Predicting Stock Price Movement Using Financial News Sentiment

Jiaying Gong, Bradley Paye, Gregory Kadlec, and Hoda Eldardiry(✉)

Virginia Tech, Blacksburg, USA
{gjiaying,bpaye,kadlec,hdardiry}@vt.edu

Abstract. A central question in financial economics concerns the degree of informational efficiency. Violations of informational efficiency represent capital miss-allocations and potentially profitable trading opportunities. Market efficiency analyses have evolved to incorporate increasingly rich public information and innovative statistical methods to analyze this information. We propose an Automatic Crawling and Prediction System (ACPS) to 1) automatically crawl online media, 2) extract useful information from a rich set of financial news, and 3) predict future stock price movements. ACPS consists of a feature selection pipeline to select an optimal set of predictive features and a sentiment analysis model to measure sentence-level news sentiment. Generated features and news sentiment data are further processed via an ensemble model based on several machine learning and deep learning algorithms to generate forecasts. Results demonstrate the robustness of our proposed model in predicting the directional movement of daily stock prices. Specifically, the model consistently outperforms existing methods on single stock prediction and it performs well across all S&P 500 stocks. Our results indicate the potential value of rich text analysis and ensemble learning methods in a real-time trading context.

Keywords: Stock prediction · Sentiment analysis · Machine learning

1 Introduction

The question of how rapidly and completely financial security prices reflect value-relevant news is a central issue in financial economics. This question is formalized by the 'market efficiency' hypothesis where value relevant information is instantly incorporated into security prices. Violations of market efficiency are of great interest to academics and practitioners, as they have key implications for capital allocation in the economy and trading strategies that can enhance investment performance. On the one hand, competition among investors should ensure that prices reflect value-relevant information over time. However, there are reasons to expect that inefficiencies exist, most notably the notion that investors will not rationally incur the expenses of gathering and processing financial information unless they expect to be rewarded by higher returns [1]. Also, investors'

L. Iliadis et al. (Eds.): EANN 2021, INNS 3, pp. 503–517, 2021.
https://doi.org/10.1007/978-3-030-80568-5_41

efforts to correct inefficiencies are likely to be met with practical limitations relating to funding, frictions, and risk [2]. Early empirical market efficiency studies focused on high frequency (daily) predictability in security prices/returns using past prices/returns [3]. The literature then shifted to lower-frequency relations between stock returns and valuation ratios [4]. However, there is renewed interest in short-horizon predictability based on technical indicators and fundamental information culled in real time from financial media sources [5]. A parallel trend involves the application of recently developed machine learning and deep learning methods to stock return prediction [6]. Despite increasing sophistication, tests of informational efficiency invariably focus on a subset of the complete set of information available to investors. Indeed, the sheer vastness of the set of public financial information potentially inhibits informational efficiency, as even professional investors face constraints on scarce cognitive resources and must rationally allocate attention to a subset of information [7]. A multitude of potential indicators can be constructed from past price and volume data. In addition, there exists a massive amount of online financial news, much of it unstructured and potentially redundant.

In this study, we aim to create an automated prediction system that obtains rich sets of technical and textual information and efficiently processes this information to produce informative forecasts to serve as inputs to investment decisions. To this end, we face significant challenges: 1) There is a vast number of potential technical indicators of interest and it is difficult to determine which are useful. 2) Large quantities of unstructured online financial news appears each day, and only a portion of it might be value-relevant. 3) Financial news stories are often specific to particular companies, so it is necessary to assign news content to relevant companies. 4) Information in technical indicators and financial news is likely correlated, and thus, it is important to consider nonlinear relations and interactions between technical indicators and news sentiment measures.

To address these challenges, we propose an Automatic Crawling and Prediction System (ACPS) to automatically crawl online news and extract useful information. ACPS automatically parses financial data and constructs a high-dimensional information set consisting of recent price and related trading data as well as textual data consisting of financial news data related to the company. A particular novelty involves the extraction of lexicon-based sentiment measures at a rich sentence-level based on four different dictionaries including a finance-specific sentiment dictionary [8]. This contrasts with related methods that limit attention to short text snippets such as tweets and headlines. To the best of our knowledge, the resulting combination of technical indicators and sentence-level news sentiment represents the richest set of stock and news data used to construct stock price forecasts. In addition, the model implements a combination of feature selection steps that eliminate irrelevant technical indicators and keep the most helpful features for stock price prediction. Experiments show that our proposed model forecasts the directional change in daily returns relatively well, with a blended prediction accuracy rate 61.88%. A simulated trading exercise illustrates the potential of the model to deliver economically significant investment gains. **The key contributions of this paper include:** 1) a novel automatic crawling and prediction system to collect and preprocess text data and make predictions, 2) a feature selection pipeline to select the most predictive features, 3) a sentiment analysis model to calculate sentiment based on each sentence in news instead of only headlines, and 4) several ensemble models combining ML and DL algorithms to improve predictions.

