Financial Innovation

RESEARCH Open Access

A hybrid model for stock price prediction based on multi-view heterogeneous data

Wen Long[1,2,3], Jing Gao[1,2,3*], Kehan Bai[4] and Zhichen Lu[1,2,3]

*Correspondence:
gaojing21@mails.ucas.ac.cn

[1] School of Economics and Management, University of Chinese Academy of Sciences, Beijing 100190, People's Republic of China
[2] Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences, Beijing 100190, People's Republic of China
[3] Key Laboratory of Big Data Mining and Knowledge Management, Chinese Academy of Sciences, Beijing 100190, People's Republic of China
[4] Department of Mathematics, Beijing Jiaotong University, Beijing 100044, People's Republic of China

Abstract

Literature shows that both market data and financial media impact stock prices; however, using only one kind of data may lead to information bias. Therefore, this study uses market data and news to investigate their joint impact on stock price trends. However, combining these two types of information is difficult because of their completely different characteristics. This study develops a hybrid model called MVL-SVM for stock price trend prediction by integrating multi-view learning with a support vector machine (SVM). It works by simply inputting heterogeneous multi-view data simultaneously, which may reduce information loss. Compared with the ARIMA and classic SVM models based on single- and multi-view data, our hybrid model shows statistically significant advantages. In the robustness test, our model outperforms the others by at least 10% accuracy when the sliding windows of news and market data are set to 1–5 days, which confirms our model's effectiveness. Finally, trading strategies based on single stock and investment portfolios are constructed separately, and the simulations show that MVL-SVM has better profitability and risk control performance than the benchmarks.

Keywords: Market data, Financial news, Support vector machine, Multi-view learning, Heterogeneous data

Introduction

Stock price predictions have always been a focus of financial research. Existing research on stock price prediction is primarily based on two data types. One is structured historical market data, and the other is unstructured text, such as financial news.

Stock market data, such as returns and trading volumes, play a vital role in stock price prediction, and many studies have used market data to predict stock price trends. White (1988) was the first to successfully predict the time series of the stock market using the Back Propagation Neural Network (BP-NN). Subsequently, Kolarik and Rudorfer (1994) compared the prediction results of Artificial Neural Network (ANN) with those of Autoregressive Integrated Moving Average model (ARIMA), showing that the ANN model was more effective. Bildirici and Ersin (2009) studied the historical stock data of the Istanbul stock market over the past 30 years by combining the Autoregressive Conditional Heteroskedasticity model (ARCH) or the Generalized Autoregressive Conditional Heteroskedasticity model (GARCH) with

ANN and found that the hybrid model of GARCH and ANN had better prediction results than the hybrid model of ARCH and ANN. Hammad et al. (2007) applied the multi-layer BP-NN to predict the stock price, which showed a better prediction performance than other methods. In recent years, deep learning has been introduced into stock price predictions. Chen et al. (2015) realized the prediction of stock returns with the Long Short-Term Memory (LSTM) model. Fischer and Krauss (2018) used LSTM to predict stock prices and drew a short-term investment strategy. Long et al. (2019) put forward a multi-filters neural network model using deep learning methodologies and applied it to the Chinese stock market index CSI 300. Some studies have also utilized reinforcement learning for financial prediction; however, the algorithm usually requires training and testing over a very long period. Tan et al. (2011) developed a non-arbitrage algorithmic trading system based on reinforcement learning, which was tested on more than 20 stocks over 13 years from 1994 to 2006. Suhail et al. (2022) employed a reinforcement learning network to guide stock market trading, which used 11 years of Apple stock data from 2006 to 2016. Additionally, the performance of reinforcement learning is sometimes unsatisfactory. Li et al. (2007) adopted actor-only and actor–critic reinforcement learning to develop two prediction systems; however, both systems were unable to generate significant improvements. Kanwar (2019) also showed that deep reinforcement learning was less successful in capturing the dynamic changes in the stock market than originally thought.

Moreover, many studies have shown that in addition to market data, financial news has an impact on stock prices. News contains information about the company's fundamentals and activities; hence, it will affect market participants' expectations of future price changes, thus driving stock price movements. Dyck and Zingales (2003) proved that issuing earnings announcements through news media could increase volatility in the stock price. Shiller (2015) also held that media can fuel the fluctuations of the stock market. Hence, deriving information affecting stock prices from media coverage is very important. Wüthrich et al. (1998) chose the news in the most influential financial newspapers, such as the Wall Street Journal, as the object of empirical research and explored the forecasting effect of the news on market indexes. Lavrenko et al. (2000) constructed an e-Analyst news recommendation system to study the correlation between news and stock price time series. This system can recommend news that has a predictive effect on future stock price trends. Gidofalvi and Elkan (2001) applied the Bayesian text classifier and found that news indicators had a certain predictive effect on the stock price within 20 min before and after the news is released. Mittermayer and Knolmayer (2006) built a NewsCATS system to predict the intraday real-time price fluctuations of stocks caused by news. Compared with other automated text categorization algorithms, this system was found to have better predictive performance and higher system trading profitability. Schumaker and Chen (2009) found that news could be used by the SVM algorithm to make excellent predictions of stock prices 20 min after the news was released, and the prediction results could be used to guide trading. Long et al. (2019) proposed a new kernel S &S to study the impact of news on stock prices, which considered the information structures among news in addition to the news contents. With SVM algorithms, the new kernel outperformed other common kernels, such as the linear kernel, by at least 5% accuracy.

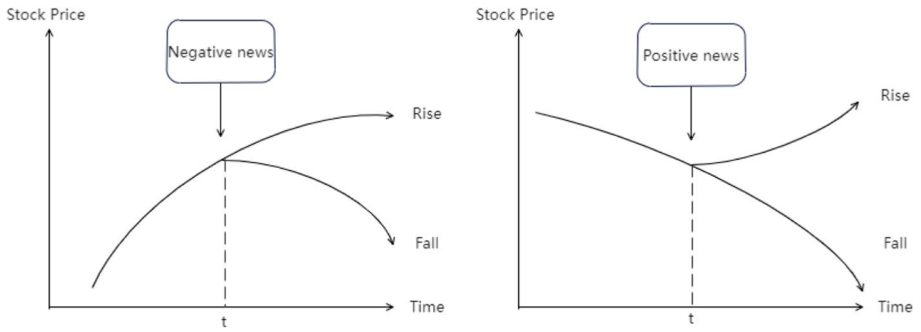


Fig. 1 Possible market scenarios

The aforementioned research works are based on single-view data, but stock price movement can be affected by both financial news and historical market data. Market and news data can be independently used to predict stock prices; however, if the model only uses single-view information, information deviation may occur. Figure 1 shows possible market scenarios. If the model only uses historical market data, the rational prediction in the left figure will be "rise," and the rational prediction in the right figure will be "fall." Therefore, if we witness an actual "fall" in the left figure or "rise" in the right figure, the prediction performance of the model is weakened. Moreover, if the model uses only financial news, it fails to explain why stock prices continue to increase when negative news is released in the left figure and why stock prices still fall when positive news is released in the right figure. Thus, analyzing the impact of multi-view data on stock prices comprehensively is important; only by this method can the model send the correct signal.

Many studies have tried incorporating the two kinds of data to improve the predicting performance. However, owing to the different structures of the two, combining them directly into a model is difficult. To solve this problem, studies usually apply indexing modeling; that is, they use textual data to compile indexes so that textual data are structured to predict stock prices together with market data (Deng et al. 2011a; Mohan et al. 2019; Li et al. 2020; Kesavan et al. 2020). Although this approach successfully fuses structured and unstructured text data, there are some limitations to this indexing treatment. Because abundant news text data are condensed into an index by directly using structured information about text (Deng et al. 2011a), such as news frequency, or by processing vectorized text into a structured sentiment index (Mohan et al. 2019; Li et al. 2020; Kesavan et al. 2020), inevitably, some information contained in the text will be lost. However, if common algorithms directly use the text vector with stock market data to perform prediction, the complicated news information with stock prediction information, a large amount of unrelated information and noise may potentially decrease the prediction performance (Lin et al. 2022). Accordingly, the appropriate extraction and exploitation of hidden information within raw multi-view heterogeneous data, including news and market data, to make accurate predictions becomes a challenging problem.

To solve the problem, this paper develops a hybrid model of stock price fluctuation prediction via a combination of multi-view learning for directly fusing different-structured data and a machine learning method called support vector machine for stock price trend classification. This model, called MVL-SVM, can maximize the consistency between the multi-view information learned from financial news or market data and therefore, not only reduces the information loss in news text processing, but also solves the difficulty in integrating complicated news information with market data. To evaluate the performance of the model, the time series method and classic SVMs were introduced for comparison. Finally, a series of trading strategies were constructed based on this algorithm and applied to three trading scenarios.

This paper contributes in the following three aspects. (1) The proposed hybrid model based on the framework of multi-view learning can input heterogeneous information influencing stock price fluctuations, such as financial news and market data, into the prediction model simultaneously, which not only enriches the information types for stock price prediction but also reduces the information loss in the process of prediction. Most previous studies only considered single-view data; some related to multi-view data tend to adopt the strategy of indexing modeling, which will inevitably lead to a large loss of information. (2) This study also investigates the lag effect of news and market data on stock-price forecasts. Usually, the news cannot be fully absorbed by the stock price on the day of the news release, which means it may further affect the stock price the next day; however, few studies consider the time lag of this impact. We solve this problem by studying a prediction problem with lag and different time windows to observe the ability of MVL-SVM to capture the information contained in multi-view heterogeneous data after or over a certain period. (3) This study constructs a series of trading strategies based on the proposed hybrid model and compares them with other prediction-based and common strategies. The simulation results show that MVL-SVM has better profitability and risk control ability than other models, which provides more favorable proof for evaluating the performance of our model. At the same time, related studies only focus on prediction accuracy.

The rest of the paper is organized as follows. “Literature review” section reviews the main existing methods related to stock price forecasting based on multi-view heterogeneous data. “Methods” section introduces the methods used in this study. “Experimental test” section presents our datasets and shows the results of the MVL-SVM model, which are compared with a time series model and some classic SVM models. “Robust test” section further evaluates the performance of the model under different sliding windows. “Trading strategy” section discusses a series of trading strategies designed with our model to assess its practical efficiency, and “Conclusion” section presents our conclusions.

Literature review

Some significant attempts have been made in the finance domain to incorporate news and market data in predicting stock prices. Relevant methods can be divided into two categories: indexing modeling and direct fusion methods.

Indexing modeling methods involve constructing indexes with news information and thus fusing the structured index with numerical market data for prediction. Frequently used methods include the statistical method, which calculates the frequency of the text data, and sentiment analysis, which uses the processed text from language preprocessing to provide polarity scores for social media data and news. Deng et al. (2011a) predicted the price movement with overall sentiment analysis and frequency based on news and comments, and technical analysis of historical market data. Mohan et al. (2019) extracted numerical data called text polarity from text articles and combined it with stock prices for prediction. Li et al. (2020) extracted sentiments from news and represented stock prices by technical indicators. Then, a layered deep learning model was used to learn the multi-view information, and a fully connected neural network was employed for stock predictions. Kesavan et al. (2020) represented news articles and social media contents by sentiment vectors and then used deep learning techniques to incorporate the polarity of the sentiments with financial time-series data to predict stock prices. This approach succeeded in fusing structured and unstructured textual data. However, when the structured information about text is directly used and constructed into some indexes, such as news frequency, or when text is first vectorized and then processed into a structured sentiment indicator, it may face information loss.

Direct fusion method aims to directly integrate structured and unstructured data to extract information or solve classification and prediction problems. Li et al. (2016) applied the extreme learning machine (ELM) to make stock price predictions based on the market news and stock prices concurrently and found that the accuracies of RBF ELM and RBF SVM are similar but higher than that of BP-NN; the prediction speeds of the two algorithms are also much faster than that of BP-NN. Wang et al. (2019) proposed a hybrid time-series predictive neural network to combine the daily K-line data with the news vectors and succeeded in stock volatility prediction. Ronaghi et al. (2022) predicted market index with COVID-19-related Twitter data and historical market data via a deep fusion framework consisting of two parallel paths, one based on CNN and another that integrates CNN with bi-directional LSTM (BLSTM). Lin et al. (2022) developed a spatial-temporal attention-based convolutional network, which successfully extracted text and numerical information for stock price prediction using the attention mechanism, CNN, and LSTM. However, the aforementioned studies were mostly based on the neural network framework. Considering that a neural network is prone to fall into a local minimum, we attempt to develop a new model based on a different framework to fuse the two data types.

Multi-view learning proposed by de Sa (1994) is a machine learning algorithm that can directly input heterogeneous data for training and usually has an excellent performance. Unlike the method of constructing an index from text, it substitutes labels using different views. It minimizes the inconsistency between the model outputs from distinct views to minimize classification errors. Yarowsky (1995) and Blum and Mitchell (1998) indicated that multi-view learning outperformed single-view learning in light of classification. Blum and Mitchell (1998) improved the algorithm by co-training distinct views when studying web page classification. Collins and Singer (1999) measured the consistency between distinct views by constructing an objective function. By maximizing the objective function, Dasgupta et al. (2001) presented an upper limit for the generalization error of multiple views. Multi-view learning has been applied to a variety of learning methods, such as dimensionality reduction (Sun et al. 2010) and classification methods (Han et al. 2022). Many scholars have noticed its usefulness and begun combining it with other traditional algorithms to obtain excellent performance. Xiao et al. (2022) utilized the multi-view learning to solve the data uncertainty; thus, successfully improving the Ordinal Regression classifier (OR). Lv et al. (2021) developed a prediction model with market data by integrating multi-view learning with the classic RBF network, which showed excellent performance in forecasting stock prices. However, it only uses market data and excludes financial news information, which has been proven to be predictive by many studies.

The intuition for building this hybrid model is as follows. On the one hand, SVM is a classic machine learning classification algorithm and is often used to predict stock prices with financial news (Schumaker and Chen 2009; Long et al. 2019). Many studies have shown that SVM performs better in financial forecasting when compared with some neural network frameworks (Kim 2003; Cao and Tay 2001, 2003; Li et al. 2016; Meesad and Thanh 2014). On the other hand, multi-view learning can learn common feature spaces or shared patterns by combining multiple data sources (Yan et al. 2021). Therefore, these two algorithms can be combined to use multi-view heterogeneous data to predict stock prices. Some scholars have theoretically proved the effectiveness of the multi-view model over the single-view model (Sun et al. 2022), and literature shows that this hybrid model has achieved excellent performance in fusing data with different structures for classification (Zhang et al. 2010; Xu et al. 2015; Ceci et al. 2015; Wang and Zhou 2021).

However, this model has not been widely applied in the financial field, and the limited related research can be divided into three categories. First, most studies based on this method considered only single-view data. Shynkevich et al. (2015b2015a) used the model to predict the stock price based on financial news and found that fusing different news categories could improve the prediction performance. However, they only considered the impact of media on stock prices and did not consider the impact of market data. Second, although some studies simultaneously included numerical and textual data with a multi-view learning framework, they transformed the text into structured indexes to fuse text with numerical data (Deng et al. 2011a, b, 2014). As stated earlier, this approach increases information loss. Third, a few studies applied a multi-view learning framework to merge historical stock prices with financial news vectors for stock price prediction (Li et al. 2011; Wang et al. 2012); however, they neither took into account the lag effect of news and market data on stock prices nor built trading strategies; hence, the model's actual application performance in the financial market could not be judged.

Methods

Chinese news text processing

Considering that news is unstructured and involves a lot of noise or redundant information, we must eliminate noise and extract representative features containing the most useful information for accurate prediction. Therefore, this section introduces the method of transforming news text into a structured feature vector for training through news preprocessing, data cleaning, text representation, and feature extraction.

News preprocessing

Because trading in the Chinese stock market ends at 15:00, news released after 15:00 on trading day_t can be assumed to not affect the fluctuation of the stock price on day_t; similarly, the news released on weekends, holidays, and other closed days have no effect on the prices. Therefore, news released on the closed day or after 15:00 on each trading day is included in the news of the next trading day. We then sorted the news by the reorganized date for processing.

Data cleaning

In data cleaning, we first removed the punctuation and garbled characters in the news, and then used the jieba package of Python to perform Chinese word segmentation. Finally, we used word filtering to filter out unimportant words from the Baidu Stop Word List. This step can help remove stop words and leave representative words such as nouns, verbs, and adjectives.

Text representation

For a news article, the value of each word was calculated according to its classification importance. Words that were more important for classification were assigned higher weights. Thus, each article can be represented as a vector of word values. The bag-of-words model is a commonly used text representation method that represents the text as a bag of words, regardless of word order and grammar, while maintaining multiplicity. Based on the bag-of-words model, Salton et al. (1975) proposed a vector space model commonly used in text classification. Because news contains many new concepts and words, it is appropriate to assume the independence of each word in this model. Each news item is then represented as a vector composed of the weight of each word, and the weight is determined by the word's importance in the news. According to Salton and Buckley (1988), the importance of words can be determined by TF-IDF, which supposes that words that rarely appear in the entire document but frequently appear in a text are of greater importance for classification. However, in practice, text length affects the weights obtained from this method. To better quantify the characteristic words, the influence of the length should be reduced. Therefore, we used the ltc method (Buckley et al. 1995) in this study, which combines length normalization ("l"), term frequency ("t") and collection frequency ("c") to calculate the weights of words. By normalizing the weight of words, the influence of article length can be avoided, and the importance of word frequency is weakened to a certain extent. This represents news articles in the following form:

$$news_t = \left(w_{t,1}, w_{t,2}, \ldots, w_{t,M}\right), \tag{1}$$

where $news_t$ represents the news vector on day_t and $w_{t,m}$ is the weight of $word_m$ in $news_t$, which is expressed as

$$w_{t,m} = \frac{\left(log\left(f_{t,m}\right) + 1.0\right) * log\frac{1}{F_m}^{\;2}}{\sum_{j=1}^{M}\left[\left(log(f_{t,j}) + 1.0\right) * log\frac{1}{F_j}\right]^2}, \; m = 1, 2, \ldots, M, \tag{2}$$

where $f_{t,m}$ represents the occurrence frequency of $word_m$ in $news_t$ (term frequency), F_m is the occurrence frequency of $word_m$ in the news corpus (collection frequency). All symbols used in the equations are listed in the Appendix (see Table 22).

