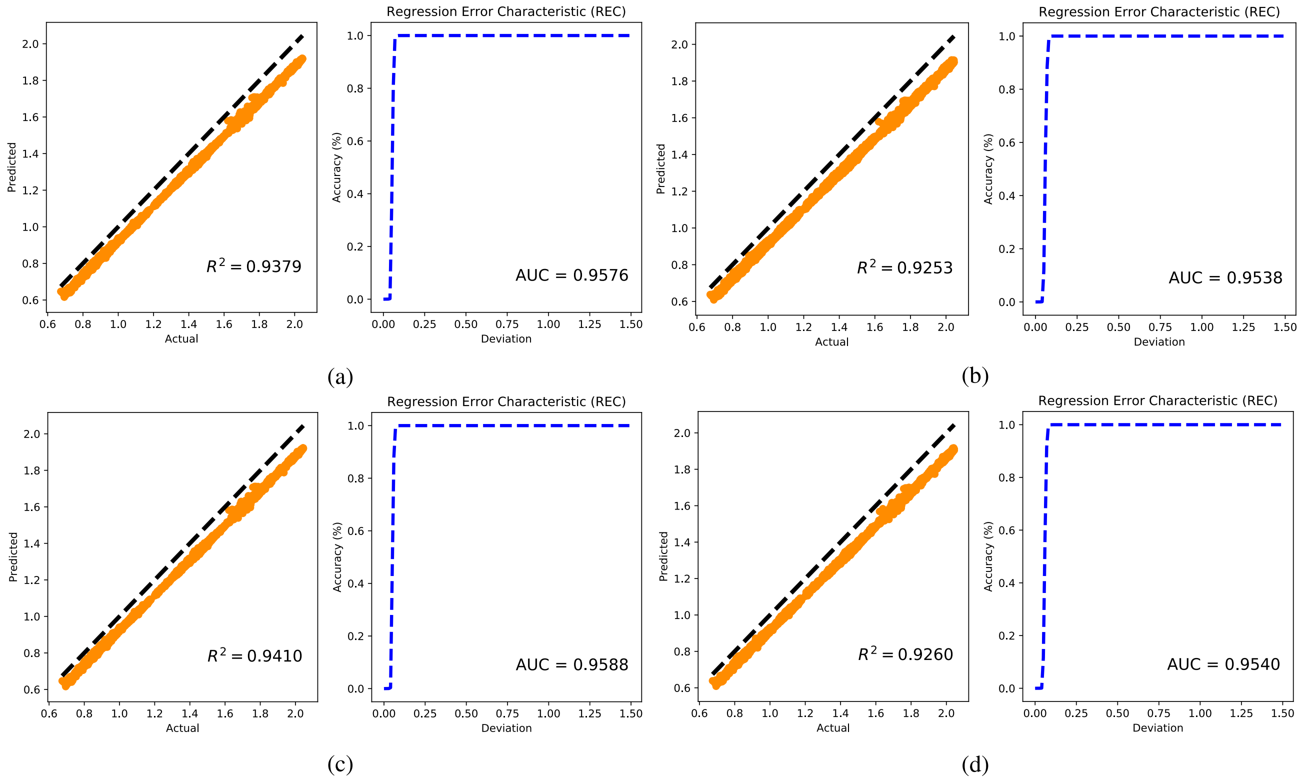


FIGURE 10. The comparison between actual and predicted results from SVR-RBF when using different sets of features in the out-of-time validation.



(a)

(b)

(c)

(d)

FIGURE 11. The predicted/actual scatter plot and REC curve from SVR-RBF in the out-of-time validation. (a) SP500. (b) LDA&SP500. (c) d-FinLDA&SP500. (d) c-FinLDA&SP500.

TABLE 3. Metrics of SVR-RBF when using the simple train-test split from 2-year dataset in the out-of-time validation.

Features	R^2	MSE	MAE	MAD
SP500	0.9378	0.0085	0.0904	**0.0146**
LDA&SP500	0.9253	0.0102	0.0991	0.0161
d-FinLDA&SP500	**0.9409**	**0.0081**	**0.0881**	**0.0146**
c-FinLDA&SP500	0.9260	0.0101	0.0988	0.0157

the predicted value from SVR-RBF when using four different sets of features. The gray line in Fig. 10, which is a bit closer to black line of the actual price than the other lines of predicted prices, shows that d-FinLDA gave a bit additional benefit to the prediction. We compare the benefit among the four different sets of features by using the measurements in Table 3. The coefficient of determination and the other metrics in Table 3 show that the performance from SVR-RBF was the best when we used the combination of features from d-FinLDA and SP500 (d-FinLDA&SP500) as input features. Although Fig. 11 shows only a bit different performance among the four sets of features, AUC was the best when using SVR-RBF with d-FinLDA&SP500.

b: RESULTS OF SVR-RBF IN THE WALK-FORWARD VALIDATION

The results from SVR-RBF when using 10 datasets from the walk-forward testing approach are shown separately for

TABLE 4. The coefficient of determination of SVR-RBF when using each of 10 overlapping datasets in the walk-forward testing.

Step	1	2	3	4	5	6	7	8	9	10	Avg	SD
SP500	0.9626	0.9952	0.9761	0.9900	0.9960	0.9858	0.9776	0.9954	0.9956	0.9571	0.9831	0.0143
LDA&SP500	0.9614	0.9951	0.9757	0.9906	0.9955	**0.9898**	0.9783	0.9952	0.9954	0.9530	0.9830	0.0155
d-FinLDA&SP500	**0.9661**	0.9953	**0.9764**	**0.9934**	**0.9973**	0.9881	0.9793	**0.9955**	**0.9961**	**0.9665**	**0.9854**	**0.0124**
c-FinLDA&SP500	0.9358	**0.9955**	0.9721	0.9720	0.9959	0.9027	**0.9831**	0.9941	0.9938	0.9239	0.9669	0.0340

TABLE 5. MSE of SVR-RBF when using each of 10 overlapping datasets in the walk-forward testing.