2 Related Work

Approaches to forecasting short term stock price movements can be classified based on the information used for predictions. Several studies base forecasts on 'technical analysis', which conditions on past stock price, volume, and related trading data. Another 'fundamental analysis' approach takes external unstructured data as input, such as company financial reports or news articles, and extracts predictive information from this data.

Technical Analysis: Machine Learning and Deep Learning in Stock Market Prediction. Technical analysis has a long history, dating back to at least the 1930s [3]. A recent trend in technical analysis incorporates ML and DL algorithms. For example, Pahul et al. [9] applied various ML methods to identify the most important technical attributes in stock market prediction. Nagaraj et al. [10] proposed a regression prediction model by considering combinations of technical indicators and selecting gold features. Mehak et al. [11] predicted stock market movements using different ML techniques. Other related contributions include Singh et al. [12], Chang et al. [13], and Mingyue et al. [14]. Broadly, these studies focus on optimizing ML or DL algorithms to select golden features from past trading data and improve the performance. Although these approaches leverage a large amount of data, they omit potentially relevant information contained in a rich body of textual data consisting of financial news media. Textual data are likely to add forecasting value because news stories are likely to contain additional 'soft' information not fully represented in or extracted from quantitative trading data [15].

Fundamental Analysis: Stock Trend Prediction Using News Sentiment. A prominent early study addressing connections between news and stock price movements links pessimistic media coverage with temporary stock price declines, followed by a reversion to fundamentals [16]. This evidence indicates that media tone reflects investor sentiment. Subsequent research finds that textual media content captures otherwise hard-to-quantify aspects of firm performance, indicating that media coverage captures fundamental news as well as sentiment [15]. More recent papers address several themes. One strand of literature augments conventional predictive return models by incorporating textual news data [17–19]. Other papers focus on sifting through large quantities of daily news to detect relevant news events [20]. A third strand of literature focuses on developing richer measures to capture the sentiment of financial news from public news headlines [21]. Our study provides a more comprehensive treatment of real-time news, including the assessment of company-specific news sentiment at the granular sentence level. Another branch of related literature proposes prediction models that, in the spirit of this paper, combine measures of the sentiment of financial news with historical stock market data [22,23]. However, these studies classify sentiment based on keywords extracted from the news, whereas our study evaluates sentiment using the complete text. Finally, and perhaps most closely related to this paper, Joshi et al. [24] created classification models that incorporate sentiment values of financial news to study the relation between financial news and stock price movements, and Shah et al. [25] develop a sentiment analysis dictionary for financial news. These studies apply their approaches to a single or a very small number of well-known companies. In contrast, we apply our proposed approach to all individual stocks included in the S&P 500 index. This aspect of our contribution is important because it is unclear whether optimistic results obtained in earlier literature for a small number of stocks will generalize to success over a broad set of stocks. This is a potentially important issue because the apparent success of these approaches may simply reflect good luck or particular characteristics of selected stocks. In addition, equity investors typically make allocation decisions across thousands of US and international firms. We develop a model that can be deployed at scale to generate useful investment signals for realistic investment choices involving hundreds of securities.

3 Methodology

In order to develop informative signals for investment decisions, we propose a novel Automatic Crawling and Prediction System (ACPS) as shown in Fig. 1, which combines both historical stock data and corresponding financial news. The model contains four main components: Stock Data Analysis, Financial News Analysis, Prediction Model and Decision Making. The system selects the most helpful features through machine learning methods to improve the accuracy of forecasts of directional stock price movements. It also analyzes concurrent financial news and measures the sentiment value of the news. The inputs are historical stock prices (open, close, high, low and volume features) and news contents, the output of the system is the stock price movement prediction value.

3.1 Feature Selection

Feature selection methods help to reduce computational time, improve predictions and better understand data [26]. The ACPS applies up to four feature selection methods to identify useful technical signals. In particular, we compute many technical indicators based on the original five features and select the best combination of feature selection steps for each ML or DL algorithm. The feature selection process involves the selection of particular methods, parameter tuning choices, and the choice of ordering of methods. We follow three criteria in implementing feature selection; 1) conduct exploratory analysis; 2) optimize the trade-off between computational complexity and performance; 3) apply methods in a simple and intuitive manner. The following discusses the feature selection.

