Playing with News Context for Algorithmic Trading

Anonymous ACL submission

Abstract

The application of reinforcement learning for algorithmic trading in the spot market using numerical data is a well-studied problem. However, news data consists of hard-to-quantify information which the investors use to base their trading decisions. Thus factoring in news data for algorithmic trading can improve the trading performance of the RL agent. This paper proposes an RL-based framework that performs algorithmic trading in the futures market by combining news data and price data. We propose two approaches for representing the context of the news data: sentiment-aware approach and context-aware approach. We investigate the effect of these approaches on the trading performance of the RL agent. We further compare the performance of on-policy and off-policy RL algorithms. The models are evaluated by trading in the NIFTY 50 index. The evaluation of the models show that using context-aware approach for representation of news data significantly improves the return (%) and also reduces the maximum drawdown of the trading model during a trading session.

1 Introduction

The stock market follows the efficient market hypothesis (Fama, 1970), which states that the stock value reflects all available information. This information is both numerical and non-numerical. The objective of algorithmic trading is to maximize the profits by learning to exploit the hidden signals from diverse datasources and open a long or short position before the information reflects in the stock price and exit the position once the stock price has reached its potential. The stock market index is highly temporal as the emergence of new information over time affects it. The algorithmic trading strategies need to operate in this temporal setting.

The current literature on algorithmic trading in the stock market uses a reinforcement learning (RL) framework to design the trading model. The agent aims to maximize the profit by learning a policy through exploration and exploitation by interacting with the trading environment. Using price data to represent the state is a well-studied problem, wherein price data comprises OHLCV and technical indicator values (Jeong and Kim, 2019; Lei et al., 2020; Yang et al., 2020; Hirchoua et al., 2021; Théate and Ernst, 2021; Taghian et al., 2022; Yang et al., 2023). Recent works have also explored the use of non-numeric data in the form of news data and have used a combination of news data and price data to represent the state of the market (Koratamaddi et al., 2021; Chen and Huang, 2021), where in the news data is represented using the news sentiment.

Due to lack of a benchmark dataset for evaluating the trading models, no comparison is possible between the existing works as each work chooses a different set of individual stocks and different stock markets. In some cases, the authors have used spot trading to trade directly in an index (Jeong and Kim, 2019; Lei et al., 2020; Théate and Ernst, 2021; Hirchoua et al., 2021), whereas, as per market regulations, we can trade in an index only through futures trading. In most of the works, the RL agent trades only once a day before the market closes and uses the data of the previous day to determine the trading action which does not simulate the market conditions, while some works perform intraday trading in the share market (Chen and Huang, 2021).

In this paper, we propose an RL-based framework that combines news data and price data to perform futures trading [1]. We propose two approaches for representing the news data: 1. Sentiment-aware approach 2. Context-aware approach. The sentiment-aware approach uses news sentiment to represent the context in the news data. The context-aware approach uses text representation schemes to encode the context of the news articles. We perform

[1] urlhttps://zerodha.com/varsity/module/futures-trading/

trading in the NIFTY 50 index in a minute-wise time series setting where the agent can take multiple actions in a single day. We also compare the trading performance of different on-policy and off-policy based algorithms. Our proposed approach uses PPO as the RL algorithm and uses a feature extraction module to extract the features from the state. Our experiments show that factoring in the news data leads to improvement in the trading performance of the RL algorithm.

The summary of the contribution of our work are as follows:

- We propose an RL-framework that factors in the contextual information of news data and combines it with price data for performing high frequency trading (HFT) in the futures market.
- We perform extensive experiments to establish the effectiveness of using news data in improving the trading behaviour of the RL agent when performing HFT and also compare the performance of the RL agent when we use different approaches to represent the news data.
- We provide a comparison of the trading performance of the RL agent when using off-policy based and on-policy based RL algorithms.
- We release our dataset as a benchmark dataset to enable comparison of existing and future works on algorithmic trading. We also release our RL environment for simulating futures trading and all the codes required for running the experiments of this paper [2].

2 Related Work

The literature on the use of RL framework for algorithmic trading primarily consists of price data only approach and combination of news data and price data approach. In the price data only approach the state is represented using OHLCV values (Théate and Ernst, 2021; Hirchoua et al., 2021; Taghian et al., 2022), technical indicator values (Lei et al., 2020; Li et al., 2020; Wu et al., 2020; AbdelKawy et al., 2021; Yang et al., 2020) and difference between the close prices (Jeong and Kim, 2019). In the combination of news data and price data approach the state is represented using news data and price data wherein the news data is represented using the news sentiment (Koratamaddi et al., 2021; Chen and Huang, 2021). In these works the authors use the VADER sentiment analyzer to get the sentiment of the news articles. Chen and Huang (2021) calculate the news influence at time step t which is sum of sentiments from $t-r$ to $t+r$ to represent the state. However this approach introduces a data leakage as the action of the agent at time step t should be based on only the events preceding time step t.