Feature extraction

Because the news corpus involves many words, but only a small portion of words is contained in each news item, we use χ^2 statistics (Yang and Pedersen 1997) to extract features of the text. Instead of using all the words in the news corpus, it selects words that contribute more to text classification to make computation easier and prevent overfitting. For word w and category c, we define A as the number of times w and c co-occur, B as the number of times w occurs without c, C as the number of times c occurs without w, D as the number of times neither c nor w occurs, and n is the sample size.

$$\chi^2(w,c) = \frac{n \times (AD - BC)^2}{(A+B) \times (A+C) \times (B+D) \times (C+D)}. \tag{3}$$

Then, for a word w, the χ^2 score is obtained by combining the scores of each category:

$$\chi^2(w) = P(-1) \times \chi^2(w,-1) + P(1) \times \chi^2(w,1), \tag{4}$$

where $P(c)$ represents the frequency of the category $c \in \{-1, 1\}$ in the news corpus. Words with higher χ^2 scores are considered more informative for prediction, so we use χ^2 scores to select the optimal number of words with the best prediction performance as the dimension of news. The prediction accuracy maximization was determined by calculating the prediction accuracy in different dimensions separately.

MVL-SVM algorithm

The proposed MVL-SVM algorithm combines a support vector machine and multi-view learning, which can apply multi-view learning for multi-view data fusion and then use a support vector machine for classification or prediction. These two components are discussed further in this section. Moreover, SVM will also serve as a benchmark to test the effectiveness of our hybrid model.

Support vector machine (SVM)

SVM (Cortes and Vapnik 1995; Deng et al. 2012) has been widely applied to solve classification problems owing to its performance. It can learn from a set of two-class training instances and divide new instances into one of the classes to solve classification problems.

We denote $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ as a two-class training dataset, where x_i for $i = 1, \ldots, n$ represents a p-dimensional real vector, and $y_i \in (-1, 1)$ represents to which class x_i belongs. According to the classification method, SVM can be divided into linear and nonlinear SVM. The main idea of linear SVM is to find a " maximum margin hyperplane," defined as $g(x) = \omega^T x + b$ so that the two classes of samples can be accurately classified by this hyperplane and the sum of distances between the hyperplane and the closest point of each class is maximized. Mathematically, the classification problem is equivalent to solving the minimization problem as follows:

$$\begin{aligned} & min \frac{1}{2}\omega^T \omega + C \sum_{i=1}^{n} \xi_i, \\ s.t. \quad & y_i(\omega^T x_i + b) + \xi_i \geq 1, \forall 1 \leq i \leq n, \\ & \xi_i \geq 0, \forall 1 \leq i \leq n. \end{aligned} \tag{5}$$

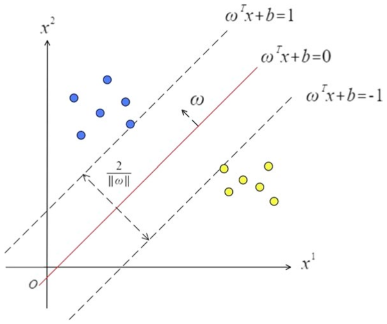


Fig. 2 A 2-dimensional classification instance using SVM

where n refers to the sample size, ξ_i is a slack variable, and C is a penalty term that controls the cost of misclassification of samples. The larger C is, the more intolerant the model is to classification errors, which are prone to overfitting. On the contrary, when C is smaller, there is more tolerance; therefore, the model is prone to underfitting. A 2-dimensional example is shown in Fig. 2 to clearly demonstrate the workings of the linear SVM. Here, samples of different colors come from different classes. The red line represents the maximum margin hyperplane obtained by training the samples.

However, in practice, not all the samples are linearly separable. Therefore, a nonlinear SVM can be introduced to solve this problem. A nonlinear SVM can implicitly map samples into a high-dimensional space with $\phi(x)$ to find a maximum margin hyperplane in this high-dimensional space. The optimization problem is as follows:

$$\begin{aligned} &min\frac{1}{2}\omega^T\omega + C\sum_{i=1}^{n}\xi_i,\\ s.t.\quad & y_i(\omega^T\phi(x_i)+b)+\xi_i \geq 1, \forall 1 \leq i \leq n,\\ &\xi_i \geq 0, \forall 1 \leq i \leq n. \end{aligned} \tag{6}$$

With Lagrange duality, the original problem can be transformed into the following dual problem.

$$\begin{aligned} max \quad W(\alpha) &= \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j(\phi(x_i)\cdot\phi(x_j))\\ &= \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j k(x_i,x_j)),\\ s.t.\quad & \sum_{i=1}^{n} y_i\alpha_i = 0,\\ & 0 \leq \alpha_i \leq C, \forall i = 1,\ldots,n. \end{aligned} \tag{7}$$

where α_i is a Lagrangian multiplier corresponding to sample x_i and $k(x_i,x_j)=\phi(x_i)\cdot\phi(x_j)$ is a kernel function that is a symmetric positive definite function that satisfies Mercer's conditions. By solving the above optimization problem, we can obtain the solutions α_i^* and b^*; the decision function is obtained as

$$f(x) = sgn\left\{\sum_{i=1}^{n} y_i\alpha_i^* k(x_i, x_j) + b^*\right\}. \tag{8}$$

Kernel function $k(x_i, x_j)$ determines the performance of the model. The linear kernel function is often used to solve linear classification problems, and the Gaussian kernel function is used to solve nonlinear classification problems.

1) Linear kernel function

$$k_{lin}(x_i, x_j) = x_i^T x_j. \tag{9}$$

2) Gaussian kernel function

$$k_{Gau}(x_i, x_j) = exp\left(-\gamma ||x_i - x_j||^2\right). \tag{10}$$

where γ is a Gaussian kernel parameter, which is important in determining kernel performance. When γ is small, the model is prone to underfitting, whereas when γ is large, the model is prone to overfitting.

Multi-view learning

Generally, single-view data can be easily used in machine learning methods for classification, whereas using multi-view data in these methods is difficult. Multi-view learning algorithms appear to solve this problem.

Multi-view learning designs a function for each perspective. All functions are optimized by maximizing the consistency between redundant views, and the model's performance is improved. Owing to its outstanding performance in multi-view data applications, multi-view learning has gradually attracted increasing attention. There are three types of existing algorithms.

1) Co-training: Maximizing mutual agreement on different views of unlabeled data through alternate learning.
2) Multiple kernel learning: Linearly or non-linearly combining kernels for each view to improve training efficiency.
3) Subspace learning: Acquiring an appropriate subspace under the assumption that multiple views are generated from this appropriate subspace.

This study uses a multiple kernel learning framework to build a stock price prediction model. By selecting the appropriate kernels and kernel combination for training, each data source can be trained with the corresponding optimal kernel function; therefore, the model can perform better than the single-kernel model (Xu et al. 2013). As illustrated in Fig. 3, distinct kernels are selected for distinct views, and multiple pieces of information can be fused by combining distinct kernels. There are many combination methods that can be grouped into two categories: linear and nonlinear combinations.

However, no empirical results show that a nonlinear combination can improve the model's performance, which raises the question of whether the nonlinear combination method is necessary and efficient. Therefore, we only used linear combination methods in this study. There are two basic categories.

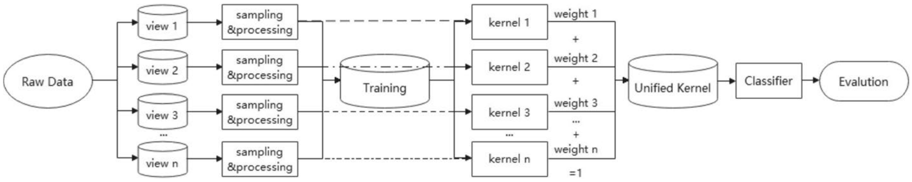


Fig. 3 Sketch map of multiple kernel learning

1) Direct summation

$$K(x_i, x_j) = \sum_{m=1}^{M} k_m(x_i, x_j), \tag{11}$$

where $k_m(x_i, x_j)$ denotes the m-th kernel.

2) Weighted summation

$$K(x_i, x_j) = \sum_{m=1}^{M} \beta_m k_m(x_i, x_j). \tag{12}$$

Here, β_m represents the weight of kernel $k_m(x_i, x_j)$.

As different types of information have different importance for prediction/classification, using the direct summation method, which assigns equal priority to each kernel, is not ideal. In comparison, we choose the weighted summation kernel in this study, and the kernel function can be written as

$$K(x_i, x_j) = \sum_{m} \beta_m k_m(x_i, x_j), \beta_m \geq 0, \sum_{m} \beta_m = 1. \tag{13}$$

The weight β_m of the kernel $k_m(x_i, x_j)$ can be determined using kernel learning. By applying SVM with the above kernel function, we can obtain the decision function of MVL-SVM, as shown in Equation (14).

$$f(x) = sgn\left\{ \sum_{i=1}^{n} \alpha_i^* y_i \sum_{m} \beta_m k_m(x_i, x_j) + b^*. \right\} \tag{14}$$

ARIMA model

ARIMA model (Box et al. 2015) is a commonly used time series model that can input historical data sequences for prediction; therefore, we use this model to design one of the benchmarks based on single-view data. It contains three terms: the autoregression term, the integrated term, and the moving average term. A nonstationary data series can be converted into a stationary one by differencing to remove the impact of nonstationarity. The first-order differencing of a data series z_t is expressed as

$$o_t = z_t - z_{t-1}. \tag{15}$$

The stationarity of the time series was tested using the ADF method. After converting the data series into a stationary one through d-order difference, we use the stationary time series to conduct a model with a combination of the autoregression model and the moving average model and obtain the future value by d-order integration. The autoregression model captures the impact of historical time-series values on the current value by performing linear regression. Because time series are usually affected by random disturbances in noisy environments, the moving average method is further introduced to observe the influence of random disturbances on future time series. Then, the ARIMA(p, d, q) model with three parameters, including the autoregression order p, differencing order d and moving average order q, can be expressed as

$$o_t = \sum_{i=1}^{p} \phi_i o_{t-1} + \sum_{j=1}^{q} \theta_j e_{t-1} + \epsilon_t, \tag{16}$$

where ϕ_i is the ith autoregression parameter, θ_j is the jth moving average parameter, and ϵ_t is the error term at time t. In practice, the autoregression order p and moving average order q can be determined using partial autocorrelation and autocorrelation diagrams, respectively.

Experimental test

Section “Introduction” shows that financial news and market data are significant in predicting price trends. Because the MVL-SVM method can integrate multiple information sources for classification, we now apply it to predict whether prices will rise or fall based on financial news and market data. Subsequently, the results were compared with classic SVMs using single-view and mixed data.

Data sources

The Shanghai Stock Exchange 50 index (SSE 50 index) comprises the most representative 50 stocks of the Shanghai Stock Exchange. This indicates the overall situation of several leading companies with the greatest market influence in the Chinese stock market. As these enterprises have the most active news reports and can thus provide sufficient news samples, we choose the constituent stocks of the SSE 50 index for empirical analysis. Due to the limitations of data sources, the period investigated in this study was from January 1, 2018 to December 31, 2020.

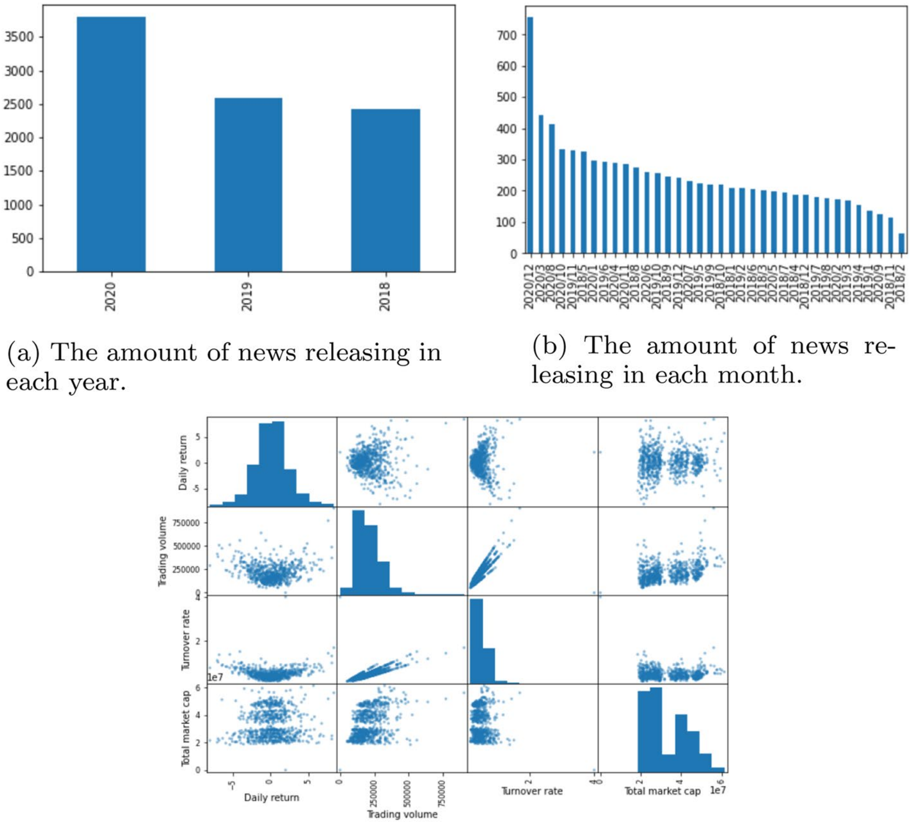
Table 1 shows the total number of days with news release for each stock. As the price of the newly listed stocks fluctuates unstably, we exclude stocks listed after January 1, 2016. Furthermore, because of less news, three stocks, including 600745. SH, 600690. SH and 601888.SH was not considered for the sample. Consequently, the research object consisted of 37 stocks.

For the structured data, considering the selected market data should comprehensively reflect the stock information, such as price changes, transaction activity, market liquidity, scale, and so on, we choose four widely used variables, including stock daily return (r), trading volume (tv), turnover rate (tr) and total market cap (mc) from Wind database (https://www.wind.com.cn/), to predict the stock price. We denote $md_1, md_2, \ldots, md_t, \ldots$ as the market data sequences, and $md_t = (r_t, tv_t, tr_t, mc_t)$ represents the four market variables on day_t. The daily returns used in this study are all log returns, and all data are daily.

Table 1 Days of news releasing

Stock code	Firm name	Days of news release
600519.SH	Kweichow Moutai	967
601211.SH	Guotai Junan Securities	692
601166.SH	Industrial Bank	691
601398.SH	Industrial and Commercial Bank of China	691
601318.SH	Ping an Insurance	689
600030.SH	CITIC Securities	688
600837.SH	Haitong Securities	685
600050.SH	China United Network Communications	685
601688.SH	Huatai Securities	684
600028.SH	China Petroleum & Chemical Corporation	684
600036.SH	China Merchants Bank	683
601288.SH	Agricultural Bank of China	683
601857.SH	PetroChina	683
601628.SH	China Life Insurance	682
600016.SH	China Minsheng Banking	680
600000.SH	Shanghai Pudong Development Bank	673
601818.SH	China Everbright Bank	672
601668.SH	China State Construction Engineering	670
601601.SH	China Pacific Insurance	663
600276.SH	Jiangsu Hengrui Medicine	662
600585.SH	Anhui Conch Cement	661
600104.SH	SAIC Motor	659
601336.SH	New China Life Insurance	658
600031.SH	Sany Heavy Industry	650
601186.SH	China Railway Construction	643
600196.SH	Shanghai Fosun Pharmaceutical	642
600048.SH	Poly Development and Holdings Group	636
600887.SH	Inner Mongolia Yili Industrial Group	615
601012.SH	Longi Green Energy Technology	614
600547.SH	Shandong Gold-Mining	609
600588.SH	Yonyou Network Technology	604
601088.SH	China Shenhua Energy	596
600309.SH	Wanhua Chemical Group	565
600703.SH	Sanan Optoelectronics	562
600570.SH	Hundsun Technologies	534
600009.SH	Shanghai International Airport	526
603288.SH	Foshan Haitian Flavouring and Food	506
600745.SH	Wingtech Technology	452
600690.SH	Haier Smart Home	432
601888.SH	China Tourism Group Duty-Free Corporation	390

For financial news, the news release time, summary, and text were collected from the Uqer database (https://uqer.datayes.com/). Uqer database provides a news API that collects news from 223 news websites, including reports on the company and coverage related to the macroeconomic environment. Here for each stock, we selected the news containing the stock's name as its stock news; we collected 496,014 pieces of



(a) The amount of news releasing in each year.

(b) The amount of news releasing in each month.

(c) Grid of scatter plot of the four market data.

