

Deep Learning for Stock Prediction using Stock Embeddings and Temporal Correlation Measures

Thomas Greninger
ETH Zurich
grethoma@student.ethz.ch

David Yenicelek
ETH Zurich
yedavid@ethz.ch

Konstantinos Bardis
ETH Zurich
bardis@lav.mavt.ethz.ch

Lukas Ryll*
University of Cambridge
lr487@cam.ac.uk

Miller Mendoza*
ETH Zurich
mmendoza@ethz.ch

Abstract—The prediction of stock market assets is an increasingly difficult task in times of big data, and requires quick analysis and automation of prediction methods. Traditional mathematical stochastic process models and simple machine learning models prevail in this community. In this paper we demonstrate that neural network models can be just as competitive as traditional machine learning models in the area of stock price prediction.

Our novel contributions are two-fold: 1. We introduce a novel neural architecture model through the use of stock-embeddings, which achieves state-of-the-art accuracy in the given task; 2. We experiment with a feature selection algorithm commonly used to predict earthquakes and solar flares through temporal correlation measurements that can be used as additional input to predict stock prices.

These contributions pave the way for future research in the direction of combining deep learning models with temporal correlation measurements, as well as using embeddings in domains different than natural language processing.

Index Terms—stock-embeddings, temporal correlation measures, deep learning, machine learning

I. INTRODUCTION

Artificial neural networks are becoming increasingly common in the financial markets thanks to their ability to handle large amounts of data that show non-linear behaviour, include discontinuities, and high-frequency polynomial components. Different approaches have been presented in the past, including but not limited to the use of: multi-layer feed forward neural network trained with back-propagation [12], [13], [18], support vector machines [14], [15] and hybrid methods [16], [17].

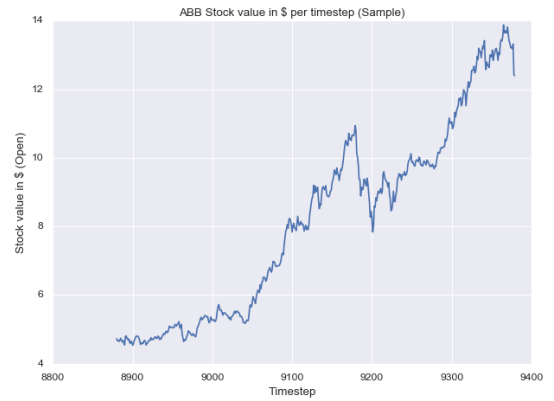
Up until now the application of deep learning in financial markets is less established [11]. Predicting complex, high-dimensional and noisy time series data is a challenging task. Some previous works in this area that are worth mentioning include [8]–[10].

The probability density of returns of stock markets often resembles a power-law distribution [23]. This property is also found in the distribution of magnitudes of earthquakes and solar flares [21], [22]. This similarity opens up the possibility that the statistical tools used to predict these physical phenomena can be applied to predict stock prices. Therefore in this work, we explore this possibility and use it in the context of neural networks.

*advisor

II. DATA

Fig. 1. A sample of the ABB Stock value in \$ at each timestep (day).

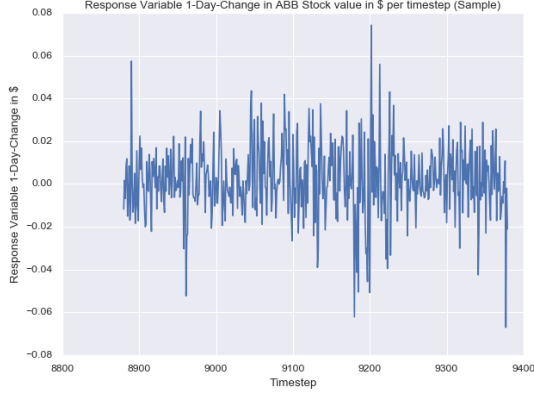


The following dataset from Kaggle was used for the present project <https://www.kaggle.com/borismarjanovic/price-volume-data-for-all-us-stocks-etfs>. It contains the historical daily price and volume data for 7195 US-based stocks between 1/1962 and 1/2017, including the open, high, low and close values. The data have been adjusted for dividends and splits. Applying a 80%-10%-10% train-validation-test split, there are in total 12'023'501 samples in the training set, and 1'335'945 samples in the validation and test set respectively. A sample of the "open" prices can be seen in Fig. 1.