Step	1	2	3	4	5	6	7	8	9	10	Avg	SD
SP500	0.0021	0.0111	0.0027	0.0023	0.0012	0.0011	0.0050	0.0042	0.0009	0.0011	0.0032	0.0031
LDA&SP500	0.0022	0.0116	0.0028	0.0022	0.0013	**0.0008**	0.0049	0.0044	0.0010	0.0012	0.0032	0.0032
d-FinLDA&SP500	**0.0019**	0.0110	**0.0026**	**0.0016**	**0.0008**	0.0009	0.0046	**0.0041**	**0.0008**	**0.0009**	**0.0029**	0.0032
c-FinLDA&SP500	0.0036	**0.0107**	0.0032	0.0066	0.0012	0.0075	**0.0038**	0.0054	0.0013	0.0020	0.0045	**0.0030**

TABLE 6. MAE of SVR-RBF when using each of 10 overlapping datasets in the walk-forward testing.

Step	1	2	3	4	5	6	7	8	9	10	Avg	SD
SP500	0.0349	0.0776	0.0383	0.0434	0.0274	0.0282	0.0518	0.0473	0.0225	0.0259	0.0397	**0.0165**
LDA&SP500	0.0333	0.0796	0.0388	0.0402	0.0287	**0.0225**	0.0507	0.0488	0.0233	0.0263	0.0392	0.0173
d-FinLDA&SP500	**0.0282**	0.0771	**0.0382**	**0.0342**	**0.0211**	0.0254	0.0492	**0.0471**	**0.0210**	**0.0221**	**0.0364**	0.0177
c-FinLDA&SP500	0.0515	**0.0763**	0.0418	0.0758	0.0278	0.0831	**0.0456**	0.0547	0.0274	0.0326	0.0517	0.0207

TABLE 7. MAD of SVR-RBF when using each of 10 overlapping datasets in the walk-forward testing.

Step	1	2	3	4	5	6	7	8	9	10	Avg	SD
SP500	**0.0255**	0.0777	0.0384	**0.0190**	**0.0201**	**0.0154**	0.0478	**0.0463**	**0.0209**	**0.0211**	**0.0332**	0.0195
LDA&SP500	0.0304	0.0793	0.0387	0.0231	0.0224	0.0159	0.0496	0.0474	0.0217	0.0240	0.0352	0.0192
d-FinLDA&SP500	0.0256	0.0768	**0.0381**	0.0191	0.0203	0.0156	0.0476	0.0464	0.0210	0.0219	**0.0332**	0.0192
c-FinLDA&SP500	0.0256	**0.0761**	0.0417	0.0228	0.0227	0.0196	**0.0432**	0.0544	0.0249	0.0326	0.0364	**0.0179**

TABLE 8. Average measurements of SVR-RBF and their standard deviation (in parentheses) when using 10 overlapping datasets in the walk-forward testing.

Features	R^2	MSE	MAE	MAD
SP500	0.9831 (0.0143)	0.0032 (0.0031)	0.0397 (0.0165)	**0.0332** (0.0195)
LDA&SP500	0.9830 (0.0155)	0.0032 (0.0032)	0.0392 (0.0173)	0.0352 (0.0192)
d-FinLDA&SP500	**0.9854** (0.0124)	**0.0029** (0.0032)	**0.0364** (0.0177)	**0.0332** (0.0192)
c-FinLDA&SP500	0.9669 (0.0340)	0.0045 (0.0030)	0.0517 (0.0207)	0.0364 (0.0179)

each measurement in Table 4 – 7 and plotted in graphs in Fig. 12. The average results from SVR-RBF when using 10 overlapping datasets are shown in Table 8. The results in Table 4 – 6 show that the prediction when using the combination of features from d-FinLDA and SP500 (d-FinLDA&SP500) as input features were better than that when using SP500 alone for all 10 datasets and were the best for 7 out of 10 datasets. MADs in Table 7 show that the results from SVR-RBF when using SP500 alone were better in some datasets but the results are only one last digit different from d-FinLDA&SP500. Furthermore, the average MADs from both of them are the same. As shown in Table 8, the average performance from SVR-RBF was the best when we used the combination of features from d-FinLDA and SP500 (d-FinLDA&SP500) as input features.

Even though we adjusted C and g parameters to be optimum for SVR-RBF when using SP500 alone, the performance from SVR-RBF when using the combination of features from d-FinLDA and SP500 (d-FinLDA&SP500) as input features was still better than that when using SP500 alone. Accordingly, the additional features from d-FinLDA give some value to the prediction when using SVR-RBF. However, SVR-RBF seems to be able to get only little benefit from our FinLDA.

2) BPNN

MLPRegressor in the scikit-learn library implements BPNN and uses the square error as the loss function. Its output is a set of continuous values. In class MLPRegressor, we used the default values of all parameters [49], except the shuffle parameter, because a 'false' for the shuffle parameter is appropriate for time series prediction. We repeated the experiment 1,000 times to obtain average R^2, MSE, MAE and MAD values.

a: RESULTS OF BPNN IN THE OUT-OF-TIME VALIDATION

For the first approach of backtesting with the simple train-test split from 2-years dataset, Table 9 shows that the average performance of BPNN when using the combination of features from c-FinLDA and SP500 (c-FinLDA&SP500) was 5.5% better than that when using SP500 alone and 4.1% better than that when using the combination of features from LDA and