Fig. 4 Data visualization for stock code 600276.SH

Table 2 Descriptive statistics of the involved market variables

Variable	Mean	Standard deviation	Minimum	Maximum
Daily return	0.0074	0.0099	− 0.0364	0.0351
Trading volume	71,034	94,172	51,572	903,875
Turnover rate	0.1533	0.2036	0.1831	1.7068
Total market cap	9,146,267	10,258,138	18,443,524	61,320,078

news for 37 stocks in total. To illustrate our data in detail, we randomly selected a stock of 600276. SH to visualize the data (see Fig. 4). Table 2 presents the basic statistical characteristics of the market variables. From Fig. 4a and b, we find that the amount of news in 2020 is the largest. Each year the amounts of news in January and February are relatively small owing to holidays. In addition, from Fig. 4c, we find that except for the linear relationship between the turnover rate and trading volume, other data are basically irrelevant.

Considering the spread of the Coronavirus in 2019, we further investigate if this issue impacts our data and the market trend. The total amount of news on COVID-19 in our sample was 3,467. The COVID-19 outbreak occurred at the end of December

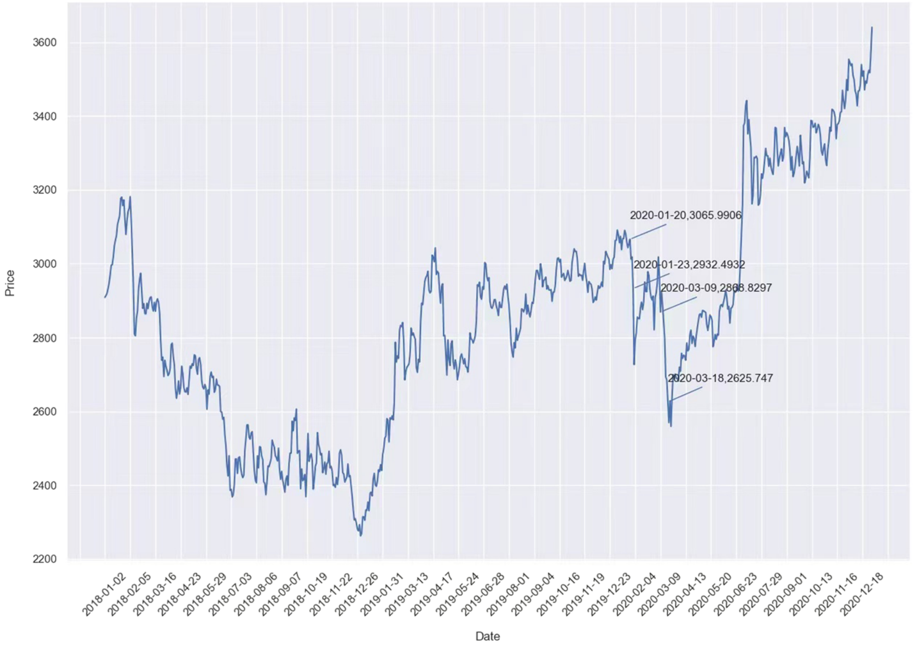


Fig. 5 The impact of COVID-19 on SSE 50 index

2019. However, the disease was not considered serious in the early stages; hence, it did not cause large stock market fluctuations. In January 20, 2020, a report pointed out that COVID-19 was a human-to-human fast-spreading communicable disease for which a cure had not been found; panic began to spread, and the stock market began to fall, as shown in Fig. 5. On January 23, 2020, Wuhan was announced to shut down, and the stock market fell sharply. Affected by the epidemic, the US stock market experienced four circuit breakers in March 2020, including March 9, March 12, March 16, and March 18. This also affected A-share investors, leading to a sharp drop in SSE 50 index. Choosing 2018 to 2020 as our research period can also help us explore whether the results of our model remain robust under special circumstances, such as epidemics.

The data must be normalized before being input into the model for training. Because the market data can take both positive and negative values, they are transformed by Equation (17) to satisfy the normalization requirements.

$$norm(md_t^k) = \frac{md_t^k}{max\{|md^k|\}}, k = 1, \ldots, 4, t = 1, \ldots, n. \tag{17}$$

Here, md_t^k denotes the k-th market variable on day_t and $max\{|md^k|\}$ refers to the maximum value of the k-th market variable. The values range from -1 to 1 after normalization. There is no need to normalize the news data because they are already normalized through the ltc method, as shown in Eq. (2).

In this study, the high-dimensional news vector obtained from the Chinese news text processing methods and the four market data were all input to the MVL-SVM algorithm. After training on the training set, the algorithm can choose the optimal kernel for each

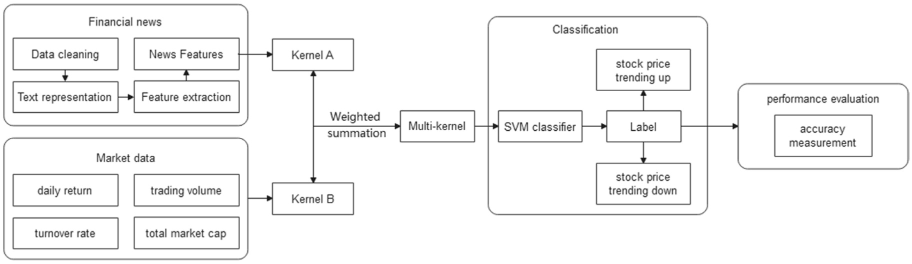


Fig. 6 Multi-view learning framework of stock price prediction model

data source and the optimal kernel combination weights. Therefore, combining multiple kernels can successfully fuse the multi-view data, and the new kernel can be input into the SVM classifier for classification. The framework of this multi-view stock price prediction model is illustrated in Fig. 6.

Experimental analysis and comparison

In this section, we consider the joint influence of structured market data and unstructured financial news and apply the MVL-SVM model to predict the stock price trend on the day or the next. The obtained sample is labeled according to the daily stock return r_{t+i}, as shown in Equation (18).

$$tag_{t+i} = \begin{cases} 1, if \ r_{t+i} > 0 \\ -1, if \ r_{t+i} \leq 0 \end{cases} \tag{18}$$

Here, r_{t+i} represents the daily return of the stock on day_{t+i}, tag_{t+i} indicates that the daily return on day_{t+i} is used to label the sample, and $i = 0$ means that the model aims to predict the stock price of the day, whereas $i = 1$ means to predict that of the next day. Because it is meaningless to use r_t to predict the price fluctuation of day_t, the market data can only be used to predict the next day's price movement, while the news released before the closing of the stock market on day_t can be used to predict the rise and fall of stock prices on both day_t and day_{t+1}. Therefore, $i = 0, 1$ is valid for financial news, and $i = 1$ is valid for market data.

As shown in Table 3, we used a confusion matrix to show the classification results, and the accuracy is given by

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{19}$$

Table 3 Confusion matrix

Model outcome	Positive (actual) ($P = TP + FN$)	Negative (actual) ($N = TN + FP$)
Positive	True positive (TP)	False positive (FP)
Negative	False negative (FN)	True negative (TN)

where TP means true positive, which indicates a stock price trends up and the model correctly predicts the upward trend, TN (true negative) means a stock price trends down and the downward trend is also correctly predicted while FN (false negative) occurs when the actual stock price is rising but the model mistakes it as a downward trend, and similarly, FP (false positive) refers to that the actual stock price is falling but the model mistakes it as an upward trend. Using Formula (19), we obtained the percentage of correctly predicted samples in the total sample.

Stock price prediction based on one-day news and four market data

Here, we build an MVL-SVM model to predict stock price movement using one-day market data and financial news. In this section, the prediction accuracy is compared with that of the classic SVM models to evaluate the predictive performance.

As explained in Section "Experimental analysis and comparison", news released before 15:00 on day_t can be used to predict the stock returns of day_t and day_{t+1}. However, market data on day_t can only be used to predict stock returns on day_{t+1}. Therefore, we consider two experimental settings: lag=0 (predict the price on the day of the news release) and lag=1 (predict the price on the next day of the news release).

In the case of lag=0, the sign of r_t is predicted by md_{t-1} and $news_t$, where $md_{t-1} = (r_{t-1}, tv_{t-1}, tr_{t-1}, mc_{t-1})$ represents the market data of day_{t-1} and $news_t = (w_{t,1}, w_{t,2}, \ldots, w_{t,M})$ represents the news vector on day_t, which is obtained from "Chinese news text processing". The input matrix of market data *MD* and News *News* is formulated in Equation (20), where each row of *MD* and *News* denotes a vector, and the labels of these vectors are also shown in Equation (20).

$$MD = \begin{bmatrix} r_1 & tv_1 & tr_1 & mc_1 \\ r_2 & tv_2 & tr_2 & mc_2 \\ . & . & . & . \\ . & . & . & . \\ r_{n-2} & tv_{n-2} & tr_{n-2} & mc_{n-2} \\ r_{n-1} & tv_{n-1} & tr_{n-1} & mc_{n-1} \end{bmatrix}, News = \begin{bmatrix} news_2 \\ news_3 \\ \cdots \\ \cdots \\ news_{n-1} \\ news_n \end{bmatrix}, Label = \begin{bmatrix} tag_2 \\ tag_3 \\ \cdots \\ \cdots \\ tag_{n-1} \\ tag_n \end{bmatrix} \tag{20}$$

When lag $= 1$, the sign of r_{t+1} is predicted by md_t and $news_t$. The input matrices *MD* and *News* and output vector *Label* are shown in Equation (21).

$$MD = \begin{bmatrix} r_1 & tv_1 & tr_1 & mc_1 \\ r_2 & tv_2 & tr_2 & mc_2 \\ . & . & . & . \\ . & . & . & . \\ r_{n-2} & tv_{n-2} & tr_{n-2} & mc_{n-2} \\ r_{n-1} & tv_{n-1} & tr_{n-1} & mc_{n-1} \end{bmatrix}, News = \begin{bmatrix} news_1 \\ news_2 \\ \cdots \\ \cdots \\ news_{n-2} \\ news_{n-1} \end{bmatrix}, Label = \begin{bmatrix} tag_2 \\ tag_3 \\ \cdots \\ \cdots \\ tag_{n-1} \\ tag_n \end{bmatrix} \tag{21}$$

For the SVM model, three cases are considered: (i) inputting only the market data (SVMMD), (ii) inputting only financial news (SVMFN), and (iii) inputting concatenated multi-view data of market data and financial news (SVMMV). Considering our sample size, we adopted three training/testing proportions: 60%/40%, 75%/25%, and 90%/10%. Because the model parameters will considerably affect the training performance, five-fold cross-validation and the grid search method are applied in the training set to select the optimal parameters. For the penalty term C, the initial values are set to $C = 10^a$ where we have $a \in \{-3, -2, -1, 0, 1, 2, 3\}$ and for the Gaussian kernel parameter γ, the

Table 4 The predicting accuracy of four models with different lags

Model	Introduction	Training (%)	Statistics	Lag0	Lag1
SVMMD	SVM based on market data	60	Average	–	0.5204
			Median	–	0.5147
		75	Average	–	0.5095
			Median	–	0.5050
		90	Average	–	0.5077
			Median	–	0.5000
SVMFN	SVM based on financial news	60	Average	0.5418	0.5291
			Median	0.5418	0.5240
		75	Average	0.5387	0.5328
			Median	0.5956	0.5275
		90	Average	0.6123	0.6170
			Median	0.6027	0.6180
SVMMV	SVM based on news and market data	60	Average	0.5533	0.5232
			Median	0.5411	0.5240
		75	Average	0.5545	0.5331
			Median	0.5464	0.5301
		90	Average	0.5536	0.5495
			Median	0.5556	0.5479
MVL-SVM	MVL-SVM based on news and market data	60	Average	0.8467	0.8504
			Median	0.8527	0.8562
		75	Average	0.8635	0.8588
			Median	0.8571	0.8634
		90	Average	0.8691	0.8741
			Median	0.8630	0.8767

initial values are $\gamma = 2^b$, where $b \in \{-4, -3, -2, -1, 0, 1, 2, 3, 4\}$. Therefore, for a nonlinear SVM, there were 63 combinations of parameters. A grid search can build a grid, where each node refers to a parameter combination. This method can traverse all the grid nodes to determine the optimal parameter combination of the model. We used five-fold cross-validation for each parameter combination to observe the model's performance. That is, the training set was divided into five parts, and each part was set as the validation set once, while the other four parts were used for training the model. Finally, the average prediction accuracy of the five experiments was considered as the model performance under this parameter combination. Among all parameter combinations, the one with the best performance was chosen to allocate the final model for independent testing. We used Python and the Sklearn package in this study to implement the model. The prediction accuracies for the 37 stocks are shown in Table 4.

Table 4 indicates that the MVL-SVM model always had the highest accuracy, despite the training set proportion. Surprisingly, the prediction accuracy of the MVL-SVM model was approximately 30% higher than that of the SVM model when both types of data were input. In particular, when we predict the price trend one day after the news release with a 90% training proportion, the average prediction accuracy of the SVM model based only on market data or financial news is 50.77% and 61.70%, respectively, whereas the MVL-SVM model can reach nearly 88% accuracy. This shows that the MVL-SVM model significantly outperformed the other baseline models.

For the three SVM models in Table 4, on the one hand, we can find that the predicting accuracy is improved when adding news data to the market data, indicating the news information contributes to price prediction. On the other hand, when the training set proportion was 60% and 75%, the accuracy of the SVMMV model was close to that of the SVMFN, but when the proportion was 90%, the SVM model with multi-view data seemed worse than the SVMFN model. However, when using the MVL-SVM model, there was an obvious improvement in the prediction accuracy. This shows that heterogeneous data, which can be used to predict the stock price alone, cannot obtain ideal results if they are simply and roughly concatenated and inputted into the SVM model because of the differences in their data characteristics, such as dimensions. This further demonstrates the advantages of the proposed model. Because the MVL-SVM model can learn and minimize the inconsistency from distinct views, it can effectively combine the data with different structures to make reasonable predictions on stock price trends. The experiment also shows that the prediction performance of MVL-SVM is stable because the average prediction accuracy of the MVL-SVM model only changes by 2.37% when the training proportion is changed, whereas that of the SVM model based only on financial news changes by 8.79%. Further, the training/testing proportion of 60%/40% refers to training from January 2, 2018, to October 23, 2019, when COVID-19 had not emerged. The testing period is from October 24, 2019, to December 31, 2020, when COVID-19 broke out and affected the stock market. The models with the other two training/testing proportions were trained using data on COVID-19. However, from Table 4, we can see that although our models were trained without COVID-19-related data, the average prediction accuracy was still approximately 85%. Considering the case of lag = 0 and lag = 1, the differences between the average prediction accuracy of the models with 75%/25% training/testing proportions and those with 60%/40% training/testing proportions, are only 1.68% and 0.84%, respectively, indicating that the spread of Coronavirus has little impact on the prediction performance of our model.

Here, we have further simplified the follow-up research. Because roughly concatenating heterogeneous data will weaken the training efficiency of the model, considering the SVMMV model is not necessary. Therefore, we only considered the prediction performance of the SVM models with single-view data and the MVL-SVM model. Additionally, as shown in Table 4, using one training/testing proportion was sufficient to illustrate the effectiveness of the different models. It appears that there is little difference between the prediction accuracies with lag = 0 and lag = 1. Considering that the result only based on market data with a 0-day lag is unavailable and for the convenience of explanation, it is better to choose the same lag period for multi-view data in the MVL-SVM model. Accordingly, in this section, we will only provide the results with a training proportion of 90% and consider the case with a 1-day lag.

Stock price prediction based on one-day news and daily return

We use four market variables to predict the stock price movement in the above study. Still, literature shows that many studies forecast stock price trends based only on historical stock daily returns (Jarrett and Schilling 2008; Sun 2017; Vo and Ślepaczuk 2022), showing that researchers pay more attention to the daily returns among the four market variables. Therefore, we changed the input of the four market data sequences to only

Table 5 The results of ADF test

Code	Dickey-Fuller	p-value	Code	Dickey-Fuller	p-value
600000	− 8.6404	< 0.01	600837	− 8.7622	< 0.01
600009	− 8.9744	< 0.01	600887	− 10.262	< 0.01
600016	− 8.9641	< 0.01	601012	− 7.8914	< 0.01
600028	− 10.166	< 0.01	601088	− 9.9771	< 0.01
600030	− 8.9646	< 0.01	601166	− 9.5602	< 0.01
600031	− 10.226	< 0.01	601186	− 9.4826	< 0.01
600036	− 9.6393	< 0.01	601211	− 8.6362	< 0.01
600048	− 9.7613	< 0.01	601288	− 9.6537	< 0.01
600050	− 9.777	< 0.01	601318	− 9.6763	< 0.01
600104	− 9.8113	< 0.01	601336	− 9.2443	< 0.01
600196	− 7.9495	< 0.01	601398	− 9.7018	< 0.01
600276	− 9.4207	< 0.01	601601	− 9.7303	< 0.01
600309	− 8.4273	< 0.01	601628	− 9.1452	< 0.01
600519	− 9.7002	< 0.01	601668	− 9.6028	< 0.01
600547	− 8.7423	< 0.01	601688	− 9.4543	< 0.01
600570	− 9.4835	< 0.01	601818	− 9.1276	< 0.01
600585	− 10.533	< 0.01	601857	− 8.6273	< 0.01
600588	− 9.4533	< 0.01	603288	− 8.8391	< 0.01
600703	− 9.3076	< 0.01			

daily return sequences to discuss the predictive ability of the MVL-SVM model in this case. Considering that ARIMA is a commonly used time-series model that can input historical return sequences for prediction, we use this model to build a benchmark based on single-view data in our study. The ARIMA model was implemented using the R language.