The following data are extracted from the aforementioned variables: 1) return one day ahead, $r_{i+1} = \frac{o_{i+1} - o_i}{o_i}$; 2) return of previous day, $r_{i-1} = \frac{o_i - o_{i-1}}{o_{i-1}}$; 3) return of previous two days, $r_{i-2} = \frac{o_i - o_{i-2}}{o_{i-1}}$; 4) return of previous five days, $r_{i-5} = \frac{o_i - o_{i-5}}{o_{i-5}}$, where r is the return and o_i the opening value on the i -th day, accordingly, as can be seen in Fig. 3. These extracted variables are appended to the data table. Further, the stock symbols and dates are both encoded into integer values. Finally, all NaN values are removed from the data.

The first generated indicator is the projected return for the following days opening price. Its sign indicates if the stock will go up or down, and this is our response variable. We construct a classification- instead of a regression problem, as the literature suggests that the former performs better than the latter in predicting financial market data [19] [20].

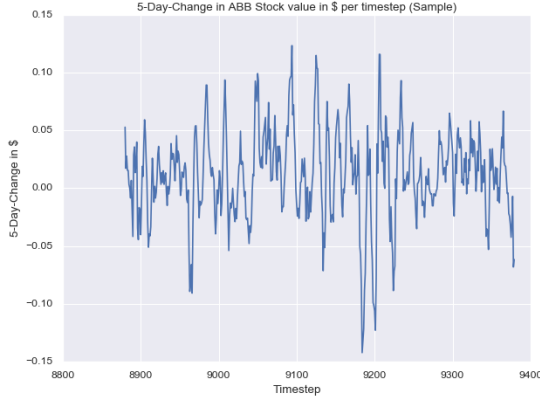
Fig. 2. A sample of the response variable for a given time period. This response variable is the stock value change in \$ for the next day.



III. FEATURE SELECTION

Feature selection, by removing the irrelevant and/or redundant features, plays an important role in compressing the data dimension as well as improving learning accuracy, reducing learning time, and simplifying the learning results [4]. Various approaches exist and have been applied in many fields. Particularly with regard to stock market prediction, previous feature selection and feature engineering approaches included: 1) the calculation of technical attributes, such as moving averages, volatility, on balance volume, momentum, relative strength index [5]; 2) the application of filters and wrappers [6], [7]; as well as 3) hybrid methods [2], [3].

Fig. 3. Change in stock price over a 5 day period (over a sampled time period)



In this study, a novel method to assess the importance of individual features in predicting the target variable (as shown in Fig. 2) is assessed. It has previously been used in the field of earthquake and solar flare predictions [21], [22]. The method's strengths lie in its weak assumptions. It involves the computation of the probability that the return will be greater than a given threshold $r > r_0$ given that the i -feature exceeds a threshold $f_i > f_0$. This conditional probability reads as follows

$$P(r > r_0 | f_i > f_0) = \frac{N(r_0, f_0)}{N(f_0)} \quad (1)$$

where $N(r_0, f_0)$ is the number of subsequent events with both $r > r_0$ and $f_i > f_0$ while $N(f_0)$ is the number of couples with $f_i > f_0$. The time series of the target variable is reshuffled and the conditional probability $P(r^* > r_0 | f_i > f_0)$ where r^* refers to an individual realization of the reshuffled time series. When $P(r > r_0 | f_i > f_0)$ lies sufficiently outside of the confidence interval of $P(r^* > r_0 | f_i > f_0)$ distribution, a significant statistical relevance of the f_i feature is inferred.

