Learning Sentiment-aware Trading Strategies for Bitcoin leveraging Deep Learning-based Financial News Analysis

N. Passalis, S. Seficha, A. Tsantekidis, and A. Tefas

Artificial Intelligence and Information Analysis Lab, Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece
{passalis, sseficha, avraamt, tefas}@csd.auth.gr

Abstract. Even though Deep Learning (DL) models are increasingly used in recent years to develop trading agents, most of them solely rely on a restricted set of input information, e.g., price time-series. However, this is in contrast with the information that is usually available to human traders that, apart from relying on price information, also take into account their prior knowledge, sentiment that is expressed regarding various markets and assets, as well as general news and forecasts. In this paper, we examine whether the use of sentiment information, as extracted by various online sources, including news articles, is beneficial when training DL agents for trading. More specifically, we provide an extensive evaluation that includes several different configurations and models, ranging from Multi-layer Perceptrons (MLPs) to Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), examining the impact of using sentiment information when developing DL models for trading applications. Apart from demonstrating that sentiment can indeed lead to improved trading efficiency, we also provide further insight on the use of sentiment-enriched data sources for cryptocurriences, such as Bitcoin, where its seems that sentiment information might actually be a stronger predictor compared to the information provided by the actual price time-series.

Keywords: Financial Trading · Sentiment Analysis · Deep Learning · Sentiment-aware Trading

1 Introduction

Deep Learning (DL) methods are increasingly used in recent years for developing intelligent agents for financial trading [5, 7, 11, 14, 18, 20], superseding, to a large extent, traditional methods, such as rule-based strategies. Indeed, powerful DL formulations led to models with enormous learning capacity, while their ability to seamlessly integrate with Reinforcement Learning (RL) methodologies allowed for directly optimizing trading policies to maximize the expected profit, even in the volatile and uncertain conditions that often exist in real markets [5, 9, 18]. Despite the encouraging results reported in the literature, current approaches

operate on a restricted set of input information, i.e., they mainly rely on time-series information regarding the price of assets. This is in contrast with the information that is usually available to human traders that, apart from observing price-related information, also take into account their prior knowledge, sentiment that is expressed regarding various markets and assets, as well as general news and forecasts.

Contrary to this, trading using DL is typically tackled as a problem that can be solved solely on relying on a single modality, i.e., price time-series, without taking into account any additional external information. Indeed, the additional complexity and development cost regarding collecting the appropriate data, pre-processing them and then transforming them into a form that can be exploited by DL models, often discourage DL researchers and companies from exploiting these valuable sources of information. Recent evidence suggests that external information, mainly provided in the form of sentiment regarding various financial assets [3, 4, 12, 19] and typically collected from social media, often has a positive effect on the accuracy of trading agents. However, little work has been done so far towards this direction, especially for exploiting large-scale datasets that contain news articles regarding financial assets.

The main contribution of this work is to examine whether the use of sentiment information, as extracted by various online sources, including news articles, is beneficial when training DL agents for trading. More specifically, in this paper, we aim to evaluate the impact of using sentiment information when training a wide variety of deep learning models, ranging from Multi-layer Percetrons (MLPs) to Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). To this end, we go beyond the existing literature that typically just evaluate a few handpicked models, with and without sentiment information, and we providing an extensive evaluation, often including more than 50 different configurations per architecture. Apart from confirming our initial hypothesis, we also provide some surprising results, demonstrating that for cryptocurriences, such as Bitcoin, sentiment information might actually be a stronger predictor compared to the information provided by price time-series.

The rest of the paper is structured as follows. First, we introduce the used notation and analytically describe the proposed method in Section 2. Then, we provided an extensive experimental evaluation in Section 3. Finally, Section 4 concludes the paper and discusses possible future research directions.

2 Proposed Method

In this Section we introduce the proposed data processing and fusion pipeline, as well as the employed financial forecasting setup. For the rest of this Section, we assume that both the forecasting, as well as the sampling time-step is set to one day. This is without loss of generality, since the proposed method can be trivially extended to work on longer or smaller time horizons, given that the appropriate data are collected.