The RL algorithms used in the agent are divided into three approaches: off-policy based and on-policy based. The papers that use the off-policy based approach widely use DQN (Jeong and Kim, 2019; Li et al., 2020; Wu et al., 2020; Théate and Ernst, 2021; AbdelKawy et al., 2021; Taghian et al., 2022) and DDPG (Koratamaddi et al., 2021) as the RL algorithm. The papers that use the on-policy based approach use policy gradient (Lei et al., 2020; Wu et al., 2020; Chen and Huang, 2021), PPO (Hirchoua et al., 2021).

In some studies the agent uses a feature extraction module to extract features from the state instead of directly using the raw features of the state to determine a trading action. The feature extraction module in these studies use encoders such as GRU (Lei et al., 2020; Wu et al., 2020), MDRNN (Chen and Huang, 2021), CNN (Taghian et al., 2022), LSTM (AbdelKawy et al., 2021) to extract the features from the state. Taghian et al. (2022) show that the performance of an RL agent using a feature extraction module improves only when the test years have a similar price movement as the train years.

The reward function used in the literature calculate the reward of an action using the relative difference between the previous and current close price (Jeong and Kim, 2019; Théate and Ernst, 2021; Taghian et al., 2022), absolute difference in close prices (Lei et al., 2020; Hirchoua et al., 2021; Chen and Huang, 2021), difference between the portfolio values (AbdelKawy et al., 2021; Koratamaddi et al., 2021). The actions of the agent are generally defined as discrete actions such as buy, sell or hold, long or short. Further, the authors define the number of shares associated with the action of an agent. The evaluation of the trading models are performed using total profit, return (%), Sharpe ratio, Sortino ratio, VaR, volatility, maximum drawdown.

[2] https://anonymous.4open.science/r/futures_trading-8BE4/

3 Proposed Approach

Our proposed approach is an RL framework that performs futures trading in a minute-wise time setting. In this approach we combine news data and price data to represent the state. The environment simulates futures trading using Algorithm 1, wherein it executes the action taken by the agent. The agent can open and close positions within the same day or carry forward a position to the next day. In this work, we consider all the contracts as near-month contracts, i.e., the contract will expire on the last Thursday of every month. Thus, we break the sequence of agent-environment interactions into episodes wherein an episode ends on the last Thursday of every month when the market closes. When an episode ends, the open positions of the agent are closed. We describe the components of the RL framework in further detail in sections 3.1, 3.2, 3.3, 3.4.

3.1 State (s_t)

We use price data (P) and news data (T) from $t-w$ to t ticks to represent the state (s_t), where $w \in \mathbb{Z}+$ indicates the window size. Technical indicators capture the trends from historical prices and indicate the market condition. We use the the technical indicators values[3]: ADX, MACD, MOM, ATR, RSI, Slow %K, Williams %R, Bollinger Bands (BBAND), and EMA to represent the price data at each tick i ($i \in [t-w, t]$), by forming a price vector ($price_i$) which comprises of the technical indicator values at tick i. We use these price vectors in sentiment-aware approach and context-aware approach.

3.1.1 Sentiment-aware approach

The sentiment-aware approach uses the sentiment of the news data to represent the market sentiment. We use FinBERT (Araci, 2019) to analyze the sentiment of a news article based on the title of the news article. The probability score quantifies the extent to which a news article is positive, negative, or neutral. We select the label with the highest probability score and use the probability score to represent the news article. To represent the news data at tick i, we form the news vector ($news_i$), which consists of the total news sentiments from $i-x$ to i ticks, where x ($x \in \mathbb{Z}^+$) is the number of ticks preceding i.

[3] https://www.fidelity.com/learning-center/trading-investing/technical-analysis/technical-indicator-guide/overview

$$total\_sent_i = \frac{\sum_{j \in t-x}^{t} p_j^{pos} - \sum_{j \in t-x}^{t} p_j^{neg}}{n_+ + n_- + n_o} \quad (1)$$

We then calculate the total news sentiment ($total\_sent_i$) at tick i using Equation 1, which is similar to that used in Allen et al. (2019), where p_j^{pos} and p_j^{neg} denotes the probability of a news article having positive and negative sentiment, respectively, n_+, n_-, n_o denotes the number of positive, negative and neutral news articles available between $t-x$ to t ticks.

$$u_i = price_i \bigoplus news_i \quad (2)$$

At each tick i (where $i \in [t-w, t]$), we concatenate $price_i$ and $news_i$ to form a combined vector u_i as shown in Equation 2, which adds the news data to the price data. The state s_t in sentiment-aware approach is thus a sequence of vectors $[u_{t-w}, \ldots, u_t]$, which represents the price data and news data from ticks $t-w$ to t.

