# News Feed for Stock Movement Prediction

Andriy Krysovatyy[1][000-0003-1545-0584], Oleksandra Vasylchyshyn[1 [0000-0002- 9948-5532]], Oksana Desyatnyuk[1] and Svitlana Galeshchuk[1,2][0000-0002-6706-3028]

[1]Faculty of Finance, Ternopil National Economic University, Ternopil 46006, Ukraine
[1] Governance Analytics, Paris Dauphine University, Paris 75016, France
rector@tneu.edu.ua, volexandra@gmail.com,
desyatnyuk.oksana@tneu.edu.ua, svitlana.galeshchuk@dauphine.fr

**Abstract.** The study aims at predicting 10-day stock return movements using heterogeneous data over the timespan of 5 years such as historical stock performance at the market and the news feed with information on the particular firm's asset. Feature engineering helps reduce the number of variables used in the classification model as it excludes multicollinearity. A suite of parametric and non-parametric machine learning methods has not provided satisfactory accuracy, i.e., the random forest ensemble gives only 66% precision at the out-of-sample data using all features and 51% with only historical data from the stock market. It motivated us to develop the convolutional neural network architecture which delivered significantly better results.

**Keywords:** classification, stock market, prediction, machine learning, convolutional neural networks



## 1 Introduction

Stock exchange prediction is a longstanding challenge that spurs interest in time-series modelling, pattern detection, analysis of macroeconomic and market data among both academics and practitioners. Also, our research contributes to the domain with its *general objective* to predict the directional change of stock exchange returns.

Predictability of stock prices from the past and current information is a fundamental basis for modern trading technics with implications in investing. It constitutes one of the most profound controversies between academics and market participants. Despite that, fundamental and technical analyses are still used by foreign exchange professionals to predict movements in the currency market due to the belief that price fluctuations will reflect known patterns.

Technical analysis implies three main principles (Neely & Weller (2011): (1) assets price history uses all relevant information, so any research assets fundamentals is pointless; (2) assets prices are moving with trends, and that is a circumstantial factor for academic investigation due to the fact that trends imply predictability and allow the traders to get the profits; (3) history tends to be repeated itself. The traders use it into adherence to some patterns with similar conditions.

Fundamental analysis involves the use of economic data (e.g., production, consumption, disposable income) to forecast prices.

However, researchers often do not take into account nonlinearities between economic data, political, behavioural factors and financial markets. Heaton et al. (2016) point out that the possibly relevant data for financial markets prediction is extensive, while the importance of the data and the potentially complex interactions in the data are not well specified by financial economic theory (see also Engel, 2013). Behavioural factors are frequently omitted in the models.

Since financial markets are complex, evolutionary, noisy, and nonlinear dynamic system (Huang & Tsai, 2009), more adaptive and flexible mechanisms are required to improve forecasting accuracy (Cavalcante et al., 2016). This motivates researchers to investigate the ability of more flexible methods to study financial markets, in particular, machine learning methods (see Chen et al., 2015, Patel et al., 2015).

Nevertheless, not only methodology defines the experimental outcomes. The quality and richness of data together with feature engineering play a crucial role in the high accuracy at the out-of-sample set. The choice of variables and their tuning paves the way to the convincing results.

The domain experts usually determine the set of features based on the prior knowledge of the dependent variable. As we mentioned above, researchers tend to build on either macroeconomic indicators or historical market data or both of them. However, a significant part of economic society believes behavioural factors such as news and public reaction may have a significant influence on stock prices. We decide to test this mainstream of economic thoughts by developing our predictive model with the extant machine learning methods. This conclusion along with the available data shaped and specified our general objective mentioned at the beginning of this section. Now we define it as follows: This *paper* particularly *aims* at developing a prediction method for the directional change of stock exchange 10-day returns with the cutting-edge machine learning approaches by integrating historical market features and the news data.

The paper is organized as follows: section 2 introduces the data and its descriptive analysis. Section 3 elaborates on the methodological set-up of the study and the evaluation technics considered. Section 4 presents the results of our model and its comparison with the outputs provides by the other existent methods. Section 5 concludes with comments and directions for future research.

