Training ensembles of faceted classification models for quantitative stock trading

(Article begins on next page)

Noname manuscript No.
(will be inserted by the editor)

Training ensembles of faceted classification models for quantitative stock trading

Luca Cagliero · Paolo Garza · Giuseppe Attanasio · Elena Baralis

Received: date / Accepted: date

Abstract Forecasting the stock markets is among the most popular research challenges in finance. Several quantitative trading systems based on supervised machine learning approaches have been presented in literature. Recently proposed solutions train classification models on historical stock-related datasets. Training data include a variety of features related to different facets (e.g., stock price trends, exchange volumes, price volatility, news and public mood). To increase the accuracy of the predictions, multiple models are often combined together using ensemble methods. However, understanding which models should be combined together and how to effectively handle features related to different facets within different models are still open research questions. In this paper we investigate the use of ensemble methods to combine faceted classification models for supporting stock trading. To this aim, separate classification models are trained on each subset of features belonging to the same facet. They produce trading signals tailored to a specific facet. Signals are then combined together and filtered to generate a unified, multi-faceted recommendation. The experimental validation, performed on different markets and in different conditions, shows that, in many cases, some of the faceted models perform as good as or better than models trained on a mix of different features. An ensemble of the faceted recommendations makes the generated trading signals more profitable yet robust to draw-down periods.

Keywords Quantitative stock trading · Classification · Ensemble methods · Financial application

PACS PACS code1 · PACS code2 · more

Mathematics Subject Classification (2010) 68U35

L. Cagliero · P. Garza · G. Attanasio · E. Baralis ·
Dipartimento di Automatica e Informatica, Politecnico di Torino, Corso Duca degli Abruzzi, 24, 1022, Torino, Italy
E-mail: name.surname@polito.it

1 Introduction

With the advent of online trading, stock market forecasting has received increasing attention from the data mining community. The aim is to develop data-driven, automated solutions to support traders and investors in decision making by recommending potentially profitable trades. Quantitative trading systems [2] analyze historical stock data to generate profitable trading signals (e.g., buy a stock at given price). Predictions are based on a variety of indicators, which are usually derived from (i) fundamental analyses (i.e., deep analyses of the intrinsic value of financial assets), (ii) technical analyses (i.e., statistical analyses of past market prices and exchanged volumes), or (iii) news reports. The characteristics and frequency of the generated recommendations depend on the horizon of investment. This work specifically addresses multiple-day stock trading (no less than one day).

Classification algorithms are among the most popular machine learning techniques used to forecast the stock markets [5]. The models trained on historical stock-related data (e.g., time series, news) are exploited to predict the direction of a stock in the near future. In this work we address the prediction of the next-day direction of a stock based on historical price, technical indicators (e.g., trend and volume indicators, price oscillators) and news sentiment (e.g., number of positive and negative news published on that day).

A common strategy to make predictions more robust to noise and variance is to ensemble multiple classifiers [20]. In literature many attempts to apply ensemble methods combining different models or the same model on different data samples have already been made (e.g., [8, 20]). They have considered large sets of (heterogeneous) features, possibly filtered by means of correlation analyses. However, deciding which models should be combined together and how to select/combine data features to train each model in the ensemble are still open research issues.

To overcome the aforesaid issues, this paper proposes a complementary strategy to generate ensembles of classification models for stock price prediction, which is proved to be effective regardless of the classifier used in the model ensemble. Instead of training all the models on the same (sub)set of features, we proposed to partition the feature set into semantically correlated facets. Facets describe the properties of a stock pertinent to different aspects (e.g., price trend, sentiment). Training models separately on faceted data allows us to tailor the generated recommendation to the given category. The faceted recommendations (buy, hold, sell) are then combined together to generate reliable yet profitable trading signals.

We backtested the proposed trading system on data acquired in different markets and market conditions. The results confirmed that, in most cases, any of the faceted models performs better than a strategy based on mixed features. We then combined faceted classification models, thus achieving performance superior, in terms of one-year total return and maximum draw-down, to both faceted and mixed-feature models independently of the classifier used.

