SENTIMENT-AWARE DISTILLATION FOR BITCOIN TREND FORECASTING UNDER PARTIAL OBSERVABILITY

Georgios Panagiotatos, Nikolaos Passalis, Avraam Tsantekidis, and Anastasios Tefas

Computational Intelligence and Deep Learning Group
Artificial Intelligence and Information Analysis Lab., Department of Informatics,
Aristotle University of Thessaloniki, Thessaloniki, Greece
$\{$*panagiotga, passalis, avraamt, tefas*$\}$*@csd.auth.gr*

ABSTRACT

Deep Learning (DL) models are increasingly used for financial forecasting problems, such as price or trend prediction of a financial asset. However, most methods either rely solely on price information or require difficult to implement data harvesting pipelines, e.g., from social media, to deploy them. The main contribution of this paper is a method that exploits sentiment information as a source of additional supervision during the training process, allowing for improving the profitability of the developed strategies compared to baseline agents, while also allowing for operating the agent under partial observability, i.e., without requiring sentiment information as input during inference. As demonstrated in the conducted experiments on the Bitcoin-USD currency pair, this approach can indeed lead to significant improvements in the performance of DL agents, as well as help reduce the overfitting phenomena that often occur when training such agents.

***Index Terms*—** Financial Trading, Knowledge Distillation, Deep Learning, Bitcoin Forecasting

1. INTRODUCTION

Recent advances in Deep Learning (DL) fueled the research interest in developing methods for numerous financial applications, including financial trading [1]. Indeed, the vast amount of data generated from financial markets, along with the ability of DL models to efficiently ingest this information led to methodologies that superseded traditionally used approaches [2, 3, 4], such as hand-crafted trading strategies. These methods typically work by either modelling the financial environment and formulating trading as a Reinforcement Learning (RL) problem [1, 3, 5], or following a forecasting-based approach, where the DL models attempt to predict the future price movements [6, 7]. The signals arising from these models can be then exploited in heterogeneous strategies to perform profitable trades, even when operating under the (often) volatile and noisy financial environment.

Despite the success of these approaches, most of them rely solely on price-related information, ranging from coarser information, e.g., minute/hourly/daily Open-Low-High-Close (OLHC) candles, to finer one, e.g., limit order book information, such as volume and price of all requested limit and market orders. However, this is in contrast with the way that most human traders actually take decisions about selling or buying an asset. Human traders usually take into account many other factors (information sources) that expand well beyond the price/order book information that can span from sentiment and general news to prior knowledge and forecasts. Indeed, *sentiment* expressed in various environments, e.g., news and social media, regarding various assets, as well as regarding various sectors of the economy, can significantly affect the price of various financial assets [8, 9, 10].

Recent works have demonstrated that approaches that rely on sentiment-related modalities, typically harvested from online sources, can achieve very promising results, often outperforming approaches that rely solely on price information [11, 12, 13]. However, this comes with the additional cost of developing the necessary data collection pipelines. This can be especially challenging, since this information is usually collected from various online sources and social media and requires further pre-processing to extract useful information that can be exploited by the DL models. As a result, this often hinders the integration of such approaches in production systems, due to the high cost of developing, setting up and ensuring the correct operation of such pipelines.

The main research question posed in this paper is whether we can exploit the sentiment information that has already been collected to improve DL models for financial trading, while - at the same time - operating these models without

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).

access to such information. In other words, we aim to examine the ability to operate such DL models *under partial observability* (i.e., by observing only price information) of the environment during inference, yet train the models under full observability (i.e., by observing both sentiment and price information). Such an approach would allow for significantly reducing the cost of integrating and operating DL models, since the already collected sentiment information (e.g., from social media datasets) can be used during the training process with little to no cost. At the same time, no change in the deployment pipeline is needed, since the models still operate solely on the price modality.

The main contribution of this paper is a method that exploits sentiment information as a source of additional supervision during the training process. To this end, we employ a *neural network distillation* [14]-based method for transferring the knowledge encoded by an agent trained to perform financial trading, while observing sentiment-related information, to an agent that performs trading based on price input. As demonstrated in the conducted experiments, this approach can indeed lead to significant improvements in the performance of DL agents, as well as help reduce the overfitting phenomena that often occur when training such agents. It is worth noting that, to the best of our knowledge, this is the first methodology that can effectively exploit sentiment information for improving the performing of DL-based cryptocurrency trading agents, while allowing for operating such agents without using sentiment information during inference.

