JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 39, 439-455 (2023)
DOI: 10.6688/JISE.202303_39(2).0012

Bond Price Prediction Using Technical Indicators and Machine Learning Techniques

SHU-YING LIN[1,+] AND HUI-YU LIN[2]
[1]*Department of Finance*
Minghsin University of Science and Technology
Hsinchu, 304 Taiwan
E-mail: sylin@must.edu.tw[+]
[2]*Institute of Information Management*
National Chiao Tung University
Hsinchu, 300 Taiwan

Price prediction in financial markets has long been a difficult task. However, while many attempts have been made to improve stock market predictability, there are few studies of bond markets. Unlike stocks, most bonds do not trade on exchanges. Consequently, the bond market usually lacks transparency and liquidity, making any estimation of bond prices an especially risky endeavor. Even so, the average daily trading volume of corporate bonds was more than 30 billion dollars. Evidently, the bond market is enormous and the need for improved prediction models that can forecast bond prices and support trading decisions cannot be overestimated. This paper proposes a novel approach to building bond price predictive models based on the technical indicators in financial markets and improving their computing efficiency by applying the machine learning techniques on Apache Spark framework. Our predictive models are constructed in three phases. First, we expand the feature set of each model by transforming the original price time series into a set of technical indicators; the number of features is then reduced by applying dimensionality reduction methods. Second, we employ machine learning algorithms to build predictive models. Finally, we compare the prediction results of different models by evaluating their MAE and RMSE. The data used in this research is a competition dataset from Kaggle containing corporate bond transactions. The experimental results show that our proposed approach considering technical indicators and dimensionality reduction outperforms the baseline for bond price prediction.

Keywords: data analytics, bond price prediction, technical indicators, machine learning, apache spark, financial markets

1. INTRODUCTION

Corporate bonds are one of the main channels for raising capital among corporations. Investors lend issuing companies money by purchasing their corporate bonds, which in return provide investors with a fixed income in the form of a periodical interest payment. In addition, investors also invest in corporate bonds to acquire capital gains, diversify their portfolio, or protect themselves against economic slowdown and deflation. According to statistics from the Securities Industry and Financial Markets Association (SIFMA) [3], a large volume of corporate bonds are traded every day.

Price prediction is a crucial and difficult task when making bond investment strategies

Received November 30, 2021; accepted December 24, 2021.
Communicated by Churn-Jung Liau.

[+] Corresponding author.

[11]. The price for a bond fluctuates with the constantly changing economic environment and the financial condition of the issuing company, as the stock price also fluctuates. In recent years, new approaches using machine learning techniques to predict the financial market have been widely introduced [7, 10, 16, 17, 20, 21, 28]. With a solid price predictive model, investors can not only assess their position before trading, but they may also find opportunities to exploit mispricing and take arbitrage profit. But the factors that affect bond prices and those that affect stock prices are different, since the two instruments are essentially disparate. Given the economic incentive to forecasting stock price, it is no surprise that many studies about how to improve stock price prediction have been made in the past. However, there is hardly any study dedicated to price prediction in the bond market, especially considering technical indicators, which are used to carry out technical analysis and make investment strategies. Perhaps the main reason stems from the fact that the bond market has lower liquidity and is less transparent than the stock market, since most bonds are traded over-the-counter whereas most stocks are traded in exchange.

In this research, we propose a novel approach to building bond price predictive models based on technical indicators in financial markets [26] and using machine learning techniques [12, 28]. To enable processing with a large dataset and improve the computing efficiency, we employ a Hadoop cluster and use Apache Spark as the computing framework [1]. The dataset used in this research is a competition dataset from Kaggle [19], provided by Benchmark Solutions, a bond trading information provider. The dataset contains 762,678 observations and each observation represents a bond trade described by 61 variables, including a ground truth trade price. In our proposed approach, the construction of a bond price predictive model contains three stages. First, the original feature set is expanded by transforming historical prices into a set of technical indicators, which are used to carry out technical analysis and make investment strategies in financial markets. Second, the dimensionality of the dataset is reduced by applying dimensionality reduction methods, including Chi-square feature selection, Random Forest feature selection, and Principal Component Analysis. Finally, a machine learning algorithm is employed to build a predictive model, which is evaluated by calculating its Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). In this way, we compare the performance of Linear Regression, Random Forest, and Gradient Boosting Regression models. The experimental results show that our proposed approach considering technical indicators and dimensionality reduction outperforms the baseline for bond price prediction.

The remainder of this paper is organized as follows. In Section 2, we summarize related studies on financial market prediction, dimensionality reduction methods, machine learning algorithms, and the Spark on YARN computing framework. In Section 3, we introduce the proposed approach to building a bond price predictive model. We demonstrate the experiments undertaken and compare the performance of different models in Section 4, and present a summary of our findings in Section 5.

2. RELATED WORK

This section presents an overview of the related literature on price prediction in financial markets.

2.1 Price Prediction in Financial Markets

According to the Efficient Market Hypothesis [14], the prices of stocks, bonds, or any other securities fully reflect all known information. The hypothesis also assumes these prices follow random walks [22], which makes it impossible to predict them. However, market participants are not always rational and information is asymmetric in the real world. Previous studies have shown that financial markets can be predicted to some extent [11, 25, 34, 38].

Traditionally, there have been two mainstream methodologies regarding price prediction in financial markets: fundamental analysis and technical analysis. Fundamental analysis [4] is conducted on the assumption that price is determined by various economic factors such as macroeconomic indicators, the financial condition of the relevant industry, and the accounting-related information of the company. Since fundamental analysis depicts the big picture of a company's potential, it is usually used in developing long-term investment strategies. Technical analysis [26], on the other hand, is the study of market action; accordingly, it focuses on historical time-series information, which includes price, volatility, and trading volume. In contrast to fundamental analysis, technical analysis is often used for shorter timeframe prediction (*e.g.*, weeks, days or even minutes).