In our case r_0 is 0 and $P(r_0 > 0 | f_i > f_0)$ refers to the a posteriori probability of our response variable (stock return from today to the next day) is positive, given our feature takes a certain value ($f_i > f_0$). On the other hand by the reshuffling simulation we estimate the zero-line $P(r^* > r_0 | f_i > f_0)$ and its standard deviation. If the former value ($P(r_0 > 0 | f_i > f_0)$) deviates enough from the later estimation ($P(r^* > r_0 | f_i > f_0)$) we consider the feature f_i as relevant for our prediction problem. The example in the table below shows the calculations for ABB data from February 2005 to November 2017. As response we take, as always, the return from today to the next day, and as potential feature the return from the previous day to today.

TABLE I
EXAMPLE RELEVANCE CALCULATION OF FEATURE

	Number
Number of Days	3192
$f < -0.0087$	799
$f < -0.0087$ and $r_0 > 0$	429
$P(r_0 > 0 f_i > f_0)$	0.53692
Estimation of $P(r^* > r_0 f_i > f_0)$	0.5194
Estimation of Standard-deviation	0.015

In the example above the difference between $P(r_0 > 0 | f_i > f_0)$ and $P(r^* > r_0 | f_i > f_0)$ is 1.14 standard-deviations. For it alone this is a weak indication for the relevance of the feature.

IV. MODEL

A. Baseline

The baseline model was taken from <https://www.kaggle.com/christofhenkel/market-data-nn-baseline> and appropriately adapted for the present application. It is a fully connected neural network that includes an embedding layer [24]. The activation units are *relu*'s after each connected layer while a *logistic sigmoid* function is applied in the output layer.

B. Proposed

Our proposed neural network follows the aforementioned simple baseline network. The following table describes the architecture of the network we use. We use a sequential model, and will also elaborate on how the embeddings are used. We use Keras [1] to implement the model. As such, our terminology will consist of Keras modules to describe the layers we use.

The model has two inputs, one numerical input (numerical features like 1-day-difference, 2-day-difference etc.), and label

input (i.e. stock symbols that are used as part of the embedding). These are split into two different streams. Stream 1 and it's underlying sequential sub-model for the numerical items, as well as stream 2 and it's underlying sequential sub-model for the embedded items.

TABLE II
MODELS OVERVIEW

Layer No.	Type	Latent Dimensions
1.1	Input (Numerical)	-
1.2	Batch Normalization	-
1.3	Fully Connected	128
1.4	Fully Connected	64
2.1	Input (Stock Label for Embedding)	-
2.2	Embedding	10
2.3	Fully Connected	32
3	Concatenate (Output of 1.4 and 2.3)	-
4	Fully Connected	64
5	Fully Connected	1

We use *relu* activation units after each fully connected layer, except for the output layer for which a *sigmoid* function was used. The last layer predicts the probability that the stock will exhibit a positive return on the following day.

V. RESULTS & DISCUSSION

A. Futures

The section "Feature selection" presents the results from the application of the proposed feature selection algorithm for a given stock. In the example there we find the probability that the return of the stock for the next day given the return of the previous day assumes a certain value is significant.

The application of this method does not have to be limited to the same stock but can be also be generalized to multiple stocks. It can also be applied to a wide range of other feature time series such as interest rates changes, commodity price changes. The method is useful in financial markets because of its weak assumptions. In particular, no distribution assumptions are made and no linearity assumptions are made. Any such assumptions have proved false in financial markets anyway. A good feature selection algorithm takes a burden of any downstream prediction algorithm

B. Training

Before we discuss the training accuracy, it is important to notice some significance levels. Because we employ a binary classification task (whether the underlying asset will increase in returns, or decrease in values), we model the prediction task as a Bernoulli distribution (as can also be validated using the Random Model). Assuming a Bernoulli distribution, the mean and standard deviation of a Bernoulli distribution follows the following formulae:

$$\mu = p \quad (2)$$

$$\sigma = \sqrt{p \times (1 - p)} \quad (3)$$

where p denotes the probability that the event will have outcome "1" (in our case, will predict an underlying asset to have positive returns). Using these formulae, we use $p = 0.5$ as our hypothesis (which is validated by the random model). Given that we use more than 1 million samples for testing alone, the significance of the found accuracies is assumed to be negligible. As such, we round the accuracy values to 2 decimal places (as percentages), and assume a noise value of 0.1. Any bigger changes can be considered "significant" (as per definition of the standard deviation).