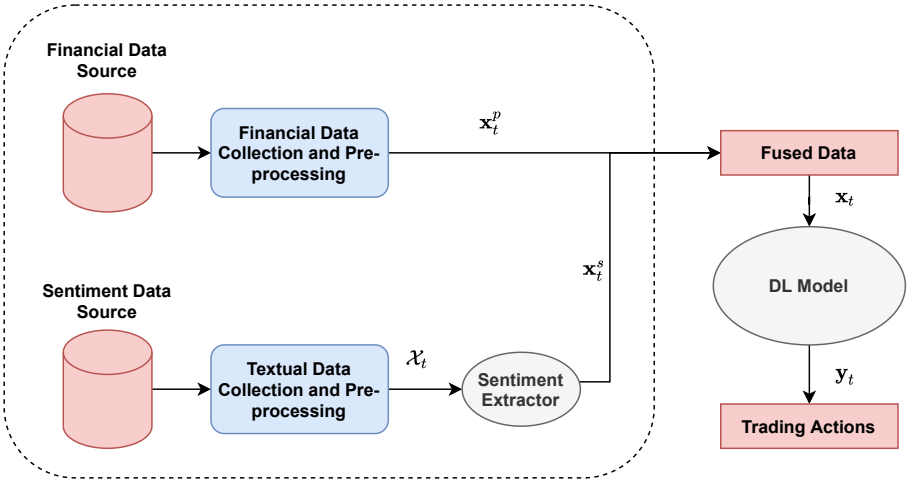

Fig. 1. Proposed trading pipeline: The trained DL models rely on two different information sources: a) financial data sources, that provide price information, as well as b) sentiment data sources, that provide sentiment information. After appropriately preprocessing both of them, the fused data are fed to a DL model that provides the trading signals.

The proposed data processing pipeline, along with the forecasting model are shown in Fig. 1. First, the DL model receives the raw price candles from a financial data source, e.g., an exchange. These data are then preprocessed in order to obtain a single scalar value for each time-step t that corresponds to the percentage change of the price of an asset as:

$$x_t^p = \frac{c_t}{c_{t-1}} - 1, \tag{1}$$

where c_t denotes the close price at time t. It is worth noting that this is among the most well-established financial data preprocessing approaches for extracting stationary features [17, 10]. Then, these percentage changes are aggregated for a windows of length L to form a vector that describes the price behavior of a specific asset during the last L steps:

$$\mathbf{x}_t^p = [\frac{c_{t-L-1}}{c_{t-L-2}} - 1, \ldots, \frac{c_t}{c_{t-1}} - 1] \in \mathbb{R}^L. \tag{2}$$

Note that we use bold notation to refer to the vector that contains the *history* of the previous L percentage changes at time t, i.e., $\mathbf{x}_t^p$, while we use a regular font to refer to the scalar percentage change at time t, i.e., x_t^p .

In this work, we propose to also employ sentiment information about a financial asset, as expressed in various online sources, to extract additional information that can be useful for predicting the future behavior of the said asset.

Let $\mathcal{X}_t$ denote a collection of textual documents that refer to the asset at hand and collected at time t, i.e., after time-step $t-1$ and until time-step t. Also, let $f_s(\mathbf{x}_d)$ denote a sentiment extractor that returns the sentiment of a document $\mathbf{x}_d$, where $\mathbf{x}_d$ is an appropriate representation of a textual document for the task of sentiment analysis, e.g., a sequence of the words that appear in the corresponding document [2]. In this work, we assume that the sentiment is represented as a scalar value that ranges from -1 (negative sentiment) to $+1$ (positive) sentiment. A sentiment extractor can be always used to extract such a scalar value as follows. For regression-based sentiment extractors we can directly use the extracted value, after appropriately normalizing it. On the other hand, for classification-based sentiment extractors, we can calculate the sentiment score simply by subtracting the confidence for the negative class from the confidence for the positive class. In this paper, we follow the latter approach, as we further explain in Section 3, by employing a state-of-the-art deep learning-based extractor that is fine-tuned on financial documents [2].