In recent years, new approaches using machine learning techniques to predict financial markets have been implemented by several scholars [7, 10, 16-18, 21]. Dhar [11] has developed an equity index trading strategy by building decision trees to find small disjuncts with high returns. In another study, Shynkevich *et al.* [33] use historical prices and 16 different technical indicators, including simple moving average (SMA), moving average convergence/divergence (MACD), momentum, relative strength index (RSI), and price rate-of-change, as input features to predict the direction of stock price movement using a support vector machine. Zimbra *et al.* [37] have extracted firm-related information from web forums and performed sentiment and stakeholder analyses in order to predict stock return. Seng and Yang [31] apply sentiment analysis to discover the association between stock price volatility and financial news. These studies show the promising results of using machine learning techniques to predict financial markets; these advanced techniques not only capture comprehensive information, but they also reveal implicit and complex patterns that have emerged in the past, which are critical in forecasting future prices.

2.2 Dimensionality Reduction

The original dataset used in this research contains 61 variables; we further expand the feature set to 64 variables by calculating various technical indicators from historical time-series price information. However, certain concerns arise when dealing with large dimensionality datasets. First, as the dimensionality of the dataset increases, its predictive power may be reduced. Second, there could be redundant or irrelevant features in a large dataset, which might result in noise and further deteriorate the performance of the predictive model. Third, the efficiency of the predictive model is dragged down with a large dataset, since more computation time and storage is required to process the information.

To mitigate the problems addressed above, we exploit some dimensionality reduction methods, after first expanding the dataset, to determine the most adequate feature subset. This determination is made by comparing the performance of various subsets in improving the result of predictive models. The feature subset generation methods used for reducing dimensionality include the feature selection approach and feature extraction approach. Feature selection is the process of selecting *n* variables from an original feature set of m variables, where $n \leq m$. The selected subset must contain variables that manifest high relevance with respect to the independent variable, or minimum redundancy with respect to other variables [24]. There are various methods of determining the representative variables. The Chi-square method can be utilized, which reduces the dimensionality of the feature space [13, 36]. The Chi-square method tests the dependency of two events. Features with the lowest values are eliminated from the feature set.

Alternatively, feature extraction approaches, which transform original features into a new set of features, can also be used. The most common of these approaches, Principal Component Analysis (PCA) [35], transforms data linearly, mapping data from an original *d* dimension space to a new *d′* dimension space, where $d' \leq d$. PCA can be done by eigenvalue decomposition or SVD (Singular Value Decomposition). In both cases, an orthogonal transformation is used to transform a data matrix with possibly correlated variables into a new data matrix with linearly uncorrelated variables. Such variables are called Principal Components (PCs), which are linear combinations of all original features. Fahad *et al.* [13] compared different dimensionality reduction methods, including PCA, Information Gain, Chi-square, and Correlation-based Feature Selection, and found that even though one method might outperform all others with respect to a particular dataset, it might be worse with a different dataset. The choice of a reduction method depends on both the dataset being employed and the choice of a machine learning algorithm. In this study, we compare different dimensionality reduction methods in order to find the most adequate feature subsets for each machine learning algorithm.

2.3 Machine Learning Algorithms

Machine learning algorithms have been widely applied to build prediction models for various applications [5, 6, 8, 23, 28, 29]. Moreover, deep learning methods have been applied to build prediction models for financial time series forecasting [27, 32]. To build the predictive models, we exploit and compare three machine learning algorithms: Linear Regression (LR), Random Forest (RF), and Gradient Boosting Decision Trees (GBDT). Linear Regression [30] models the linear relation between a dependent variable and multiple independent variables. The method of least squares is applied to fit the model by minimizing the sum of squared residuals of predictions and observations. The second algorithm, Random Forest [9], is an ensemble learning method consisting of multiple tree predictors $h(x, \Theta_k)$, $k = 1, ..., K$, where *x* denotes the vector of explanatory variables and (Θ_k) are independent identically distributed random vectors. As the number of trees increases, the law of large numbers ensures the convergence of prediction, thus preventing overfitting. The Random Forest algorithm involves three steps: (1) Draw a bootstrap sample from the original dataset and repeat it for *N* times, where *N* represents the number of trees; (2) Grow a decision tree for each bootstrap sample. When growing each tree, select *m* variables at random for a split, where *m* is less than or equal to the number of all variables; (3) Aggregate the results from all the *N* trees to predict new data. The final prediction is made by majority voting in a classification problem, whereas it is determined by the unweighted average in a regression problem.

The Gradient Boosting Decision Tree (GBDT) [15] is similar to the Random Forest algorithm in that both are ensemble methods consisting of multiple decision trees. The major difference between Random Forest and Gradient Boosting Decision Trees is that the construction of every tree in GBDT is based on the refined result of previously built trees. A gradient descent algorithm is applied, in which the objective function considers multiple prediction results of classification trees. This objective function consists of two components: a loss function and a regularization term, which controls the model complexity to avoid overfitting. The final prediction is determined by the weighted average of the prediction values of each individual tree.

2.4 Spark on YARN Computing Framework

Spark [1] is an open-source distributed computing framework, which is able to process large-scale data efficiently by distributing the computing tasks among a cluster of computers. The computing in Spark is relatively fast due to Resilience Distributed Datasets (RDD) and the Directed Acyclic Graph (DAG) execution engine, which allow in-memory computing and cyclic data flow. We launch the Spark applications on YARN (Yet Another Resource Negotiator), a resource management framework in Hadoop. Each Spark application invokes an Application Master process, which then requests resources from the YARN Resource Manager and instructs Node Managers to start containers.

3. PROPOSED APPROACH

3.1 Overview

In order to predict future bond prices, we propose an approach for building predictive models and compare the performance of different machine learning algorithms, using HDFS (Hadoop Distributed File System) [2] as the distributive storage system and Apache Spark as our computing framework.