3.1.2 Context-aware approach

The context-aware approach represents the hard-to-quantify contextual information of the news articles. At each time step t, we select k latest news article titles published between $t-w'$ to t time step wherein w' is the window size. Thus the news data consists of a sequence of news article titles $[news_1, news_2, \ldots, news_k]$. We use different LLM-based text representation schemes to represent the context of a news article title $news_j$ ($j \in [1, k]$). We represent $news_j$ using the token representation (v_j) of the last token in sequence of tokens. Thus the news data is represented as sequence of vectors $[v_1, \ldots, v_k]$. The state s_t in context-aware approach is thus sequence of news vectors $[v_1, \ldots, v_k]$ and sequence of price vectors $[p_{t-w}, \ldots, p_t]$.

3.2 Agent

The agent uses PPO (Schulman et al., 2017) as the deep RL algorithm, which uses a feature extraction module (FEM) to extract features from the state s_t to form a feature vector (f_t). PPO predicts the next action using f_t. The value and policy network of PPO shares the parameters of FEM. The value and policy network consists of three fully connected neural layers and uses f_t as input. The last layer of the value network gives the value function, while the last layer of the policy network gives the action value. The feature extraction module used in sentiment-aware approach and context-aware approach are discussed in the subsequent subsections.

3.2.1 Sentiment-aware approach

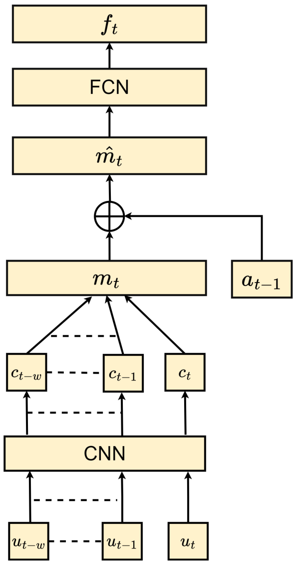

Figure 1: Architecture of FEM in sentiment-aware approach

The architecture of FEM in sentiment-aware approach is shown in Figure 1. In the FEM, the vectors in s_t are passed through a 1D CNN layer to get the context vectors $[c_{t-w}, \ldots, c_t]$. The context vectors capture the contextual relationship between the vectors in s_t. It then takes a sum over the context vectors to get the relation vector (m_t). The relation vector encodes the contextual information captured in the context vectors. The relation vector (m_t) is then concatenated with the previous action (a_{t-1}) of the agent to get the vector $(\hat{m}_t)$. The vector $\hat{m}_t$ is then passed through a fully connected neural (FCN) layer to obtain f_t. We term this model as PPO_FEM_PT_Senti.

3.2.2 Context-aware approach

The architecture of FEM in context-aware approach is shown in Figure 2. The news vectors $[v_1, \ldots, v_k]$ in s_t are passed through a 1D CNN layer to get the context vectors $[c_1, \ldots, c_k]$. The context vectors capture the local relationship between the events mentioned in the news articles. It then takes a sum over the context vectors and passes the vector through two fully connected neural layers to form the news sequence vector n_v. It then applies the sigmoid function over n_v to get the news context value n_{cv} which quantifies the context of the sequence of news articles.

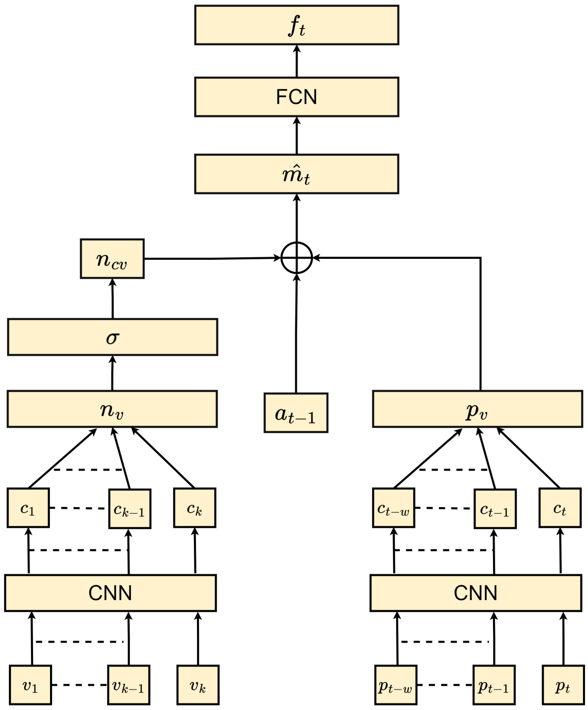

Figure 2: Architecture of FEM in context-aware approach

The price vectors $[p_{t-w}, \ldots, p_t]$ are passed through a 1D CNN layer to get the context vectors $[c_{t-w}, \ldots, c_t]$. It then takes a sum over the context vectors to get the price sequence vector p_v which encodes the context of the prices. It then concatenates n_{cv}, p_v and a_{t-1} to form the vector $\hat{m}_t$, which is then passed through a single fully connected neural network to obtain f_t. We term this model as PPO_FEM_PT_Context.