The contribution of the paper can be summarized as follows: (i) It proposes a new ensemble method to forecast future stock prices, which combines predictions based on separate facets of the stock data. (ii) Unlike previous approaches, it analyzes the separate impact of price trends, volatility and momentum indices, and news sentiment by combining them using an ensemble method. (iii) It empirically evaluates the performance of the proposed approach in various market conditions. (iv) It shows that combining per-facet classification models allows achieving better performance than training classifiers on mixed features.

The rest of the paper is organized as follows. Section 2 discusses the position of this work in the state-of-the-art literature. Section 3 describes the proposed trading system. Section 4 summarizes the experimental results, while Section 5 draws conclusions and discusses the future research perspectives of this work.

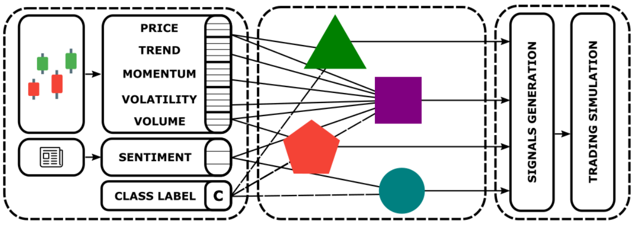
2 Related works

A huge body of work has been devoted to applying supervised machine learning techniques to automatically generate signals for intraday stock trading. Most research efforts have been devoted to predicting the most likely direction of a stock rather than its price level, because it has often resulted in more accurate trading results [5]. In this context, the applied techniques include (i) Bayesian Networks (e.g., [17]), (ii) deep learning models (e.g., [15,18]), (iii) metaheuristic optimization algorithms (e.g., [7,9]), (iv) pattern recognition techniques (e.g., [13]), and (v) K-Nearest Neighbor and Support Vector Machines classifier [1,14]. In this work the aim is to apply ensembles of classification algorithms to predict next-day stock price directions.

Using ensembles of classification models has proved to achieve performance superior to single classifiers in many different data mining domains [19]. In financial applications, combining multiple classifiers to take trading decisions has become an established strategy because it reduces the variance of estimation errors as well as improves the robustness and profitability of the proposed trading systems [20]. For instance, in [20] the authors combined both multiple instances of the same classification model as well as different models by using majority voting and bagging. Heterogeneous classifier ensembles offered slightly better performance than the homogeneous ones (independently of the strategy used to ensemble models). However, choosing the combination of classifiers that fits best the analyzed data distribution may be challenging. The system presented in [8] combines rough set theory with fuzzy time series and genetic algorithms, respectively. Several hybrid methods based on Artificial Neural Networks have been proposed, e.g., [6,16].

Selecting the right features to use for classifier training is a parallel yet relevant issue. Previous approaches adopted either ad hoc techniques to filter out non-correlated features (e.g., [10,21]) or weighting strategies to differentiate features according to their importance (e.g., [3]). Unlike [3,4,10,21], we train an ensemble of faceted classification models separately for each stock. To the

Fig. 1: System architecture.



best of our knowledge, this work is the first attempt to address stock trading by combining faceted classification models.

3 The trading system

To verify the ability of ensembles of faceted classifiers to generate profitable stock trading signals, we developed a new trading system. The system architecture is depicted in Figure 1. The system analyzes historical data about all the stocks of a user-specified stock exchange index (e.g., the U.S. Standard & Poor 500 index).The system consists of the following modules: (i) **Data acquisition and preparation.** This module acquires historical stock-related prices and news data, extracts the values of a subset of relevant features, and stores them into relational datasets (see Section 3.1). (ii) **Faceted classification.** This module takes as input the relational dataset and a feature categorization into facets (given by the domain experts). For each stock in the stock market index considered for trading purposes, the module trains a separate classification model per facet on past data. These models are applied to predict the next-day direction of all the stocks in the index (see Section 3.2). (iii) **Trade and money management.** This module is responsible for generating trading signals based on classifier predictions. A trading signal on a stock is generated if and only if, on a given day, all the faceted models are concordant (e.g., they all recommend BUY). Based on the generated signals, old and new trading positions are managed in compliance with the money management strategy adopted by the trader (see Section 3.3).