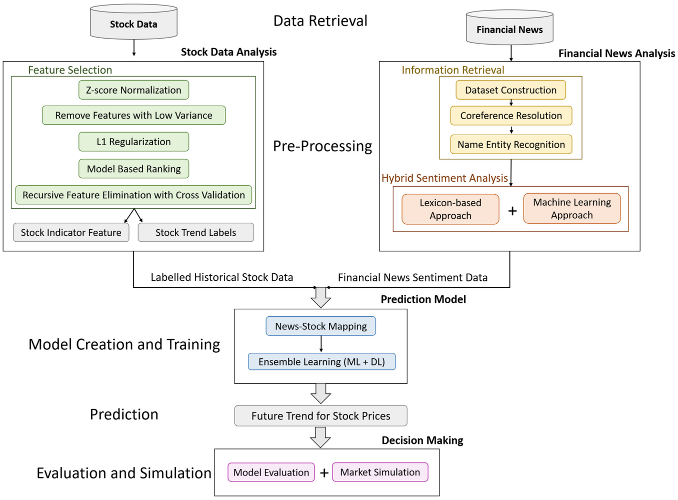

Fig. 1. Automatic crawling and prediction system

Removing Features with Low Variance. This method removes features with variance below a threshold 0.8 in order to boost performance on a high-dimensional dataset. Results were similar for modest threshold perturbations.

L1 Regularization/Lasso. Linear regression models penalized with the L1 norm deliver sparse solutions such that many coefficients or weights become zero. We first define a linear model with weight w and bias b and a loss function L as the squared error which is the difference between the true value (labels) y and the predicted value. The loss function with L1 regularization $L_1 = (wx+b-y)^2+\lambda|w|$ where λ is the regularization parameter and $|w|$ is the sum of the absolute regression weights w. The penalty term $\lambda|w|$ effectively bounds the L_1-norm of the coefficient vector w. The Lasso regression acts both as a form of shrinkage estimator and a variable selection tool, as only a subset of coefficients in w will receive nonzero weight. The regularization parameter λ is set to 100.

Model-Based Ranking. Model-based ranking is a filtering method that selects variables based on performance scores on a training dataset. We apply the random forest classifier, which fits various decision tree classifiers on different subsets of data. This approach uses averaging to improve accuracy and avoid over-fitting and selects features based on variable-importance ranking on a training set. The impurity-based feature importance is computed based on Gini importance [27]. The higher the Gini importance, the more relevant the feature is. A threshold 0.01 is set for feature selection.

Recursive Feature Elimination with Cross Validation. Recursive feature elimination (RFE) is a wrapper method which uses the predictor as a black box and the predictor performance as the objective function to evaluate the variable subset [26]. RFE selects features by recursively considering smaller and smaller sets of features. The estimator is trained on the initial set of features and the importance of each feature is obtained. Then, in each subsequent iteration, the least important feature is eliminated from current set of features. This procedure repeatedly removes the weakest features. Our application uses machine learning (linear regression) methods with features that survived previous feature selection steps as inputs to calculate validation error for all subsets. The selected feature is the subset which has the least validation error (the highest cross validation score). If the number of features is n, the number of all subsets is $2^n - 1$. Hence, recursive feature elimination with cross validation is the final step for our feature selection procedure. The output is a collection of selected features.

3.2 News Data Processing

This section describes how to use Information Retrieval and Natural Language Processing techniques to process text data and do sentiment analysis. The approach consists of five main components: Data Collection, Data Pre-processing, Coreference Resolution, Name Entity Recognition and Sentiment Analysis. We describe these parts in detail below.

Dataset Construction. A comprehensive and representative dataset of news and its sentiment analysis is the basis for analyzing and establishing connections between news and stock prediction. We first build an automatic crawler to collect information from Reuters Financial section through Wayback Machine [28]. The crawler is built based on Beautiful Soup [29] to extract different features. After we crawl the source code of webpages containing different news, we build the dataset for news sentiment analysis, which contains the following seven major parts: title, date, keywords, content, country, news source and news URL.

Coreference Resolution. Coreference resolution is the task of finding all expressions that refer to the same entity in a text [30]. Before extracting company names in news content, coreference resolution is done in order not to miss company names in news content. We use NeuralCoref, a state-of-the-art coreference resolution technique based on Neural Networks and SpaCy, to find out all pronouns and return back to their original company names or organizations.