3.3 Action (a_t)

The action (a_t) denotes the number of lots that the agent can buy, sell or hold. In order to avoid the curse of dimensionality due to using discrete actions (Lillicrap et al., 2015) and to ensure that the agent can be scaled to trade in higher number of lots, we define a continuous action space $(\mathcal{A})$ which lies in the range $[-1, +1]$. Algorithm 1 needs a discrete value in num_lots. So we use Equation 3 to get the num_lots, where max_num_lots indicates the maximum number of lots that the agent can trade.

$$num\_lots = \lfloor max\_num\_lots \times a_t \rfloor \quad (3)$$

3.4 Reward Function

The reward function considers two aspects: 1. The goodness of an action w.r.t. the change in close price from tick t to $t+1$. 2. The effect of an action on the balance of the agent from tick t to $t+1$. The reward function is shown in Equation 4 wherein $balance_t$ and c_t denote the balance of the agent and the close price at tick t respectively. λ $(0 < \lambda < 1)$ assigns some weightage to both parts of the equation.

$$r_t = \lambda \times (num\_lots \times (c_{t+1} - c_t)) + (1 - \lambda) \times (balance_{t+1} - balance_t) \tag{4}$$

4 Experiments

4.1 Dataset

We use tick data (OHLC values) of NIFTY 50 [4] from 2010-2021 as the source of price data. The tick data consists of date, time and OHLC values. We select the minute data from 9:15 hrs to 15:15 hrs and calculate the technical indicators values from the OHLC values. We also add indicators of contract expiry to the price data. Further, we perform z-normalization over the technical indicator values. We news articles scraped from the Economic Times [5] as the source of our news data. To remove unwanted noise from the data, we use a proprietary classifier to select only financial news articles and select news article published between 8:15 hrs to 15:15 hrs. The news data consists of unique hash id, publication data and time and the news title. The data from 2010-2016 is the training data and data from 2017-2021 is the test data. The statistics of the dataset is shown in Table 1.

	Price data	News data
Training data	624647	81400
Test data	444769	114518

Table 1: Statistic of size of price data and news data in training data and test data

4.2 Evaluation Metrics

1. Return (%): Return (%) is the percentage relative difference between the trading balance at the start of the trading session and end of the trading session.

2. Maximum Drawdown (MDD): MDD is the maximum loss incurred by the trading model between the highest peak and the lowest trough that follows it before a new peak is achieved. The duration of the MDD is the number of days between the two peaks, thus indicating the time for which the model will face a loss. We use equation 5 to calculate the MDD wherein the L is the return at the lowest trough and P is the return at the highest peak.

$$MDD = \frac{L - P}{P} \times 100 \tag{5}$$

3. Volatility: Volatility is the risk associated with investment. Volatility is calculated using equation 6, wherein σ is the std. deviation in daily return and T is the number of days in the trading session.

$$\text{Volatility} = \sigma\sqrt{T} \tag{6}$$

4.3 Baselines

4.3.1 Price-only approach

1. DQN_P: The agent uses DQN (Mnih et al., 2015) as the RL algorithm. The state is represented using technical indicator values at time t and a_{t-1}. The action space consists of discrete values which indicates the number of lots to buy, sell or hold. The agent uses the raw features of the state to determine the action.

2. DQN_FEM_P: We use the technical indicator values from tick $t - w$ to t to represent the state s_t. The FEM has the same architecture as used in PPO_FEM_PT_Senti. The model uses the same state and action space used in DQN_P.

3. PPO_P: The agent uses PPO as the RL algorithm. The state is the same as that used in DQN_P. The agent uses the raw features of the state to determine the action.

4. PPO_FEM_P: It uses the same state space used in DQN_FEM_P. The FEM has the same architecture as used in PPO_FEM_PT_Senti.

4.3.2 Sentiment-aware approach

1. PPO_PT_Senti: We use the technical indicator values and news sentiments at time step t and the previous action taken by the agent to represent s_t. The agent uses the raw features to determine the action.

[4] https://www.kaggle.com/datasets/nishanthsalian/indian-stock-index-1minute-data-2008-2020

[5] https://economictimes.indiatimes.com/archive.cms

2. Variants of PPO_FEM_PT_Senti: We use trading models that use only a single sentiment (positive (Pos), negative (Neg)) or combination of two news sentiments (positive and negative (Pos_Neg), negative and neutral (Neg_Neu), positive and neutral (Pos_Neu)) to represent the news data.