3.1 Data acquisition and preparation

This module acquires and stores the historical price series of all the stocks in the index as well as the content of news reports ranging over each stock in the considered time period.

Data acquisition. Let T be the reference time period[1]. Let $\{s_1, s_2, \ldots, s_N\}$ be the set of N stocks composing the index. For each stock s_j $[1 \leq j \leq N]$

[1] For the sake of simplicity, throughout the paper we have considered yearly periods.

the modules acquires the samples in T of the following time series: (i) daily opening prices (OP_j^T), (ii) daily closing prices (CP_j^T), (iii) daily maximum prices ($MAXP_j^T$), (iv) daily minimum prices ($MINP_j^T$), and (v) exchange volume (V_j^T).

Let NEW_q^T $[1 \leq q \leq N]$ be the set of news reports ranging over the q-th stock within the reference period T. It is deemed as relevant to trace the salient events and the public mood on the stock [2]. The textual content of each news report $new \in NEW_q^T$ is modelled as a Bag-Of-Words, which consists of the set of words occurring at least once in the text.

Data modelling. Stock price and news data are stored in relational datasets. Let F=$\{f_1, f_2, \ldots, f_L\}$ be a set of features describing stock prices and news content under various aspects. A relational dataset consists of a set of records characterized by a fixed schema F. The module generates a separate dataset per stock, namely $D^T(s_j)$. It collects the values of a set of descriptors (described later on) associated with stock s_j and a reference time period T. Each record $r_i \in D^T(s_j)$ corresponds to a different time point $t_i \in T$ and takes exactly one value per feature in F.

To avoid introducing a bias in the training phase, relational data are cleaned prior to running the classification process. Since missing values affect the majority of the features, stocks that present at least one missing value are removed from the analysis. Furthermore, textual news are cleaned by removing the English stopwords in the NLTK dictionary [11].

Currently, the systems integrates a large number of stock features. They are categorized, according to their semantic meaning, into the following facets:

- **Price history:** This facet describes the history of the stock price series. It includes the daily variations of the opening, closing, maximum, and minimum prices in the last ten trading days.
- **Price trend:** This facet includes various price trend descriptors established in technical analysis, i.e., the relative difference between the Simple/Exponential Moving Averages computed over different periods (i.e., 5 periods vs. 20, 8 vs. 15, 20 vs. 50), the Moving Average Convergence/Divergence indicator, the Aroon Oscillator, the Average Directional Index, the difference between the positive and negative directional oscillators.
- **Volatility:** This facet concerns the analysis of the stock volatility. It consists of two established volatility indices, i.e., the Chande Momentum Oscillator and the Average True Range.
- **Volume:** This facet describes the volumes of exchange related to each stock. it consist of the daily variations of the exchange volume, the Percentage Volume Oscillator, the Accumulation Distribution Line, and the On Balance Volume.
- **Sentiment**: This facet analyzes the sentiment of the news related to a given stock by means of a selection of numerical descriptors, i.e., the number of published news, and the numbers of news with positive/negative sentiment.

A more thorough description of the analyzed features is given in [12].

To predict the next-day direction of a stock, a class label C labels each record in each dataset. At each point of time t_i the class label C takes value (i) *Increase*, if the closing price of stock s_j has increased by at least 1% with respect to the previous time point (t_{i-1}), (ii) *Decrease*, if the closing price has decreased by at least 1%, or (iii) *Stationary*, if the daily price variation has been between -1% and +1%. Using 3 classes allows us to decide whether the predicted price variation is worth considering for stock trading.