Name Entity Recognition. Understanding the context of a document is of very high value, especially for information extraction. In any form of document, there are particular terms representing particular entities which are more informative in a unique context. In the context of analyzing the relationship between news and stock prices, company names extracted for each news text are of great importance. We use Named Entity Recognition (NER) to identify and segment the named entities, to extract company names and major events in news. We first tokenize the text of each piece of news and lemmatize all the words. Next, we use Spacy, a pre-trained statistical model supporting tokenization and tagging, to extract company names and major event topics on titles, keywords and contents. Then we classify them under various predefined classes.

Sentiment Analysis. Sentiment analysis is a Natural Language Processing task directed at the automatic identification and analysis of people's opinions, sentiments, evaluations, appraisals, attitudes and emotions towards events [31]. Understanding the sentiment (positive and negative) in news facilitates a better understanding of influence of news on stock trends. However, only considering the sentiment value for the whole news article is not enough, because one piece of news may contain several different companies or organizations. The final sentiment value based on the whole news article usually relates to the company which appears the most times in the corresponding news. Other companies or organizations that appear in the news may have an opposite or different sentiment value. As a concrete example, consider an article about the development of new electric vehicle technology at General Motors (positive sentiment) that might increase competitive pressure on Tesla (negative sentiment). Therefore, instead of only considering the sentiment value of the whole news article, we also compute the sentiment value for each sentence in the news article. We use a hybrid approach (both lexicon-based approach and machine learning (Textblob) approach) to automatically detect the sentiment of text within news articles. The lexicon-based approach is very robust, because not only does it provide

Algorithm 1: Sentiment Calculation

```
Input: News Articles $N_p$, Various Dictionaries $d_0 \cdots d_n$, Company List $C_l$,
       Threshold $t$
Output: Company Hybrid Sentiment Value List $V_c$
while $N_p$ exists do
    Break $N_p$ into sentences $N_s$;
    Tokenize $N_s$ into word vectors $N_v$;
    Extract company name $c$ from $N_v$;
    if $c \in C_l$ or $\frac{c \cdot C_{l_i}}{\|c\|\|C_{l_i}\|} > t$ then
        $v_0 \cdots v_n$ calculation;
    else
        continue;
    end
    if $v$ from same company then
        Aggregation $v = \sum v_i$;
        $V_c = \{V_{c_0}, \cdots V_{c_n}\}$ where $V_{c_i} = \{v_0 \cdots v_n\}$;
    else
        continue;
    end
end
```

good cross-domain performance, but it also incorporates sentiment difficult to detect using the machine-learning approach. Our implementation of the lexicon-based approach uses four different dictionaries to label the training dataset. One dictionary is based on finance-specific words with its polarity defined by McDonald's research [8] that classifies words into seven different sentiments. We match the news articles' words with the finance-specific word list and count numbers of words appearing in the dictionary and calculate the sentiment score of that news article. The other three dictionaries include the Liu and Hu opinion lexicon [32], AFFIN lexicon [33] and MPQA Subjectivity Lexicon [34]. We use the same method to calculate the sentiment score of news based on these different dictionaries. The algorithm to calculate the final sentiment value of one company each day is given in Algorithm 1. We assume companies are the same if the cosine similarity between the company names are less than a threshold of 0.6 because articles sometimes misspell words, use plural forms, or company abbreviations. Sentiment value for each sentence can be calculated by both [35] and a financial lexicon-based method: $S_t = S_s \times S_p$; $S_l = \frac{2^{(C_l+C_s+C_c)}}{2^{(C_u+C_w)}}(C_p - C_n)$ where S_s is the sentiment value, S_p is the polarity value, C_l, C_s, C_c, C_u, C_w, C_p, C_n is count number of litigious, strongModal, constraining, uncertainty, weakModal, positive and negative sentiment appearing in each sentence. The sentiment score considering all dictionaries equals the difference between the count of positive and negative polarities.

4 Experiments and Evaluation

This section compares the performance of our proposed model ACPS with different machine learning and deep learning models in order to demonstrate that the proposed hybrid ensemble learning model improves upon alternative machine learning models in predicting directional daily stock price movement.

4.1 Experiment Settings

In the experiments, several comparisons are made to evaluate the incremental predictive value associated with feature selection and news sentiment analysis.

Dataset Description. Two datasets are collected: financial news and historical stock data. News data are collected from Reuters. The news articles were published in the same time period as the the time period of collected historical stock data. For historical stock data, we collected daily stock data in the S&P 500 index from Yahoo Finance over a period ranging from February 08, 2013 to February 07, 2018. The second historical dataset is a subset of the full dataset, which excludes those trading dates without relevant financial news on a company-by-company basis. The target output is derived from the daily stock return, defined as the ratio of the closing price to the opening price. The output of the model is a binary indicator that takes the value 1 if the next day return is positive, and zero otherwise. We use 80% of the data for training and reserve the remaining 20% for testing and evaluation.