According to the ADF test (see Table 5), the p-values for all stocks are significant, so the series of daily returns are stationary. The orders of the ARIMA model can be determined using autocorrelation and partial autocorrelation diagrams. The results are presented in Table 6, and the average prediction accuracy for the 37 stocks is 49.61%.

Before evaluating the MVL-SVM model, we first observe whether the performance of SVM model changes if daily returns replace the market data. The experimental settings are as follows. We consider the case of lag = 1, that is, we use md_t and $news_t$ to predict r_{t+1}. Instead of using all four variables, we input r_t as md_t to conduct the experiment. The results of the SVM model based on daily returns (SVMDR) are presented in Table 7. The performance was compared with that of the SVM model based on four-market data (SVMMD).

As shown in Table 7, there are 22 of 37 stocks (59.46%) whose prediction accuracy of the SVMDR model is higher than that of the SVMMD model. The average prediction accuracies of SVMMD and SVMDR were 50.77% and 51.72%, respectively. This indicates that the SVMDR model is better than the SVMMD and ARIMA models and implies that historical data of daily stock returns contain most of the price fluctuation information among the market data.

The prediction accuracy of MVL-SVM is given in Table 8. For simplicity, we name the MVL-SVM model based on news and market data MVL-SVMMD and the MVL-SVM model based on news and the daily return MVL-SVMDR.

Table 6 The results of ARIMA models

Code	Model	Accuracy	Code	Model	Accuracy
600000	ARIMA(4,0,4)	46.99	600837	ARIMA(4,0,3)	48.63
600009	ARIMA(4,0,4)	50.27	600887	ARIMA(4,0,4)	48.63
600016	ARIMA(3,0,3)	49.73	601012	ARIMA(3,0,3)	51.93
600028	ARIMA(7,0,7)	49.18	601088	ARIMA(4,0,4)	48.09
600030	ARIMA(3,0,2)	50.56	601166	ARIMA(4,0,4)	50.82
600031	ARIMA(8,0,7)	52.46	601186	ARIMA(4,0,3)	49.18
600036	ARIMA(8,0,4)	46.45	601211	ARIMA(6,0,3)	56.28
600048	ARIMA(4,0,4)	49.18	601288	ARIMA(4,0,4)	35.52
600050	ARIMA(1,0,1)	40.98	601318	ARIMA(4,0,4)	50.82
600104	ARIMA(7,0,7)	47.54	601336	ARIMA(4,0,4)	43.17
600196	ARIMA(4,0,4)	48.09	601398	ARIMA(5,0,4)	46.99
600276	ARIMA(3,0,1)	46.99	601601	ARIMA(4,0,6)	56.28
600309	ARIMA(3,0,3)	51.92	601628	ARIMA(6,0,0)	53.01
600519	ARIMA(5,0,5)	52.46	601668	ARIMA(3,0,1)	54.64
600547	ARIMA(5,0,5)	52.20	601688	ARIMA(3,0,3)	45.90
600570	ARIMA(3,0,1)	53.01	601818	ARIMA(5,0,5)	50.82
600585	ARIMA(6,0,6)	50.27	601857	ARIMA(8,0,6)	48.09
600588	ARIMA(4,0,4)	56.28	603288	ARIMA(1,0,4)	48.09
600703	ARIMA(4,0,0)	54.10			

Table 7 The predicting accuracy of SVM models based on one-day news and market data or daily return

Code	SVMMD	SVMDR	Code	SVMMD	SVMDR
600000	0.5000	0.5167	600837	0.5328	0.5164
600009	0.4918	0.5328	600887	0.5230	0.5146
600016	0.5205	0.5492	601012	0.4897	0.5185
600028	0.5369	0.5246	601088	0.4848	0.4805
600030	0.5000	0.5992	601166	0.5350	0.4938
600031	0.4467	0.4836	601186	0.4508	0.5205
600036	0.4508	0.4754	601211	0.5821	0.5082
600048	0.5556	0.5597	601288	0.5697	0.5123
600050	0.5422	0.5822	601318	0.5000	0.5123
600104	0.4631	0.5369	601336	0.4754	0.5041
600196	0.4836	0.4631	601398	0.5287	0.5164
600276	0.5451	0.5082	601601	0.4713	0.4426
600309	0.4932	0.4977	601628	0.5164	0.5246
600519	0.5082	0.5533	601668	0.5205	0.5082
600547	0.5000	0.4916	601688	0.4836	0.4918
600570	0.5246	0.5615	601818	0.6066	0.5861
600585	0.5000	0.5082	601857	0.5041	0.5328
600588	0.4754	0.4836	603288	0.5123	0.5697
600703	0.4590	0.4549			

The average prediction accuracy of MVL-SVMMD and MVL-SVMDR is 87.41% and 87.89%, respectively, showing that MVL-SVMDR is slightly higher, and the performance of the MVL-SVMDR model is better than that of MVL-SVMMD model for nearly half of

Table 8 The predicting accuracy of MVL-SVM models based on one-day news and market data or daily return

Code	MVL-SVMMD	MVL-SVMDR	Code	MVL-SVMMD	MVL-SVMDR
600000	0.9041	0.9178	600837	0.7534	0.9178
600009	0.7808	0.7397	600887	0.8356	0.7945
600016	0.9041	0.9589	601012	0.7945	0.8630
600028	0.8904	0.9315	601088	0.9452	0.8219
600030	0.9583	0.9306	601166	0.8767	0.9315
600031	0.8630	0.8356	601186	0.9041	0.8904
600036	0.9452	0.9178	601211	0.8630	0.9178
600048	0.8219	0.8493	601288	0.8630	0.8767
600050	0.9315	0.9589	601318	0.9452	0.9315
600104	0.8767	0.8904	601336	0.8630	0.8904
600196	0.8630	0.8630	601398	0.9452	0.9315
600276	0.9041	0.8904	601601	0.8904	0.9041
600309	0.7937	0.7937	601628	0.9178	0.9452
600519	0.9315	0.9178	601668	0.8767	0.8493
600547	0.8630	0.8493	601688	0.9178	0.9041
600570	0.8904	0.8219	601818	0.8219	0.8082
600585	0.7945	0.8904	601857	0.9178	0.9589
600588	0.8219	0.8493	603288	0.7534	0.7671
600703	0.9178	0.8082			

the sample. Our experiments imply that after excluding the other three market variables, including total market cap, turnover rate, and trading volume, the information for prediction in the market data does not necessarily decrease and even becomes more effective due to the refinement of data and also confirms that many studies only use stock returns for prediction to be meaningful and reasonable.

Statistical analysis

The above analysis is based on a numerical comparison. Next, we further evaluated the model from the perspective of statistical analysis.

The nonparametric test (Demšar 2006) is a useful approach for classifier comparison over multiple datasets. Here, we employ two non-parametric tests, the Nemenyi test (Demšar 2006) and contrast estimation based on medians (García et al. 2010), to compare the relative performances of the pairwise algorithms. Nemenyi test can determine whether one algorithm yields competitive performance compared to the other methods. The algorithms were ranked according to their performance on multiple datasets, and the average rank of each algorithm was calculated. The performance of each pairwise model whose average rank differs by at least one critical difference (CD) is considered significantly different, and the critical difference can be calculated using the following formula:

$$CD = q_\alpha \sqrt{\frac{K(K+1)}{6N_{stock}}}, \tag{22}$$

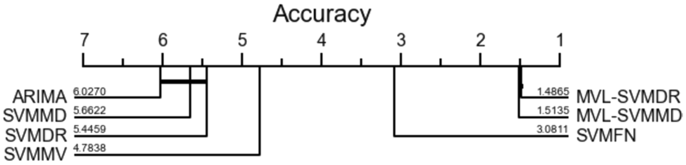


Fig. 7 Comparison of pairwise algorithms with the Nemenyi test

Table 9 Contrast estimation based on medians among all models

Model	MVL-SVMMD	MVL-SVMDR	SVMMV	SVMFN	SVMMD	SVMDR	ARIMA
MVL-SVMMD	0.0000	0.0003	0.3300	0.2663	0.3769	0.3659	0.3872
MVL-SVMDR	−0.0003	0.0000	0.3297	0.2660	0.3766	0.3656	0.3869
SVMMV	−0.3300	−0.3297	0.0000	−0.0637	0.0469	0.0359	0.0572
SVMFN	−0.2663	−0.2660	0.0637	0.0000	0.1106	0.0996	0.1210
SVMMD	−0.3769	−0.3766	−0.0469	−0.1106	0.0000	−0.0110	0.0103
SVMDR	−0.3659	−0.3656	−0.0359	−0.0996	0.0110	0.0000	0.0213
ARIMA	−0.3872	−0.3869	−0.0572	−0.1210	−0.0103	−0.0213	0.0000

where q_α is the critical value of Nemenyi test, K represents the number of algorithms involved, and N_{stock} is the number of stocks. In this section, we compare the performance of all the models involved, including ARIMA, SVMDR, SVMMD, SVMFN, SVMMV, MVL-SVMDR and MVL-SVMMD. Therefore, the K value in our test is seven, and we have $q_\alpha = 2.949$ at a significance level $\alpha = 0.05$. The accuracy of the CD diagram is shown in Fig. 7. In the figure, if the average ranks of pairwise models are within one CD, the two models will be linked in the CD diagram, whereas the performances of the unlinked models are thought to be significantly different. Clearly, the MVL-SVM algorithm is ranked first on average and is significantly different from the benchmark methods.

Moreover, contrast estimation based on medians can obtain a quantitative difference calculated from the medians between comparison algorithms over multiple datasets. Using this method, researchers can successfully estimate the difference between the performance of the two algorithms. Table 9 lists the results, where a positive value suggests that the row algorithm outperforms the corresponding column algorithm. Our model always achieves positive values concerning the baseline models.

Robust test

To further observe the ability of MVL-SVM to capture the joint impacts of news and market data on stock price trends over a certain period of time, we set the sliding window for news and market data from 1 to 5 days to predict the price trend with a 1-day lag after news releases. We define $\lambda = max\{T_1, T_2\}$, where T_1 denotes the news window, and T_2 denotes the market data window. When we obtain n days for the sample, considering the use of sliding windows, the actual length of the output is $n - \lambda$. Then, the input–output process of the model is formulated by equation (23), where md denotes the original market data sequence, and the dimension of the input matrix

Table 10 The predicting accuracy of MVL-SVMMD in different sliding windows

Sliding windows	Statistics	$T_1 = 1$	$T_1 = 2$	$T_1 = 3$	$T_1 = 4$	$T_1 = 5$
$T_2 = 1$	Average	0.8741	0.8108	0.7884	0.7497	0.7459
	Median	0.8767	0.8082	0.7945	0.7534	0.7397
$T_2 = 2$	Average	0.8842	0.8121	0.7808	0.7533	0.7366
	Median	0.8904	0.8082	0.7808	0.7534	0.7260
$T_2 = 3$	Average	0.8777	0.8272	0.7741	0.7588	0.7438
	Median	0.8904	0.8219	0.7671	0.7534	0.7397
$T_2 = 4$	Average	0.8758	0.8346	0.7703	0.7506	0.7357
	Median	0.8767	0.8356	0.7671	0.7534	0.7397
$T_2 = 5$	Average	0.8767	0.8117	0.7728	0.7485	0.7355
	Median	0.8767	0.8082	0.7671	0.7534	0.7260

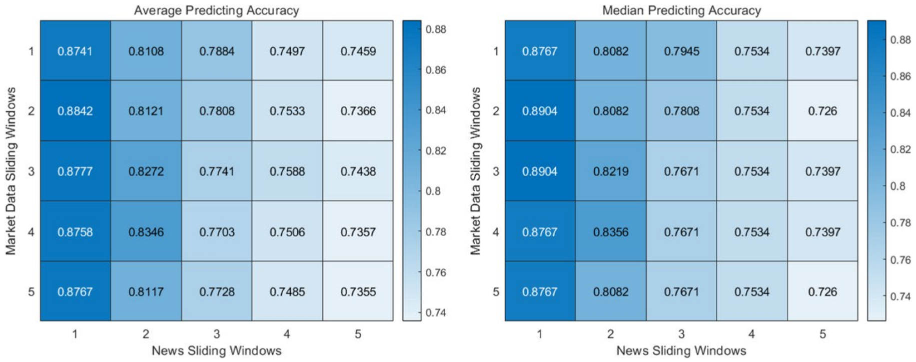


Fig. 8 The heatmaps of the predicting accuracy of MVL-SVMMD models

MD is expanded according to the sliding window. $news_t^{T_1}$ is obtained by gathering the news between day_t and day_{t-T_1+1} and inputting it into the Chinese news text processing algorithm. $w_{t,m}^{T_1}$ represents the weight of $word_m$ obtained from T_1-days of news.

$$md = \begin{bmatrix} md_1 \\ md_2 \\ \cdots \\ md_{n-1} \\ md_n \end{bmatrix} MD = \begin{bmatrix} md_\lambda & md_{\lambda-1} & md_{\lambda-2} & \cdots & md_{\lambda-T_2+1} \\ md_{\lambda+1} & md_\lambda & md_{\lambda-1} & \cdots & md_{\lambda-T_2+2} \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ md_{n-2} & md_{n-3} & md_{n-4} & \cdots & md_{n-T_2-1} \\ md_{n-1} & md_{n-2} & md_{n-3} & \cdots & md_{n-T_2} \end{bmatrix},$$
$$News = \begin{bmatrix} news_\lambda^{T_1} \\ news_{\lambda+1}^{T_1} \\ \cdots \\ \cdots \\ news_{n-2}^{T_1} \\ news_{n-1}^{T_1} \end{bmatrix}, where\ news_t^{T_1} = \begin{bmatrix} w_{t,1}^{T_1} \\ w_{t,2}^{T_1} \\ \cdots \\ \cdots \\ w_{t,M}^{T_1} \end{bmatrix}', Label = \begin{bmatrix} tag_{\lambda+1} \\ tag_{\lambda+2} \\ \cdots \\ \cdots \\ tag_{n-1} \\ tag_n \end{bmatrix}. \quad (23)$$

Notably, the case that both sliding windows are one day has been discussed in "Experimental test" section. The prediction results of MVL-SVMMD are listed in Table 10.

Heatmaps can show the results more clearly. In Fig. 8, the horizontal axis represents the sliding windows of financial news, whereas the vertical axis represents market data. We find that the color darkens when the sliding window of financial news changes from

Table 11 The predicting accuracy of SVMMD and SVMFN models in different sliding windows

Model	Statistics	T = 1	T = 2	T = 3	T = 4	T = 5
SVMMD	Average	0.5077	0.5238	0.5184	0.5096	0.5164
	Median	0.5000	0.5185	0.5168	0.5103	0.5185
SVMFN	Average	0.6170	0.6189	0.6118	0.6144	0.6338
	Median	0.6180	0.6164	0.6164	0.6027	0.6301

five days to one day. This means MVL-SVMMD can reach the highest prediction accuracy when using one-day news, indicating that it can capture the prompt impact of news on stock prices.

We also find that the average prediction accuracy of the MVL-SVMMD model shows little change for different sliding windows of market data, indicating that the impact of market data is persistent, and the largest prediction accuracy is up to 88%, showing that this model has good performance. Furthermore, we want to determine whether the model is superior to the SVM models and whether the single-view data are sufficient to predict stock prices. Hence, we used the SVMMD and the SVMFN model in different sliding windows to conduct the experiment and compare them with the MVL-SVM model. The results are illustrated in Table 11, indicating that the MVL-SVM model performed much better than the SVM models. The prediction accuracy of the SVMMD model was between 51% and 53%, that of the SVMFN model was between 61% and 64%, and that of the MVL-SVM model was between 73% and 88%, which was at least 10% higher than that of the two SVM models. This shows that the MVL-SVM model can successfully combine data with different structures and extract information about stock price rise and fall.

Furthermore, as mentioned in "Stock price prediction based on one-day news and daily return" section, removing the total market cap, turnover rate, and trading volume from the market data may improve the performance of the MVL-SVM model. Therefore, in this section, we attempt to refine the model in the same way and set the sliding windows for news and daily returns to 1–5 days separately.

Before discussing the results of MVL-SVM model, we observe the performance change of the SVM model in different sliding windows after replacing the four-market data with daily returns. From Table 12, we can see that in most cases, the average prediction accuracy of SVMDR is higher than that of the ARIMA and SVMMD models. Moreover, with an increase in the length of the sliding window, the average prediction accuracy of both models shows the same trend of first rising, then falling, and then rising (shown in Fig. 9). They reached the highest average prediction accuracy when the sliding window was 2 days.

Table 12 The predicting accuracy of SVMMD and SVMDR in different sliding windows

Model	Statistics	T = 1	T = 2	T = 3	T = 4	T = 5
SVMMD	Average	0.5077	0.5238	0.5184	0.5096	0.5164
	Median	0.5000	0.5185	0.5168	0.5103	0.5185
SVMDR	Average	0.5172	0.5262	0.5159	0.5150	0.5252
	Median	0.5146	0.5287	0.5205	0.5103	0.5267

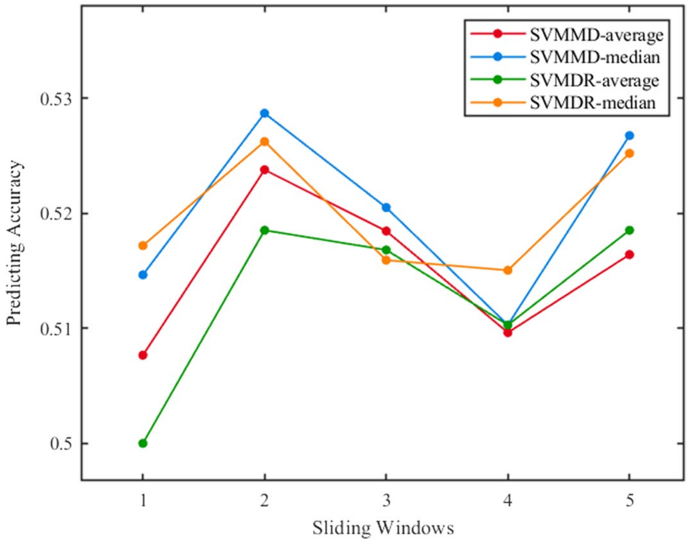


Fig. 9 Average and median predicting accuracy of SVMMD and SVMDR models

Because the performance of the SVM model is improved by changing the four-market data into daily returns, we infer that the same adjustment will lead to the same improvement as the MVL-SVM model. The prediction accuracies of the MVL-SVM model for different sliding windows are presented in Table 13.