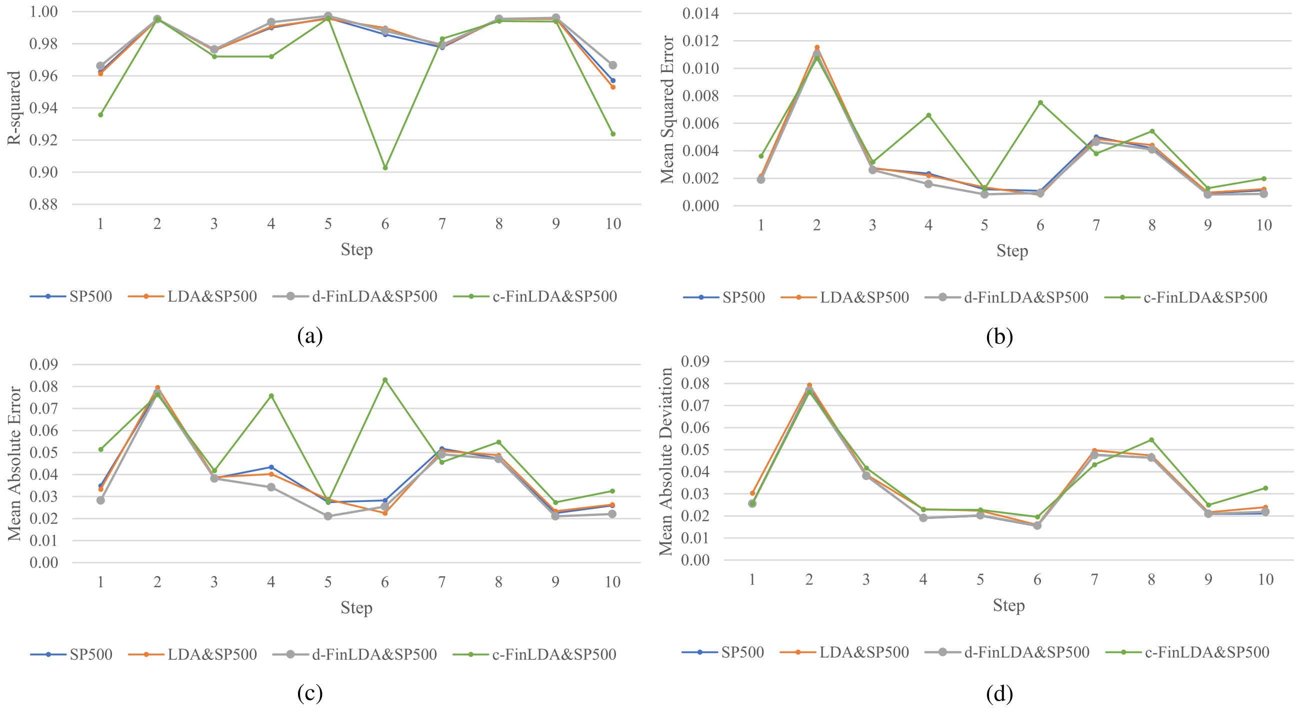


FIGURE 12. The measurements of SVR-RBF when using each of 10 overlapping datasets in the walk-forward testing. (a) R-squared. (b) MSE. (c) MAE. (d) MAD.

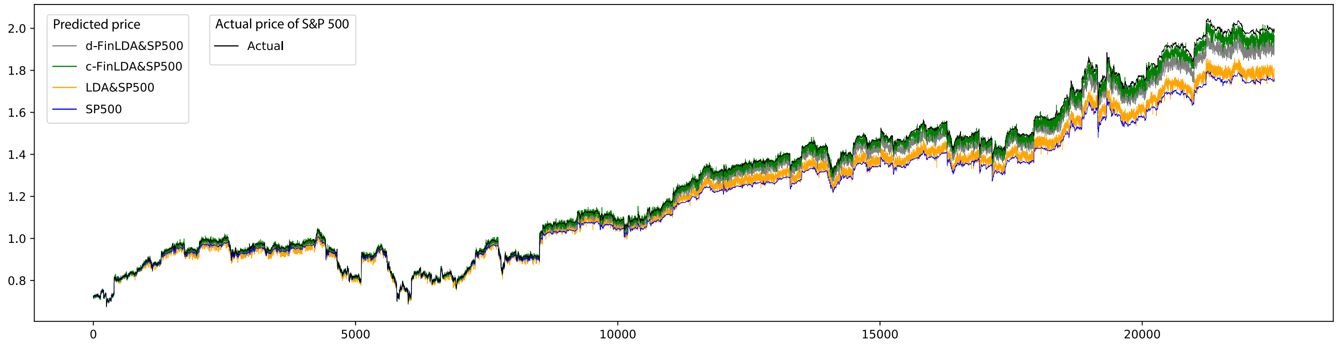


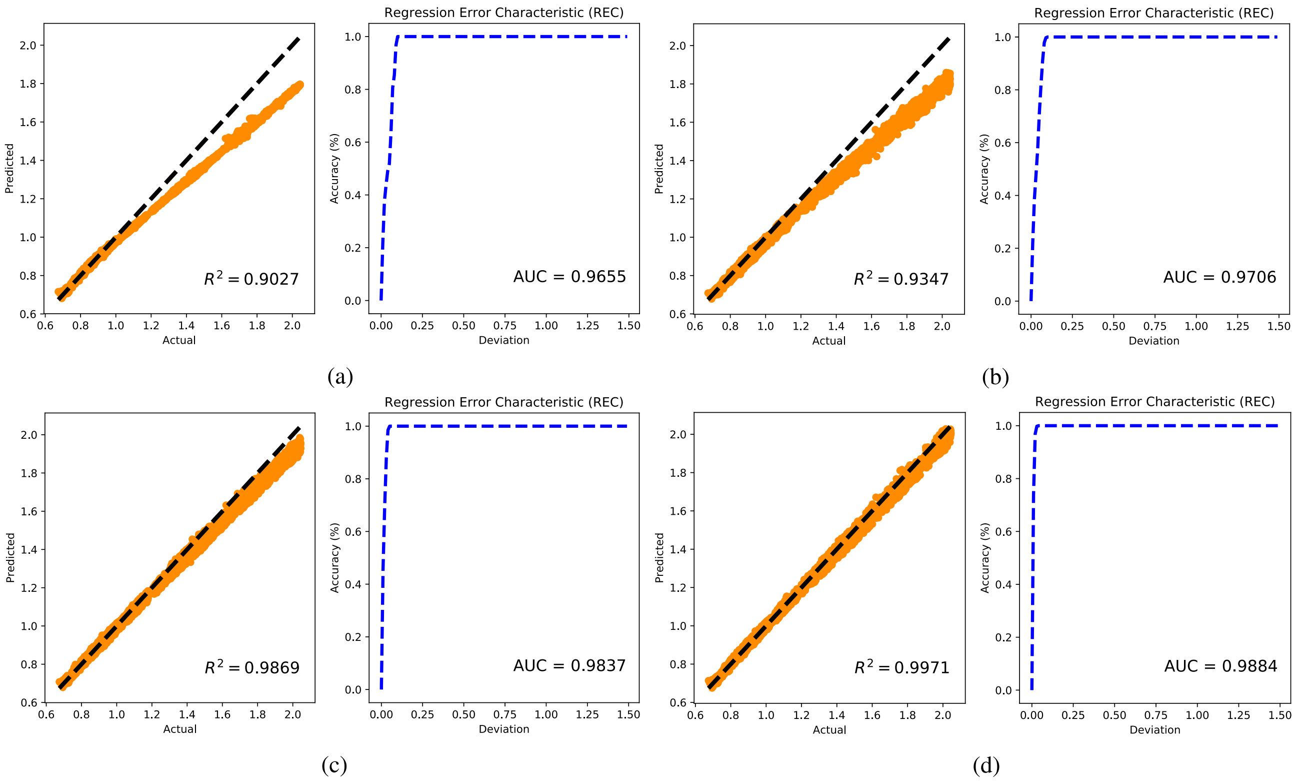
FIGURE 13. The comparison between actual and predicted results from BPNN when using different sets of features in the out-of-time validation (an example result from one of a thousand times of the experiment).