4.4 Experimental Settings

Year	2017	2018	2019	2020	2021
Initial Balance	615757.5	2369632.5	2448382.5	2745483.75	3149122.5

Table 2: Initial balance at start of each test year (NIFTY 50)

We perform all our experiments on NVIDIA RTX 2080Ti, and for inferenceing Llama 3 8B and Gemma 7B we use NVIDIA RTX 4090 Ti. The configuration of the feature extraction modules, Q network and target network of DQN, policy network and value network of PPO are shared in Appendix C and the hyperparameters are shared in Appendix D. In the context-aware approach, for the text representation schemes we use Gemma 2B, Gemma 7B (Team et al., 2024), Llama 2 7B (Touvron et al., 2023), Mistral 7B (Jiang et al., 2023), and Llama 3 8B [6]. Since we are running our experiments in GPU resource poor environment, we use AWQ (Lin et al., 2023) versions of Llama 2 7B and Mistral 7B and use bitsandbytes (Dettmers et al., 2022) for 4 bit quantization of Gemma 2B, Gemma 7B and Llama 3 8B.

The max_num_lots is set to 3, so the num_lots_held of the agent will always be between -3 to 3, and the num_lots that the agent can buy or sell will lie between -3 to 3. The inital balance of the agent before starting the trade in a year is the product of max_num_lots, close price of the first tick of the year and lot_size. The initial balance at the start of each test year for NIFTY 50 is shown in Tables 2. As per Indian stock market regulations the lot_size from 2010-2017 is 25 and lot_size from 2018-2021 is 75.

5 Results

The results of the price data only approach and sentiment-aware approach is shown in Table 3. In the price-only approach we observe that PPO_P has the highest avg. return (%) and lowest avg. MDD among the price data only approach models. DQN_FEM_P has the lowest return among all price data only models. In terms of avg. return (%) DQN_P only performs marginally better than DQN_FEM_P but has the highest avg. MDD. Further, the results show that adding a feature extraction module (FEM) degrades the performance of the trading models. We observe that the off-policy based trading models give much lower average returns than on-policy based trading models. In off-policy based approach, the agent uses rewards from trajectories of previous policies to update the current policy. While in on-policy based approach, the agent uses the rewards from the trajectory of the current policy to update the same policy. As the futures market is highly temporal, the on-policy based approach allows the agent to learn a stable and dynamic policy that can factor in this temporal nature.

In the sentiment-aware approach, the comparison of the trading performance of PPO_P and PPO_PT_Senti on the basis of avg. return (%) and avg. MDD shows that using news sentiment along with price data improves the return (%) as compared to using only price data while also reducing the duration of loss that the model will face. The performance of PPO_FEM_PT_Senti shows that using a feature extraction module is effective when we are extracting features from diverse datasources, which leads to further increase in return (%) and also reduces the MDD duration. The performance of PPO_FEM_PT_Pos and PPO_FEM_PT_Neg show that using only negative sentiment is more effective than using only positive sentiment. Thus negative news sentiment plays an important role in influencing the trading decisions of the model. Using combination of neutral annd positive or negative sentiment degrades the performance of the trading model. However, the performance of PPO_FEM_PT_Pos_Neg show that using only positive and negative is sufficient for ensuring higher returns. But the use of positive and negative sentiments can over emphasize the impact of the positive and negative news on the stock market. Thus using neutral news sentiment along with positive and negative news sentiments provides a balance of the importance of the positive and negative sentiments, which evident from the return (%) and MDD of PPO_FEM_PT_Senti.

The results of context-aware approach is shown in Table 4. In context-aware approach, we observe that using LLM-based text representation for repre-

[6] https://github.com/meta-llama/llama3

Price Data Only Approach					
Data	**Model**	**Avg. Return (%)**	**Avg. MDD (%)**	**Avg. MDD Duration (Days)**	**Avg. Volatility**
Price Data	DQN_P	2.50	28.27	218.80	1.13
	DQN_FEM_P	-0.68	28.85	116.20	**0.90**
	PPO_P	**25.75**	**26.81**	**47.6**	1.48
	PPO_FEM_P	6.89	32.63	159.20	1.15
Sentiment-aware Approach					
Data	**Model**	**Avg. Return (%)**	**Avg. MDD (%)**	**Avg. MDD Duration (Days)**	**Avg. Volatility**
Price Data + News Title Sentiments	PPO_PT_Senti	32.45	30.31	65.00	2.17
	PPO_FEM_PT_Senti	**52.82**	29.69	**41.60**	2.05
	PPO_FEM_PT_Pos	6.52	31.82	153.60	1.34
	PPO_FEM_PT_Neg	12.85	33.23	134.20	1.75
	PPO_FEM_PT_Pos_Neg	42.12	**27.76**	104.80	2.14
	PPO_FEM_PT_Neg_Neu	-32.77	59.12	226.80	2.60
	PPO_FEM_PT_Pos_Neu	15.11	31.30	135.40	**1.28**