We evaluated the validity of the MVL-SVM model based on daily returns and news by comparing its prediction results with those of the ARIMA model, SVMDR, and SVMFN models, as shown in Tables 6, 11, and 12. The prediction accuracy of the MVL-SVM

Table 13 The predicting accuracy of MVL-SVM models in different sliding windows

Sliding windows	Model	Statistics	$T_1 = 1$	$T_1 = 2$	$T_1 = 3$	$T_1 = 4$	$T_1 = 5$
$T_1 = 1$	MVL-SVMMD	Average	0.8741	0.8108	0.7884	0.7497	0.7459
		Median	0.8767	0.8082	0.7945	0.7534	0.7397
	MVL-SVMDR	Average	0.8789	0.8336	0.7905	0.7499	0.7423
		Median	0.8904	0.8356	0.7945	0.7534	0.7397
$T_1 = 2$	MVL-SVMMD	Average	0.8842	0.8121	0.7808	0.7533	0.7366
		Median	0.8904	0.8082	0.7808	0.7534	0.7260
	MVL-SVMDR	Average	0.8797	0.8197	0.7860	0.7673	0.7391
		Median	0.8904	0.8056	0.7808	0.7671	0.7397
$T_1 = 3$	MVL-SVMMD	Average	0.8777	0.8272	0.7741	0.7588	0.7438
		Median	0.8904	0.8219	0.7671	0.7534	0.7397
	MVL-SVMDR	Average	0.8815	0.8018	0.7845	0.7517	0.7426
		Median	0.8904	0.8219	0.7808	0.7534	0.7534
$T_1 = 4$	MVL-SVMMD	Average	0.8758	0.8346	0.7703	0.7506	0.7357
		Median	0.8767	0.8356	0.7671	0.7534	0.7397
	MVL-SVMDR	Average	0.8722	0.8231	0.7831	0.7621	0.7497
		Median	0.8767	0.8356	0.7808	0.7671	0.7397
$T_1 = 5$	MVL-SVMMD	Average	0.8767	0.8117	0.7728	0.7485	0.7355
		Median	0.8767	0.8082	0.7671	0.7534	0.7260
	MVL-SVMDR	Average	0.8777	0.8134	0.7804	0.7520	0.7449
		Median	0.8904	0.8219	0.7808	0.7534	0.7397

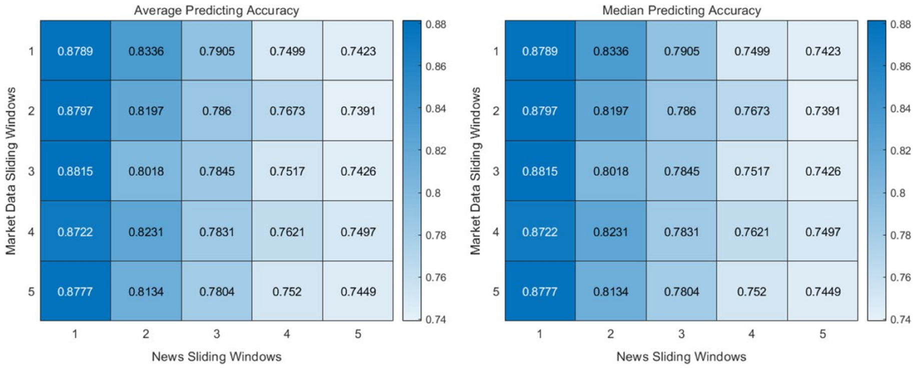


Fig. 10 The heatmaps of the predicting accuracy of MVL-SVMDR models

model with daily returns and news is much higher than that of the other three baseline models, which is the same as the results of the MVL-SVM model based on four market variables and news. The heatmaps of average and median prediction accuracies of the MVL-SVMDR model are shown in Fig. 10. We speculate from the heatmaps that the accuracy of the MVL-SVM models is related to the sliding windows of financial news because the prediction accuracy shows a downward trend with the increase in the length of the news sliding window. In particular, when the sliding window of financial news is set to one day, the model shows the best average and median accuracy performance.

In addition, by comparing the average and median prediction accuracy of MVL-SVMDR with those of MVL-SVMMD, we find that for most cases, the prediction accuracy of the MVL-SVM model based on news and daily returns is slightly higher than that based on news and four market data. This confirms our conjecture that using only daily returns in the market data can improve the model's performance.

Trading strategy

From the above experimental analysis, news and market data play an important role in stock price prediction, and the model trained by MVL-SVM has the best predictive performance. To test its effectiveness in practical applications, we design and evaluate a series of trading strategies based on this model in this section.

The trading setup is as follows. In Section "Stock price prediction based on oneday news and four market data", we adopt three training/testing proportions, including 60%/40%, 75%/25% and 90%/10%, respectively. To maximize the number of samples in the training and testing sets, we chose the middle training/testing proportion of 75%/25% to implement the trading strategy. We divide the data set into 75% training and 25% testing and train the models on the training set to predict future price trends of the 37 stocks from April 4, 2020, to December 31, 2020. We apply five-fold cross-validation and the grid search method on the training set to find the best parameters of the model that can achieve the highest average predicting accuracy on the validation set. Then, the model with adjusted parameters was applied to the testing set, and the prediction results were used as the signal to guide trading. If the stock price is predicted to rise on day_{t+1}, we buy it at the closing price on day_t and sell it at that on day_{t+1}. If the stock price is predicted to fall on day_{t+1}, no operation will be carried out. For convenience, we assume that the transaction has no cost, which is common in trading simulations. Moreover, from the analysis in Section "Robust test" the prediction performance of the MVL-SVMMD model varies with different sliding windows, and the model can achieve the highest prediction accuracy when the sliding windows of news and market data are one and two days, respectively. Therefore, we choose an optimal sliding window in the trading strategy.

All previously designed models are considered in this section for comparison, including ARIMA, SVMMD, SVMDR, SVMFN, SVMMV, MVL-SVMMD, and MV-SVMDR models. Additionally, we also introduce the momentum trading strategy, buy-and-hold strategy and randomly buy strategy to compare with the above seven models. For the momentum trading strategy, we consider absolute momentum, also known as price momentum. This strategy is based on the momentum effect, which holds that future returns are positively correlated with past returns. This strategy measures the average stock return over the past period and assumes that when the average stock return over the past period is positive, the stock price will rise.

In Sections "Experimental test" and "Robust test", we focus on prediction accuracy, which can measure the predicting ability of different models. However, investors are concerned about whether they can profit from these strategies. Therefore, we introduce the annual return rate (AR) to measure the model's profitability. The equation used is as follows:

$$AR = \frac{250}{t}\sum_{i=1}^{t} r(i), \tag{24}$$

where 250 is the total average number of trading days in a year, t is the number of trading days for testing during the simulation, and $r(i)$ represents the return obtained on trading day_i from the trading strategy, which is calculated by

$$r(i) = r_i \times signal_i, i = 1, \ldots, t, \tag{25}$$

where r_i represents the stock's daily return on trading day_i and $signal_i$ is a dummy variable representing the corresponding strategy signal. When $signal_i$ is 0, the strategy predicts that the price will fall on day_i and we take no action; on the contrary, when $signal_i$ is 1, the price is predicted to trend up on day_i so we buy it on day_{i-1} and have a short position on day_i. Therefore, the cumulative return from trading day_1 to day_t can be used to observe the real-time performance of each strategy.

Investment risk plays a vital role in evaluating the performance of a trading system; therefore, annual volatility (AV) and maximum drawdown (MDD) are measured to evaluate the risk. AV can be calculated using the following equation:

$$AV = \sqrt{\frac{250}{t-1}\sum_{i=1}^{t}[r(i) - \bar{r}]^2}, \tag{26}$$

where

$$\bar{r} = \frac{1}{t}\sum_{i=1}^{t} r(i). \tag{27}$$

MDD is the maximum loss from a peak to a trough before a new peak is attained. A lower MDD indicates a lower maximum possible loss during a trading period.

The annual sharp ratio (ASR) is often regarded as risk-adjusted profit and introduced for the stability assessment of trading systems. It is the ratio of average excess returns to the volatility of excess returns. The formula of ASR is as follows:

$$ASR = \sqrt{250} \times \frac{\bar{r_e}}{\sigma_e}, \tag{28}$$

where $\bar{r_e}$ and σ_e represent the average and volatility of daily excess returns during the simulation period, respectively.

$$\begin{aligned} \bar{r_e} &= \frac{1}{n}\sum_{i=1}^{n}[r(i) - r_f(i)], \\ \sigma_e &= \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}[r(i) - r_f(i) - \bar{r_e}]^2}. \end{aligned} \tag{29}$$

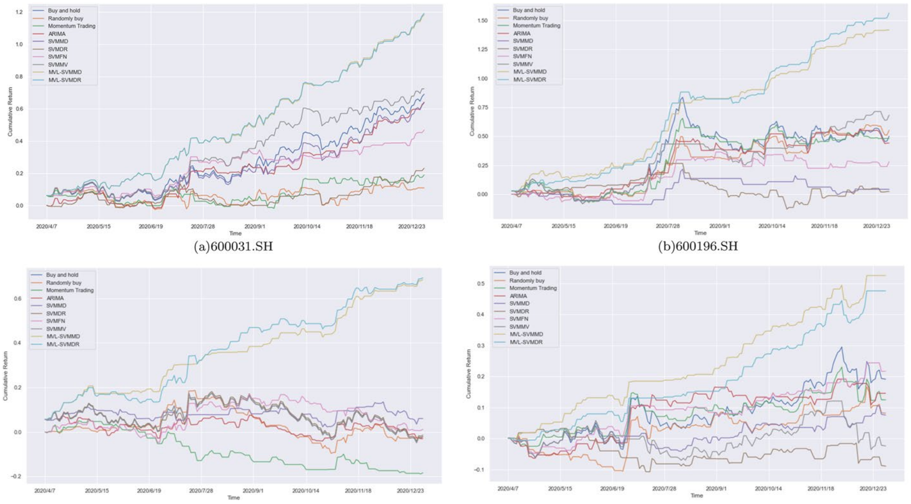
Here, $r_f(i)$ represents the risk-free interest rate on trading day_i. From Equation (29), a large ASR indicates that investors can obtain high profits under unit risk. This also implies a more stable trading system.

The four indicators of AR, ASR, MDD, and AV obtained from the different trading strategies are shown in Table 14. Moreover, we randomly selected four stocks to demonstrate the cumulative return curves in Fig. 11, which can be used to observe the performance of different strategies in detail.

From Fig. 11, we find that the MVL-SVM strategy proposed in this study is significantly more profitable than the other strategies. Although both SVM and MVL-SVM use multi-view heterogeneous data, the SVM strategy behaves in an unstable manner. In some cases, the performance of SVM based on multi-view data is third only to the two MVL-SVM strategies, such as stock 600031. SH and 600196.SH. However, in some cases, its performance is similar to or worse than that of the random buy strategy, such as stock 600585. SH and 601088.SH. In comparison, the MVL-SVM model exhibits good results in almost all cases. Even if the daily return is used to replace the four market data,

Table 14 The average AR, ASR, MDD and AV of different strategies

Strategy	AR (%)	ASR	MDD (%)	AV (%)
Buy and hold	33.45	0.86	19.59	31.52
Randomly buy	15.53	0.51	15.29	22.63
Momentum trading	17.39	0.63	16.31	24.94
ARIMA	21.52	0.67	14.42	22.54
SVMMD	14.31	0.83	7.68	14.18
SVMDR	15.20	0.49	12.70	18.92
SVMFN	19.24	0.75	10.68	16.66
SVMMV	24.79	0.90	8.21	14.89
MVL-SVMMD	96.62	4.73	6.57	20.58
MVL-SVMDR	104.26	4.75	6.76	21.84



(a)600031.SH

(b)600196.SH

(c)600585.SH

(d)601088.SH

Fig. 11 The cumulative return curves of different trading strategies on four randomly selected stocks

it can still obtain a much higher return than the buy-and-hold strategy, randomly buy strategy, momentum trading strategy, and other prediction-based strategies.

Then, we specifically analyze the performance of each strategy according to AR, ASR, AV, and MDD. Clearly, from Table 15, only MVL-SVM strategies (both MVL-SVMMD and MVL-SVMDR) can achieve positive returns for all stocks. And among the ten strategies, MVL-SVM strategies always have the highest AR except for one stock (code 600276.SH). The average AR of the MVL-SVMMD strategy is higher than that of the buy-and-hold and randomly buy strategies by 63.17% and 81.09%, respectively. There are 36 in 37 stocks (97.30%) whose ARs of MVL-SVM are the highest and surpass those of traditional algorithms based on single-view data, including momentum trading strategy, ARIMA, SVMMD, and SVMFN. However, although the average AR of SVM based on multi-view heterogeneous data is better than that of the single-view models, there are 24 in 37 stocks (64.86%) whose ARs of SVMMV are lower than those based on single-view models. This confirms that a rough connection between heterogeneous data leads to unsatisfactory results.

As shown in Table 16, there are 26 in 37 stocks (70.27%) whose ASR of the MVL-SVM strategy is at least twice that of other strategies. From the average value shown in Table 14, the average ASR of MVL-SVMMD is higher than that of the buy-and-hold strategy, randomly buy strategy, and momentum trading strategy by 3.87, 4.22 and 4.10, respectively. In particular, the average ASR based on time series and traditional machine learning strategies is lower than 1, whereas that of MVL-SVM is still above 4. This demonstrates that MVL-SVM has higher stability and can gain higher profits under unit risk.

Regarding risk, Table 14 shows that the MVL-SVM models have the lowest average MDD and relatively lower average AV among the strategies. For most stocks, the maximum loss from a peak to a trough of MVL-SVM is less than that of most traditional strategies including buy-and-hold, randomly buy, momentum trading, ARIMA, and SVMFN strategies (shown in Table 17), and for most stocks, the AV of MVL-SVM is lower than that of buy-and-hold, randomly buy, momentum trading and ARIMA strategies (see in Table 18). This implies that MVL-SVM has a relatively good capacity to send stable trading signals and a relatively lower risk.