The rest of the paper is structured as follows. Section 2 introduces and presents the proposed method. Then, in Section 3 we present the experimental setup and results. Finally, Section 4 concludes this paper.

2. PROPOSED METHOD

The proposed method is introduced in this Section. First, we briefly provide the background on training DL agents for detecting price change indicators using either price or sentiment input. Then, we present the proposed method that allows for training improved DL agents for trading using sentiment-aware knowledge distillation.

2.1. Background

Let $f_{\mathbf{W}}(\mathbf{x})$ denote a DL model that can be used to detect financial price change indicators, where $\mathbf{W}$ are the trainable parameters of the model and $\mathbf{x}$ is the input provided to the model. In this work we consider the input to be either features extracted from the price time series, denoted by $\mathbf{x}^{(p)}$, or a sentiment time series, denoted by $\mathbf{x}^{(s)}$. More specifically, for models operating on the price time series the input to the model consists of L more recent percentage changes, i.e.,

$$\mathbf{x}_t^{(p)} = \left[\frac{p_{t-L+1}}{p_{t-L+2}} - 1, \dots, \frac{p_t}{p_{t-1}} - 1\right]^T \in \mathbb{R}^L, \quad (1)$$

where p_t denotes the close price of an asset at time t. The training target for the model is defined as:

$$l_t = \begin{cases} 1, & \text{if } \frac{p_{t+1}}{p_t} - 1 > \delta \\ -1, & \text{if } \frac{p_{t+1}}{p_t} - 1 < \delta \\ 0, & \text{otherwise} \end{cases}, \quad (2)$$

where δ denotes the threshold for considering a price movement significantly. Usually, δ is set according to the trading commission and accounts for possible price slippage that might occur during order execution. Setting a higher value for δ leads to models that perform less trades, while a smaller value of δ provides more trading signals. Typical values range around $\delta = 10^{-3}$.

A DL model can be also trained to provide trading signals using as input a sentiment time series. To this end, a sentiment mining system can be employed, e.g., collecting tweets about a currency and then averaging their sentiment polarity, as further explained in Section 3. More advanced aggregation methods, such as using the Bag-of-Features models [15], can be also used to this end, since the actual methodology for extracting the sentiment time series does not affect the proposed method. Therefore, a sentiment time series $\mathbf{x}_t^{(s)}$ can be extracted for each time step:

$$\mathbf{x}_t^{(s)} = [s_{t-L+1}, \dots, s_t]^T \in \mathbb{R}^L, \quad (3)$$

where s_t provides the sentiment at the time step t. Then, the DL model can be trained using the same targets as those provided in (2).

2.2. Proposed Methodology

The main contribution of this paper is to use a multi-modal distillation approach to transfer the knowledge from models that operate on different modalities. Let $f^{(s)}_{i,\mathbf{W}^{(s)}}(\mathbf{x}_t^{(s)})$ denote the i-th agent (of an ensemble) trained on sentiment input, while $f^{(p)}_{i,\mathbf{W}^{(p)}}(\mathbf{x}_t^{(p)})$ denote the i-th agent trained on price input. First, instead of transferring the knowledge from a single agent, we propose employing ensembles of agents. This approach has been proven especially beneficial in noisy domains, such as financial trading [16]. Therefore, we denote the output of the sentiment ensemble for a given input $\mathbf{x}_t^{(s)}$ as:

$$\mathbf{d}_t^{(s)} = \frac{1}{N} \sum_{i=1}^{N} f^{(s)}_{i,\mathbf{W}^{(s)}}(\mathbf{x}_t^{(s)}), \quad (4)$$

where N is the number of models used for creating the teacher ensemble. Similarly, the output of price ensemble for a given input $\mathbf{x}_t^{(p)}$ as:

$$\mathbf{d}_t^{(p)} = \frac{1}{N} \sum_{i=1}^{N} f^{(p)}_{i,\mathbf{W}^{(p)}}(\mathbf{x}_t^{(p)}). \quad (5)$$

Neural network distillation [14] allows for transferring the knowledge encoded in large ensembles of teacher models in a smaller model, called student. Let $f_{\mathbf{W}}(\mathbf{x})$ denote the student model, which always operates on price input. Then, we can define a price distillation loss as:

$$\mathcal{L}_{p,t} = -\sum_{j=1}^{3} [\mathbf{d}_t^{(p)}]_j \log[f_{\mathbf{W}}(\mathbf{x}_t^{(p)})]_j, \tag{6}$$

where the notation $[\mathbf{x}]_i$ is used to denote the i-th element of a vector $\mathbf{x}$. This loss expresses the objective of transferring the knowledge from teachers that operate on the same modality (price) into the single student model. Also, note that the cross entropy loss spans over three classes, since the models have three outputs according to (2). The temperature of the softmax function, which is part of the model, can be also raised to provide a smoother output distribution, as proposed in [14]. However, recent works have shown that in many tasks this have only a negligible effect on the effectiveness of distillation [17].