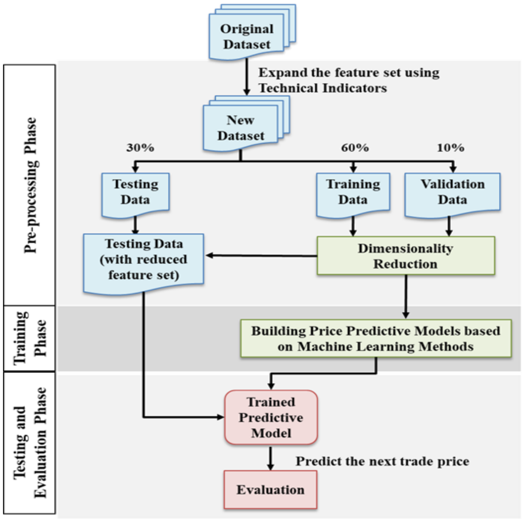

Original Dataset
Expand the feature set using Technical Indicators
New Dataset
30%
60%
10%
Testing Data
Training Data
Validation Data
Testing Data (with reduced feature set)
Dimensionality Reduction
Building Price Predictive Models based on Machine Learning Methods
Trained Predictive Model
Predict the next trade price
Evaluation
Pre-processing Phase
Training Phase
Testing and Evaluation Phase

Fig. 1. An overview of the proposed approach.

Fig. 1 illustrates our proposed framework, which is composed of three phases. First, we expand the feature set by transforming the original price time series into a set of technical indicators, and reduce the dimensionality by applying feature extraction methods. Then we utilize machine learning algorithms, namely, Linear Regression, Random Forest, and Gradient Boosting Decision Trees, to build predictive models. Finally, we compare the prediction results from different models by evaluating their mean absolute error and root mean square error. The details of each phase are discussed in the following sections.

3.2 Data Description

In this research, a competition dataset from Kaggle is used that contains 762,678 U.S. corporate bond transactions. It was provided by Benchmark Solutions Inc., a bond-pricing data provider. Each transaction in the dataset is described by 61 variables, which include basic information about the traded bond (*e.g.*, coupon rate, time to maturity, and trade type) as well as information about previous trades of the same bond; the target is to predict the price at which the trade occurred. Table 1 lists the details of each column in this dataset, and Table 2 contains the variables about the last ten trades; thus, each description in Table 2 refers to 10 variables in the dataset.

Interest rate is a key factor in bond valuation, and a bond rating reflects the default risk of a bond. While this information is crucial, it was not disclosed in the dataset. Instead, the intermediate calculation result was provided as a reference price without revealing its exact formulation. Since we have no knowledge about the calculation process of the reference prices, we removed the columns of *curve_based_price, curve_based_price_last1, ..., curve_based_price_last10*. The columns id and bond_id were also removed since they do not carry meaningful information, and 48 variables were left in the dataset to be processed.

Table 1. Description of variables in original dataset.

Column Name	Description
Id	Sequence of data
Bond_id	The unique id for each bond
Trade_price	The price at which the trade occurred (This is the column to predict in testing data.)
weight	Calculated as the square root of the time since the last trade and then scaled so the mean is 1
current_coupon	The coupon of the bond at the time of the trade
time_to_maturity	The number of years until the bond matures at the time of the trade
is_callable	A binary value indicating whether or not the bond is callable by the issuer
reporting_delay	The number of seconds after the trade occurred that it was reported
Trade_size	The notional amount of the trade
Trade_type	2 = customer sell, 3 = customer buy, 4 = trade between dealers. We would expect customers to get worse prices on average than dealers.
Curve_based_price	A fair price estimate based on implied hazard and funding curves of the issuer of the bond

Table 2. Description of variables in original dataset (continued).

Column Name	Description
received_time_diff_last {1-10}	The time difference between the time of trade and that of the previous ten trades
Trade_price_last {1-10}	The trade price of the last ten trades
Trade_size_last {1-10}	The notional amount of the last ten trades
Trade_type_last {1-10}	The trade type of the last ten trades
Curve_based_price_last {1-10}	The curve based price of the last ten trades

3.3 Data Pre-Processing

In this study, the data pre-processing phase is composed of two parts. First, we implant some domain knowledge into the variables by transforming the time-series information of prices into a set of technical indicators, thus expanding the feature set to include a total of 64 variables. Next, we reduce the dimensionality of the dataset by separately applying the Chi-square feature selection method, Random Forest feature selection method, and Principal Component Analysis. The final feature subset is determined by selecting the best subset among the outcomes of the three approaches.

3.3.1 Technical indicators

Technical indicators are analytic tools that have been extensively used in forecasting future prices and supporting trading decisions in financial markets. They only capture market activity, such as price momentum and trading volume, and do not analyze any fundamental state, including economic indices or the profitability of a company.

We use the columns of *trade_price_last{1-10}*, which indicate the trade prices of the ten most recent trades, as the data employed in the calculation of technical indicators. In Murphy's study [26], the author introduced various technical indicators. We select four basic indicators that are commonly used in technical analysis. For each indicator, we apply different numbers pertaining to observation period N, where $N = 3, 5, 8, 10$.

Exponential Weighted Moving Average (EWMA): This is calculated as the weighted average of the N past data points; the weighting for each older price decreases exponentially, with the result that EWMA focuses more on recent prices. The weights are derived from a constant α, where $\alpha = 2/(N+1)$. With the objective of predicting the next trade price P_{t+1}, P_t as the most recent price, and N representing the number of observations, we calculate EWMA as Eq. (1):

$$EWMA = \frac{\sum_{i=0}^{N-1}(1-\alpha)^i P_{t-i}}{\sum_{i=0}^{N-1}(1-\alpha)^i}. \tag{1}$$

Rate of Change (ROC): This measures the percentage price change within N periods, and can be formulated as Eq. (2):

$$ROC = 100 \times \frac{P_t}{P_{t-i}} \tag{2}$$

where P_t represents the price of the most recent trade, and P_{t-i} is the price of the previous *i*th trade.