# 2 Data

## 2.1 Data Sources

The data for stock exchange performance is publicly available. Hence, we do not experience any challenges in getting it. However, collecting the news and their processing is a time-consuming and labor-intensive task. Many datasets are now available for training the models and Kaggle [1]contributes to the machine learning society by

[1] https://www.kaggle.com/c/two-sigma-financial-news

publishing some data from trustworthy sources. Kaggle competition "Two Sigma: Using News to Predict Stock Movements" includes the market and news data from 2007 to 2016. Moreover, Thomson Reuters, the mass media and information firms with a longstanding tradition of news procurement, is a point of supply for this dataset.

The market data reflects the following indicators for the US-listed firms and their assets:

1) raw open-to-open daily returns, market-residualized open-to-open returns;

2) 10-day raw open-to-open daily returns, 10-day market-residualized open-to-open daily returns;

3) raw close-to-close daily returns, market-residualized close-to-close daily returns;

4) 10-day raw close-to-close returns, 10-day market-residualized close-to-close returns;

5) daily trading volume in shares;

6) daily open price;

7) daily close price;

8) 10-day forward market-residualized open-to-open daily returns.

The news table comprises the data on the articles published concerning the particular company and its assets: the title, source, sentiment (negative, neutral, positive) of a story, words count, novelty vis-à-vis previous news (12-hour novelty, 24-hour novelty, 3 –day, 5-day, 7-day)., volume of news (12-hour volume, 24-hour volume, 3 –day, 5-day, 7-day), news relevance, sentiment scores (positive score, negative, neutral, general (binary)), news urgency.

Our task is to predict the directional change for the 10-day forward market-residualized open-to-open daily returns (whether it will go down, stay stable, go up). For more details on the dataset and its variables please follow the link.[2]

### 2.2 Descriptive Statistics and Feature Engineering

Market dataset accounts for circa 4 mln observations (3,979,902) for more than 2000 firms. We first create some new variables as "price difference" (the ratio between the difference in the close and open prices and open price), "volume percentage change", "absolute change. Having this rich dataset, the problem of missing values occurred. We simply impute the rows containing the 'nan' values.

Linear correlation analysis (Fig. 1 a) does not show clear dependencies between 10-day forward open returns and the other variables. However, it provides insights into possible multicollinearity to avoid in developing the model. Jaccard index measures the non-linear dependencies between the sets of data. We calculated first the directional change for each variable as:

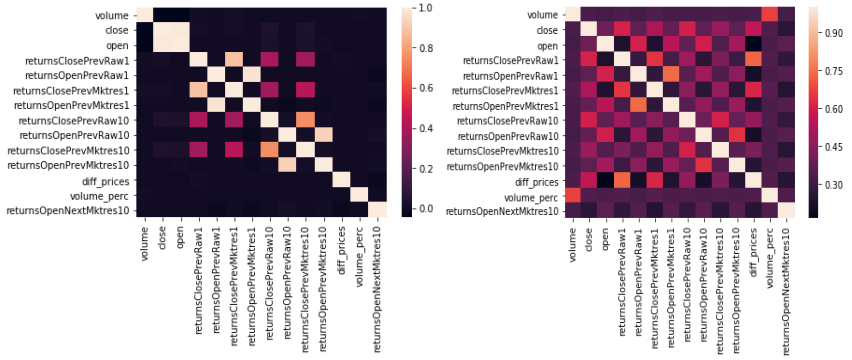
$$b[t] = 1 \text{ if } x[t+1]>x[t] \text{ and } 0 \text{ otherwise}$$

The Jaccard index for two Boolean arrays may in our case be defined as:

[2] https://www.kaggle.com/c/two-sigma-financial-news/data

$$J(X,Y) = \frac{C_{0X} + C_{Y0}}{C_{XY} + C_{0X} + C_{Y0}}$$

where $C_{XY}$ represents the number of occurrences when both X and Y are equal to 1. $C_{0X}/C_{Y0}$ if X/ Y are equal to 0. Fig.1 b depicts the results that support the conclusion about the non-linear relationship between 10 day forward returns and the rest of open-to-open returns. Moreover, interestingly the difference in close and open prices shows relates to the 1-day close-to-close returns.




a) b)

**Fig.1.** a) Correlation between the variables; b) Jaccard index measure of the similarity between the sets of data

Data distribution on Fig.2 shows that close-to-close returns generally stay close to the mean, while open-to-open ones are more dispersed. The plot also detects the outliers: we impute the data with 10-day future returns that violate the boundaries [-800; 800] filtering away circa 16 000 observations. Based on the output of descriptive statistics, we decide to ignore close-to-close returns in our experimental set-up.