3.2 Faceted classification

This module forecasts the next-day class value separately for each stock of the considered index. On each trading day t_x, the module trains an ad hoc ensemble of classification models to forecast the direction of each stock s_j at t_{x+1} by applying an expanding window approach [19]. More specifically, the training dataset $D^T(s_j)$ includes all the historical data within the reference time period (T), both in the short- and in the medium-term.

An ensemble of classification models is generated. To specialize each model on a different data facet, features belonging to different facets are selected prior to running classifier training. Specifically, for each facet a separate model is trained on a dataset exclusively including the features belonging to that facet. To perform multivariate time series forecasting, within each faceted model the feature describing the facet are joined with the *Price history* features.

The module integrates various established classification models, among which Multi-Layer Perceptron (MLP) [19], Support Vector Classifier (SVC), Multinomial Naive Bayes (MNB), Random Forest Classifier (RFC) and K-Nearest Neighbor. Currently, the implementations provided by Scikit-learn (https://scikit-learn.org) are integrated.

3.3 Trade and money management

This module generates the per-stock trading signals (e.g., BUY stock s_i, SELL stock s_j, HOLD stock s_z) based on classifier predictions. In order to minimize the number of false signals, the class label predictions made by each faceted model for an arbitrary stock s_j are combined together as follows: If *all* the predictions are *Increase* then a *Buy* signal for stock s_j is generated. Else if *all* the predictions are *Decrease* then a *Sell* signal for stock s_j is generated. Otherwise, a *Hold* signal for stock s_j is generated.

The generated trading signals are exploited to dynamically open and close multi-day long- and short-selling positions. On each trading day the trading system examines the generated signals for all the stocks, decides which long-/short-selling positions need to be opened, and reconsiders all the previously opened positions. Trading positions are opened and reconsidered at the opening of the t_{i+1} day according to the predictions made at t_i. Specifically, at the opening time of day t_{i+1} the trading system performs the following actions:

(A) Opens a new multi-day long position (betting on a significant stock price increase in the next day) for every stock for which (i) no positions have already been opened and (ii) a *Buy* signal has been generated at t_i.
(B) Opens a new multi-day short-selling position (betting on a significant stock price decrease in the next day) for every stock for which (i) no positions have already been opened and (ii) a *Sell* signal has been generated at t_i.
(C) Closes any previously opened multi-day long position for the stock for which a *Sell* signal has been generated at t_i.
(D) Closes any previously opened multi-day short-selling position for the stocks for which a *Buy* signal has been generated at t_i.

An open position on a stock is kept open until a explicit contrary indication is given. However, to preserve the equity against excessive losses, a trailing stop loss signal is automatically executed whenever the stock price moves in the unfavourable direction by more than 0.5% thus yielding a reward/risk ratio of the trading strategy equal to or above 2.

4 Experiments

We backtested the trading system on historical stock data crawled through the APIs of Yahoo! Finance (https://finance.yahoo.com/). Specifically, we separately analyzed the stocks of two different markets (i.e., the U.S. Standard & Poor 500 and the Italian FTSE MIB 40). To analyze the system performance in different market conditions, for each index we generated three different datasets collecting price series related to three different years (2011, 2013, 2015). Year 2011 was a representative period of bearish market, year 2013 was mainly a period of bullish market, while year 2015 was a mix of bullish and bearish sub-periods. In these years some of the stocks' price series presented missing values. Note that these anomalies occurred in less than the 8% of the stocks in the worst case (year 2011) and in less than the 2% in the best case (year 2015). To integrate news data for U.S. stocks we crawled the news reports from Thomson Reuters (https://reuters.com). The analyzed historical data contains about 251 trading days as rows, separately for each year (the exact number depends on holidays and weekends). Each day has five numerical values: the open, close, high and low prices and the volume exchanged. The total number of news related to U.S. stocks depends on the considered year. On average, about $260k$ news per year were processed.