Table 3: The performance of price data only and sentiment-aware approaches in terms of average return (%), average MDD (%), average MDD duration (days), and average volatility

senting the news title leads to a significant improvement in the return (%) and also reduces the MDD. Further, the results show that FEM can exploit the relationship between the news events and quantify the context of the news data using the sigmoid function. Using Llama 2 7B for representing the news titles and combining it with news data yields the highest return (%). We also observe that using Gemma 2B gives a similar performance as Llama 2 7B in terms of return (%) and MDD, which shows the effectiveness of using smaller LLM models for trading. However, Gemma 7B has a much lower performance compared to Gemma 2B. We also observe that Mistral 7B has a lower return (%) than Llama 2 7B, however the MDD (%) and duration is lowest among all the models. The use of quantized version of Llama 3 8B adversely affects the performance which is evident from the lowest return (%) and highest MDD. This is consistent with the observation that quantization of Llama 3 8B affects its performance (Huang et al., 2024). Overall, for all the three approaches we observe that the volatility the model increases when the return (%) increases, as the model needs to take higher risks to ensure higher returns which is also mentioned in the efficient market hypothesis (Fama, 1970). Additional results on the year-wise performance of models in price data only, sentiment-aware approach and context-aware approach are added in Appendix E

In Figure 3, we plot the balance during contract expiry for each month in the year 2020 for models that use sentiment-aware approach. We observe that PPO_FEM_PT_Pos_Neg has a sharp rise and fall in the entire trading session, while

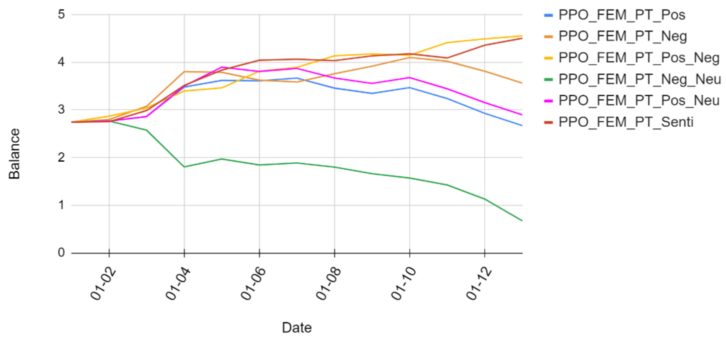

Figure 3: Movement of balance in the test year 2020 of models in sentiment-aware approach. Balance is scaled to $1e6$.

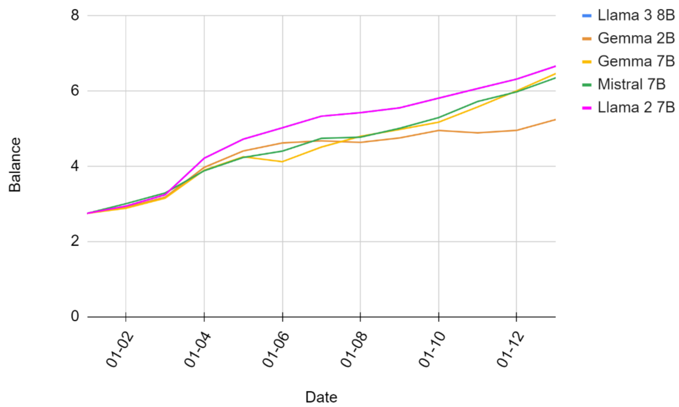

Figure 4: Movement of balance in the test year 2020 of PPO_FEM_PT_Context when using different text representation schemes. Balance is scaled to $1e6$.

Context-aware Approach						
Data	**Text Representation Schemes**	**Model**	**Avg. Return (%)**	**Avg. MDD (%)**	**Avg. MDD Duration (Days)**	**Avg. Volatility**
Price Data + News Article Titles	Gemma 2B	PPO_FEM_PT_Context	75.46	27.64	38	2.47
	Gemma 7B		68.01	27.62	32.2	**2.16**
	Llama 2 7B		**78.33**	27.81	38.8	2.18
	Mistral 7B		73.47	**27.27**	**29.6**	2.36
	Llama 3 8B		26.27	29.10	95.2	2.02

Table 4: The performance of context-aware approach when using different text representation schemes to represent the news data in terms of average return (%), average MDD (%), average MDD duration (days), and average volatility

PPO_FEM_PT_Senti has smoother overall rise in balance over the entire trading session. Thus confirming the importance of using neutral sentiment along with positive and negative news sentiments. In case of the other models, we observe that the models start facing a loss as they receive only partial signals from the news data. Overall, PPO_FEM_PT_Senti ends with a slightly higher balance than PPO_FEM_PT_Pos_Neg.