Similarly, we can define the sentiment distillation loss as:

$$\mathcal{L}_{s,t} = -\sum_{j=1}^{3} [\mathbf{d}_s^{(p)}]_j \log[f_{\mathbf{W}}(\mathbf{x}_t^{(p)})]_j. \tag{7}$$

This loss expresses the objective of transferring the knowledge from teachers that operate on a different modality (sentiment) into a student model that operates on price input. Note that the inputs that are used for distillation still have to be temporally aligned and synchronized by performing the appropriate sub-sampling. Therefore, the same time step should be used for both the student and teacher models when performing distillation.

As previously described, in this paper we aim to transfer the knowledge from models operating on both modalities. Therefore, the final loss used to train the student model is defined as:

$$\mathcal{L} = \sum_{t=1}^{M} \mathcal{L}_{gt,t} + \mathcal{L}_{p,t} + \mathcal{L}_{s,t}, \tag{8}$$

where M is the total number of training time steps, $\mathcal{L}_{gt,t}$ is the cross-entropy loss arising from the ground truth labels defined as:

$$\mathcal{L}_{gt,t} = -\sum_{j=1}^{3} [\mathbf{l}_t]_j \log[f_{\mathbf{W}}(\mathbf{x}_t^{(p)})]_j, \tag{9}$$

and $\mathbf{l}_t$ is the one-hot encoding of l_t, as defined in (2). Then, the model can be trained using gradient descent to minimize this loss:

$$\Delta \mathbf{W} = -\eta \frac{\partial \mathcal{L}}{\mathbf{W}}, \tag{10}$$

where η is the used learning rate. More advanced optimization methods, such as Adam [18], can be also employed.

The proposed method provides several benefits over existing approaches. First, it allows for reducing the inference complexity by improving the accuracy of the student agent without having to query the ensemble during the deployment. Note that the teacher ensemble is used only during the training process. As a result, any additional cost arising from it will only occur once. Furthermore, it enables the student agent to operate under partial observability of the environment, since no access to sentiment information is required during inference. Despite this, it can provide improved performance compared to similar agents that operate on the same input, as demonstrated in Section 3. However, the quality of the collected sentiment can affect the training process. Finally, the distillation process allows for better shielding the models against overfitting phenomena, which is an especially important issue in the financial domain, as demonstrated both in the literature [3], as well as in the conducted experiments.

3. EXPERIMENTAL EVALUATION

In this Section we provide the experimental evaluation of the proposed method. First, we introduce the used datasets and evaluation metrics. Then, we present the employed network architectures and evaluation setups. Finally, we present and discuss the experimental results.

3.1. Datasets and Evaluation Setup

For all conducted experiments, we used the Bitcoin-USD currency pair. This choice was mainly motivated by the availability of sentiment information regarding cryptocurancies in various online sources, as well as the uninterrupted operation of such markets. The data collection period ranges from 2015 to 2020. We collected OLHC day candles, while the price time series was compiled using the close price of the collected data. The first four years were employed for training the DL models. The testing dataset consisted of the last year (2020) and was used for performing the backtesting of the models.

We also used a dataset published by BDC Consulting [19], which contains over 200,000 titles of financial articles that have been collected from various online sites, such as Cointelegraph and CoinDesk. We used this dataset for extracting a sentiment time series to estimate the polarity of opinions regarding cryptocurrencies in online sources. To extract the sentiment time series we employed a pre-trained FinBERT model [20]. More specifically, let $\mathcal{X}_t$ denote a collection of the titles of textual documents that were collected at time t, i.e., after time-step $t-1$ and until time-step t, and refer to the asset at hand. Then, we can use the pre-trained FinBERT model to extract a sentiment value for each document in $\mathcal{X}_t$, following the process described in [13]. The sentiment at time t (s_t) is calculated as the average sentiment extracted for all the collected documents in $\mathcal{X}_t$ and ranges between -1 (negative sentiment) and 1 (positive sentiment).