Relative Strength Indicator (RSI): RSI was developed by J. Welles Wider [34]. It reveals the strength or weakness of a bond from the range of prices in *N* periods. Relative strength is defined as the average value of upward changes (*up*) over *N* periods divided by the average value of downward changes (*down*). RSI is calculated as Eq. (3):

$$RSI = 100 - \frac{100}{1+RS}; \quad RS = \frac{\frac{1}{N}\sum_{i=0}^{N-1} up_{t-i}}{\frac{1}{N}\sum_{i=0}^{N-1} down_{t-i}}. \tag{3}$$

The range of RSI is on a scale of 0 to 100. A bond is considered overbought when the RSI is above 80, and oversold when the RSI is below 20.

Larry William's %R (W%R): This measures the latest price in relation to the price range over a given period of time, *N*. W%R is given by Eq. (4):

$$W\%R = \frac{Highest_N - P_t}{Highest_N - Lowest_N} \times (-100) \cdot \tag{4}$$

In Eq. (4), *Highest*$_N$ indicates the highest price among *N* trading days, while *Lowest*$_N$ indicates the lowest price.

3.3.2 Dimensionality reduction

The process of dimensionality reduction in this work is illustrated in Fig. 2. To begin with, the dataset is divided into testing data, training data, and validation data, with each chunk accounting for 30%, 60%, and 10% of the dataset, respectively. The training data is used to apply dimensionality reduction methods and build predictive models; the validation data is used for the purpose of tuning parameters in predictive models and determining the final feature subset; finally, the testing data is used to evaluate each model.

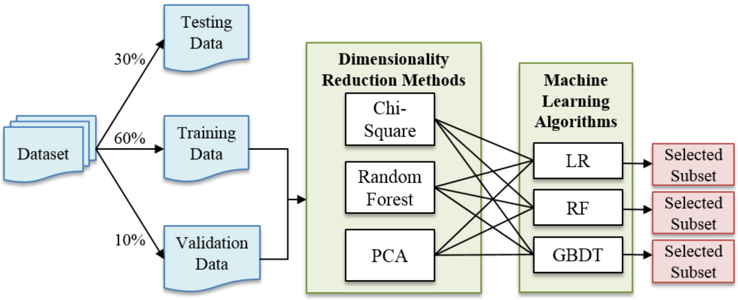


Fig. 2. The process of dimensionality reduction.

We utilize Chi-squared feature selection, Random Forest feature selection, and Principal Component Analysis to obtain the reductive feature subset. The former two methods are feature selection methods, which rank the variables (features) according to their importance; the least important variables are subsequently eliminated. With the Chi-squared method, the value of each variable is determined by conducting a correlation test between the variable input and output, whereas with the Random Forest method the importance of each variable is determined by the mean decrease in node impurity when the variable is assigned to a split point. The third method, Principal Component Analysis (PCA), does not rank the original features, but transforms the original dataset into a new dataset consisting of linear combinations (PCs) of original variables.

After obtaining the subsets of features from the aforementioned methods, we apply the machine learning algorithms to the subsets to build bond price predictive models. We compare the performance of each dimensionality reduction method with different machine learning algorithms; moreover, we also test different numbers of features. For each machine learning algorithm, we select the fittest method, which is able to reduce the most MSE and RMSE. Finally, the optimal combination of dimensionality reduction method and machine learning algorithm is employed to build the predictive models.

3.4 Predictive Models

In this section, we discuss the process of building predictive models based on Linear Regression, Random Forest, and Gradient Boosting Decision Trees. Two approaches are used. The general predictive approach builds one predictive model for all trade types. The trade-type predictive approach separates trades according to their type and builds one predictive model for each trade type.

Most corporate bonds are traded over-the-counter (OTC), and generally OTC markets are segmented into a customer market and inter-dealer market as shown in Figure 3. The difference between these two is that the latter is comprised solely of other dealers, whereas the former is more heterogeneous. The price a dealer quotes to a potential buyer in the customer market differs from the price quoted to another dealer, and in most cases the bid-ask spread is wider in the customer market than in the inter-dealer market. As a result, it is expected that customers get worse prices on average than dealers; the price of a bond, for instance, is most likely higher in the customer market than in the inter-dealer market at any point in time, and the price for which a customer will sell the bond is usually lower than the inter-dealer price.

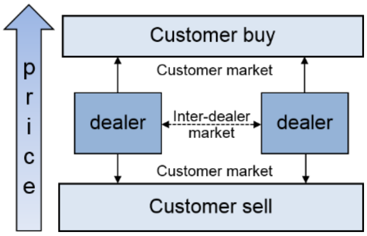


Fig. 3. Three types of trade in the bond market.

Other variables may also affect the trade price in different types of trading markets. One obvious example is the trading volume. With a larger trading volume, a customer can buy a bond at a lower price and sell it at a higher price than what is possible with a lower trading volume. We build models separately for three different trade types, which are customer buy, customer sell, and inter-dealer trade. The proportion of customer buy, customer sell, and inter-dealer trade in the dataset is 35%, 20%, and 45%, respectively.

Parameter tuning: Parameters play an important role in affecting the model results. Therefore, when training a machine learning model, it is important to finely tune the parameters to improve the overall prediction performance.

There are some parameters that can be tuned in a Linear Regression Model. One is the type of regularization. In our experiment, we use elastic net regularization, which linearly combines L1 and L2 method, as shown in Eq. (5):

$$\alpha\left(\lambda\|w\|_1\right)+(1-\alpha)\left(\frac{\lambda}{2}\|w\|_2^2\right),\quad \alpha\in[0,1],\lambda\geq 0 \tag{5}$$

where α controls the ratio of L1 and L2 regularization. The equation $\alpha = 0$ is equivalent to an L1 regularization, while $\alpha = 1$ is equivalent to an L2 regularization. Finally, λ defines the trade-off between minimizing the training error and avoiding overfitting.