SP500 (LDA&SP500). The average performance of BPNN when using d-FinLDA&SP500 was only a bit lower than that when using c-FinLDA&SP500. The standard deviation (SD) of R^2 when using SP500 alone is the worst and 630% worse than that when using c-FinLDA&SP500. Even though SD of R^2 when using LDA&SP500 is better than that when using SP500 alone, it is still a lot worse than those when using c-FinLDA&SP500 and when using d-FinLDA&SP500. The average MSE from d-FinLDA&SP500, 0.0017, and c-FinLDA&SP500, 0.0014, are significantly better than those from SP500 alone, 0.0085, and LDA&SP500, 0.0069. The average MAE and MAD also show the same results - see Table 9. Thus, the additional features from LDA gave only a little value added to the prediction but the additional features from d-FinLDA and c-FinLDA clearly improved the prediction.

TABLE 9. Average measurements of BPNN and their standard deviation (in parentheses) when using the simple train-test split from 2-year dataset.

Features	R^2	MSE	MAE	MAD
SP500	0.9379 (0.0628)	0.0085 (0.0086)	0.0612 (0.0328)	0.0463 (0.0247)
LDA & SP500	0.9498 (0.0451)	0.0069 (0.0061)	0.0560 (0.0256)	0.0428 (0.0191)
d-FinLDA & SP500	0.9876 (0.0127)	0.0017 (0.0017)	0.0266 (0.0134)	0.0219 (0.0105)
c-FinLDA & SP500	**0.9894** (0.0100)	**0.0014** (0.0013)	**0.0246** (0.0119)	**0.0204** (0.0093)

As we experimented BPNN 1,000 times to get average results, we picked an example of predicted results from one of a thousand experiments to show the differences between actual price of S&P 500 and the predicted values by BPNN when using different sets of features in Fig. 13. The example



(a)

(b)

(c)

(d)

FIGURE 14. The predicted/actual scatter plot and REC curve from BPNN in the out-of-time validation (an example result from one of a thousand times of the experiment). (a) SP500. (b) LDA&SP500. (c) d-FinLDA&SP500. (d) c-FinLDA&SP500.

TABLE 10. Average of the coefficient of determination of BPNN and their standard deviation (in parentheses) when using each of 10 overlapping datasets in the walk-forward testing.

Step	1	2	3	4	5	6	7	8	9	10	Avg	SD
SP500	0.9554 (0.0180)	0.9926 (0.0028)	**0.9733** (0.0017)	0.8725 (0.1130)	0.7162 (0.2256)	0.8847 (0.1041)	0.9683 (0.0117)	0.9917 (0.0035)	0.9869 (0.0086)	**0.9623** (0.0059)	0.9304	0.0863
LDA&SP500	0.9390 (0.0198)	0.9908 (0.0036)	0.9700 (0.0022)	0.8615 (0.1043)	0.6206 (0.2577)	0.8659 (0.0956)	0.9627 (0.0134)	0.9904 (0.0032)	0.9821 (0.0083)	0.9381 (0.0132)	0.9121	0.1127
d-FinLDA&SP500	**0.9570** (0.0115)	0.9934 (0.0020)	0.9722 (0.0015)	0.9270 (0.0665)	0.7937 (0.1781)	0.9323 (0.0519)	0.9702 (0.0087)	**0.9927** (0.0021)	**0.9890** (0.0057)	0.9556 (0.0067)	0.9483	0.0591
c-FinLDA&SP500	**0.9570** (0.0106)	**0.9937** (0.0017)	0.9684 (0.0017)	**0.9312** (0.0631)	**0.8178** (0.1513)	**0.9372** (0.0512)	**0.9742** (0.0090)	0.9923 (0.0016)	0.9862 (0.0054)	0.9482 (0.0069)	**0.9506**	**0.0516**

in the figure shows that the predicted values from BPNN with SP500 and LDA&SP500 departed from the actual value when the time passed, while the predicted values from BPNN with c-FinLDA&SP500 and d-FinLDA&SP500 were still close to the actual price. The example result is also shown in the predicted/actual scatter plot and Regression Error Characteristic curve in Fig. 14. Both Fig. 13 and Fig. 14 illustrate that the performances of BPNN when using c-FinLDA&SP500 and d-FinLDA&SP500 were a lot better than those when using SP500 and LDA&SP500.

b: RESULTS OF BPNN IN THE WALK-FORWARD VALIDATION

For the second approach of backtesting with the walk-forward testing from 10 overlapping of 6-month dataset, we also repeated BPNN 1,000 times to get average results from each dataset and show the average measurements in Table 10 – 13 and plot them in Fig. 15. Table 10 shows that the average performances of BPNN when using c-FinLDA&SP500 and d-FinLDA&SP500 were better than that when using SP500 in 8 out of 10 steps of walk-forward testing, and c-FinLDA&SP500 was the best in 6 out of 10 steps. Table 11 and 12 show that the average performance of BPNN when using d-FinLDA&SP500 was better than that when using SP500 in 8 out of 10 steps of walk-forward testing, and c-FinLDA&SP500 was better than SP500 in 7 out of 10 steps. Table 13 shows that the average performance of BPNN when using d-FinLDA&SP500 was better than that when using SP500 in 7 out of 10 steps of walk-forward testing, and c-FinLDA&SP500 was better than SP500 in 5 out of 10 steps. Additionally, the graphs in Fig. 15 display a better angle of the comparison that the benefit of c-FinLDA and d-FinLDA was appeared distinctly when only features from S&P 500 were not good enough. Furthermore, the worse

TABLE 11. Average MSE of BPNN and their standard deviation (in parentheses) when using each of 10 overlapping datasets in the walk-forward testing.