Table 15 The results of AR

Code	Buy and hold (%)	Randomly buy (%)	Momentum trading (%)	ARIMA (%)	SVMMD (%)	SVMDR (%)	SVMFN (%)	SVMMV (%)	MVL-SVMMD (%)	MVL-SVMDR (%)
600000	0.77	1.76	13.19	− 14.75	12.27	− 1.36	11.53	− 0.43	60.89	54.69
600009	25.67	− 3.44	29.81	16.19	1.82	− 0.32	7.57	37.94	88.14	73.31
600016	− 4.5	8.37	13.17	− 2.85	10.8	0.00	− 3.06	0.00	39.47	47.86
600028	− 5.23	1.05	0.97	− 4.51	0.00	2.83	0.02	0.00	36.16	43.91
600030	38.63	56.15	27.87	34.78	− 0.56	26.5	38.63	45.33	140.03	145.39
600031	94.73	14.86	25.92	87.94	87.37	31.65	64.13	99.3	162.42	163.53
600036	47.14	49.32	33.41	8.64	24.58	31.12	0.00	36.73	113.32	123.54
600048	13.4	44.12	− 4.23	1.08	− 0.71	− 4.93	0.00	29.64	82.01	97.2
600050	− 20.99	− 9.15	− 19.48	− 23.3	− 13.12	− 1.88	0.00	− 11.75	57.13	55.83
600104	39.11	− 10.98	0.20	15.38	0.00	14.3	60.1	0.00	127.89	146.28
600196	68.75	75.89	66.05	60.59	5.73	3.12	38.55	93.8	194.91	215.02
600276	49.17	29.02	12.77	− 25.29	48.51	22.21	53.02	49.17	27.9	35.56
600309	115.02	75.66	37.87	57.62	64.01	89.95	87.47	133.92	138.37	173.9
600519	78.7	31.82	60.15	46.43	17.77	7.77	19.82	0.00	141.45	137.03
600547	− 4.83	− 3.2	− 55.29	69.5	17.39	− 6.69	5.37	− 14.03	108.19	107.73
600570	54.52	− 13.43	56.66	28.16	18.08	21.4	− 3.59	39.22	91.48	84.94
600585	− 2.77	− 3.17	− 25.14	− 4.15	8.34	− 1.89	1.67	− 2.77	94.09	95.22
600588	39.63	− 17.63	11.85	42.16	7.98	25.93	23.35	12.29	110.06	128.04
600703	29.08	− 19.47	3.81	49.06	− 11.77	− 4.76	0.00	0.00	99.73	142.35
600837	2.55	33.56	− 9.77	9.75	3.8	44.9	0.00	0.00	88.16	99.38
600887	56.9	24.59	10.81	39.77	56.9	36.75	45.26	56.9	101.18	109.99
601012	178.83	110.53	114.29	85.33	14.32	89.63	69.34	68.1	232.5	229.57
601088	26.29	11.29	17.15	20.02	10.39	− 12.25	29.79	− 3.31	72.23	65.45
601166	42.14	17.43	37.51	12.58	− 2.38	24.01	11.78	0.00	107.99	103.58
601186	− 24.53	− 8.9	− 14.08	− 25.99	− 4.36	0.00	− 36.59	0.00	39.43	40.72
601211	12.48	− 24.59	10.09	25.78	13.2	− 9.78	22.91	3.07	100.33	98.84

Table 15 (continued)

Code	Buy and hold (%)	Randomly buy (%)	Momentum trading (%)	ARIMA (%)	SVMMD (%)	SVMDR (%)	SVMFN (%)	SVMMV (%)	MVL-SVMMD (%)	MVL-SVMDR (%)
601288	− 3.96	− 8.23	5.25	− 15.67	0.00	− 13.03	0.00	0.00	22.51	21.77
601318	35.91	− 4.73	25.48	15.75	24.23	19.62	36.29	0.00	114.76	127.85
601336	48.83	45.88	27.97	− 35.75	20.68	28.81	− 2.43	64.67	139.06	144.92
601398	2.91	0.92	2.94	4.84	− 2.24	8.29	0.00	0.00	39.31	46.34
601601	50.44	37.16	27.54	64.93	33.26	− 14.58	24.61	0.00	103.87	106.68
601628	54.51	28.46	46.22	80.05	31.29	− 18.1	0.00	35.55	115.2	165.74
601668	− 2.23	9.29	− 6.55	31.31	5.57	2.8	0.00	0.00	46.3	56.07
601688	5.98	− 27.16	− 5.74	3.76	5.25	3.17	30.54	44.15	83.76	108.09
601818	22.36	− 9.42	26.70	42.37	17.11	19.56	11.12	0.00	81.37	76.88
601857	− 9.81	− 22.2	− 0.83	− 5.35	0.00	− 0.58	0.00	0.00	40.95	42.02
603288	82.2	53.04	38.73	0.00	4.07	98.2	64.75	99.75	132.33	142.34

Table 16 The results of ASR

Code	Buy and hold	Randomly buy	Momentum trading	ARIMA	SVMMD	SVMDR	SVMFN	SVMMV	MVL-SVMMD	MVL-SVMDR
600000	0.04	0.11	0.78	− 1.08	2.25	− 0.10	0.95	− 0.03	6.17	5.45
600009	0.79	− 0.16	1.37	0.67	0.96	− 1.17	0.45	1.76	3.54	3.02
600016	− 0.28	0.64	1.03	− 0.23	2.10	0.00	− 0.33	0.00	4.30	3.53
600028	− 0.36	0.09	0.09	− 0.36	0.00	0.46	0.01	0.00	4.28	5.04
600030	1.04	2.23	1.00	1.18	− 1.18	0.83	1.04	2.17	4.59	4.75
600031	2.63	0.54	0.99	3.23	2.47	1.36	2.63	3.11	5.38	5.41
600036	1.68	2.09	1.56	0.41	1.16	1.74	0.00	2.08	5.43	5.85
600048	0.46	2.10	− 0.18	0.06	− 0.12	− 0.31	0.00	1.87	4.41	5.45
600050	− 0.96	− 0.63	− 1.23	− 1.80	− 0.68	− 0.35	0.00	− 1.32	5.22	5.19
600104	1.01	− 0.39	0.01	0.60	0.00	0.62	2.61	0.00	4.57	5.15
600196	1.20	1.86	1.49	1.47	0.22	0.10	1.11	2.33	6.12	5.97
600276	1.62	1.39	0.50	− 0.91	1.59	0.90	1.78	1.62	1.20	1.60
600309	2.81	2.68	1.24	2.09	2.22	3.03	2.91	4.65	4.90	5.15
600519	2.97	1.78	3.04	2.16	2.59	0.45	1.54	0.00	7.39	7.11
600547	− 0.14	− 0.13	− 1.16	1.31	0.78	− 0.27	0.16	− 0.53	4.48	4.26
600570	1.37	− 0.45	2.09	0.95	2.05	0.70	− 0.13	1.03	3.39	3.31
600585	− 0.10	− 0.15	− 1.50	− 0.21	0.50	− 0.07	0.07	− 0.10	5.57	4.12
600588	0.90	− 0.57	0.27	1.38	1.50	0.99	0.73	0.83	5.54	4.47
600703	0.58	− 0.56	0.11	1.48	− 0.47	− 0.21	0.00	0.00	3.04	4.02
600837	0.07	1.34	− 0.36	0.44	0.20	2.33	0.00	0.00	5.00	4.96
600887	1.56	0.89	0.38	1.67	1.56	1.36	1.30	1.56	3.86	4.08
601012	3.43	2.85	2.65	2.38	1.63	2.4	2.31	1.95	5.93	5.85
601088	1.01	0.58	0.79	1.11	0.66	− 0.82	2.02	− 0.15	4.59	4.27
601166	1.63	0.97	1.88	0.70	− 0.18	1.29	0.85	0.00	6.24	5.58
601186	− 1.13	− 0.60	− 0.92	− 1.59	− 0.68	0.00	− 2.36	0.00	3.29	2.91
601211	0.44	− 1.35	0.46	1.33	0.60	− 0.90	1.41	0.15	4.55	4.47

Table 16 (continued)

Code	Buy and hold	Randomly buy	Momentum trading	ARIMA	SVMMD	SVMDR	SVMFN	SVMMV	MVL-SVMMD	MVL-SVMDR
601288	− 0.29	− 0.74	0.50	− 2.34	0.00	− 1.77	0.00	0.00	3.40	3.21
601318	1.32	− 0.25	1.28	0.88	1.07	0.96	1.53	0.00	7.29	8.17
601336	1.15	1.38	0.90	− 1.22	1.38	1.07	− 0.09	3.12	4.63	4.22
601398	0.17	0.07	0.19	0.34	− 1.17	0.59	0.00	0.00	5.62	4.81
601601	1.42	1.50	1.01	2.27	2.69	− 0.63	0.92	0.00	4.99	5.11
601628	1.10	0.74	1.33	2.32	0.67	− 0.57	0.00	1.23	4.63	5.73
601668	− 0.11	0.73	− 0.42	2.01	0.56	0.21	0.00	0.00	3.12	3.47
601688	0.19	− 1.22	− 0.23	0.16	2.04	0.18	1.49	1.92	3.58	4.53
601818	0.80	− 0.58	1.29	2.31	1.49	1.41	0.60	0	4.37	4.31
601857	− 0.61	− 2.46	− 0.06	− 0.52	0.00	− 1.17	0.00	0.00	5.69	5.75
603288	2.35	2.47	1.13	0.00	0.27	3.42	2.40	3.92	4.83	5.54

Table 17 The results of MDD

Code	Buy and hold (%)	Randomly buy (%)	Momentum trading (%)	ARIMA (%)	SVMMD (%)	SVMDR (%)	SVMFN (%)	SVMMV (%)	MVL-SVMMD (%)	MVL-SVMDR (%)
600000	20.44	14.28	11.50	14.33	1.63	12.01	7.99	14.06	2.19	2.19
600009	21.79	16.06	13.40	13.46	0.27	0.23	15.05	10.79	8.34	9.19
600016	16.91	8.88	5.33	11.67	1.54	0.00	10.97	0.00	3.50	5.54
600028	12.13	8.15	8.96	7.90	0.00	3.45	0.99	0.00	1.47	1.47
600030	17.61	7.46	16.54	20.17	0.40	14.13	17.61	6.84	7.24	7.24
600031	11.11	12.55	13.61	9.89	11.11	13.34	8.43	11.11	5.85	5.85
600036	14.25	9.68	10.49	15.71	11.12	8.28	0.00	6.59	6.34	5.65
600048	16.23	8.56	16.13	10.62	3.67	13.14	0.00	6.28	7.54	3.96
600050	22.52	14.95	18.30	20.89	22.81	4.39	0.00	10.54	2.57	2.57
600104	19.26	21.45	24.14	16.19	0.00	12.26	10.23	0.00	10.61	5.91
600196	39.47	18.26	24.22	15.78	20.36	33.73	14.54	19.83	6.97	9.54
600276	17.88	8.70	14.08	28.10	17.88	19.77	14.81	17.88	18.70	13.03
600309	13.02	9.75	9.30	11.85	13.02	8.29	8.59	6.59	7.04	9.37
600519	10.95	9.55	6.07	16.41	1.03	13.99	6.15	0.00	4.93	4.93
600547	29.45	24.93	42.57	23.91	16.03	22.16	30.17	27.52	8.60	6.83
600570	31.28	33.88	10.01	22.90	1.32	25.03	35.27	27.13	14.15	10.18
600585	20.96	24.93	22.63	12.70	9.66	20.95	15.55	20.96	3.79	6.81
600588	29.91	24.68	27.58	14.35	0.00	17.38	16.33	8.95	3.62	14.35
600703	29.67	34.29	21.52	14.50	17.87	23.63	0.00	0.00	13.01	9.74
600837	25.72	10.23	32.08	17.41	13.06	6.18	0.00	0.00	7.36	8.11
600887	15.16	12.38	17.99	11.58	15.16	8.31	16.83	15.16	10.54	10.26
601012	25.03	15.48	13.92	15.68	1.36	23.36	10.06	18.05	9.96	9.96
601088	13.03	11.03	10.80	8.13	8.02	13.23	4.86	14.99	7.21	7.21
601166	12.96	8.19	7.53	11.95	10.64	7.63	6.11	0.00	2.99	4.66
601186	21.43	12.38	13.83	22.46	6.49	0.00	26.80	0.00	3.34	4.52

Table 17 (continued)

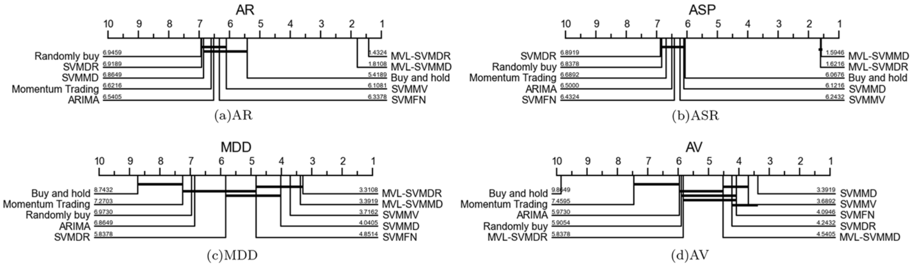
Code	Buy and hold (%)	Randomly buy (%)	Momentum trading (%)	ARIMA (%)	SVMMD (%)	SVMDR (%)	SVMFN (%)	SVMMV (%)	MVL-SVMMD (%)	MVL-SVMDR (%)
601211	20.55	24.15	22.74	7.80	15.09	11.05	6.89	15.05	6.44	6.44
601288	12.71	9.35	5.32	11.95	0.00	11.86	0.00	0.00	2.45	2.45
601318	13.50	16.94	14.84	11.78	12.56	11.54	17.45	0.00	2.33	1.34
601336	19.15	13.78	21.30	32.69	7.09	14.95	31.51	7.09	8.18	16.65
601398	12.84	6.71	7.65	7.74	1.63	7.09	0.00	0.00	2.83	2.83
601601	13.60	12.01	17.89	8.65	1.10	22.09	19.55	0.00	7.04	7.04
601628	29.08	23.06	13.46	16.68	27.72	30.40	0.00	15.88	6.35	6.26
601668	10.62	5.37	11.77	6.67	5.91	5.57	0.00	0.00	4.93	4.73
601688	29.13	28.58	26.56	22.75	0.39	17.13	9.21	13.73	6.14	6.14
601818	20.06	17.92	9.99	11.38	0.00	3.76	18.01	0.00	5.06	3.76
601857	12.98	16.51	7.88	6.86	0.00	0.42	0.00	0.00	0.47	0.22
603288	22.38	10.72	31.69	0.00	8.18	9.26	15.20	8.90	13.04	13.04

Table 18 The results of AV

Code	Buy and hold (%)	Randomly buy (%)	Momentum trading (%)	ARIMA (%)	SVMMD (%)	SVMDR (%)	SVMFN (%)	SVMMV (%)	MVL-SVMMD (%)	MVL-SVMDR (%)
600000	19.63	16.63	16.97	13.60	5.46	13.68	12.18	15.23	9.87	10.04
600009	32.34	22.06	21.82	24.31	1.90	0.27	16.99	21.53	24.88	24.26
600016	16.13	13.07	12.82	12.22	5.13	0.00	9.25	0.00	9.18	13.55
600028	14.54	11.25	11.31	12.53	0.00	6.21	1.43	0.00	8.44	8.71
600030	37.1	25.17	27.86	29.45	0.48	31.87	37.10	20.86	30.48	30.61
600031	36.09	27.67	26.10	27.25	35.37	23.25	24.37	31.95	30.2	30.21
600036	28.14	23.55	21.41	21.13	21.11	17.84	0.00	17.63	20.87	21.11
600048	28.86	20.97	23.57	18.27	5.82	16.10	0.00	15.81	18.62	17.84
600050	21.96	14.64	15.88	12.92	19.27	5.40	0.00	8.93	10.95	10.75
600104	38.72	27.98	29.88	25.53	0.00	22.89	23.05	0.00	28.01	28.38
600196	57.22	40.71	44.35	41.33	26.23	32.79	34.89	40.19	31.85	36.00
600276	30.42	20.91	25.52	27.82	30.42	24.80	29.74	30.42	23.18	22.27
600309	40.92	28.26	30.54	27.62	28.77	29.68	30.10	28.78	28.24	33.80
600519	26.53	17.90	19.77	21.47	6.86	17.20	12.90	0.00	19.13	19.27
600547	35.05	25.28	47.49	52.97	22.21	24.36	32.85	26.47	24.15	25.28
600570	39.80	30.09	27.06	29.64	8.84	30.76	27.52	38.25	27.00	25.65
600585	28.18	21.52	16.72	19.71	16.80	27.85	22.52	28.18	16.88	23.12
600588	43.86	30.91	43.66	30.58	5.33	26.24	31.84	14.72	19.87	28.64
600703	49.78	34.75	34.97	33.23	25.03	22.82	0.00	0.00	32.85	35.38
600837	34.24	24.99	26.84	21.94	19.14	19.27	0.00	0.00	17.64	20.05
600887	36.56	27.59	28.62	23.89	36.56	26.94	34.69	36.56	26.20	26.99
601012	52.06	38.72	43.16	35.91	8.79	37.37	30.03	35	39.18	39.25
601088	26.01	19.52	21.80	18.04	15.70	14.98	14.76	21.95	15.72	15.31
601166	25.85	17.9	19.97	17.99	13.08	18.56	13.92	0.00	17.31	18.55
601186	21.75	14.73	15.29	16.39	6.42	0.00	15.51	0.00	11.99	13.98
601211	28.28	18.25	22.16	19.41	22.07	10.90	16.22	20.23	22.06	22.11

Table 18 (continued)

Code	Buy and hold (%)	Randomly buy (%)	Momentum trading (%)	ARIMA (%)	SVMMD (%)	SVMDR (%)	SVMFN (%)	SVMMV (%)	MVL-SVMMD (%)	MVL-SVMDR (%)
601288	13.54	11.16	10.52	6.7	0.00	7.35	0.00	0.00	6.62	6.78
601318	27.16	18.76	19.91	17.82	22.74	20.38	23.77	0.00	15.75	15.64
601336	42.61	33.25	31.17	29.32	14.95	26.82	27.85	20.74	30.06	34.32
601398	16.88	14.21	15.38	14.29	1.91	14.16	0.00	0.00	7.00	9.64
601601	35.62	24.78	27.22	28.64	12.36	23.00	26.87	0.00	20.81	20.89
601628	49.51	38.56	34.80	34.43	46.68	32.00	0.00	28.94	24.86	28.95
601668	19.70	12.64	15.49	15.55	10.00	13.36	0.00	0.00	14.83	16.15
601688	32.17	22.23	24.98	23.48	2.57	17.85	20.44	22.94	23.41	23.85
601818	28.03	16.26	20.67	18.38	11.50	13.88	18.61	0.00	18.60	17.84
601857	16.01	9.03	12.78	10.37	0.00	0.50	0.00	0.00	7.19	7.31
603288	34.93	21.44	34.39	0.00	15.15	28.67	26.95	25.44	27.40	25.68



(a)AR

(b)ASR

(c)MDD

(d)AV

Fig. 12 Comparison of pairwise algorithms with the Nemeyi test in terms of each evaluation metrics

Moreover, we compared the practical performances of MVL-SVMMD and MVL-SVMDR. Table 15 shows that nearly 70% of stocks whose AR of MVL-SVMDR is higher than that of MVL-SVMMD. In general, the average AR of MVL-SVMDR is higher than that of MVL-SVMMD by 7.64%, as shown in Table 14, while the average values of the other indicators have little difference. This indicates that in contrast to the MVL-SVMMD model, the MVL-SVMDR model can provide more appropriate guidance for investors and help them obtain higher returns.

Finally, to evaluate the statistical significance of the advantages of MVL-SVM over other baseline strategies, we apply nonparametric statistical analyses again, including the Nemenyi test and contrast estimation based on medians. As demonstrated in Fig. 12, MVL-SVM strategies are ranked first in the performance of AR, ASR, and MDD, and are significantly different from other baseline strategies in the profitability metrics, including AR and ASR. This illustrates the profitability of our model and its ability to control risks, which can also be demonstrated by the positive values in rows “MVL-SVMMD” and “MVL-SVMDR” in Table 19. In addition, the results also show that the performance of MVL-SVM can be improved by transforming market data into daily returns, as MVL-SVMDR significantly outperforms MVL-SVMMD according to AR. In contrast, other metrics do not show significant differences.