In Figure 4, we plot the balance during contract expiry for each month in the year 2020 for models that use context-aware approach. We observe that using Llama 2 7B for text representation allows the agent to learn an optimal policy, as the trading balance of the agent improves over the months and the line graph of the trading balance of Llama 2 7B is much higher than the line graph of the balances of the other LLM models.

In Table 5 we provide a summary of the best performing models from each approach based wherein the models are selected based on the avg. return (%). PPO_P from price data only approach, PPO_FEM_PT_Senti from sentiment-aware approach and PPO_FEM_PT_Context (Llama 2 7B) from context-aware approach. We observe that adding news sentiment to the price data improves the returns of the trading model compared to using only price data only approach while also reducing the MDD duration. Further using text representation schemes for representing the news data further improves the returns of the trading model. We also observe that this reduces the MDD (%) and duration as compared to the sentiment-aware approach. Thus demonstrating the advantage of using news data for improving the trading behaviour of the RL agent.

	Avg. Return (%)	**Avg. MDD (%)**	**Avg. MDD Duration (Days)**	**Avg. Volatility**
Price Data Only Approach	25.75	26.81	47.6	1.48
Sentiment-aware Approach	52.82 (+27.07)	29.69 (+2.88)	41.6 (-6)	2.05
Context-aware Approach	78.33 (+25.51)	27.81 (-1.88)	38.8 (-2.8)	2.18

Table 5: Summary of best performing model in the price data only approach, sentiment-aware approach, and context-aware approach. (The values in bracket is difference between the current row and the previous row.)

6 Conclusion and Future Work

In this work, we have performed RL-based algorithmic trading at high frequency in the futures market. We performed algorithmic trading using price-only approach, sentiment-aware approach and context-aware approach. We showed that the performance of the trading models improves when the RL agent combines news data with price data for trading. Further, we get the best results by using context-aware approach as this approach can effectively harness the hard-to-quantify information of the news data and use it for trading. We experimented with different models to show that on-policy based RL agents perform better in algorithmic trading than off-policy based RL agents.

Limitations

News data consists of some lag between when the information is available and when news is published. As the market already factors in the information even before the news is published, relying only on news data as the data source will lead to the agent receiving delayed signals, which will, in turn, impact the agent's performance. Therefore, further research should focus on using diverse data sources, especially multimodal data, and effectively reduce the lag in information. Given the advent of generative AI, the multimodal data will contain AI-generated content, which can contain fake information in text, video, or audio form. This fake information can adversely impact the agent, so future investigations should also explore techniques for adversarial training of the trading agent to prevent this impact. In this work we used only news titles to represent the news data, we did not examine the effectiveness of using news summary on the trading performance of the RL agent. The reward function employed in this study is designed to reward the immediate actions of the agent. However, in the trading domain, the true value of an action is often realized only when a position is closed. This study assumes the absence of transaction costs in the actions of the RL. Previous research has addressed this by adjusting the reward function to account for transaction costs, deducting them from the reward. However, applying this methodology in our study led to non-convergence of the model. Therefore, future investigations should focus on developing a reinforcement learning framework capable of managing delayed rewards. Such a framework should incorporate a reward function that effectively balances long-term and short-term rewards, providing a more realistic and practical approach to financial trading scenarios.

References

Rasha AbdelKawy, Walid M Abdelmoez, and Amin Shoukry. 2021. A synchronous deep reinforcement learning model for automated multi-stock trading. *Progress in Artificial Intelligence*, 10(1):83–97.

David E. Allen, Michael McAleer, and Abhay K. Singh. 2019. Daily market news sentiment and stock prices. *Applied Economics*, 51(30):3212–3235.

Dogu Araci. 2019. Finbert: Financial sentiment analysis with pre-trained language models. *Preprint*, arXiv:1908.10063.

Yu-Fu Chen and Szu-Hao Huang. 2021. Sentiment-influenced trading system based on multimodal deep reinforcement learning. *Applied Soft Computing*, 112:107788.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale. *Advances in Neural Information Processing Systems*, 35:30318–30332.

Eugene F Fama. 1970. Efficient capital markets: A review of theory and empirical work. *The journal of Finance*, 25(2):383–417.

Badr Hirchoua, Brahim Ouhbi, and Bouchra Frikh. 2021. Deep reinforcement learning based trading agents: Risk curiosity driven learning for financial rules-based policy. *Expert Systems with Applications*, 170:114553.

Wei Huang, Xudong Ma, Haotong Qin, Xingyu Zheng, Chengtao Lv, Hong Chen, Jie Luo, Xiaojuan Qi, Xianglong Liu, and Michele Magno. 2024. How good are low-bit quantized llama3 models? an empirical study. *arXiv preprint arXiv:2404.14047*.