Therefore, for each time-step we can extract the *average polarity* regarding the asset at hand as follows:

$$x_t^s = \frac{1}{|\mathcal{X}_t|} \sum_{\mathbf{x}_d \in \mathcal{X}_t} f(\mathbf{x}_d), \tag{3}$$

where $|\mathcal{X}_t|$ denotes the number of text documents collected at time-step t. Then, we can similarly define the time-series that describes the sentiment over a horizon of L time-steps as:

$$\mathbf{x}_t^s = [x_{t-L-1}^s, \ldots, x_t^s] \in \mathbb{R}^L. \tag{4}$$

The most straightforward way to combine the price information ($\mathbf{x}_t^p$), with the sentiment information ($\mathbf{x}_t^s$) is to simply concatenate the corresponding vectors into a tensor $\mathbf{x}_t = [\mathbf{x}_t^p; \mathbf{x}_t^s] \in \mathbb{R}^{L \times 2}$. Then, this tensor is fed to the corresponding DL model, as shown in Fig. 1.

Several different approaches have been proposed in the literature for training DL models for financial trading, ranging from classification-based methods [16] to complex reinforcement learning setups which aim to simulate the trading environment [5]. In this work, we opt for following a classification-based setup, where a DL model is trained to predict the price movements that are more likely to lead to profit. More specifically, the ground labels for training the DL model are generated as:

$$l_t = \begin{cases} 1 & \text{if } \frac{c_{t+1}}{c_t} - 1 > c_{thres} \\ -1 & \text{se } \frac{c_{t+1}}{c_t} - 1 < -c_{thres} \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

where c_{thres} denotes the threshold for considering that a price movement is a potential candidate for performing a profitable trade. Therefore, the label "1" corresponds to a long position, the label "-1" to a short position, while the label "0" indicates market conditions that probably do not allow the specific agent to perform profitable trades, i.e., the agent should exit the market. Typically, c_{thres} is set to a value high enough to overcome any commission fees, as well as to

account for price slippage that might occur. Please note that during back-testing, the consecutive "long" or "short" positions do not lead to multiple commissions (since the agent simply keeps the already existing position open), while the exit position ("0") closes the currently open position and materializes any gain/loss acquired.

After generating the labels, the DL model can be directly trained using the cross entropy loss, i.e.,

$$\mathcal{L} = -\frac{1}{N}\sum_{t=1}^{N}\sum_{j=1}^{3}[\mathbf{l}_t]_j \log([g_{\mathbf{W}}(\mathbf{x}_t)]_j), \tag{6}$$

where $g_{\mathbf{W}}(\cdot)$ denotes the DL model employed for 3-way classification, $\mathbf{l}_t$ is the one-hot encoding of l_t, the notation $[\mathbf{x}]_j$ is used to refer to the j-th element of a vector $\mathbf{x}$, and N is the total number of time-steps for the training time-series, assuming that the time-series is continuous. Then, the model can be readily trained using gradient descent, i.e.,

$$\mathbf{W}' = \mathbf{W} - \eta\frac{\partial\mathcal{L}}{\partial\mathbf{W}}, \tag{7}$$

where $\mathbf{W}$ denotes the parameters of the model $g_{\mathbf{W}}(\cdot)$. In this work, mini-batch gradient descent is used, while the Adam algorithm is employed for the optimization [8]. Please also note that the main aim of this work is to evaluate whether using sentiment information can have a positive impact on the trading performance of a DL agent. To this end, we used three different models, i.e., a) a Multilayer Perceptron (MLP) (after appropriately flattening the input tensor into a vector), b) a 1-D Convolutional Neural Network and c) a Long-Short-Term Memory (LSTM)-based Network. All of these network architectures are widely used for training agents that can provide trading signals [15, 16, 20]. As we explain in detail in Section 3, we performed several experiments in order to evaluate the impact of using sentiment information on trading for a wide range of different setups and architectures.