Baselines and Proposed Models. We propose four new models to improve the performance on the baseline model. The following identifies each model: 1)**Baseline**: Input of five original technical indicators and machine learning prediction model. 2)**All Features**: Input of all sixty-nine technical indicators such as MACD, ROC, etc. and machine learning prediction model. 3) **Feature Selection (FS)**: Input of technical indicators after feature selection and machine learning prediction model. 4) **Ensemble-FS**: Input of stock individual predictors and ensemble machine learning prediction model. 5)**Ensemble-FSN**: Input of both stock and news individual predictors and ensemble machine learning prediction model. The baseline model consists of only five features: opening, closing, high, low price and volume. First, for each model, we developed sixty-four additional technical indicators [36] based on the five features above and added them to the model. Second, we selected the best set of features using the feature selection methods described in Sect. 3.1 and then analyzed the performance of the corresponding model ('FS'). We next proposed an ensemble learning model to improve the performance based on FS Model. We combined the decisions from multiple machine learning (Decision Tree, K Nearest Neighbors, Linear Discriminant Analysis, Logistic Regression, Naive Bayes, Neural Networks, Random Forest and Support Vector Machine) and deep learning (Artificial Neural Network and Long Short-Term Memory) models. We treated individual predictors as model input features and took a majority vote ('Ensemble-FS'). Finally, we combined Ensemble-FS Model with news sentiment features ('Ensemble-FSN'). We also combined the results of multiple models to get the final prediction result.

4.2 Results and Discussion

This section describes results for predicting directional movement in stock prices using technical indicators and financial news sentiment measures. The experiments consist of three phases. The first phase describes the results of feature selection based on the entire dataset to show that predicting stock movement by using only one company's stock data is not sufficiently representative. The second phase describes the results of feature selection based on the dataset which filters out those dates without news. The third phase describes the result of the news sentiment analysis.

Feature Selection Based on Entire Dataset. We compare the performance of eight machine learning algorithms on each of the S&P 500 companies. For company-specific stock price movement prediction, companies perform quite differently. Table 1 illustrates the average performance, best performance and worst performance among 500 companies over the eight machine learning algorithms. There is a large variance between the maximum and minimum accuracy value, which implies that the ML approaches perform differentially across companies. Consequently, the average performance across stocks is more informative concerning the merits of various approaches relative to extreme company-specific performance outcomes. The results in Table 2 indicate that on average the prediction success rate for most approaches is quite close to 50%, which is random

Table 1. Company-specific performance accuracy

Models	Baseline			All features		
	Mean	Max	Min	Mean	Max	Min
DT	49.42%	60.64%	39.76%	49.97%	83.33%	36.84%
KNN	50.04%	69.23%	33.33%	49.56%	60.71%	38.94%
LDA	50.31%	66.67%	36.84%	49.82%	63.16%	33.33%
LR	51.22%	62.50%	33.33%	50.16%	66.67%	38.96%
NB	50.25%	62.65%	32.14%	49.89%	61.54%	37.75%
RF	49.67%	63.16%	32.14%	49.84%	66.67%	34.62$
SVM	51.27%	60.64%	38.46%	51.28%	60.64%	38.46%

chance. We next examine whether the feature selection steps we propose are useful for improving prediction accuracy rates. Figure 2 shows that feature selection both reduces variance in performance across companies and improves average accuracy. However, the associated improvements appear to be somewhat modest, suggesting that additional gains might be feasible via incorporating financial news sentiment information.

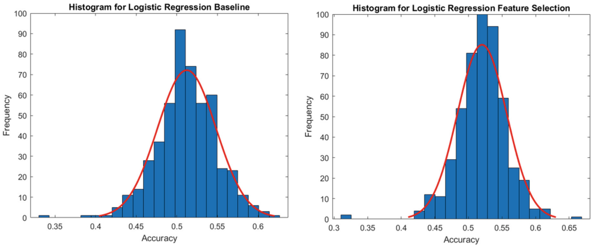

Fig. 2. Comparison of histogram for LR accuracy

Feature Selection Based on a Subset of Data. After matching stock data with financial news, we drop trading days of stock data without any news. We then implement a more general stock price movement prediction (model trained on all companies' data) instead of company-specific prediction. Figure 3 illustrates the accuracy and F1-score comparison over eight ML algorithms and two DL algorithms mentioned in section dataset description respectively. In both cases, the model using features after feature selection performs much better than the baseline model. Table 2 shows results illustrating how different combinations of feature selection steps perform for each algorithm. Except for the NB and SVM, all other ML and DL algorithms exhibit a more than 5% accuracy increase after applying the optimal feature selection approach for that method. In addition, all machine learning algorithms achieve a more than 5% increase on