The next step explores the news dataset. The table contains many possible variables but some feature engineering is necessary to avoid overfitting and multicollinearity. We create five new features taking into consideration relative importance of each indicator:

(i) sentiment_positive: ('sentimentPositive'*'relevance')/('urgency'*0.2 "noveltyCount12H"*0.1noveltyCount24H)

(ii) sentiment_negative: 'sentimentNegative'*'relevance')/('urgency'*0.2 "noveltyCount12H"*0.1noveltyCount24H)

(iii) sentiment neutral: 'sentimentNeutral'*'relevance')/('urgency'*0.2 "noveltyCount12H"*0.1noveltyCount24H"

The volume indicators are in absolute values. We merged market and news data on dates and firm names. We used Scikit-Learn Python library to scale the features.

We run stratified split data on train and test sets with a ratio: 80:15. Thus we obtain same ratio of 0 and 1 classes for the target variable in every set.

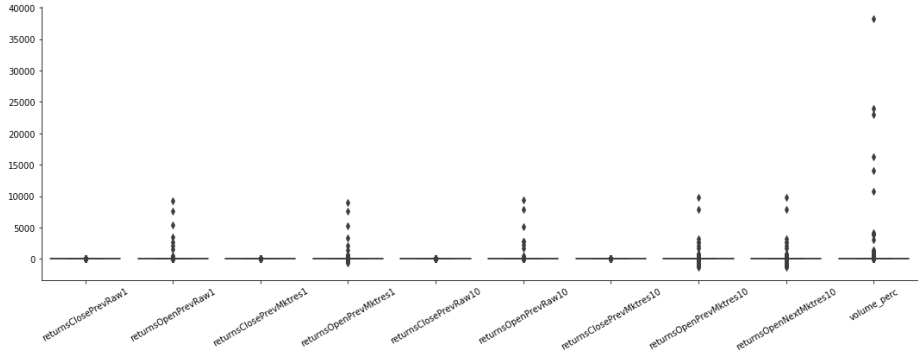



**Fig.2.** The distribution of data

# 3 Methodology

Recall from the Introduction that our goal is a directional prediction for the 10-day forward rate of stock return. We use the following formal definition of the directional change: Define the direction of change $z_k(t)=1$, if the rate increases, i.e. if $y_{(t+1)}-y_t>0$; otherwise, $z_k(t)=0$. A k-period ($k$ = 1 in our set-up) forward prediction model is evaluated by its classification accuracy on out-of-sample observations, where classification accuracy is defined as the percentage of test cases for which the predicted direction of change $\hat{z}_k(t)$ equals the true direction of change $z_k(t)$.

This section elaborates on the baseline technics used, developed method and the evaluation of the results.

## 3.1. Baseline Methods

We apply the number of extant classification methods to help in prediction of directional movement for 10-day forward returns. The following methods are used[3]:

*Fixed effects linear regression* reveals the linear dependencies among the dependent and independent variables vis-à-vis each firm. Python library *Linearmodels*[4] includes fixed-effects panel regression models. Our model may be summarized by the following equation:

$$\hat{y} = \alpha + \beta X + \delta F + \varepsilon,$$

where $\alpha$ is an intercept, X is a vector of dependent variables, F represents firms' fixed effects, $\varepsilon$ is an error. Then we compute the difference between prognosed and the previous value to determine the directional change.

Simple *logistic regression* calculates weighted sum of input variables (like linear regression) outputting the probability of each instance to belong to a positive class.

[3] we skip random walk model since it has been previously implemented by the Kaggle competiotion founders to show its poor accuracy

[4] https://pypi.org/project/linearmodels/

Our set-up exploits all the available training instances to train logistic regression relaxing on the firm's individual effects. Python Library Scikit Learn (linear models)[5] allows logit estimations.

*Decision Trees* usually handle well linearities and non-linearities in the data.[6] The method is versatile and simple in interpretation. We do not need to run feature scaling while training the data. Scikit Learn provides us with decision trees implementation. The method is, however, prone to be sensitive to the data variation

*Random forest* helps overcome the disadvantages of single decision tree by summarizing and averaging predictions over the number of trees. It is an ensemble learning approach that uses the outputs of the individual predictors as votes. If positive class gets more votes, the method will return the corresponding result. Again, Scikit Learn comprises random forest as a part of its ensemble methods[7].