3 Experimental Evaluation

In this Section, we provide the experimental evaluation of the proposed sentiment-aware trading pipeline. First, we briefly introduce the employed setup and hyper-parameters used for the conducted experiments, while we provide some additional information regarding the dataset used as a source of financial sentiment information for the conducted experiments. Then, we present and discuss the experimental results that confirm our initial hypothesis that sentiment can be a valuable source of information for training DL agents for financial trading.

Regarding the financial data source, we use the daily close prices for Bitcoin-USD (United States Dollar) currency pair. For extracting sentiment information we used a dataset published by BDC Consulting [1], which contains over 200,000 titles of financial articles collected from various sites that publish articles on

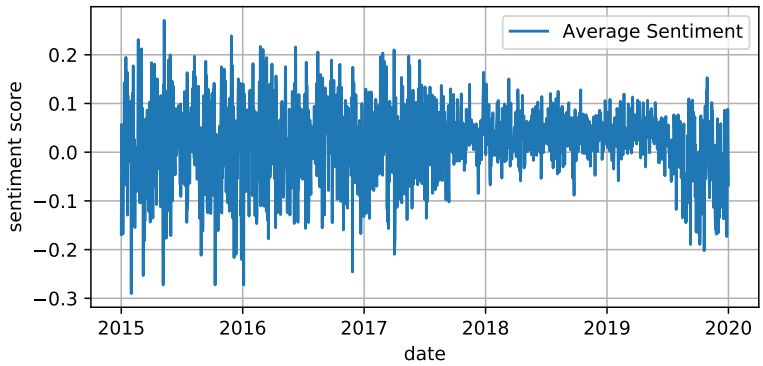

Fig. 2. Average sentiment score per day, as expressed by the documents contained in the BDC Consulting dataset. The finBERT model was used for extracting the sentiment of the titles of news articles published each day. Note that -1 corresponds to the most negative sentiment, while 1 corresponds to the most positive sentiment.

cryptocurrencies, such as Cointelegraph and CoinDesk. This dataset provides data for 5 years, from 2015 to 2020. Therefore, we used the first four years for training the DL models (2015-2020), while the last year (2019-2020) was used for performing the evaluation/back-testing of the trading agents. For both the training and testing datasets, we carefully aligned the textual data and price data, using the corresponding timestamps to ensure that no information from the future can leak into each training window.

The main motivation of this work is to evaluate the impact of using sentiment information across a wide range of DL models and configurations. To this end, we did not limit the evaluation on a smaller number of handpicked DL models. Instead, we evaluated a wide range of models for different hyper-parameters, including different number of layers, neurons per layer, learning rates and dropout rates. More specifically, for the MLP model we evaluated models with 1, 2 and 3 layers and 8, 16, 32, 64 and 128 neurons per layer. For the CNN models we evaluated models with 1, 2 and 3 convolutional layers (all followed by a final classification layer) and 4, 8 and 16 filters per layer and kernel sizes equal to 3, 4 and 5. Finally, for the LSTM models we experimented with 1, 2, 3 layers and 8, 16, 32, 64 and 128 neurons per LSTM layer. For all the configurations we used the Adam optimizer [8]. Therefore, we trained and evaluated the models with three different learning rates, i.e., 10^{-2}, 10^{-3} and 10^{-4}, as well as different dropout rates for the layers [13], i.e., 0.1, 0.2 and 0.4. All possible model configurations that were produced by the different combinations of the aforementioned parameters were trained and evaluated.