In Random Forest and Gradient Boosting Decision Trees, the operator can decide on the number of trees and number of bins used when discretizing continuous features, as well as the fraction of training data used for training the decision tree, and the number of features included as candidates for splitting at each node.

4. EXPERIMENT AND EVALUATION

In this section, we conduct several experiments to determine the parameters and compare the performance of each predictive model.

4.1 Experiment Setup

The dataset used in this experiment contains 762,678 observations and 64 variables, including the technical indicators introduced in the previous section. We divide the dataset into training data, testing data, and validation data, with each accounting for 60%, 30%, and 10% of the dataset, respectively.

To identify the most adequate feature subset for training a predictive model with a specific machine learning algorithm, we carry out an experiment to compare the performance of models with subsets acquired from different dimensionality reduction methods. In this experiment, we apply dimensionality reduction methods on the training data, and use the reductive data to train predictive models with machine learning algorithms. In addition, different numbers of features in the subset are also tested. Next, the validation data is used to evaluate each model, and we compare the subsets obtained from each of the methods for every machine learning algorithm. Each algorithm is designated with a feature subset, which shows the least MAE and RMSE in the experiment. This combination of algorithm and feature subset is further used to build a predictive model in the training phase.

In the training phase, we use Linear Regression, Random Forest, and Gradient Boosting Decision Trees to build predictive models for every type of trade. In each algorithm, parameters are tuned by evaluating the model with validation data. The trade price in testing data is predicted through the tuned predictive models. Lastly, we compare the performance of different models.

To evaluate the accuracy of a prediction, we use two measures: Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). MAE measures the discrepancies between predictions and actual prices by calculating the average of the absolute errors. In our experiments, we calculate the unweighted average; the formula is shown as Eq. (6):

$$MAE = \frac{1}{M}\sum_{i=1}^{M}\left|y_i - \hat{y}_i\right| = \frac{1}{M}\sum_{i=1}^{M}\left|e_i\right|. \tag{6}$$

Another measure, RMSE, represents the standard deviation of the discrepancies between predictions and actual prices; the RMSE is calculated as Eq. (7):

$$RMSE = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\left(y_i - \hat{y}_i\right)^2} = \sqrt{\frac{1}{M}\sum_{i=1}^{M}e_i^2} \tag{7}$$

where y_i represents the i_{th} observed value of the dependent variable, $\hat{y}_i$ represents the i_{th} predicted value, and the difference between y_i and $\hat{y}_i$ is known as the residual e_i.

4.2 Dimensionality Reduction and Parameter Tuning

In this section, we compare different dimensionality reduction methods in each machine learning algorithm model. First, we apply the training data to the PCA, Chi-square and Random Forest feature selection methods. The top 30 features selected by Chi-Square and Random Forest are listed in Table 3. EWMAs and historical prices are the most important features according to the Chi-square method, while trade type, EWMAs and ROCs have the highest rankings in the Random Forest method. As for features extracted from PCA, the original features are replaced by the Principle Components (PCs), which are linear combinations of original features.

Table 3. Top-30 features selected by Chi-square and random forest methods.

Rank	Chi-squared	Random Forest	Rank	Chi-squared	Random Forest
1	EWMA.3	trade_type	**16**	current_coupon	trade_price_last3
2	EWMA.5	EWMA.3	**17**	is_callable	trade_price_last9
3	EWMA.8	trade_type_last1	**18**	received_time_diff_last8	RSI.8
4	EWMA.10	EWMA.5	**19**	received_time_diff_last7	trade_price_last2
5	trade_price_last1	EWMA.8	**20**	received_time_diff_last10	trade_price_last5
6	trade_price_last2	EWMA.10	**21**	received_time_diff_last6	time_to_maturity
7	trade_price_last3	ROC.5	**22**	received_time_diff_last5	ROC.8
8	trade_price_last4	trade_price_last10	**23**	received_time_diff_last9	WR.10
9	trade_price_last5	trade_price_last6	**24**	received_time_diff_last4	trade_type_last2
10	trade_price_last6	ROC.10	**25**	trade_type_last2	RSI.10
11	trade_price_last7	trade_price_last7	**26**	received_time_diff_last3	current_coupon
12	trade_price_last9	trade_price_last4	**27**	trade_type_last10	received_time_diff_last1
13	trade_price_last10	trade_price_last8	**28**	trade_type_last4	received_time_diff_last10
14	trade_price_last8	trade_price_last1	**29**	received_time_diff_last1	received_time_diff_last9
15	time_to_maturity	trade_size	**30**	trade_type_last7	WR.8

Different numbers of features are also tested in the experiment, and the feature set with the least MAE and RMSE is chosen to be further used in the training phase. The results of each dimensionality reduction method for various machine learning methods are evaluated. Based on the results obtained for each model respectively, we use the top 50 features selected from the Chi-square method for training Linear Regression models. For parameter tuning in Random Forest, we apply the top 50 features selected from the Random Forest feature selection method. Finally, we apply the whole feature set obtained from Gradient Boosting Decision Trees for parameter tuning in GBDT.

4.2.1 Parameter tuning in linear regression

To test different sets of parameters in Linear Regression, we use the top 50 features obtained from the Chi-square method in the previous experiment. The parameters to be tuned in this experiment are the elastic net parameter and regularization parameter, which represent the α and λ in Eq. (5). For training Linear Regression models, we fix the elastic net parameter as 0 and the regularization parameter as 1, based on the result in Fig. 4.