The experiments were run on a machine equipped with Intel® Xeon® X5650, 32 GB of RAM and running Ubuntu 18.04.1 LTS. For each algorithm we set the most appropriate configuration setting using a grid search[2].

Due to lack of space, hereafter we will report and discuss a selection of the results achieved with all the algorithms and on all the datasets. The complete result set (with all the tables and plots) is provided as additional material.

[2] Recommended configuration settings: SVC (Rbf kernel. C=1. Gamma=$\frac{1}{|D|}$), MNB (α=1.0), K-NN (K=5), RFC (Criterion=*Gini*, Max_depth=*none*, num_estimators=100), MLP (hidden_layer_sizes=20, solver=$lbfgs$, n_iter_no_change=2)

4.1 Analysis of the equity lines

We compared the performance of (i) the newly proposed ensemble of faceted classifiers (namely *Ensemble*), (ii) each of the faceted models (namely, *Trend, Volume, Momentum*, and *Volatility*), (iii) a strategy using all the features (namely, *All-Features*). Figure 2 shows the equity lines generated by running a representative classifier, i.e., Support Vector Classifier (SVC), on all the datasets. In the simulation we assume (i) a starting equity equal to 100,000, (ii) 10% fixed-amount trades, and (iii) 0.5% of transaction cost. Figure 3 compares the equity lines generated by different classifiers on the S&P500 dataset (i.e., the largest among the two analyzed indices traded in a mixed-trend period).

Ensemble achieved maximal total return and minimum one-year draw-down on 4 out of 6 datasets. On the FTSE MIB 2015 dataset it is the only profitable strategy. In the remaining two cases, one faceted model performed better than *Ensemble* in terms of total return, but worse in terms of maximal draw-down. *All-Features* performed worse than *Ensemble* in all the cases. Among the tested classifiers, MLP overperformed the others in terms of total return, but underperformed SVC in terms of one-year maximal draw-down.

Table 1 summarizes some relevant statistics about the trading simulations based on SVC on S&P500 2015. Profitable trades are mostly multi-day, while the money management strategy preserves the equity by using the stop loss in case of wrong forecasts. The average return per trade of *Ensemble* is 25% higher than those of *All-Features*.

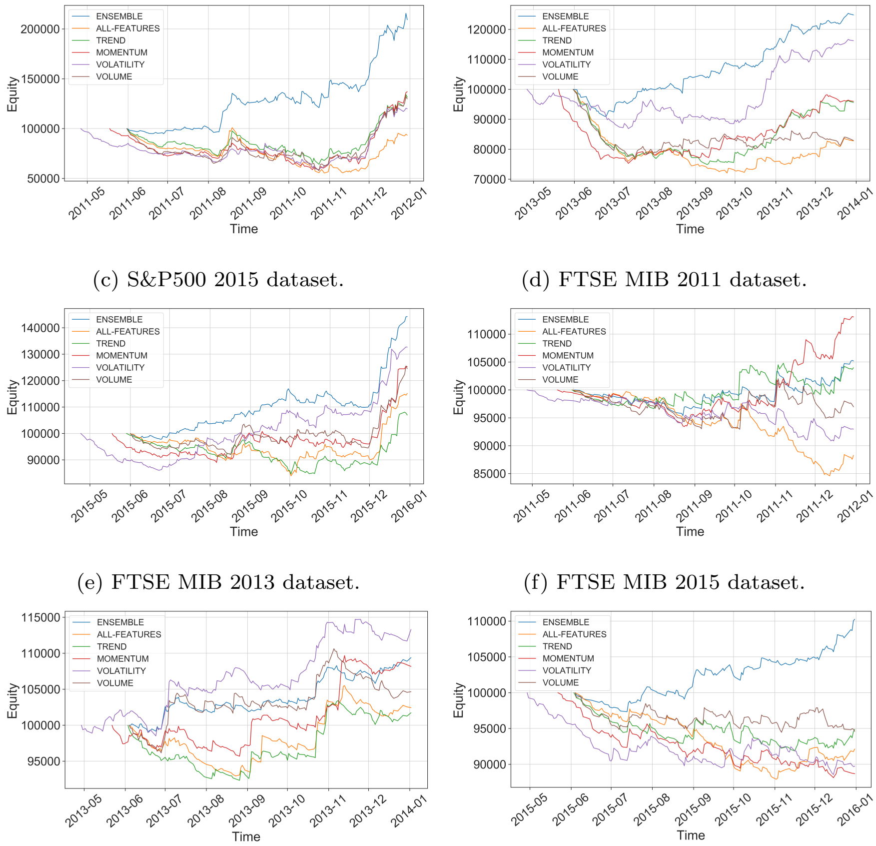
Table 1: Trading statistics, SVC Classifier, S&P500 2015 dataset.