*Shallow multilayer perceptron* (MLP) has the ability capture non-linearity between features. We use the following architecture to train the model: 16-300-1, where 1' is a number of input neurons, 300 neurons in the hidden layer and we have binary classification problem, hence 1 final output. Adam is used for the model optimization.

We use Randomized Search to tune the parameters for decision trees, random forest, MLP (e.g., it determines 512 as an optimal number of trees in the Random forest classification).

### 3.1. Convolutional Neural Network

The Convolutional Neural Networks (CNN) belongs to a family of deep learning methods with empirically proved classification ability on large datasets. CNN are capable to learn complex patterns due to the idea of receptive fields when each hidden neuron is connected not with all input neurons but corresponding local part of them. Moreover, CNN detect learn patterns anywhere in the input data, they have fewer parameters that the vanilla deep learning networks which makes CNN less prone to overfitting. It motivates us to use the CNN architecture in our study. We define the structure of the CNN by trials and errors. The architecture that provides the highest accuracy on the test set is describe on Fig. 3.

[5]https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
[6] https://scikit-learn.org/stable/modules/tree.html
[7]https://scikitlearn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.htm

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d_1 (Conv1D)            (None, 15, 64)            192
_________________________________________________________________
conv1d_2 (Conv1D)            (None, 14, 64)            8256
_________________________________________________________________
max_pooling1d_1 (MaxPooling1 (None, 7, 64)             0
_________________________________________________________________
dropout_1 (Dropout)          (None, 7, 64)             0
_________________________________________________________________
conv1d_3 (Conv1D)            (None, 6, 128)            16512
_________________________________________________________________
max_pooling1d_2 (MaxPooling1 (None, 3, 128)            0
_________________________________________________________________
dropout_2 (Dropout)          (None, 3, 128)            0
_________________________________________________________________
conv1d_4 (Conv1D)            (None, 2, 256)            65792
_________________________________________________________________
conv1d_5 (Conv1D)            (None, 1, 256)            131328
_________________________________________________________________
dense_1 (Dense)              (None, 1, 2)              514
=================================================================
Total params: 222,594
Trainable params: 222,594
Non-trainable params: 0
```

**Fig. 3.** The CNN Architecture

## 4 Results and Conclusion

Table 1 describes the output accuracy of the predictors on the test sets. Fixed-effects linear regression model explains only small part of the data variance as $R^2$ is insignificant equal on the training set. Thus, we do not need to verify it on the out-of-sample data.

Table 1. Classification accuracy on the test set (%)

| Dataset/Classifier | Logistic Reg. | Decision Tree | Rand Forest | MLP | CNN |
|---|---|---|---|---|---|
| **Market+News Data** | 53.7 | 54.1 | 61.0 | 66.4 | 73.4 |
| **Market Data** | 48.8 | 48.3 | 51.0 | 51.6 | 61.2 |

As you can see the prediction accuracy of the developed CNN is higher than those of the other methods. We try to run same models for the market data only without counting the news data. The results even with the CNN architecture is significantly poorer. It empirically proofs that at our dataset which comprises the data from stock exchange market over 10 years with over 4 mln observations the news information is essential in the forward market prediction.

## 5 Further Research

The results make contribution to the market theory proving that the news data is significant for prediction of stock exchange. We plan to extend the dataset with the data from Google Trends as we believe it has a predictive significance. Moreover, we envisage using the LSTM with attention mechanism in our future studies.

## References


1. Cavalcante, R. C., Brasileiro, R. C., Souza, V. L., Nobrega, J. P., & Oliveira, A. L. (2016). Computational intelligence and financial markets: A survey and future directions. Expert Systems with Applications, 55, 194-211.
2. Chen, K., Zhou, Y., & Dai, F. (2015, October). 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.
3. Engel, C. (2014). Exchange rates and interest parity. In Handbook of international economics (Vol. 4, pp. 453-522). Elsevier.
4. Huang, C. L., & Tsai, C. Y. (2009). A hybrid SOFM-SVR with a filter-based feature selection for stock market forecasting. Expert Systems with Applications, 36(2), 1529-1539.
5. Heaton, J. B., Polson, N. G., & Witte, J. H. (2016). Deep learning in finance. arXiv preprint arXiv:1602.06561.
6. Neely, C. J., & Weller, P. A. (2011). Technical analysis in the foreign exchange market. Federal Reserve Bank of St. Louis Working Paper No.
7. Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock market index using fusion of machine learning techniques. Expert Systems with Applications, 42(4), 2162-2172.