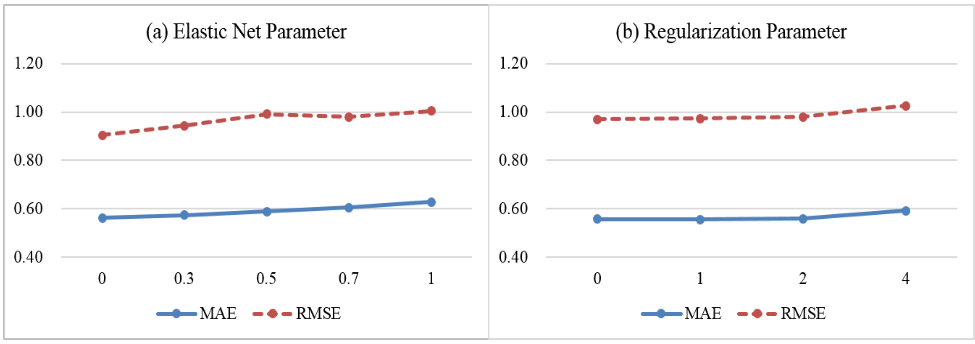


Fig. 4. Comparison of different parameter settings in linear regression.

4.2.2 Parameter tuning in random forest

We adjust two parameters in Random Forest: the maximum number of bins used in splitting a node and the number of trees. Increasing the maximum number of bins may lead to more accurate predictions, but it also requires more computation time. With respect to the second parameter, we fix the number of trees as 50. Then we test different numbers of bins. The MAE and RMSE decrease as the maximum number of bins increases, as shown in Fig. 5 (a). However, the changes become insignificant when the number exceeds 300, so we set the maximum number of bins as 300. Next, we test different numbers of trees and the results appear to be constant, as shown in Fig. 5 (b), so we set the number of trees as 100.

4.2.3 Parameter tuning in gradient boosting decision trees

For Gradient Boosting Decision Trees, we tune the same parameters as those in Random Forest, *i.e.*, the maximum number of bins and the number of trees (or iterations in GBDT). We set the maximum number of bins as 300 and iterations as 50, based on the result shown in Fig. 6.

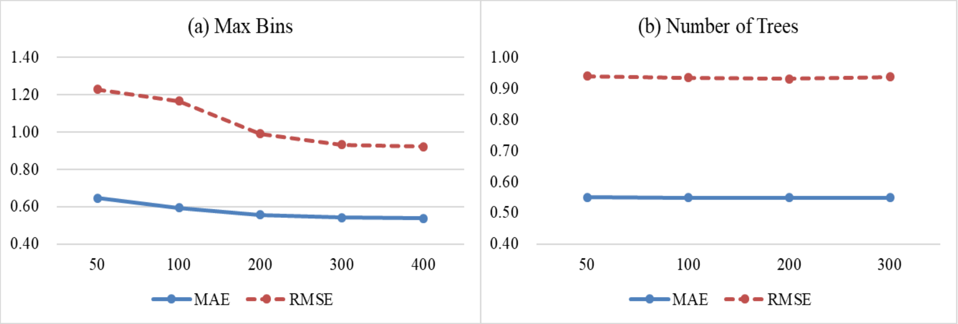


Fig. 5. Comparison of different parameter settings in random forest.

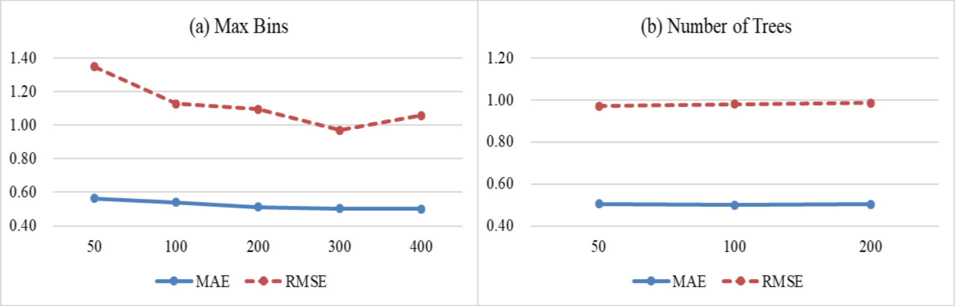


Fig. 6. Comparison of different parameter settings in gradient boosting decision trees.

4.3 Experimental Results

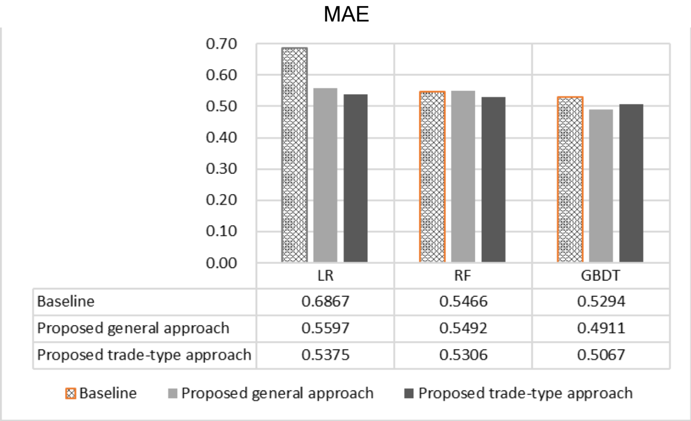
With adequate feature subsets and parameters obtained from the previous experiment, we use testing data to evaluate the performance of each predictive model. We compare the performance of models trained by different machine learning algorithms, and for each algorithm we compare three different methods. These are described as follows:

Proposed trade-type approach: As explained in previous sections, the proposed trade-type approach includes expanding the feature set with technical indicators, performing dimensionality reduction, and building different models for different types of trade.
Proposed general approach: Slightly different from the proposed trade-type approach, this method trains a model for all types of trade.
Baseline method: To prove the effectiveness of our proposed approaches, we compare them with a baseline method, in which original features are applied to train a model. In the baseline method, technical indicators are excluded and none of the dimensionality reduction methods is performed.