Method	**Avg trade duration (hours)**	**Avg num trades x stock**	**Avg return x trade (%)**	**1Y max draw-down (%)**	**1Y max gain (%)**	**Num Stop losses**	**Num trades**	**Tot return (%)**
Ensemble	70.9	11.5	0.5	-2.2	44.2	764	1046	**44.2**
Volatility	111.6	12.9	0.4	-0.14	32.7	1546	2018	32.7
Momentum	111.9	11.1	0.4	-11.0	25.3	1295	1649	25.3
Volume	122.5	9.7	0.5	-8.0	25.5	1258	1606	24.5
All-Features	122.9	9.7	0.4	-16.0	15.1	1104	1390	15.1
Trend	123.2	9.6	0.4	-15.3	8.1	1221	1546	7.0

4.2 Impact of the news facet

We performed further tests by adding a new facet (*Sentiment*) whose values are independent of the stock prices and exchanged volumes. Figure 4 shows the equity lines of SVC on the S&P500 2015 dataset. The results show that *Ensemble* achieved maximal one-year return and draw-down, and the *News* faceted model performed roughly as good as *All-Features*, but significantly worse than *Ensemble*.

Fig. 2: Equity lines, SVC classifier.



(a) S&P500 2011 dataset.

(b) S&P500 2013 dataset.

(c) S&P500 2015 dataset.

(d) FTSE MIB 2011 dataset.

(e) FTSE MIB 2013 dataset.

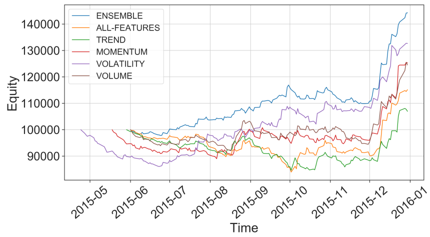
(f) FTSE MIB 2015 dataset.

4.3 Statistical significance tests

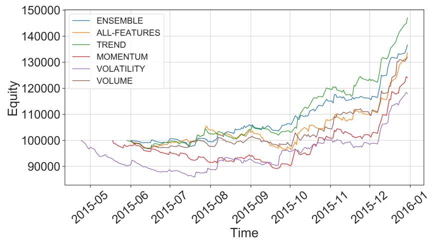
We validated the statistical significance of the performance gaps between the considered methods using the Friedman test. The test was performed separately for each algorithm by applying the following procedure. (i) Separately for each dataset, the methods are sorted by decreasing value of the reference evaluation metric. (ii) For each method, its average ranking over all the considered datasets is computed. (iii) The observed differences between the average rankings of the considered methods are compared with the critical difference threshold CD that establishes whether the difference is statistically significant. By setting the significance level to 95%, the corresponding value of CD is 0.17.

Fig. 3: Comparison among the equity lines obtained using different classifiers. S&P500 2015 dataset.

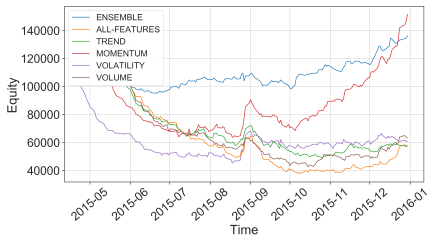
(a) SVC classifier.