Step	1	2	3	4	5	6	7	8	9	10	Avg	SD
SP500	0.0025 (0.0010)	0.0172 (0.0066)	**0.0030** (0.0002)	0.0300 (0.0266)	0.0862 (0.0686)	0.0089 (0.0080)	0.0071 (0.0026)	0.0076 (0.0032)	0.0027 (0.0018)	**0.0010** (0.0002)	0.0166	0.0260
LDA&SP500	0.0034 (0.0011)	0.0216 (0.0084)	0.0034 (0.0003)	0.0326 (0.0246)	0.1153 (0.0783)	0.0103 (0.0074)	0.0084 (0.0030)	0.0088 (0.0030)	0.0037 (0.0017)	0.0016 (0.0003)	0.0209	0.0346
d-FinLDA&SP500	**0.0024** (0.0006)	0.0156 (0.0047)	0.0032 (0.0002)	0.0172 (0.0157)	0.0627 (0.0541)	0.0052 (0.0040)	0.0067 (0.0019)	0.0067 (0.0020)	**0.0023** (0.0012)	0.0012 (0.0002)	0.0123	0.0185
c-FinLDA&SP500	**0.0024** (0.0006)	**0.0148** (0.0040)	0.0036 (0.0002)	**0.0162** (0.0148)	**0.0554** (0.0460)	**0.0048** (0.0039)	**0.0058** (0.0020)	**0.0071** (0.0015)	0.0029 (0.0011)	0.0013 (0.0002)	**0.0114**	**0.0163**

TABLE 12. Average MAE of BPNN and their standard deviation (in parentheses) when using each of 10 overlapping datasets in the walk-forward testing.

Step	1	2	3	4	5	6	7	8	9	10	Avg	SD
SP500	0.0370 (0.0099)	0.0971 (0.0178)	**0.0414** (0.0017)	0.1431 (0.0710)	0.2174 (0.0945)	0.0655 (0.0307)	0.0642 (0.0128)	0.0673 (0.0142)	0.0380 (0.0112)	**0.0234** (0.0022)	0.0794	0.0597
LDA&SP500	0.0447 (0.0089)	0.1099 (0.0204)	0.0443 (0.0019)	0.1523 (0.0632)	0.2569 (0.0943)	0.0739 (0.0269)	0.0703 (0.0129)	0.0735 (0.0123)	0.0457 (0.0094)	0.0307 (0.0034)	0.0902	0.0689
d-FinLDA&SP500	0.0360 (0.0066)	0.0922 (0.0134)	0.0424 (0.0014)	0.1067 (0.0529)	0.1826 (0.0845)	0.0518 (0.0196)	0.0625 (0.0100)	**0.0636** (0.0099)	**0.0359** (0.0082)	0.0258 (0.0021)	0.0700	0.0471
c-FinLDA&SP500	**0.0356** (0.0062)	**0.0902** (0.0118)	0.0448 (0.0014)	**0.1031** (0.0511)	**0.1719** (0.0778)	**0.0495** (0.0196)	**0.0582** (0.0103)	0.0644 (0.0075)	0.0402 (0.0073)	0.0276 (0.0020)	**0.0685**	**0.0434**

TABLE 13. Average MAD of BPNN and their standard deviation (in parentheses) when using each of 10 overlapping datasets in the walk-forward testing.

Step	1	2	3	4	5	6	7	8	9	10	Avg	SD
SP500	**0.0328** (0.0068)	0.0951 (0.0167)	**0.0402** (0.0012)	0.0475 (0.0186)	0.1516 (0.0645)	0.0472 (0.0203)	0.0575 (0.0088)	0.0599 (0.0108)	0.0364 (0.0121)	**0.0229** (0.0017)	0.0591	0.0380
LDA&SP500	0.0403 (0.0063)	0.1072 (0.0196)	0.0436 (0.0017)	0.0558 (0.0160)	0.1788 (0.0647)	0.0530 (0.0172)	0.0628 (0.0093)	0.0649 (0.0089)	0.0433 (0.0096)	0.0305 (0.0034)	0.0680	0.0442
d-FinLDA&SP500	0.0336 (0.0048)	0.0901 (0.0122)	0.0418 (0.0011)	**0.0432** (0.0129)	0.1287 (0.0587)	0.0389 (0.0130)	0.0572 (0.0071)	**0.0580** (0.0069)	**0.0350** (0.0087)	0.0254 (0.0019)	**0.0552**	0.0316
c-FinLDA&SP500	0.0336 (0.0046)	**0.0885** (0.0106)	0.0442 (0.0011)	0.0434 (0.0124)	**0.1234** (0.0543)	**0.0386** (0.0130)	**0.0531** (0.0075)	0.0613 (0.0051)	0.0380 (0.0076)	0.0273 (0.0019)	**0.0552**	**0.0296**