The results are presented in Figs. 7 and 8, which show the comparison of different methods. The overall performance of Gradient Boosting Decision Trees is better than the performance of Linear Regression and Random Forest in terms of MAE and RMSE, but Linear Regression, which takes only 10 second to train a model, has the highest efficiency. It takes 8 minutes and 15 minutes, respectively, in Random Forest and Gradient Boosting Decision Trees, to train a predictive model. With respect to the proposed trade-type approach and general approach, the Random Forest model has a lower MAE and higher

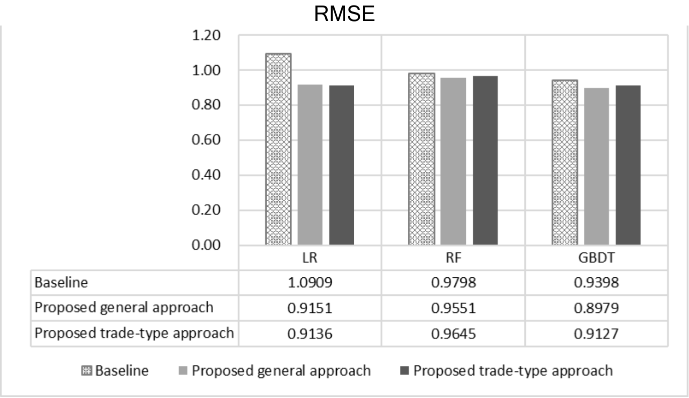
RMSE than the Linear Regression model, indicating that the errors in Random Forest models are fewer on average than those in Linear Regression models, although the standard deviation of errors is higher in Random Forest models.

Regardless of the choice of machine learning algorithm, the proposed general approach and trade-type approach outperform the baseline method, which does not consider technical indicators or dimensionality reduction. The results imply that the proposed approaches are effective in improving the accuracy of bond price predictions. In Linear Regression and Random Forest, the proposed trade-type approach performs the best among these three methods in MAE, whereas in Gradient Boosting Decision Trees, the general approach performs slightly better than the trade-type approach in MAE and RMSE. The proposed general approach with the GBDT model achieves the best predictive performance in MAE and RMSE. The results imply that the predictive performance of the two approaches varies in different machine learning methods. The trade-type approach relies on segmenting the markets into different trade-types, including customer buy, customer sell, and inter-dealer trade. In practice, the market makers may not be able to anticipate whether the next order is going to be a buy or sell. The uncertainty may affect the predictive performance of the trade-type approach.



	LR	RF	GBDT
Baseline	0.6867	0.5466	0.5294
Proposed general approach	0.5597	0.5492	0.4911
Proposed trade-type approach	0.5375	0.5306	0.5067

Fig. 7. Comparison of all methods in MAE.



	LR	RF	GBDT
Baseline	1.0909	0.9798	0.9398
Proposed general approach	0.9151	0.9551	0.8979
Proposed trade-type approach	0.9136	0.9645	0.9127

Fig. 8. Comparison of all methods in RMSE.

5. CONCLUSIONS

Machine learning has been widely applied to a number of domains including medical science, social media, electronic commerce, and engineering. However, few studies employing machine learning have focused on financial market prediction, especially in the bond market. In practice, most of the predictive analysis in financial markets is still done in the traditional fashion, that is, fundamentally, technically, or statistically. With machine learning techniques, more comprehensive information can be utilized, and implicit and complex patterns can be detected by machines, which transform this learning into reliable predictive models. Therefore, we combine technical analysis with machine learning techniques to improve the result of predictive models. More specifically, we exploit various technical indicators for bond price prediction with machine learning techniques that include dimensionality reduction approaches and machine learning algorithms. In addition, we employ Apache Spark as the computing framework and execute Spark applications on a Hadoop cluster in pursuit of higher computation speed and large dataset processing ability. Our experimental results show that our proposed approach considering technical indicators and dimensionality reduction outperforms the baseline for bond price prediction.

Some notable findings emerge from the experimental results. First, given a certain degree of domain knowledge, higher accuracy can be achieved through machine learning techniques. Second, the performance of a predictive model can be improved by applying an adequate dimensionality reduction method. Third, among the three machine learning algorithms used in our experiment, the prediction made by the Gradient Boosting Decision Trees model demonstrates the highest accuracy, although the computational cost in model training is rather high. The Linear Regression model shows the highest efficiency, but the accuracy of prediction is compromised. Last, the computation speed of processing a large volume of data can be improved significantly by deploying the task on a distributed framework.

REFERENCES

1. Apache Spark, http://spark.apache.org.
2. Hadoop, https://hadoop.apache.org.
3. S. I. a. F. M. A. (SIFMA), *US Bond Market Issuance and Outstanding*, ed., 2016.
4. J. S. Abarbanell and B. J. Bushee, "Fundamental analysis, future earnings, and stock prices," *Journal of Accounting Research*, Vol. 35, 1997, pp. 1-24.
5. R. Ahuja and S. Sharma, "Exploiting machine learning and feature selection algorithms to predict instructor performance in higher education," *Journal of Information Science and Engineering*, Vol. 37, 2021, pp. 993-1009.
6. B. An and Y. Suh, "Identifying financial statement fraud with decision rules obtained from modified random forest," *Data Technologies and Applications*, Vol. 54, 2020, pp. 235-255.
7. F. Barboza, H. Kimura, and E. Altman, "Machine learning models and bankruptcy prediction," *Expert Systems with Applications*, Vol. 83, 2017, pp. 405-417.
8. G. P. Bhandari, R. Gupta, and S. K. Upadhyay, "An approach for fault prediction in SOA-based systems using machine learning techniques," *Data Technologies and Applications*, Vol. 53, 2019, pp. 397-421.
9. L. Breiman, "Random forests," *Machine Learning*, Vol. 45, 2001, pp. 5-32.