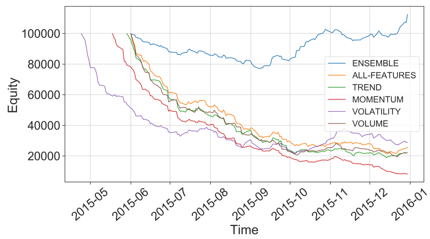
(b) MNB classifier.



(c) RFC classifier.



(d) K-NN classifier.



(e) MLP classifier.

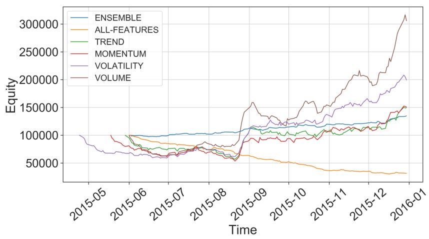
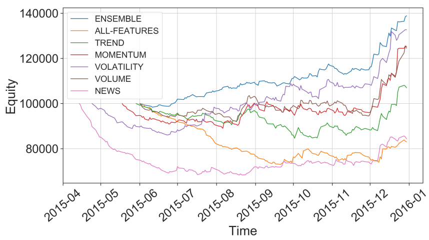


Fig. 4: Impact of news sentiment, equity lines, SVC, S&P500 2015 dataset.



The majority of the difference between the mean ranks of the considered strategies are statistically significant[3]. The *Ensemble* method achieved the best

[3] Due to the lack of space, the detailed results are given as additional material.

average mean rank value for each classifier. The rank differences are statistically significant for all the considered classifiers, except for MLP. The results confirmed that the ensemble of faceted models performed significantly better than the other methods in terms of both profit and draw-down.

4.4 Execution time and complexity analysis

The usability of machine learning-based quantitative trading strategies strongly depends on the complexity of the data analytics process. In our context, most of the computation effort is due to classifier learning, whereas data preparation and trade management phases took negligible time and space. The spatial complexity of the training phase strongly depends on the characteristics of the analyzed models. Among those tested in this research work, the most computationally intensive algorithm is MultiLayer Perceptron. In the worst case, it entails training with 200 epochs per dataset. The model consists of two matrices that sum up to $19k$ floating point values. The simpler K-Nearest Neighbor (K-NN) model computes a 125x64 distance matrix in order to assign the class label. The Support Vector Classifier computes and stores a 250x250 K matrix to apply the kernel function to the training data and to choose the class boundaries accordingly. The Multinomial Naive Bayes classifier computes and stores the prior class probabilities and the conditional probabilities of each feature value with respect to them. Hence, its spatial complexity is linear with the feature domain size. Finally, the Random Forest Classifier generates 100 trees consisting of up to $1 + 2^d$ nodes (where d is the tree depth). More details on the algorithmic complexity are given in [19].

We analyzed also the algorithm execution times by tracking the elapsed time in the back-testing phase. The average training and classification time per stock for the ensemble method ranged between 0.052s (using SVC and MNB classifiers) and 1s (using the MLP classifier). The execution times are suitable for nighttime trade recommendation by the next-day opening of the market even when coping with large stock markets (e.g., S&P500).

5 Conclusions and future work

The paper presented a quantitative stock trading strategy relying on an ensemble of classification models. Instead of training multiple models on the same features, it proposes to first partition data features into semantic facets and then combine together the models that have been trained separately on each facet. The proposed strategy performs better than using single or mixed-faceted models, independently of the classification model used.

In our research agenda, we will answer to the following questions: Which models should be combined? Which features to use? To this purpose, we will explore the use of Deep Learning models and the integration of features derived from fundamental analysis.