First, we examined the average performance of different configurations for three different kinds of inputs: a) price alone, b) sentiment alone, and c) com-

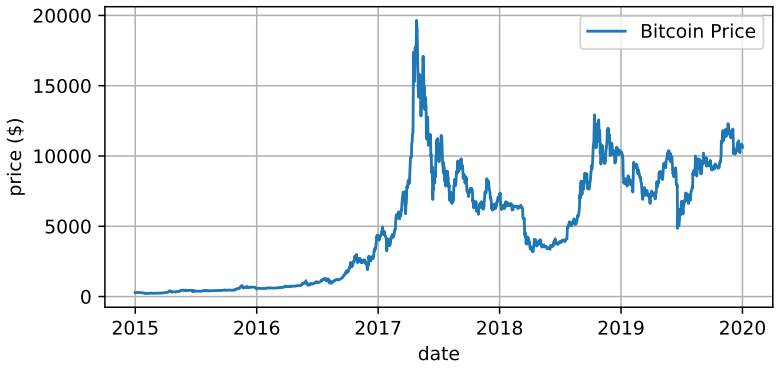

Fig. 3. BTC-USD price during the period 2015-2020

Table 1. Average Percentage (%) Profit and Loss (PnL) for the 50 top-performing configurations for each model (back-testing performed on the test set, i.e,. 2019-2020). The prediction horizon was set to 1 day. The lot size used is constant for the whole duration of the backtest regardless of accumulated profits or losses

Input Modality	**MLP**	**CNN**	**LSTM**
price	201%	219%	214%
sentiment	221%	**228%**	222%
price & sentiment	**224%**	**228%**	**224%**

bined price and sentiment. The evaluation results for the test set are provided in Table 1, where we compare the average Profit and Loss (PnL) metric [17], which allows us to estimate the expected profit and/or loss of a trading agent over a specific period of time. We report the average over the top-50 performing configurations, in order to ensure a fair comparison between the different models. Using sentiment information alone provides better PnL compared to just using the price, while combining the price and sentiment together allows for slightly improving the obtained results.

These results are also confirmed in the evaluation performed for the training set for individual agents, as provided in the left column of Fig. 4, where we also examine the convergence speed of the models by evaluating three different snapshots of the agents, i.e., on epoch 100, 200 and 300. Using price alone leads to a PnL of about 7. On the other hand, the obtained results clearly demonstrate that the DL models learn significantly faster when sentiment information is available, since there are very small differences between the three model snapshots (i.e., epochs 100, 200 and 300) and the final training PnL reaches values over

Fig. 4. Train (left column) and test (right column) PnL for MLP architectures trained on three different input sources: a) price alone, b) sentiment alone and c) combined price and sentiment.

30. This result demonstrates that sentiment-information for cryptocurriences, such as Bitcoin, might actually be a stronger predictor of its future behavior compared to the information provided by the price time-series. Combining price and sentiment information together shows a bit mixed result, possibly limiting overfitting issues that might occur when sentiment is used, since the maximum

train PnL in this case is around 20, while the models converge slower compared to only using sentiment input.

Indeed, similar results are obtained for the test evaluation, where the trained DL models are evaluated on unseen test data, as shown in right column of Fig. 4. The models that were trained using sentiment information consistently perform better compared to the corresponding models that were trained only using price information as input. Combining price and sentiment information seems to lead to slightly better behavior. Therefore, the obtained results confirmed our initial hypothesis that taking into account sentiment information can lead to agents that perform consistently better trades, since in all the evaluated cases using sentiment information as input increased the obtained PnL.

4 Conclusion

In this work we evaluated the impact of using additional information sources, that provide sentiment information, when developing DL agents for trading cryptocurrencies. The experimental results suggest that sentiment can indeed be a strong predictor of the future behavior of cryptocurrencies and can be effectively used for a wide range of DL architectures. These results indicate that similar results might be obtained even when more advanced learning algorithms are used, e.g., Deep Reinforcement Learning algorithms [18], potentially further improving their performance. Furthermore, the impact of other information sources, e.g., sentiment from related currencies or assets, can be also evaluated, since they consist a potentially useful additional pool of information, that is typically considered by human traders. Finally, more advanced ways of combining sentiment and price information, e.g., using transformer-based architectures for data fusion [6], could further improve the obtained results, since, in some cases, combining these two sources of information only led to marginal improvements over just using sentiment information.