10. R. C. Cavalcante, R. C. Brasileiro, V. L. Souza, *et al.*, "Computational intelligence and financial markets: A survey and future directions," *Expert Systems with Applications*, Vol. 55, 2016, pp. 194-211.
11. V. Dhar, "Prediction in financial markets: The case for small disjuncts," *ACM Transactions on Intelligent Systems and Technology*, Vol. 2, 2011, pp. 1-22.
12. A. A. Elhag and A. M. Almarashi, "Forecasting based on some statistical and machine learning methods," *Journal of Information Science and Engineering*, Vol. 36, 2020, pp. 1167-1177.
13. A. Fahad, Z. Tari, I. Khalil, *et al.*, "Toward an efficient and scalable feature selection approach for internet traffic classification," *Computer Networks*, Vol. 57, 2013, pp. 2040-2057.
14. E. F. Fama, "The behavior of stock-market prices," *The Journal of Business*, Vol. 38, 1965, pp. 34-105.
15. J. H. Friedman, "Stochastic gradient boosting," *Computational Statistics and Data Analysis*, Vol. 38, 2002, pp. 367-378.
16. M.-W. Hsu, S. Lessmann, M.-C. Sung, *et al.*, "Bridging the divide in financial market forecasting: machine learners vs. financial economists," *Expert Systems with Applications*, Vol. 61, 2016, pp. 215-234.
17. H. A. Ihlayyel, N. M. Sharef, and M. Z. A. Nazri, "An enhanced feature representation based on linear regression model for stock market prediction," *Intelligent Data Analysis*, Vol. 22, 2018, pp. 45-76.
18. S. Jeon, B. Hong, and V. Chang, "Pattern graph tracking-based stock price prediction using big data," *Future Generation Computer Systems*, Vol. 80, 2018, pp. 171-187.
19. Kaggle, "Benchmark bond trade price challenge," www.kaggle.com, ed., 2012.
20. M. Kraus and S. Feuerriegel, "Decision support from financial disclosures with deep neural networks and transfer learning," *Decision Support Systems*, Vol. 104, 2017, pp. 38-48.
21. C.-C. Lin, C.-S. Chen, and A.-P. Chen, "Using intelligent computing and data stream mining for behavioral finance associated with market profile and financial physics," *Applied Soft Computing*, Vol. 68, 2018, pp. 756-764.
22. B. G. Malkiel, *A Random Walk Down Wall Street*, Norton, USA, 1973.
23. R. N. Mekhaldi, P. Caulier, S. Chaabane, *et al.*, "A comparative study of machine learning models for predicting length of stay in hospitals," *Journal of Information Science and Engineering*, Vol. 37, 2021, pp. 1025-1038.
24. H. M. a. F. Mhamdi, "Feature selection methods on biological knowledge discovery and data mining: A survey," in *Proceedings of the 25th International Workshop on Database and Expert Systems Applications*, 2014, pp. 46-50.
25. S. Mohanty and R. Dash, "A support vector regression framework for Indian bond price prediction," in *Proceedings of International Conference on Applied Machine Learning*, 2019, pp. 69-72.
26. J. J. Murphy, *Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications*, New York Institute of Finance, NY, USA, 1999.
27. T. Niu, J. Wang, H. Lu, W. Yang, and P. Du, "Developing a deep learning framework with two-stage feature selection for multivariate financial time series forecasting," *Expert Systems with Applications*, Vol. 148, 2020, p. 113237.
28. M. Obthong, N. Tantisantiwong, W. Jeamwatthanachai, and G. Wills, "A survey on

machine learning for stock price prediction: algorithms and techniques," in *Proceedings of the 2nd International Conference on Finance*, *Economics*, *Management and IT Business*, 2020, pp. 63-71.

29. N. Sandhya, P. Samuel, and M. Chacko, "Feature intersection for agent-based customer churn prediction," *Data Technologies and Applications*, Vol. 53, 2019, pp. 318-332.
30. G. A. Seber and A. J. Lee, *Linear Regression Analysis*, Vol. 936, John Wiley & Sons, USA, 2012.
31. J.-L. Seng and H.-F. Yang, "The association between stock price volatility and financial news–a sentiment analysis approach," *Kybernetes*, Vol. 46, 2017, pp. 1341-1365.
32. O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, "Financial time series forecasting with deep learning: A systematic literature review: 2005-2019," *Applied Soft Computing*, Vol. 90, 2020, p. 106181.
33. Y. Shynkevich, T. M. McGinnity, S. Coleman, *et al*., "Forecasting stock price directional movements using technical indicators: investigating window size effects on one-step-ahead forecasting," in *Proceedings of IEEE Conference on Computational Intelligence for Financial Engineering and Economics*, 2014, pp. 341-348.
34. J. W. Wilder, *New Concepts in Technical Trading Systems*, Trend Research, 1978.
35. S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, Vol. 2, 1987, pp. 37-52.
36. Q. Zhu, L. Lin, M. L. Shyu, *et al*., "Feature selection using correlation and reliability based scoring metric for video semantic detection," in *Proceedings of IEEE 4th International Conference on Semantic Computing*, 2010, pp. 462-469.
37. D. Zimbra, H. Chen, and R. F. Lusch, "Stakeholder analyses of firm-related web forums: Applications in stock return prediction," *ACM Transactions on Management Information Systems*, Vol. 6, 2015, pp. 1-38.

Shu-Ying Lin (林淑瑛) is an Associate Professor of the Department of Finance at the Minghsin University of Science and Technology of Taiwan. She received the Bachelor of Finance degree from the National Taiwan University. She received the MBA and Master of Statistics degrees from the National Chung Hsing Univeristy and the University of Minnesota, respectively. She received the Ph.D. degree in Finance from the National Central University of Taiwan. Her research interests include corporate finance, financial innovation, machine learning and financial technology.

Hui-Yu Lin (林慧瑜) received the B.S. degree in Department of Information Management and Finance from the National Chiao Tung University, Taiwan. She received the M.S. degree in Institute of Information Management from the National Chiao Tung University, Taiwan. Her research interests include e-commerce, machine learning and financial technology.