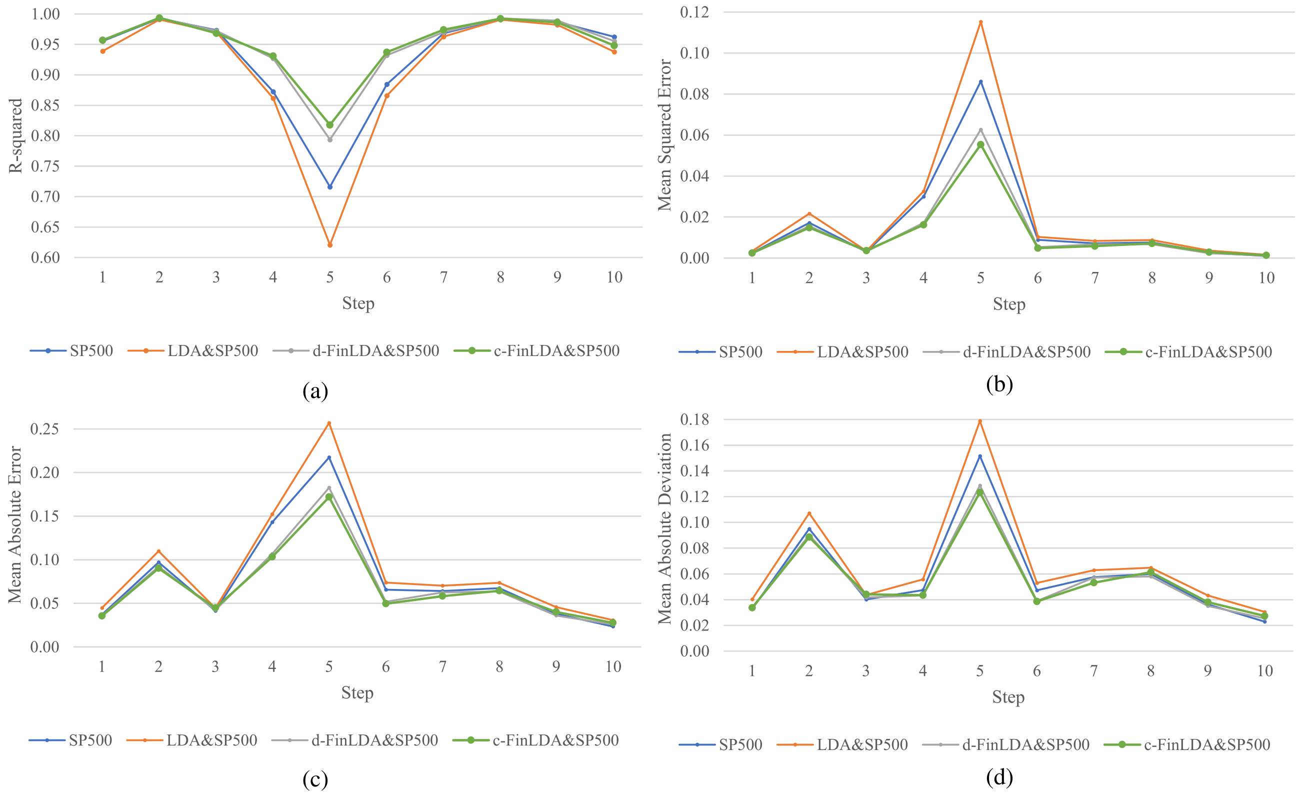
TABLE 14. Average measurements of BPNN and their standard deviation (in parentheses) when using 10 overlapping datasets in the walk-forward testing.

Features	R^2	MSE	MAE	MAD
SP500	0.9304 (0.0863)	0.0166 (0.0260)	0.0794 (0.0597)	0.0591 (0.0380)
LDA&SP500	0.9121 (0.1127)	0.0209 (0.0346)	0.0902 (0.0689)	0.0680 (0.0442)
d-finLDA&SP500	0.9483 (0.0591)	0.0123 (0.0185)	0.0700 (0.0471)	**0.0552** (0.0316)
c-finLDA&SP500	**0.9506** (0.0516)	**0.0114** (0.0163)	**0.0685** (0.0434)	**0.0552** (0.0296)

performance of BPNN when using SP500, the more the additional features from LDA decrease the performance of BPNN. Besides, the average performance from 10 datasets in Table 14 shows that the features from c-FinLDA&SP500 are the best, followed by the features from d-FinLDA&SP500, and the worst is LDA&SP500. Accordingly, the final results from BPNN empirically show the benefit of both d-FinLDA and c-FinLDA in data mining for financial time series prediction.

As the results from LDA&SP500 are worse than the results from d-FinLDA&SP500 and c-FinLDA&SP500 from BPNN, it implies that the additional features from FinLDA are better than the additional features from LDA. Accordingly, taking changes of financial time series into account in FinLDA can make the features better for the financial time series prediction than the features from LDA. Moreover, as the results from LDA&SP500 are worse than the results from SP500 alone from SVR-RBF and only a bit better than the results from SP500 alone from BPNN when using the first approach of backtesting, and worse than the results from SP500 alone from both SVR-RBF and BPNN when using the second approach of backtesting, the features extracted from text by using the model that is trained by text alone (LDA) do not seem to give any advantage to the financial prediction. On the contrary, the results from d-FinLDA&SP500 and c-FinLDA&SP500 are better than the results from SP500 alone from BPNN. It implies that taking changes of financial time series into account in FinLDA can make the normal text features become the features for financial time series prediction, esp. with BPNN.

In summary, our features from d-FinLDA and c-FinLDA empirically gave value added to the prediction when they were used in BPNN and our features from d-FinLDA empirically gave a bit value added to the prediction when they were used in SVR with RBF kernel.



(a)

(b)

(c)

(d)

FIGURE 15. The measurements of BPNN when using each of 10 overlapping datasets in the walk-forward testing. (a) R-squared. (b) MSE. (c) MAE. (d) MAD.

Moreover, the differences between the benefits from LDA and FinLDA show that incorporation of changes in financial time series in FinLDA improves features for the prediction.

V. CONCLUSIONS

We introduced FinLDA to extract better features from news articles for the prediction. This FinLDA is an extension of the Latent Dirichlet Allocation model which takes changes in financial time series into account. The extracted features can be used in any machine learning algorithm to predict financial results. In our experiment, parameters of our two FinLDA variants (one with discrete input data and the other with continuous variables describing changes) were estimated by using both news articles from Reuters and Standard & Poor's 500 Index data and the final outputs from the two FinLDA variants were used as input features in two conventional machine learning algorithms, i.e., SVR and BPNN, to validate the benefit of the features from FinLDA when comparing with other features. Although adding FinLDA resulted in only minor changes with SVR, FinLDA gave some value to the prediction. Additionally, BPNN was significantly better with FinLDA showing 5- to 6-fold drops in MSE in predictions. Accordingly, our features from FinLDA empirically give value added to the prediction when they are used in both BPNN and SVR.

As here is the first article in which we theoretically established and formalized the FinLDA, we therefore focused on the explanation of FinLDA in data preparation phase and conducted the initial experiment to show the benefits of the features from FinLDA applied in two conventional machine learning algorithms. Our future work will be on the comprehensive experiment to squeeze more value out of the features from FinLDA by focusing on other advanced machine learning algorithms, e.g., XGBoost, LightGBM, etc., in a modeling phase as well as on hyperparameter tuning.

REFERENCES

[1] E. F. Fama, "Efficient capital markets: A review of theory and empirical work," *J. Finance*, vol. 25, no. 2, pp. 383–417, May 1970.