The trading signals of the model were used to compile a trading policy by translating the up/down class into a long/short action. If the predicted price movement is within the threshold ($\delta = 10^{-3}$), then the trading agent exits the market. All models were evaluated based on the obtained Profit and Loss (PnL) metric, as calculated when back-testing the model on the test set. A fixed lot size is used for calculating the PnL.

Following recent literature on financial forecasting [3, 6], we employed a Long Short Term Memory (LSTM)-based architecture [21] for models that receive price input. For models that process sentiment input we opted for a simpler, yet equally effective according to our experiments, MLP-based architecture. The LSTM model consists of one LSTM layer with 8 hidden neurons, a fully connected layer with 20 neurons and a final 3-neuron output layer. We feed the 12 most recent observations to the LSTM model. For the MLP model we used a hidden layer of 10 neurons followed by the decision layer that is composed of 3 neurons. For all fully connected layers we employed the ReLU activation function [22]. All models were trained using the Adam optimizer [18] using a learning rate of $3 \cdot 10^{-3}$ and weight decay set to 10^{-6}. All models were trained for 100 training epochs. The teacher ensemble consists of 5 teachers (each one starting from a different initialization), while for all results we report the mean of 10 repeated runs. Using more teachers is not expected to significantly improve the performance of the resulting student model. The input data were normalized using z-score normalization (standarization) using the statistics of the training set.

3.2. Experimental Results

The experimental results are presented in Table 1, where the PnL (as a ratio over the initial amount) is reported. Apart from the proposed method, four other baseline and competitive approaches have been included. First, the results using two baseline agents ("Price - Baseline" and "Sentiment - Baseline"), that have trained on the price and sentiment modality, respectively, are reported. These agents have been trained without employing any distillation approach. Then, we also report cross-modality distillation results, where the knowledge has been transferred from teacher ensembles that operate on a different modality, i.e., "Sentiment Cross-Distillation" and "Price - Cross Distillation". Note that both the "Sentiment Baseline" and the "Sentiment - Cross Distillation" baseline require as input the sentiment features that can be expensive to acquire in practice, as previously discussed. Nonetheless, they have been included as baselines to better highlight the improvements that can be acquired using the proposed method.

Several interesting conclusions can be drawn from the results reported in Table 1. First, note that both baseline agents are prone to overfitting, since both of them achieve negative PnL when the training continues after the first 50 training epochs. Also, using distillation improves the trading performance for both the price and sentiment agents. It is actually worth noting that the sentiment agent actually achieves better performance compared to the price agent, an observation that has been also reported in previous works [13], confirming that sentiment can be a very strong predictor of the future price movements of cryptocurriences, such as Bitcoin. At the same time, the agents trained using any distillation-based approach are significantly more resilient to overfitting phenomena compared to the baseline agents. Finally, the proposed method, which combines two different knowledge sources for distillation, achieves the overall best results, outperforming the sentiment agent, despite not having access to sentiment information during the inference process. Also, the proposed method is significantly more robust to overfitting, since the PnL drop is significantly lower, as the training progresses, compared to the rest of the evaluated approaches.

Table 1. Experimental Evaluation: PnL during back testing for different training epochs and methods. The first part of the name of each method denotes the modality on which the corresponding agent operates.

Method / Epoch	**50**	**70**	**100**
Price - Baseline	0.198	-0.233	-0.398
Sentiment - Baseline	0.148	-0.348	-0.235
Price - Cross Distillation	0.946	0.483	0.048
Sentiment - Cross Distillation	1.246	1.002	0.923
Price - Proposed	**1.334**	**1.354**	**1.155**

4. CONCLUSIONS

In this paper we presented a method that exploits sentiment information as a source of additional supervision during the training process. In this way, it allows for improving the profitability of the developed strategies compared to other agents that operate on unimodal input, while also allowing for operating the agent under partial observability, i.e., without requiring sentiment information as input during the inference. As a result, the proposed method can allow for significantly reducing the cost of integrating and operating DL models, since no change in the deployment pipeline is needed, given that models still operate solely on the price modality. As demonstrated in the conducted experiments, this approach can indeed lead to significant improvements in the performance of DL agents, as well as help reduce the overfitting phenomena that often occur when training such agents. The proposed method paves the way for developing more advanced distillation methods that can also take into account more modalities, e.g., news articles, forecasts, etc., potentially further improving the performance of the developed agents.