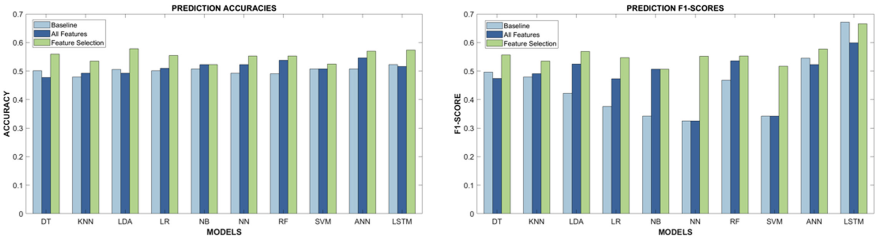

Fig. 3. Prediction performance comparison

F1-score after feature selection. We also ran an experiment to predict the level of the stock return using Linear Regression on input features generated by our feature selection pipeline. Our initial results are very promising, as we achieved a Root Mean Square Error (RMSE) of 0.02 for stock return prediction. In future work, we plan to explore further approaches and run more extensive experiments on stock return prediction.

Table 2. Feature selection performance

Models	Feature selection	Baseline			Feature selection		
		Accuracy	F1-Score	Precision	Accuracy	F1-Score	Precision
DT	L1 + MBR + RFE	50.11%	49.65%	49.97%	55.89%	55.65%	56.16%
KNN	Re + L1 + RFE	47.97%	47.97%	47.98%	53.53%	53.53%	53.52%
LDA	Re + MBR	50.54%	42.24%	49.94%	57.82%	56.89%	58.29%
LR	L1 + MBR	50.11%	37.58%	47.27%	55.46%	54.68%	55.64%
NB	Re	50.75%	34.17%	25.76%	52.25%	50.63%	52.27%
RF	Re + L1 + MBR	49.04%	46.81%	48.50%	55.25%	55.23%	55.23%
SVM	Norm + Re + L1	50.75%	34.17%	25.76%	52.46%	51.73%	52.43%
ANN	Re	50.75%	54.55%	51.30%	56.96%	57.68%	57.56%
LSTM	L1 + MBR	52.25%	67.16%	51.58%	57.39%	66.55%	55.31%

News Sentiment Analysis. We implement sentiment analysis by Textblob on news data. In addition to positive and negative sentiment, large quantities of news are classified as neutral, which implies that Textblob can only identify limited sentiment. Therefore, we implement a lexicon-based approach to further analyze the news with no sentiment value. The results show a 59% increase in sentiment finding when the lexicon-based approach is added. We use both finance and text dictionaries to calculate sentiment values in news and show a performance comparison in Fig. 4. The results indicate that a combination of both finance and text dictionaries works best. We then combine the lexicon-based approach with the ML approach. The input features are news sentiment calculated using a combination of dictionaries. Among all ML and DL algorithms, only RF and LSTM exceed 55% accuracy. The ensemble of all algorithms except

DT and NB achieves 57.17% accuracy (Table 3) We conclude that news sentiment is informative concerning future stock price movements, but that it is essential to consider news sentiment in conjunction with quantitative stock indicators: examining news sentiment in isolation does not perform well.