References

1. Baralis, E., Cagliero, L., Cerquitelli, T., Garza, P., Pulvirenti, F.: Discovering profitable stocks for intraday trading. Information Sciences **405**, 91 – 106 (2017)
2. Chan, E.: Algorithmic Trading: Winning Strategies and Their Rationale, 1st edn. Wiley Publishing (2013)
3. Chen, Y., Hao, Y.: A feature weighted support vector machine and k-nearest neighbor algorithm for stock market indices prediction. Expert Systems with Applications **80**, 340 – 355 (2017)
4. Chiang, W.C., Enke, D., Wu, T., Wang, R.: An adaptive stock index trading decision support system. Expert Systems with Applications **59**, 195 – 207 (2016)
5. Enke, D., Thawornwong, S.: The use of data mining and neural networks for forecasting stock market returns. Expert Systems with Applications **29**(4), 927 – 940 (2005)
6. Gaaken, M., Azaalaca, M., Boru, A., Dosdoayru, A.T.: Integrating metaheuristics and artificial neural networks for improved stock price prediction. Expert Systems with Applications **44**, 320 – 331 (2016)
7. Kim, M.J., Min, S.H., Han, I.: An evolutionary approach to the combination of multiple classifiers to predict a stock price index. Expert Systems with Applications **31**, 241–247 (2006)
8. Kim, Y., Ahn, W., Oh, K.J., Enke, D.: An intelligent hybrid trading system for discovering trading rules for the futures market using rough sets and genetic algorithms. Applied Soft Computing **55**, 127 – 140 (2017)
9. Kwon, Y., Moon, B.: A hybrid neurogenetic approach for stock forecasting. IEEE Transactions on Neural Networks **18**(3), 851–864 (2007)
10. Li, Q., Chen, Y., Jiang, L.L., Li, P., Chen, H.: A tensor-based information framework for predicting the stock market. ACM Trans. Inf. Syst. **34**(2), 11:1–11:30 (2016)
11. Loper, E., Bird, S.: Nltk: The natural language toolkit. In: In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics (2002)
12. Murphy, J.: Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications. New York Institute of Finance Series. New York Institute of Finance (1999). URL https://books.google.it/books?id=5zhXEqdr_IcC
13. Naranjo, R., Arroyo, J., Santos, M.: Fuzzy modeling of stock trading with fuzzy candlesticks. Expert Systems with Applications **93**, 15 – 27 (2018)
14. Nayak, R.K., Mishra, D., Rath, A.K.: A naive svm-knn based stock market trend reversal analysis for indian benchmark indices. Applied Soft Computing **35**, 670 – 680 (2015)
15. Nelson, D.M.Q., Pereira, A.C.M., de Oliveira, R.A.: Stock market's price movement prediction with lstm neural networks. In: 2017 International Joint Conference on Neural Networks (IJCNN), pp. 1419–1426 (2017)
16. Patel, J., Shah, S., Thakkar, P., Kotecha, K.: Predicting stock market index using fusion of machine learning techniques. Expert Systems with Applications **42**(4), 2162 – 2172 (2015)
17. S. Malagrino, L., Roman, N., Monteiro, A.: Forecasting stock market index daily direction: a bayesian network approach. Expert Systems with Applications **105** (2018)
18. Selvin, S., Vinayakumar, R., Gopalakrishnan, E.A., Menon, V.K., Soman, K.P.: Stock price prediction using lstm, RNN and cnn-sliding window model. In: ICACCI, pp. 1643–1647. IEEE (2017)
19. Tan, P.N., Steinbach, M., Karpatne, A., Kumar, V.: Introduction to Data Mining (2Nd Edition), 2nd edn. Pearson (2018)
20. Tsai, C.F., Lin, Y.C., Yen, D.C., Chen, Y.M.: Predicting stock returns by classifier ensembles. Applied Soft Computing **11**(2), 2452 – 2459 (2011)
21. Zhong, X., Enke, D.: Forecasting daily stock market return using dimensionality reduction. Expert Systems with Applications **67**, 126 – 139 (2017)