5 Acknowledgment

This work has been co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH - CREATE - INNOVATE (project code: T2EDK-02094).

References

1. Analyzing Crypto Headlines - BDC Consulting, https://bdcenter.digital/insights/cryptocurrency/analyzing-crypto-headlines
2. Araci, D.: Finbert: Financial sentiment analysis with pre-trained language models. arXiv preprint arXiv:1908.10063 (2019)

3. Chantona, K., Purba, R., Halim, A.: News sentiment analysis in forex trading using r-cnn on deep recurrent q-network. In: Proceedings of the Fifth International Conference on Informatics and Computing. pp. 1–7 (2020)
4. Day, M.Y., Lee, C.C.: Deep learning for financial sentiment analysis on finance news providers. In: Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. pp. 1127–1134 (2016)
5. Deng, Y., Bao, F., Kong, Y., Ren, Z., Dai, Q.: Deep direct reinforcement learning for financial signal representation and trading. IEEE Transactions on Neural Networks and Learning Systems **28**(3), 653–664 (2016)
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
7. Jeong, G., Kim, H.Y.: Improving financial trading decisions using deep q-learning: Predicting the number of shares, action strategies, and transfer learning. Expert Systems with Applications **117**, 125–138 (2019)
8. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
9. Lei, K., Zhang, B., Li, Y., Yang, M., Shen, Y.: Time-driven feature-aware jointly deep reinforcement learning for financial signal representation and algorithmic trading. Expert Systems with Applications **140**, 112872 (2020)
10. Schäfer, R., Guhr, T.: Local normalization: Uncovering correlations in non-stationary financial time series. Physica A: Statistical Mechanics and its Applications **389**(18), 3856–3865 (2010)
11. Sezer, O.B., Ozbayoglu, A.M.: Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach. Applied Soft Computing **70**, 525–538 (2018)
12. Shi, Y., Zheng, Y., Guo, K., Ren, X.: Stock movement prediction with sentiment analysis based on deep learning networks. Concurrency and Computation: Practice and Experience **33**(6), e6076 (2021)
13. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research **15**(1), 1929–1958 (2014)
14. Tran, D.T., Magris, M., Kanniainen, J., Gabbouj, M., Iosifidis, A.: Tensor representation in high-frequency financial data for price change prediction. In: Proceedings of the IEEE Symposium Series on Computational Intelligence. pp. 1–7
15. Tsantekidis, A., Passalis, N., Tefas, A., Kanniainen, J., Gabbouj, M., Iosifidis, A.: Using deep learning to detect price change indications in financial markets. In: Proceedings of the European Signal Processing Conference. pp. 2511–2515
16. Tsantekidis, A., Passalis, N., Tefas, A., Kanniainen, J., Gabbouj, M., Iosifidis, A.: Forecasting stock prices from the limit order book using convolutional neural networks. In: Proceedings of the IEEE Conference on Business Informatics (CBI). vol. 1, pp. 7–12 (2017)
17. Tsantekidis, A., Passalis, N., Tefas, A., Kanniainen, J., Gabbouj, M., Iosifidis, A.: Using deep learning for price prediction by exploiting stationary limit order book features. Applied Soft Computing **93**, 106401 (2020)
18. Tsantekidis, A., Passalis, N., Toufa, A.S., Saitas-Zarkias, K., Chairistanidis, S., Tefas, A.: Price trailing for financial trading using deep reinforcement learning. IEEE Transactions on Neural Networks and Learning Systems (2020)
19. Zhang, W., Skiena, S.: Trading strategies to exploit blog and news sentiment. In: Proceedings of the International AAAI Conference on Web and Social Media. vol. 4 (2010)

20. Zhang, Z., Zohren, S., Roberts, S.: Deeplob: Deep convolutional neural networks for limit order books. IEEE Transactions on Signal Processing **67**(11), 3001–3012 (2019)