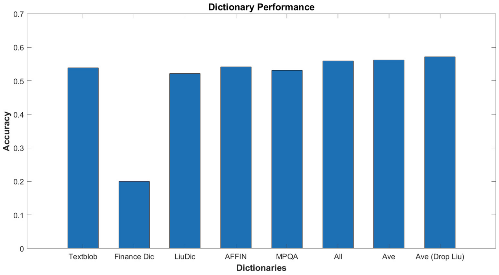

Fig. 4. Comparison of dictionaries

4.3 Evaluation

Evaluating ACPS. For assessing the performance of the proposed ACPS, we use Accuracy, Precision and the F1-score as evaluation metrics. Our proposed ACPS combines feature selection steps with news sentiment analysis. The Prediction Model consists of an ensemble learning model which treats individual predictors obtained from different ML or DL algorithms as new features. Then we take an unweighted vote to get the final prediction. Evaluation metrics for the baseline model are the average value of all ten machine learning and deep learning models. Ensemble-FS is the model consisting of five ML algorithms (DT, LDA, LR, NN and RF) and two DL algorithms (ANN and LSTM) using hard voting schema. Ensemble-FSN is a joint model that consists of Ensemble-FS and a LSTM model for news sentiment analysis. The Prediction Model of ACPS is Ensemble-FSN, which is a hybrid machine learning model. Table 4 shows the performance of these three models. Compared with the baseline mode, which achieves only 50.15% accuracy (essentially equivalent to random prediction), Ensemble-FS achieves a 59.53% accuracy rate, a more than 9% increase. After incorporating news sentiment measures, Ensemble-FSN achieves 61.88% accuracy, a roughly 11% increase compared with the baseline. The Ensemble-FS and

Table 3. Sentiment analysis

Models	Accuracy	F1-Score	Precision
RF	55.03%	55.03%	55.06%
LSTM	56.53%	64.20%	55.15%
Ensemble	57.17%	55.76%	57.89%

Ensemble-FSN generate even larger proportional gains based on the F1-score and Precision. Although we omit standard errors for brevity, the performance differences in Table 4 are statistically significant at conventional levels.

Table 4. Automatic crawling and prediction

Models	Accuracy	F1-Score	Precision
Baseline	50.15%	44.68%	42.23%
Ensemble-FS	59.53%	58.59%	60.20%
Ensemble-FSN	61.88%	61.61%	60.84%

Market Simulation. We conducted a stock market simulation trading system following a similar simulation exercise by Lavrenko et al. [37]. The analysis is intended to evaluate the potential profitability of applying our proposed model in a trading context. If the model predicts that the stock price will increase (decrease) the next day, purchase (sell short) $10,000 worth of the stock. After purchasing the stock, sell immediately if market conditions dictate that a profit of 1% ($100) or more can be made, which means we sell the stock at anytime when it reaches profit of 1%. Otherwise, sell the stock at the closing price. After selling short the stock, buy it immediately if at a price 1% ($100) lower than shorted in order to cover. Otherwise, buy the stock at the closing price. We use the same training and testing dataset described in Sect. 4.1 for market simulation. Due to limited space, Table 5 shows profit results for selected companies. Our simple trading simulation assumes traders can buy at opening price and sell at closing price exactly. However, in reality the opening and closing prices may not represent the prices at which traders could have actually purchased or sold the stock. Given that S&P 500 stocks tend to be relatively liquid, it is hoped that any associated distortions are small.

Table 5. Market simulation performance

	AMZN	BAC	FRT	GM	HCA	PFE
Daily profits	0.8%	1.3%	0.2%	0.8%	1.7%	0.4%
Accuracy	72.73%	77.78%	59.26%	80%	75%	80%

5 Conclusion and Future Work

This paper proposes a novel Automatic Crawling and Prediction System to automatically crawl online news and extract useful information. The approach applies both a feature selection pipeline to select the most informative features and a hybrid sentiment analysis model to calculate sentence-level news sentiment. Results show that our feature selection method outperforms non-feature selection models with more than 5% average increases on both accuracy and F1-score and our sentiment analysis model can can achieve 57.17% accuracy rate on directional stock price forecasts. Moreover, we proposed several ensemble models combining both ML and DL algorithms. Experimental results showed that our proposed model can successfully predict directional stock price movement with an overall accuracy of 61.88%. A market simulation analysis suggests that the application of our method can enhance trading profit. In future work, we will consider some state-of-the-art transformers such as BERT to improve sentiment classification performance. Besides, we will add experiments comparing with existing prediction models using financial news sentiment.