In addition, all models were trained, validated, and tested on the exact same sample set, which provides a solid ground for comparison between models. Please notice that all accuracy measures are given in percentages. We decided to not use the validation loss, but to use accuracy measures as a comparative value between models, as this gives 1. more intuition and is 2. directly related to the binary cross-entropy of the model. Doing this, and making sure the results are statistically significant, we weed out the possibility that exchanging the binary cross-entropy loss with the measure of accuracy diminishes any qualitative expressiveness.

TABLE III
REFERENCE MODELS (NON-NEURAL NETWORK MODELS) OVERVIEW

Model	Train Accuracy	Test Accuracy
Random	50.16	50.19
Decision Tree	53.76	53.46
XGBoost	53.93	53.56

As hyperparameters, we set "min samples leaf = 5000" for the decision tree, and kept the default hyperparameters for all the other models. As one can see, the xgboost (which employs a linear gradient algorithm) has the highest performance.

TABLE IV
ACCURACY OF DIFFERENT NEURAL NETWORK ARCHITECTURES WITH DIFFERENT CONFIGURATIONS.

Significance	Embedding	Epochs	Train Accuracy	Test Accuracy
No	Yes	3	53.55	53.48
No	No	20	53.44	53.48
No	Yes	20	54.22	54.03
Yes	No	20	53.38	53.41
Yes	Yes	20	53.79	53.35
Yes	Yes	20	54.09	53.93

One can see that the neural network model with the stock embeddings performs best, while the significance measures do not improve the model. The improvement due do the embeddings are significant (compared to non-embeddings), as this is three standard deviations away from the XGBoost model, as well as the non-embeddings model (at least 0.50 % difference, which is greater than σ). This implies that the learned results are significant, and not due to random fluctuations.

VI. SUMMARY

Deep learning architectures are increasingly winning relevance in the fields of sequence prediction and computer vision tasks. However, the use of deep learning models within the stock market is rather scarce compared to traditional machine learning algorithms and traditional stochastic prediction models. In this paper we show that neural network models are just as competitive as traditional machine learning models in the area of stock price prediction. Our contributions are two-fold. First, we compare traditional machine learning algorithms with a neural network baseline, showing that the neural network is indeed capable of achieving similar accuracy levels. Second, we introduce a novel neural architecture model through the use of stock-embeddings (similar to word-embeddings), which achieves state-of-the-art accuracy in the task of stock price prediction.

Lastly, we also show that a feature selection algorithm commonly used to predict earthquakes and solar flares through temporal correlation measurements can be used to feed in as additional input to predict stock prices. The use of this algorithm in stock market predictions has not been investigated before, to the best of our knowledge. These contributions pave the way for future research in the direction of combining deep learning models with temporal correlation measures, as well as using embeddings in domains different than natural language processing.