5. REFERENCES

[1] Yue Deng, Feng Bao, Youyong Kong, Zhiquan Ren, and Qionghai Dai, "Deep direct reinforcement learning for financial signal representation and trading," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 28, no. 3, pp. 653–664, 2016.

[2] Dat Thanh Tran, Martin Magris, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis, "Tensor representation in high-frequency financial data for price change prediction," in *Proc. IEEE Symposium Series on Computational Intelligence*, pp. 1–7.

[3] Avraam Tsantekidis, Nikolaos Passalis, Anastasia-Sotiria Toufa, Konstantinos Saitas-Zarkias, Stergios Chairistanidis, and Anastasios Tefas, "Price trailing for financial trading using deep reinforcement learning," *IEEE Trans. on Neural Networks and Learning Systems*, 2020.

[4] Zihao Zhang, Stefan Zohren, and Stephen Roberts, "Deeplob: Deep convolutional neural networks for limit order books," *IEEE Trans. on Signal Process.*, vol. 67, no. 11, pp. 3001–3012, 2019.

[5] Kai Lei, Bing Zhang, Yu Li, Min Yang, and Ying Shen, "Time-driven feature-aware jointly deep reinforcement learning for financial signal representation and algorithmic trading," *Expert Systems with Applications*, vol. 140, pp. 112872, 2020.

[6] Avraam Tsantekidis, Nikolaos Passalis, Anastasios Tefas, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis, "Forecasting stock prices from the limit order book using convolutional neural networks," in *Proc. IEEE Conf. on Business Informatics*, 2017, vol. 1, pp. 7–12.

[7] Mingxi Liu, Guowen Li, Jianping Li, Xiaoqian Zhu, and Yinhong Yao, "Forecasting the price of bitcoin using deep learning," *Finance Research Letters*, vol. 40, pp. 101755, 2021.

[8] Feng Mai, Zhe Shan, Qing Bai, Xin Wang, and Roger HL Chiang, "How does social media impact bitcoin value? a test of the silent majority hypothesis," *Journal of Management Information Systems*, vol. 35, no. 1, pp. 19–52, 2018.

[9] Andrew Burnie and Emine Yilmaz, "Social media and bitcoin metrics: which words matter," *Royal Society open Science*, vol. 6, no. 10, pp. 191068, 2019.

[10] Dominique Guégan and Thomas Renault, "Does investor sentiment on social media provide robust information for bitcoin returns predictability?," *Finance Research Letters*, vol. 38, pp. 101494, 2021.

[11] Yong Shi, Yuanchun Zheng, Kun Guo, and Xinyue Ren, "Stock movement prediction with sentiment analysis based on deep learning networks," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 6, pp. e6076, 2021.

[12] Wenbin Zhang and Steven Skiena, "Trading strategies to exploit blog and news sentiment," in *Proc. Intl. AAAI Conf. on Web and Social Media*, 2010, vol. 4.

[13] N Passalis, S Seficha, A Tsantekidis, and A Tefas, "Learning sentiment-aware trading strategies for bitcoin leveraging deep learning-based financial news analysis," in *Proc. Intl. Conf. on Artificial Intelligence Applications and Innovations*, 2021, pp. 757–766.

[14] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean, "Distilling the knowledge in a neural network," *arXiv preprint 1503.02531*, 2015.

[15] Nikolaos Passalis, Anastasios Tefas, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis, "Temporal logistic neural bag-of-features for financial time series forecasting leveraging limit order book data," *Pattern Recognition Letters*, vol. 136, pp. 183–189, 2020.

[16] Avraam Tsantekidis, Nikolaos Passalis, and Anastasios Tefas, "Diversity-driven knowledge distillation for financial trading using deep reinforcement learning," *Neural Networks*, vol. 140, pp. 193–202, 2021.

[17] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker, "Learning efficient object detection models with knowledge distillation," *Proc. Advances in Neural Information Process. Systems*, vol. 30, 2017.

[18] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint 1412.6980*, 2014.

[19] "Analyzing Crypto Headlines - BDC Consulting," .

[20] Dogu Araci, "Finbert: Financial sentiment analysis with pre-trained language models," *arXiv preprint 1908.10063*, 2019.

[21] Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber, "Lstm: A search space odyssey," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, 2016.

[22] Vinod Nair and Geoffrey E Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. Intl. Conf. on Machine Learning*, 2010.