References

1. Grossman, S.J., Stiglitz, J.E.: On the impossibility of informationally efficient markets. Am. Econ. Rev. **70**(3), 393–408 (1980)
2. Shleifer, A., Vishny, R.W.: The limits of arbitrage. J. Finan. **32**, 35–55 (1997)
3. Cowles 3rd, A., Herbert, E.J.: Some a posteriori probabilities in stock market action. Econometrica. J. Econ. Soc. **5**, 280–294 (1937)
4. Fama, E.F., French, K.R.: Business conditions and expected returns on stocks and bonds. J. Finan. Econ. **25**(1), 23–49 (1989)
5. Ho, K., Liu, R., Wang, K., Wang, W.: The relation between news events and stock price jump: an analysis based on neural network. In: 20th MODSIM (2013)
6. Gu, S., Kelly, B., Xiu, D.: Empirical asset pricing via machine learning. Rev. Finan. Stud. **33**(5), 2223–2273 (2020)
7. Kacperczyk, M., Van, S.N., Veldkamp, L.: A rational theory of mutual funds' attention allocation. Econometrica **84**(2), 571–626 (2016)
8. Loughran, T., Mcdonald, B.: When is a liability not a liability? textual analysis, dictionaries, and 10-ks. J. Finan. **66**, 35–65 (2011)
9. Kohli, P.P.S., Zargar, S., Arora, S., Gupta, P.: Stock prediction using machine learning algorithms. In: Engineering Applications of Artificial Intelligence (2019)
10. Naik, N., Mohan, B.: Optimal feature selection of technical indicator and stock prediction using machine learning technique. In: ICETCE (2019)
11. Usmani, M., Adil, S.H., Raza, K., Ali, S.S.A.: Stock market prediction using machine learning techniques. In: 2016 ICCOINS, pp. 322–327 (2016)
12. Singh, R., Srivastava, S.: Stock prediction using deep learning. Multimedia Tools Appl. **76**(18), 18569–18584 (2017)
13. Li, C., Song, D., Tao, D.: Multi-task recurrent neural networks and higher-order markov random fields for stock price movement prediction: multi-task rnn and higer-order mrfs for stock price classification. In: SIGKDD (2019)
14. Qiu, M., Song, Y.: Predicting the direction of stock market index movement using an optimized artificial neural network model. PLOS (2016)
15. Tetlock, P.C., Saar-Tsechansky, M., Macskassy, S.: More than words: quantifying language to measure firms' fundamentals. J. Finan. (2008)
16. Tetlock, P.C.: Giving content to investor sentiment: the role of media in the stock market. J. Finan. **62**(3), 1139–1168 (2007)
17. Vanstone, B.J., Gepp, A., Harris, G.: Do news and sentiment play a role in stock price prediction? Appl. Intell. **49**(11), 3815–3820 (2019)

18. Pagolu, V.S., Reddy, K.N., Panda, G., Majhi, B.:Sentiment analysis of twitter data for predicting stock market movements. In: 2016 SCOPES (2016)
19. Khedr, A.E., Salama, S.E., Yaseen, N.: Predicting stock market behavior using data mining technique and news sentiment analysis. In: IJISA (2017)
20. Ding, X., Zhang, Y., Liu, T., Duan, J.: Deep learning for event-driven stock prediction. In: International Joint Conferences on Artificial Intelligence (2015)
21. Kirange, D., Deshmukh, R.: Sentiment analysis of news headlines for stock price prediction. In: IJACT (2016)
22. Ayman, K., Salama, S.E., Nagwa, Y.: Predicting stock market behavior using data mining technique and news sentiment analysis. In: IJISA (2017)
23. Duong, D., Nguyen, T., Dang, M.: Stock market prediction using financial news articles on ho chi minh stock exchange. In: Proceedings of the 10th IMCOM (2016)
24. Kalyani, J., Bharathi, H.N., Jyothi, R.: Stock trend prediction using news sentiment analysis. In: IJCSIT (2016)
25. Shah, D., Isah, H., Zulkernine, F.: Predicting the effects of news sentiments on the stock market. In: IEEE International Conference on Big Data (2018)
26. Girish, C., Ferat, S.: A survey on feature selection methods. In: IJECE (2014)
27. Nembrini, S., König, I.R., Wright, M.N.: The revival of the Gini importance? Bioinformatics **34**(21), 3711–3718 (2018)
28. A digital archive of the world wide web. http://web.archive.org/
29. Html parser in python. http://wikipedia.org/wiki/Beautiful_Soup
30. Coreference resolution. http://nlp.stanford.edu/projects/coref.shtml
31. Guerrero, J.S., Viedma, E.H., Olivas, J.A., Romero, F.P.: Sentiment analysis: a review and comparative analysis of web services. Inf. Sci. **311**, 18–38 (2015)
32. Liu, B., Hu, M., Cheng, J.: Opinion observer: analyzing and comparing opinions on the web. In: Proceedings of the International Conference on World Wide Web (2005)
33. Nielsen, F.Å.: A new anew: evaluation of a word list for sentiment analysis in microblogs (2011)
34. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity in phrase-level sentiment analysis. In: Proceedings of the HLT/EMNLP, HLT '05 (2005)
35. Python library for processing text data. http://textblob.readthedocs.io/en/dev/
36. Padial, D.L.: Technical analysis library in python documentation (2019)
37. Lavrenko, V., Schmill, M., Lawrie, D., Ogilvie, P., Jensen, D., Allan, J.: Mining of concurrent text and time series. In: KDD (2000)