REFERENCES

- [1] Chollet, François and others, "Keras", <https://keras.io> (2015),
- [2] Tsai, Chih-Fong, and Yu-Chieh Hsiao, "Combining multiple feature selection methods for stock prediction: Union, intersection, and multi-intersection approaches." *Decision Support Systems* 50.1 (2010): 258-269.
- [3] Lee, Ming-Chi. "Using support vector machine with a hybrid feature selection method to the stock trend prediction." *Expert Systems with Applications* 36.8 (2009): 10896-10904.
- [4] Cai, Jie, et al. "Feature selection in machine learning: A new perspective." *Neurocomputing* 300 (2018): 70-79.
- [5] Vanstone, Bruce, and Gavin Finnie. "An empirical methodology for developing stockmarket trading systems using artificial neural networks." *Expert systems with applications* 36.3 (2009): 6668-6680.
- [6] Lee, Ming-Chi. "Using support vector machine with a hybrid feature selection method to the stock trend prediction." *Expert Systems with Applications* 36.8 (2009): 10896-10904.
- [7] Lin, Fengyi, et al. "Novel feature selection methods to financial distress prediction." *Expert Systems with Applications* 41.5 (2014): 2472-2483.
- [8] Kuremoto, Takashi, et al. "Time series forecasting using a deep belief network with restricted Boltzmann machines." *Neurocomputing* 137 (2014): 47-56.
- [9] Yoshihara, Akira, et al. "Predicting stock market trends by recurrent deep neural networks." *Pacific rim international conference on artificial intelligence*. Springer, Cham, 2014.
- [10] Ding, Xiao, et al. "Deep learning for event-driven stock prediction." *Ijcai*. 2015.
- [11] Cavalcante, Rodolfo C., et al. "Computational intelligence and financial markets: A survey and future directions." *Expert Systems with Applications* 55 (2016): 194-211.
- [12] Lasfer, Assia, Hazim El-Baz, and Imran Zuolkernan. "Neural Network design parameters for forecasting financial time series." *Modeling, Simulation and Applied Optimization (ICMSAO)*, 2013 5th International Conference on. IEEE, 2013.
- [13] Dhar, Satyajit, Tuhin Mukherjee, and Arnab Kumar Ghoshal. "Performance evaluation of Neural Network approach in financial prediction: Evidence from Indian Market." *Communication and Computational Intelligence (INCOCCI)*, 2010 International Conference on. IEEE, 2010.
- [14] Bao, Yukun, et al. "A comparative study of multi-step-ahead prediction for crude oil price with support vector regression." *Computational Sciences and Optimization (CSO)*, 2011 Fourth International Joint Conference on. IEEE, 2011.
- [15] Xie, Guo-qiang. "The optimization of share price prediction model based on support vector machine." *Control, Automation and Systems Engineering (CASE)*, 2011 International Conference on. IEEE, 2011.
- [16] Zhu, Bangzhu, and Yiming Wei. "Carbon price forecasting with a novel hybrid ARIMA and least squares support vector machines methodology." *Omega* 41.3 (2013): 517-524.
- [17] Patel, Jigar, et al. "Predicting stock market index using fusion of machine learning techniques." *Expert Systems with Applications* 42.4 (2015): 2162-2172.
- [18] Martinez, Leonardo C., et al. "From an artificial neural network to a stock market day-trading system: A case study on the bmf bovespa." *Neural Networks*, 2009. *IJCNN 2009. International Joint Conference on*. IEEE, 2009.
- [19] Enke, D. and Thawornwong, S. (2005). The use of data mining and neural networks for forecasting stock market returns. *Expert Systems with Applications*, 29(4):927940.
- [20] Leung, M. T., Daouk, H., and Chen, A.-S. (2000). Forecasting stock indices: A comparison of classification and level estimation models. *International Journal of Forecasting*, 16(2):173190.
- [21] E. Lippiello L. de Arcangelis and C. Godano. Influence of Time and Space Correlations on Earthquake Magnitude. *PHYSICAL REVIEW LETTERS*, published 22 January 2008
- [22] M. Mendoza, A. Kaydul, L. de Arcangelis, J.S. Andrade Jr and H.J. Herrmann, Modelling the influence of photospheric turbulence on solar flare statistics, *Nature Communications*, Published 23 Sep 2014
- [23] Gopikrishnan, Parameswaran, et al. Scaling of the distribution of fluctuations of financial market indices. *Physical Review E* 60.5 (1999): 5305.
- [24] T. Mikolov, et al. Distributed Representations of Words and Phrases and their Compositionality. *NIPS Proceedings* 2013.