[2] E. F. Fama, "Efficient capital markets: II," *J. Finance*, vol. 46, no. 5, pp. 1575–1617, 1991.

[3] R. J. Shiller, "Speculative prices and popular models," *J. Econ. Perspect.*, vol. 4, no. 2, pp. 55–65, 1990.

[4] L. Blume, D. Easley, and M. O'Hara, "Market statistics and technical analysis: The role of volume," *J. Finance*, vol. 49, no. 1, pp. 153–181, 1994.

[5] M. T. Leung, H. Daouk, and A.-S. Chen, "Forecasting stock indices: A comparison of classification and level estimation models," *Int. J. Forecasting*, vol. 16, no. 2, pp. 173–190, 2000.

[6] H. Ince and T. B. Trafalis, "Kernel principal component analysis and support vector machines for stock price prediction," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, vol. 3, Jul. 2004, pp. 2053–2058. doi: 10.1109/IJCNN.2004.1380933.

[7] R. S. Tsay, *Analysis of Financial Time Series* (Series in Probability and Statistics), 2nd ed. Hoboken, NJ, USA: Wiley, 2005.

[8] C.-F. Tsai and Y.-C. Hsiao, "Combining multiple feature selection methods for stock prediction: Union, intersection, and multi-intersection approaches," *Decis. Support Syst.*, vol. 50, no. 1, pp. 258–269, 2010.

[9] B. Wüthrich, D. Permunetilleke, S. Leung, V. Cho, J. Zhang, and W. Lam, "Daily prediction of major stock indices from textual www data," in *Proc. 4th Int. Conf. Knowl. Discovery Data Mining (KDD)*, 1998, pp. 364–368. [Online]. Available: http://dl.acm.org/citation.cfm?id=3000292.3000361

[10] W. S. Chan, "Stock price reaction to news and no-news: Drift and reversal after headlines," *J. Financial Econ.*, vol. 70, no. 2, pp. 223–260, 2003.

[11] W. F. M. De Bondt and R. Thaler, "Does the stock market overreact?" *J. Finance*, vol. 40, no. 3, pp. 793–805, Jul. 1985. [Online]. Available: http://www.jstor.org/stable/2327804

[12] R. J. Shiller, "From efficient markets theory to behavioral finance," *J. Econ. Perspect.*, vol. 17, no. 1, pp. 83–104, 2003.

[13] G. P. C. Fung, J. X. Yu, and W. Lam, "News sensitive stock trend prediction," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*. Berlin, Germany: Springer-Verlag, 2002, pp. 481–493.

[14] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval," *J. Document.*, vol. 28, no. 1, pp. 11–21, 1972.

[15] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab.*, 1967, pp. 281–297.

[16] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[17] F. Jin, N. Self, P. Saraf, P. Butler, W. Wang, and N. Ramakrishnan, "Forex-foreteller: Currency trend modeling using news articles," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*. New York, NY, USA: ACM, 2013, pp. 1470–1473.

[18] S. Feuerriegel, A. Ratku, and D. Neumann, "Analysis of how underlying topics in financial news affect stock prices using latent Dirichlet allocation," in *Proc. 49th Hawaii Int. Conf. Syst. Sci. (HICSS)*, 2016, pp. 1072–1081.

[19] F. E. H. Tay and L. J. Cao, "Improved financial time series forecasting by combining support vector machines with self-organizing feature map," *Intell. Data Anal.*, vol. 5, no. 4, pp. 339–354, Sep. 2001. [Online]. Available: http://dl.acm.org/citation.cfm?id=1294015.1294019

[20] L. J. Cao and F. E. H. Tay, "Support vector machine with adaptive parameters in financial time series forecasting," *IEEE Trans. Neural Netw.*, vol. 14, no. 6, pp. 1506–1518, Nov. 2003.

[21] F. E. Tay and L. Cao, "Application of support vector machines in financial time series forecasting," *Omega*, vol. 29, no. 4, pp. 309–317, Aug. 2001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0305048301000263

[22] Y. Guo, S. Han, C. Shen, Y. Li, X. Yin, and Y. Bai, "An adaptive SVR for high-frequency stock price forecasting," *IEEE Access*, vol. 6, pp. 11397–11404, 2018.

[23] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, San Francisco, CA, USA, 2016, pp. 785–794.

[24] D. Wang, Y. Zhang, and Y. Zhao, "LightGBM: An effective miRNA classification method in breast cancer patients," in *Proc. Int. Conf. Comput. Biol. Bioinf. (ICCBB)*. New York, NY, USA: ACM, 2017, pp. 7–11. doi: 10.1145/3155077.3155079.

[25] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: Unbiased boosting with categorical features," in *Proc. Adv. Neural Inf. Process. Syst.* Curran Associates, 2018, pp. 6638–6648. [Online]. Available: http://papers.nips.cc/paper/7898-catboost-unbiased-boosting-with-categorical-features.pdf

[26] A. V. Dorogush, V. Ershov, and A. Gulin, "CatBoost: Gradient boosting with categorical features support," Oct. 2014, *arXiv:1810.11363*. [Online]. Available: https://arxiv.org/abs/1810.11363

[27] I. Kaastra and M. Boyd, "Designing a neural network for forecasting financial and economic time series," *Neurocomputing*, vol. 10, no. 3, pp. 215–236, 1996. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0925231295000399

[28] R. Pardo and R. Pardo, *The Evaluation and Optimization of Trading Strategies* (Wiley Trading), 2nd ed. Hoboken, NJ, USA: Wiley, 2008.

[29] J. D. Kelleher, B. M. Namee, and A. D'Arcy, *Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies*. Cambridge, MA, USA: MIT Press, 2015.

[30] R. Wirth and J. Hipp, "CRISP-DM: Towards a standard process model for data mining," in *Proc. 4th Int. Conf. Practical Appl. Knowl. Discovery Data Mining*, 2000, pp. 1–11.

[31] H. Drucker, C. J. C. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, "Support vector regression machines," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 9. Cambridge, MA, USA: MIT Press, 1997, pp. 155–161.

[32] F. Rosenbaltt, "The perceptron–a perciving and recognizing automation," Cornell Aeronautical Laboratory, Ithaca, NY, USA, Tech. Rep. 85-460-1, 1957.