The aforementioned trading strategies are based on a single stock. If an investor is optimistic about a certain stock and plans to invest in it, our strategy can help them find better time nodes for trading. However, investors often tend to invest in a basket of stocks, which implies that building an appropriate trading strategy for an investment portfolio is necessary.

We further take the 37 stocks of the sample as a basket to build a trading strategy with our proposed algorithm. Considering that some investors tend to construct portfolios according to a certain market index, we add a passive trading strategy to hold the SSE 50 index for comparative evaluation. Assuming that the initial capital is 1,000,000 CNY, which is used to invest equally in each stock, we use the same operation for each stock as the single-stock trading strategy. That is, if the algorithm sends an up signal for a certain stock, our system will spend 1/37 of the capital to buy it at the day’s closing price and have a short position at the closing price the next day; if it sends a down signal, no operation will be carried out. Trading signals are executed only when the total cash balance exceeds 100,000 CNY. The division of the training/testing set and the selection of sliding windows are the same as those in the single-stock trading strategy. Therefore, the return on trading day_i is transformed into

$$r(i) = \frac{1}{N_{stock}} \sum_{j=1}^{N_{stock}} r_{i,j} \times signal_{i,j}, i = 1, \ldots, n, \tag{30}$$

Table 19 Contrast estimation based on medians among all models

	Buy and hold	Randomly buy	Momentum trading	ARIMA	SVMMD	SVMDR	SVMFN	SVMMV	MVL-SVMMD	MVL-SVMDR
AR										
Buy and hold	0.000	0.162	0.127	0.098	0.133	0.138	0.109	0.075	− 0.669	− 0.702
Randomly buy	− 0.162	0.000	− 0.034	− 0.064	− 0.029	− 0.024	− 0.052	− 0.087	− 0.831	− 0.863
Momentum trading	− 0.127	0.034	0.000	− 0.029	0.006	0.011	− 0.018	− 0.052	− 0.796	− 0.829
ARIMA	− 0.098	0.064	0.029	0.000	0.035	0.040	0.011	− 0.023	− 0.767	− 0.800
SVMMD	− 0.133	0.029	− 0.006	− 0.035	0.000	0.005	− 0.024	− 0.058	− 0.802	− 0.834
SVMDR	− 0.138	0.024	− 0.011	− 0.040	− 0.005	0.000	− 0.029	− 0.063	− 0.807	− 0.839
SVMFN	− 0.109	0.052	0.0180	− 0.011	0.024	0.029	0.000	− 0.034	− 0.778	− 0.811
SVMMV	− 0.075	0.087	0.052	0.023	0.058	0.063	0.034	0.000	− 0.744	− 0.776
MVL-SVMMD	0.669	0.831	0.797	0.767	0.802	0.807	0.778	0.744	0.000	− 0.033
MVL-SVMDR	0.702	0.863	0.829	0.800	0.834	0.839	0.811	0.776	0.033	0.000
ASR										
Buy and hold	0.000	0.267	0.160	0.120	− 0.028	0.201	0.091	0.025	− 3.942	− 3.924
Randomly buy	− 0.267	0.000	− 0.107	− 0.147	− 0.295	− 0.066	− 0.176	− 0.242	− 4.209	− 4.191
Momentum trading	− 0.160	0.107	0.000	− 0.040	− 0.188	0.041	− 0.069	− 0.135	− 4.102	− 4.084
ARIMA	− 0.120	0.147	0.040	0.000	− 0.148	0.081	− 0.029	− 0.095	− 4.062	− 4.044
SVMMD	0.028	0.295	0.188	0.148	0.000	0.229	0.119	0.053	− 3.914	− 3.896
SVMDR	− 0.201	0.066	− 0.041	− 0.081	− 0.229	0.000	− 0.110	− 0.176	− 4.143	− 4.125
SVMFN	− 0.091	0.176	0.069	0.029	− 0.119	0.110	0.000	− 0.066	− 4.033	− 4.015
SVMMV	− 0.025	0.242	0.135	0.095	− 0.053	0.176	0.066	0.000	− 3.967	− 3.949
MVL-SVMMD	3.942	4.209	4.102	4.062	3.914	4.143	4.033	3.967	0.000	0.018
MVL-SVMDR	3.924	4.191	4.084	4.044	3.896	4.125	4.015	3.949	− 0.018	0.000
MDD										
Buy and hold	0.000	− 0.045	− 0.038	− 0.047	− 0.116	− 0.066	− 0.097	− 0.114	− 0.117	− 0.116
Randomly buy	0.045	0.000	0.008	− 0.002	− 0.071	− 0.021	− 0.051	− 0.069	− 0.071	− 0.071

Table 19 (continued)

	Buy and hold	Randomly buy	Momentum trading	ARIMA	SVMMD	SVMDR	SVMFN	SVMMV	MVL-SVMMD	MVL-SVMDR
Momentum trading	0.038	− 0.008	0.000	− 0.009	− 0.078	− 0.029	− 0.059	− 0.076	− 0.079	− 0.079
ARIMA	0.047	0.002	0.009	0.000	− 0.069	− 0.019	− 0.050	− 0.067	− 0.070	− 0.070
SVMMD	0.116	0.071	0.078	0.069	0.000	0.050	0.019	0.002	− 0.001	− 0.001
SVMDR	0.066	0.021	0.029	0.019	− 0.050	0.000	− 0.030	− 0.048	− 0.050	− 0.050
SVMFN	0.097	0.051	0.059	0.050	− 0.019	0.030	0.000	− 0.018	− 0.020	− 0.020
SVMMV	0.114	0.069	0.076	0.067	− 0.002	0.048	0.018	0.000	− 0.003	− 0.002
MVL-SVMMD	0.117	0.071	0.079	0.070	0.001	0.050	0.020	0.003	0.000	0.001
MVL-SVMDR	0.116	0.071	0.079	0.070	0.001	0.050	0.020	0.002	− 0.001	0.000
AV										
Buy and hold	0.000	0.162	0.127	0.098	0.133	0.138	0.109	0.075	− 0.669	− 0.702
Randomly buy	− 0.162	0.000	− 0.034	− 0.064	− 0.029	− 0.024	− 0.052	− 0.087	− 0.831	− 0.863
Momentum trading	− 0.127	0.034	0.000	− 0.029	0.006	0.011	− 0.018	− 0.052	− 0.796	− 0.829
ARIMA	− 0.098	0.064	0.029	0.000	0.035	0.040	0.011	− 0.023	− 0.767	− 0.800
SVMMD	− 0.133	0.029	− 0.006	− 0.035	0.000	0.005	− 0.024	− 0.058	− 0.802	− 0.834
SVMDR	− 0.138	0.024	− 0.011	− 0.040	− 0.005	0.000	− 0.029	− 0.063	− 0.807	− 0.839
SVMFN	− 0.109	0.052	0.018	− 0.011	0.024	0.029	0.000	− 0.034	− 0.778	− 0.811
SVMMV	− 0.075	0.087	0.052	0.023	0.058	0.063	0.034	0.000	− 0.744	− 0.776
MVL-SVMMD	0.669	0.831	0.796	0.767	0.802	0.807	0.778	0.744	0.000	− 0.033
MVL-SVMDR	0.702	0.863	0.829	0.800	0.834	0.839	0.811	0.776	0.033	0.000

where N_{stock} is the number of stocks, $r_{i,j}$ is the daily return of stock j on trading day_i and $signal_{i,j} \in \{0, 1\}$ is the corresponding strategy signal of stock j. Figure 13 shows that MVL-SVM models performed excellently during the simulation. From Table 20, we find that our models are significantly superior to other baseline models, not only in terms of profits but also in terms of risk control, indicating that they can pick out quality ones from a basket of stocks and invest them at an appropriate trading point to make profits.

Moreover, from the simulation results of MVL-SVMMD strategies based on a single stock, we find that the AR ranges from 22.51 to 232.5%, indicating that our model does not always yield high returns. Therefore, we conduct a simulation to determine whether our model can help investors gain profit when it underperforms. Therefore, for each stock, we calculate the minimum difference between MVL-SVMMD and other baseline strategies to select five stocks with the lowest differences; that is, MVL-SVMMD does not perform well on them. The five stocks, including 600276.SH, 601288.SH, 601668.SH, 600016.SH and 600570.SH are regarded as a new equally weighted portfolio and traded with 1/5 of the capital each time, with the same operation as above. Figure 14 and Table 21 show that the ARIMA strategy has the worst performance in the portfolio of five stocks. Although the buy-and-hold strategy has the highest cumulative return in July 2020, our model performs the best in the long run. This indicates that even if the MVL-SVM model does not significantly outperform the others, it can still help investors achieve considerable returns with relatively low risk.

According to the above three trading simulations, clearly, compared with the common trading strategies, such as momentum trading strategy and prediction-based strategies, such as ARIMA and traditional SVM strategies, MVL-SVM based on multi-view heterogeneous data shows excellent performances in profitability and risk control ability. Even if some prediction-based strategies fail to exceed the passive strategy of holding the SSE 50 index and the buy-and-hold strategy, MVL-SVM can achieve much better results than the passive strategies and other benchmarks. In particular, when the dimension of the input data is decreased, that is, the four market variables are changed to one (the daily return), the predictive and profitable effectiveness can be improved owing to the reduction in redundant information.

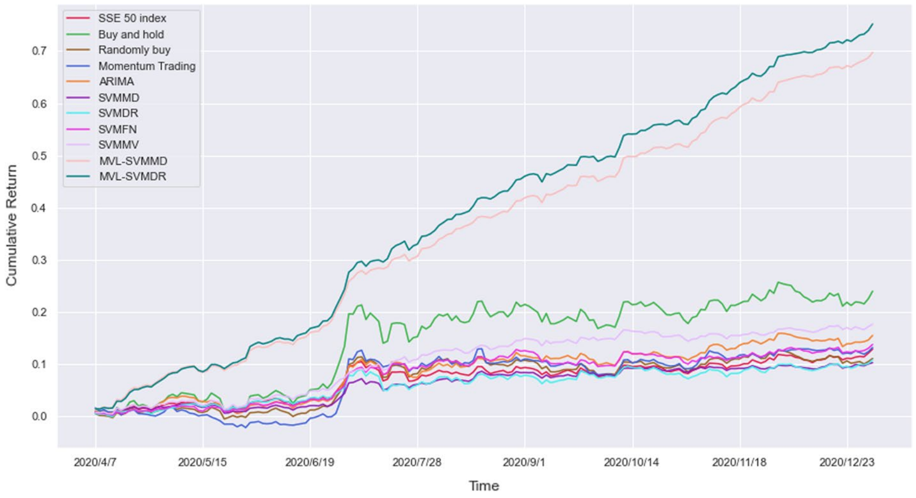


Fig. 13 The cumulative return curves of market simulations based on different trading strategies for portfolios

Table 20 Four metrics of different trading strategies for portfolios

Strategy	AR (%)	ASR (%)	MDD (%)	AV (%)
SSE 50 index	17.64	2.11	3.72	8.38
Buy and hold	32.88	1.77	7.19	18.53
Momentum trading	18.02	1.48	5.11	12.14
Randomly buy	15.15	1.5	3.90	10.11
ARIMA	21.22	2.24	3.41	9.49
SVMMD	14.04	2.31	2.22	6.08
SVMDR	14.77	1.86	4.02	7.95
SVMFN	18.84	2.41	3.22	7.81
SVMMV	24.17	3.48	2.18	6.95
MVL-SVMMD	95.82	10.43	1.36	9.19
MVL-SVMDR	103.31	10.35	1.69	9.98

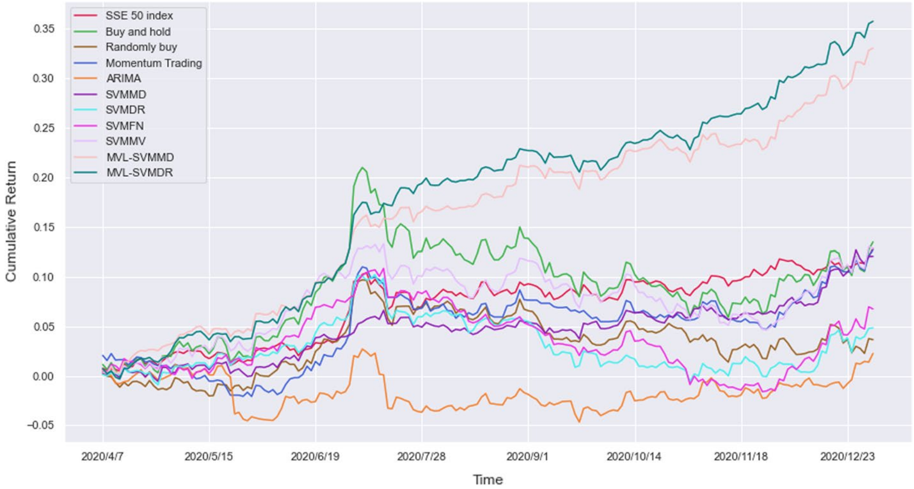


Fig. 14 The cumulative return curves of different trading strategies for an investment portfolio of 5 stocks

Table 21 Four metrics of different trading strategies for a portfolio of 5 stocks

Strategy	AR (%)	ASR	MDD (%)	AV (%)
SSE 50 index	17.64	2.11	3.72	8.38
Buy and hold	18.54	1.14	14.37	16.23
Randomly buy	5.02	0.50	8.10	10.01
Momentum trading	17.46	1.66	6.21	10.49
ARIMA	3.10	0.30	7.23	10.43
SVMMD	16.57	2.25	2.87	7.38
SVMDR	6.65	0.66	10.39	10.09
SVMFN	9.28	0.95	11.84	9.78
SVMMV	17.63	1.59	8.50	11.12
MVL-SVMMD	45.43	4.86	2.40	9.34
MVL-SVMDR	49.14	5.33	2.34	9.21

Conclusion

In this study, we propose a hybrid model for stock price prediction called MVL-SVM. It combines multi-view learning with a support vector machine to investigate the joint impact of financial news and market data on stock price movements. MVL-SVM can fuse multiple data sources directly with the multi-view learning algorithm and classify stock price fluctuations with a support vector machine, which enriches the information sources and reduces information loss in the fusion process.

In the experiment, we consider 37 constituent stocks in the SSE 50 index as the research object and use unstructured financial news and structured market data as inputs to predict the price trend of each stock. By comparing MVL-SVM with classic SVM models based on single-view and multi-view data, we found that roughly concatenating and inputting multi-view heterogeneous data yields unsatisfactory results because of the characteristic difference between the distinct views. However, the MVL-SVM model can learn and minimize the inconsistency from multiple data sources, and thus can demonstrate outstanding performance in this situation. Furthermore, we aimed to improve our model. Considering the importance of daily returns among the four market variables, we replace the four with daily return sequences to construct a new model and compare it with the ARIMA model and classic SVM models. It appears that MVL-SVM based on news and daily return sequences significantly outperforms the other baseline models. Its performance surpasses that of MVL-SVM based on news and the four market variables. This shows the important role of daily returns in market data and confirms the validity of the many studies that only use stock returns for research.

In the robustness test, we try to observe the model's ability to capture the joint impact of the multi-view data within a certain period, thus setting the sliding windows of news and market data to 1–5 days. It can be concluded from the results that MVL-SVM can capture the prompt impact of news on stock prices because the sliding window of news can influence the prediction accuracy of MVL-SVM, and the model based on 1-day news has the best performance. The comparison demonstrates that MVL-SVM surpasses the benchmarks by at least 10% accuracy, which is a meaningful improvement.

Finally, a series of trading strategies are constructed based on the predicting results of two MVL-SVM models, which are compared with other prediction-based strategies based on single-view and multi-view data, as well as three common strategies, including the buy-and-hold strategy, randomly buy strategy and momentum trading strategy. When building trading strategies for a basket of stocks, a passive strategy of holding the SSE 50 index was also considered for comparative evaluation. The results show that the MVL-SVM strategy has excellent profitability and risk-control performance in various scenarios. Moreover, its performance can be improved by changing the four market variables to daily return sequences.

In summary, from the prediction perspective, the proposed MVL-SVM model based on multi-view heterogeneous data can predict stock price movement more accurately than other models. From the perspective of trading strategy, this can help investors gain higher profits and have better risk control ability. But there are still some limitations. In this article, we only consider two information sources: market data and news. For future work, we can include more data sources in the model for discussion, such as online posts on social media, companies' financial statements, etc. In addition, when building SVM and MVL-SVM models, this study only considers two kernel functions, including linear and Gaussian kernels. In the future, we can construct the models by adding more kernel functions, such as the poly kernel and the sigmoid kernel function. As the proposed model has no restrictions on financial assets, we will further attempt to apply it to solve the problems of other financial assets.

Appendix

Variables and the corresponding definition are listed in Table 22.

Some related works are listed in Table 23.