[33] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1, D. E. Rumelhart and J. L. McClelland, Eds. Cambridge, MA, USA: MIT Press, 1986, pp. 318–362. [Online]. Available: http://dl.acm.org/citation.cfm?id=104279.104293

[34] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," in *Proc. 14th Int. Conf. Neural Inf. Process. Syst., Natural Synth. (NIPS)*. Cambridge, MA, USA: MIT Press, 2001, pp. 601–608. [Online]. Available: http://dl.acm.org/citation.cfm?id=2980539.2980618

[35] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.

[36] T. Griffiths, "Gibbs sampling in the generative model of latent Dirichlet allocation," Tech. Rep., 2002.

[37] A. Gelman, J. B. Carlin, H. Stern, D. Dunson, A. Vehtari, and D. Rubin, *Bayesian Data Analysis* (Chapman & Hall/CRC Texts in Statistical Science), 3rd ed. Boca Raton, FL, USA: Taylor & Francis, 2013.

[38] T. L. Griffiths and M. Steyvers, "Finding scientific topics," *Proc. Nat. Acad. Sci. USA*, vol. 101, no. 1, pp. 5228–5235, 2004. [Online]. Available: http://www.pnas.org/content/101/suppl_1/5228

[39] Thomson Reuters Corporation. *The Official Website of Reuters*. Accessed: Jan. 5, 2018. [Online]. Available: https://www.reuters.com/

[40] P. Cheeseman and R. W. Oldford, *Selecting Models From Data: Artificial Intelligence and Statistics IV*, vol. 89. Berlin, Germany: Springer, 2012.

[41] Finam Holdings. *Resource to Get 1 Minute-Level Intraday Market Historical Data of Standard & Poor's 500 (S&P 500) Index*. Accessed: Jan. 5, 2018. [Online]. Available: https://www.finam.ru/

[42] N. Kanungsukkasem and T. Leelanupab. *The Link to Archived News Articles by Reuters Has Been Changed Since January 2019. Our Website Provide the Old Link Used in Our Experiment and the Current Link to the Archived News Articles by Reuters*. Accessed: Jan. 5, 2018. [Online]. Available: http://www.it.kmitl.ac.th/ teerapong/finlda/

[43] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. Newton, MA, USA: O'Reilly Media, 2009.

[44] R. Řehůřek and P. Sojka. *Class Dictionary in Gensim*. Accessed: Jan. 6, 2019. [Online]. Available: https://radimrehurek.com/gensim/corpora/dictionary.html

[45] R. Řehůřek and P. Sojka, "Software framework for topic modelling with large corpora," in *Proc. LREC Workshop New Challenges NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50. [Online]. Available: http://is.muni.cz/publication/884893/en

[46] A. Asuncion, M. Welling, P. Smyth, and Y. W. Teh, "On smoothing and inference for topic models," in *Proc. 25th Conf. Uncertainty Artif. Intell. (UAI)*, 2009, pp. 27–34. [Online]. Available: http://dl.acm.org/citation.cfm?id=1795114.1795118

[47] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux. *Class SVR in Scikit-Learn Library*. Accessed: Jan. 6, 2019. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html

[48] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27:1–27:27, May 2011. doi: 10.1145/1961189.1961199.

[49] J. Q. Issam H. Laradji, and A. Mueller. *Class MLPRegressor in Scikit-Learn Library*. Accessed: Jan. 6, 2019. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html

[50] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.

[51] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. Vanderplas, A. Joly, B. Holt, and G. Varoquaux, "API design for machine learning software: Experiences from the scikit-learn project," in *Proc. ECML PKDD Workshop, Lang. Data Mining Mach. Learn.*, 2013, pp. 108–122.

[52] A. Gramfort, M. Blondel, O. Grisel, A. Joly, J. Wersdorfer, L. Buitinck, J. Nothman, K. Desai, N. Dawe, M. Kumar, M. Eickenberg, and K. Shmelkov. *Mean_Squared_Error Function in Scikit-Learn Library*. Accessed: Jan. 9, 2019. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_squared_error.html

[53] A. Gramfort, M. Blondel, O. Grisel, A. Joly, J. Wersdorfer, L. Buitinck, J. Nothman, K. Desai, N. Dawe, M. Kumar, M. Eickenberg, and K. Shmelkov. *Mean_Absolute_Error Function in Scikit-Learn Library*. Accessed: Jan. 9, 2019. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_absolute_error.html

[54] W. McKinney. *Mad Function in Class DataFrame in Pandas*. Accessed: Jan. 9, 2019. [Online]. Available: https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.mad.html

[55] W. McKinney, "Pandas: A foundational Python library for data analysis and statistics," in *Proc. Workshop Python High Perform. Sci. Comput.*, 2011, pp. 1–9.

[56] A. Tahmassebi. *Regression Error Characteristic Curve in Python*. Accessed: Apr. 11, 2019. [Online]. Available: https://github.com/amirhessam88/Regression-Error-Characteristic-Curve

[57] A. Tahmassebi, "iDeepLe: Deep learning in a flash," *Proc. SPIE*, vol. 10652, May 2018, Art. no. 106520S.

[58] A. Tahmassebi, A. H. Gandomi, and A. Meyer-Baese, "A Pareto front based evolutionary model for airfoil self-noise prediction," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2018, pp. 1–8.

NONT KANUNGSUKKASEM received the B.Eng. degree (Hons.) in computer engineering from the King Mongkut's Institute of Technology Ladkrabang (KMITL), Thailand, in 2003, and the M.Sc. degree in finance from Chulalongkorn University, Thailand, in 2010. He is currently pursuing the Ph.D. degree with KMITL.

His current research interests include time series prediction, natural language processing, information retrieval, and machine learning.

TEERAPONG LEELANUPAB received the B.Eng. degree in computer engineering from the King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand, in 2003, the M.Sc. degree (Hons.) in software system engineering from University College London, U.K., in 2007, and the Ph.D. degree in computer science from the University of Glasgow, U.K., in 2012. He is currently an Assistant Professor with the Faculty of Information Technology, King Mongkut's Institute of Technology Ladkrabang.

His current research interests include cognitive and human-centered computing, information retrieval, text analytic and mining, machine learning, and haptic feedback application.