Table 22 Variables and the corresponding definition

Variable	Definition
$f_{t,m}$	The occurrence frequency of $word_m$ in $news_t$
F_m	The occurrence frequency of $word_m$ in the news corpus
A	The number of times that word w and category c co-occur
B	The number of times that word w occurs without category c
C	The number of times that category c occurs without word w
D	The number of times that neither category c nor word w occurs
$P(c)$	The frequency of category $c \in \{-1, 1\}$ in news corpus
ξ_i	A slack variable
C	A penalty term controlling the cost to misclassification of samples
α_i	A Lagrangian multiplier corresponding to sample x_i
$k(x_i, x_j)$	The kernel function
γ	A Gaussian kernel parameter
β_m	The weight of kernel $k_m(x_i, x_j)$
o_t	The first order differencing of a data series z_t
p	The autoregression order
d	The differencing order
q	The moving average order
ϕ_i	The i-th autoregression parameter
θ_j	The j-th moving average parameter
ϵ_t	The error term at time t
r_t	Stock daily return on day_t
tv_t	Trading volume on day_t
tr_t	Turnover rate on day_t
mc_t	Market cap on day_t
$md_t = (r_t, tv_t, tr_t, mc_t)$	The market data including four market variable on day_t
md_t^k	The k-th market variable on day_t
$max\{\lvert md^k \rvert\}$	The maximum value of the k-th market variable
q_α	The critical value of Nemenyi test
K	The number of involved algorithms
N_{stock}	The number of stocks
T_1	The news window
T_2	The market data window
$w_{t,m}^{T_1}$	The weight of $word_m$ obtained from T_1-days of news
$r(i)$	The return obtained from the trading strategy on trading day_i
$signal_i \in \{0, 1\}$	A dummy variable representing the corresponding strategy signal on trading day_i
$\bar{r}_e$	The average of daily excess returns during the simulation period
σ_e	Volatility of daily excess returns during the simulation period
$r_f(i)$	The risk-free rate of interest on the trading day_i
$r_{i,j}$	The daily return of stock j on trading day_i
$signal_{i,j} \in \{0, 1\}$	The corresponding strategy signal of stock j on trading day_i

Table 23 Related works

References	Input set	Frequency	Sample period	Models	Prediction type	Evaluation	Trading strategies	Main findings
Mohan et al. (2019)	News: textual polarity, word2vec; historical prices	Daily	2013/02 to 2017/03	RNN-LSTM	Numerical	MAPE	No	Higher accuracy of stock price predictions
Li et al. (2020)	News: sentiment analysis; historical prices: technical analysis	Daily	2003/01 to 2008/03	LSTM	Numerical	Accuracy	No	Models combining news sentiments and prices outperform models only using either news sentiments or technical indicators
Kesavan et al. (2020)	News headlines and twitter: polarity score analysis; market data	Daily	2012 to 2020	LSTM	Numerical	Percentage error, accuracy and precision	No	The percentage error of the proposed method is about 3.05 which is lesser than other benchmarks
Li et al. (2016)	News: textual processing; stock prices: technical analysis	Intra-day	In year 2001	ELM, SVM, BP-NN	Directional	Accuracy and speed	Yes	RBF ELM and RBF SVM achieve faster prediction speed and higher prediction accuracy than BP-NN
Wang et al. (2019)	News: word vectors; K-line data	Daily	2017/01/01 to 2019/03/01	Hybrid time-series predictive neural network	Directional	Accuracy and MCC	No	The model's average accuracy is higher than others by nearly 5%
Ronaghi et al. (2022)	Twitter: word embedding; market data	Daily	2016/01/01 to 2020/07/30	CNN/CNN-LSTM	Directional	Sensitivity, specificity and accuracy	Yes	The accuracy of the proposed model is more than 66%
Lin et al. (2022)	News titles: word embedding; historical data	Daily	2008/08/08 to 2016/07/01	Spatial-temporal attention-based convolutional network	Numerical	MSE and correlation coefficient	No	The model outperforms CNNs and LSTMs in stock regression tasks
Lv et al. (2021)	Stock data	Daily	2018/01/02 to 2019/12/31	Multi-view RBF neural network	Directional	Accuracy	No	Generalization performance of the proposed model has a certain improvement compared with the traditional classification model
Shynkevich et al. (2015b)	News: text representation	Daily	2009/09/01 to 2014/09/01	Multiple kernel learning	Directional	Accuracy	No	Multiple kernel learning achieves the highest prediction accuracy

Table 23 (continued)

References	Input set	Frequency	Sample period	Models	Prediction type	Evaluation	Trading strategies	Main findings
Shynkevich et al. (2015a)	News: text representation	Daily	2009/01/01 to 2014/01/01	Multiple kernel learning	Directional	Accuracy	No	The proposed model achieves 79.14% accuracy and 0.48% return
Deng et al. (2011a)	News and comments: frequency and sentiment analysis; stock data: technical analysis	Daily	2006/01/01 to 2008/08/15.	Multi-view learning: multiple kernel learning regression framework	Numerical	MAE, MAPE and RMSE	No	Lower MAE, MAPE and RMSE
Deng et al. (2011b)	News and comments: frequency; comments: average/std response time, average/std comments length, number of Loyals/Outliers, number of comments rank, number of early/later responser; historical prices	Daily	2006/01/01 to 2008/08/15	Multi-view learning: multiple kernel learning regression framework	Numerical	MAE, MAPE and RMSE	No	Lower MAE, MAPE and RMSE
Deng et al. (2014)	News and comments: frequency, SMA of frequency; comments: average and standard deviation of comment length; trading data: technical analysis	Daily	2006/01/01 to 2008/08/15	Multi-view learning: multiple kernel learning regression and genetic algorithm	Numerical	RMSE, accumulated return and sharpe ratio	Yes	Good profits and consistently positive sharpe ratios
Li et al. (2011)	News: text representation; market data: technical analysis	Intra-day & inter-day	In year 2001	Multi-kernel learning	Directional	Accuracy	No	The proposed model achieves 4 best-results in 6 experiments
Wang et al. (2012)	News: TF-IDF; trading volume: technical analysis	Intra-day	In year 2001	Service-oriented multi-kernel learning approach	Directional	Accuracy	No	MKL-NP can obtain 3 second-best-results and MKL-NPV can obtain 4 best-results in 5 experiments

Abbreviations

SVM	Support vector machine
SVMMD	SVM based on market data
SVMFN	SVM based on financial news
SVMMV	SVM based on news and market data
MVL-SVM	Model integrating multi-view learning with SVM
MVL-SVMMD	MVL-SVM model based on news and market data
MVL-SVMDR	MVL-SVM model based on news and daily returns
AR	Annual return rate
ASR	Annual share ratio
MDD	Maximum drawdown
AV	Annual volatility
CD	Critical difference

Acknowledgements
We are thankful to reviewers for their valuable comments to improve the manuscripts

Author Contributions
All the authors were involved in the research that led to the article and in its writing. All the authors read and approved the final manuscript

Funding
This research was partly supported by National Natural Science Foundation of China (No. 71771204, 72231010) and the Fundamental Research Funds for the Central Universities (No. E0E48946X2).

Availability of data and materials
The market data used in this article are available in the WIND database, https://www.wind.com.cn/. And financial news is available in the Uqer database, https://uqer.datayes.com/.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 28 June 2022 Accepted: 24 May 2023
Published online: 29 February 2024

References

Bildirici M, Ersin ÖÖ (2009) Improving forecasts of GARCH family models with the artificial neural networks: an application to the daily returns in Istanbul stock exchange. Expert Syst. Appl. 36(4):7355–7362

Blum A, Mitchell T (1998) Combining labeled and unlabeled data with co-training. In: Proceedings of the eleventh annual conference on computational learning theory, pp 92–100

Box GE, Jenkins GM, Reinsel GC, Ljung GM (2015) Time series analysis: forecasting and control. Wiley, Hoboken

Buckley C, Salton G, Allan J, and Singhal A (1995) Automatic query expansion using SMART: TREC 3. NIST special publication sp, pp 69–69

Cao L, Tay F (2001) Financial forecasting using support vector machines. Neural Comput Appl 10(2):184–192

Cao LJ, Tay FEH (2003) Support vector machine with adaptive parameters in financial time series forecasting. IEEE Trans Neural Netw 14(6):1506–1518

Ceci M, Pio G, Kuzmanovski V, Džeroski S (2015) Semi-supervised multi-view learning for gene network reconstruction. PLoS ONE 10(12):e0144031

Chen K, Zhou Y, and Dai F (2015) A LSTM-based method for stock returns prediction: a case study of china stock market. In: 2015 IEEE international conference on big data (big data), pp 2823–2824. IEEE

Collins M and Singer Y (1999) Unsupervised models for named entity classification. In: 1999 joint SIGDAT conference on empirical methods in natural language processing and very large corpora

Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297

Dasgupta S, Littman ML, and Mcallester DA (2001) PAC generalization bounds for co-training. In: Advances in neural information processing systems 14 [neural information processing systems: natural and synthetic, NIPS 2001, 3–8 Dec 2001, Vancouver, British Columbia, Canada], pp 375–382

de Sa VR (1994) Learning classification with unlabeled data. Morgan Kaufmann Publishers, Burlington, pp 112–112

Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30

Deng N, Tian Y, Zhang C (2012) Support vector machines: optimization based theory, algorithms, and extensions. CRC Press, Boca Raton

Deng S, Mitsubuchi T, Sakurai A (2014) Stock price change rate prediction by utilizing social network activities. Sci World J. https://doi.org/10.1155/2014/861641

Deng S, Mitsubuchi T, Shioda K, Shimada T, and Sakurai A (2011a) Combining technical analysis with sentiment analysis for stock price prediction. In: 2011 IEEE ninth international conference on dependable, autonomic and secure computing, pp 800–807. IEEE

Deng S, Mitsubuchi T, Shioda K, Shimada T, and Sakurai A (2011b) Multiple kernel learning on time series data and social networks for stock price prediction. In: 2011 10th international conference on machine learning and applications and workshops, vol 2, pp 228–234. IEEE

Dyck A , Zingales L (2003) The media and asset prices. Technical report, Working Paper, Harvard Business School, Harvard

Fischer T, Krauss C (2018) Deep learning with long short-term memory networks for financial market predictions. Eur J Oper Res 270(2):654–669

García S, Fernández A, Luengo J, Herrera F (2010) Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power. Inf Sci 180(10):2044–2064

Gidofalvi G, Elkan C (2001) Using news articles to predict stock price movements. University of California, San Diego, p 17

Hammad AAA, Ali SMA, Hall EL (2007) Forecasting the Jordanian stock price using artificial neural network. Intell Eng Syst Through Artif Neural Netw 17:1–6

Han Z, Zhang C, Fu H, Zhou JT (2022) Trusted multi-view classification with dynamic evidential fusion. IEEE Trans Pattern Anal Mach Intell. https://doi.org/10.1109/TPAMI.2022.3171983

Jarrett JE, Schilling J (2008) Daily variation and predicting stock market returns for the frankfurter börse (stock market). J Bus Econ Manag 9(3):189–198

Kanwar N (2019) Deep reinforcement learning-based portfolio management. PhD thesis, The University of Texas at Arlington

Kesavan M, Karthiraman J, Ebenezer RT, Adhithyan S (2020) Stock market prediction with historical time series data and sentimental analysis of social media data. In: 2020 4th international conference on intelligent computing and control systems (ICICCS)

Kim KJ (2003) Financial time series forecasting using support vector machines. Neurocomputing 55(1/2):307–319

Kolarik T, Rudorfer G (1994) Time series forecasting using neural networks. ACM Sigapl Apl Quote Quad 25(1):86–94

Lavrenko V, Schmill M, Lawrie D, Ogilvie P, Jensen D, Allan J (2000) Language models for financial news recommendation. In: Proceedings of the ninth international conference on Information and knowledge management, pp 389–396

Li X, Xie H, Wang R, Cai Y, Cao J, Wang F, Min H, Deng X (2016) Empirical analysis: stock market prediction via extreme learning machine. Neural Comput Appl 27(1):67–78

Li X, Wu P, Wang W (2020) Incorporating stock prices and news sentiments for stock market prediction: a case of Hong Kong. Inf Process Manag 57:102212

Li H, Dagli CH, Enke D (2007) Short-term stock market timing prediction under reinforcement learning schemes. In: 2007 IEEE international symposium on approximate dynamic programming and reinforcement learning, pp 233–240. IEEE

Lin CT, Wang YK, Huang PL, Shi Y, Chang YC (2022) Spatial-temporal attention-based convolutional network with text and numerical information for stock price prediction. Neural Comput Appl. https://doi.org/10.1007/s00521-022-07234-0

Li X, Wang C, Dong J, Wang F, Deng X, Zhu S (2011) Improving stock market prediction by integrating both market news and stock prices. In: International conference on database and expert systems applications, pp 279–293. Springer, Berlin

Long W, Lu Z, Cui L (2019) Deep learning-based feature engineering for stock price movement prediction. Knowl-Based Syst 164:163–173

Long W, Song L, Tian Y (2019) A new graphic kernel method of stock price trend prediction based on financial news semantic and structural similarity. Expert Syst Appl 118:411–424

Lv B, Jiang Y, Li Q (2021) Prediction of short-term stock price trend based on multiview RBF neural network. Intell Neuroscience. https://doi.org/10.1155/2021/8495288

Meesad P and Thanh H (2014) Stock market trend prediction based on text mining of corporate web and time series data. J Adv Comput Intell Intell Inf. https://doi.org/10.20965/jaciii.2014.p0022

Mittermayer M-A, Knolmayer GF (2006) Newscats: a news categorization and trading system. In: Sixth international conference on data mining (ICDM'06), pp 1002–1007. IEEE

Mohan S, Mullapudi S, Sammeta S, Vijayvergia P, Anastasiu DC (2019) Stock price prediction using news sentiment analysis. In: 2019 IEEE Fifth international conference on big data computing service and applications (BigDataService), pp 205–208. IEEE

Ronaghi F, Salimibeni M, Naderkhani F, Mohammadi A (2022) COVID19-HPSMP: COVID-19 adopted hybrid and parallel deep information fusion framework for stock price movement prediction. Expert Syst Appl 187:115879

Salton G, Buckley C (1988) Term-weighting approaches in automatic text retrieval. Inf Process Manag 24(5):513–523

Salton G, Wong A, Yang C-S (1975) A vector space model for automatic indexing. Commun ACM 18(11):613–620

Schumaker RP, Chen H (2009) Textual analysis of stock market prediction using breaking financial news: the AZFin text system. ACM Trans Inf Syst (TOIS) 27(2):1–19

Shiller RJ (2015) Irrational exuberance. Princeton University Press, Princeton

Shynkevich Y, McGinnity TM, Coleman S, Belatreche A (2015a) Predicting stock price movements based on different categories of news articles. In: 2015 IEEE symposium series on computational intelligence, pp 703–710. IEEE

Shynkevich Y, McGinnity TM, Coleman S, Belatreche A (2015b) Stock price prediction based on stock-specific and sub-industry-specific news articles. In: 2015 International joint conference on neural networks (IJCNN), pp 1–8. IEEE

Suhail K, Sankar S, Kumar AS, Nestor T, Soliman NF, Algarni AD, El-Shafai W, Abd El-Samie FE (2022) Stock market trading based on market sentiments and reinforcement learning. CMC-Comput Mater Continua 70(1):935–950

Sun K et al (2017) Equity return modeling and prediction using hybrid ARIMA-GARCH model. Int J Financ Res 8(3):154–161

Sun S, Yu M, Shawe-Taylor J, Mao L (2022) Stability-based PAC-bayes analysis for multi-view learning algorithms. Inf Fusion 86:76–92

Sun L, Ceran B, Ye J (2010) A scalable two-stage approach for a class of dimensionality reduction techniques. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 313–322

Tan Z, Quek C, Cheng PY (2011) Stock trading with cycles: a financial application of ANFIS and reinforcement learning. Expert Syst Appl 38(5):4741–4755

Vo N, Ślepaczuk R (2022) Applying hybrid ARIMA-SGARCH in algorithmic investment strategies on S &P500 index. Entropy 24(2):158

Wang H, Zhou Z (2021) Multi-view learning based on maximum margin of twin spheres support vector machine. J Intell Fuzzy Syst 40(6):11273–11286

Wang Y, Liu H, Guo Q, Xie S, Zhang X (2019) Stock volatility prediction by hybrid neural network. IEEE Access 7:154524–154534

Wang F, Liu L, Dou C (2012) Stock market volatility prediction: a service-oriented multi-kernel learning approach. In: 2012 IEEE ninth international conference on services computing, pp 49–56. IEEE

White H (1988) Economic prediction using neural networks: the case of IBM daily stock returns. In: ICNN, vol 2, pp 451–458

Wüthrich B, Permunetilleke D, Leung S, Lam W, Cho V, Zhang J (1998) Daily prediction of major stock indices from textual www data. Hkie Trans 5(3):151–156

Xiao Y, Li X, Liu B, Zhao L, Kong X, Alhudhaif A, Alenezi F (2022) Multi-view support vector ordinal regression with data uncertainty. Inf Sci 589:516–530

Xu C, Tao D, Xu C (2015) Multi-view learning with incomplete views. IEEE Trans Image Process 24(12):5812–5825

Xu C, Tao D, Xu C (2013) A survey on multi-view learning. arXiv preprint arXiv:1304.5634

Yan X, Hu S, Mao Y, Ye Y, Yu H (2021) Deep multi-view learning methods: a review. Neurocomputing 448:106–129

Yang Y, Pedersen JO (1997) A comparative study on feature selection in text categorization. In: ICML, vol 97, pp 35. CiteSeer

Yarowsky D (1995) Unsupervised word sense disambiguation rivaling supervised methods. In: 33rd annual meeting of the association for computational linguistics, pp 189–196

Zhang T, Liu S, Xu C, Lu H (2010) Human action recognition via multi-view learning. In: Proceedings of the second international conference on internet multimedia computing and service, pp 23–28

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.