Gyeeun Jeong and Ha Young Kim. 2019. 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.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Prahlad Koratamaddi, Karan Wadhwani, Mridul Gupta, and Sriram G Sanjeevi. 2021. Market sentiment-aware deep reinforcement learning approach for stock portfolio allocation. *Engineering Science and Technology, an International Journal*, 24(4):848–859.

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

Yuming Li, Pin Ni, and Victor Chang. 2020. Application of deep reinforcement learning in stock trading strategies and stock forecasting. *Computing*, 102(6):1305–1322.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Xingyu Dang, and Song Han. 2023. Awq: Activation-aware weight quantization for llm compression and acceleration. *arXiv preprint arXiv:2306.00978*.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Mehran Taghian, Ahmad Asadi, and Reza Safabakhsh. 2022. Learning financial asset-specific trading rules via deep reinforcement learning. *Expert Systems with Applications*, 195:116523.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Thibaut Théate and Damien Ernst. 2021. An application of deep reinforcement learning to algorithmic trading. *Expert Systems with Applications*, 173:114632.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Xing Wu, Haolei Chen, Jianjia Wang, Luigi Troiano, Vincenzo Loia, and Hamido Fujita. 2020. Adaptive stock trading strategies with deep reinforcement learning methods. *Information Sciences*, 538:142–158.

Bing Yang, Ting Liang, Jian Xiong, and Chong Zhong. 2023. Deep reinforcement learning based on transformer and u-net framework for stock trading. *Knowledge-Based Systems*, 262:110211.

Hongyang Yang, Xiao-Yang Liu, Shan Zhong, and Anwar Walid. 2020. Deep reinforcement learning for automated stock trading: An ensemble strategy. In *Proceedings of the first ACM international conference on AI in finance*, pages 1–8.

A Links of LLM Models

- https://huggingface.co/TheBloke/Llama-2-7B-AWQ
- https://huggingface.co/TheBloke/Mistral-7B-v0.1-AWQ
- https://huggingface.co/google/gemma-2b
- https://huggingface.co/google/gemma-7b
- https://huggingface.co/meta-llama/Meta-Llama-3-8B

B Algorithm for Futures Trading

Algorithm 1: Algorithm for Futures Trading

Input:
num_lots: Number of lots agent will buy or sell
$balance$: Balance of the agent
EOC: Contract has expired (True or False)
EOD: Trading day has ended (True or False)
$contract\_value$: Initialize contract value to 0
num_lots_held: Initialize number of lots held by agent to 0
max_num_lots: Initialize maximum no. of lots the agent can hold
if EOC **:**
    Set num_lots to num_lots_held
    Calculate contract value
    Calculate margin value
    Update the balance with the margin value
    Set num_lots_held to 0
end
else:
    if $num\_lots < -max\_num\_lots$ *or* $num\_lots > max\_num\_lots$ **:**
        $num\_lots = 0$
    end
    Calculate contract value
    Calculate margin value
    Update the balance with the margin value
    Update num_lots_held with num_lots
    if EOD **:**
        Calculate price difference for M2M
        Update the balance by using the price difference
    end
end

C Model Configuration

The configuration of FEM for the price data only approach, sentiment-aware approach and context-aware approach are shown in Tables 6, 7, 8. The configuration of the Q network and value network is shown in Table 9. The configuration of the policy network and value network is shown in Table 10, where dim is dimension of vector obtained at the last layer in Table 7.

Trading Models	CNN Layer	Layer 1
DQN_FEM_P	14×20	21×14
PPO_FEM_P	14×20	21×14
PPO_FEM_PT (Senti)	15×20	21×14

Table 6: Configuration of FEM for encoding price data in DQN_FEM_P and PPO_FEM_P and for encoding news sentiment and price data in PPO_FEM_PT (Senti)

Text Representaion Model	CNN	Layer 1	Layer 2
Gemma 2B	2048×1000	1000×500	100×1
Gemma 7B	3072×1000		
Llama 2 7B	4096×1000		
Mistral 7B			
Llama 3 8B			

Table 7: Configuration of FEM for encoding the news articles in context-aware approach

D Hyperparamters

In all the approaches, the window size (w) for selecting the price data is 5 mins. In sentiment-aware approach at each tick i we consider news articles published in last 1 hr. In context-aware approach for representing the news data we set the window size w' to 60 mins. We select the 10 latest news articles and set the value of k to 10. In the reward function, we set the value λ to 0.85. We determined the optimal λ value by training PPO_FEM_PT_Senti on data from 2010-2015 for values of λ which range from 0.15 to 0.95 and validated the model on data of 2016. The graph of return (%) for different values of λ is shown in Figure 5.

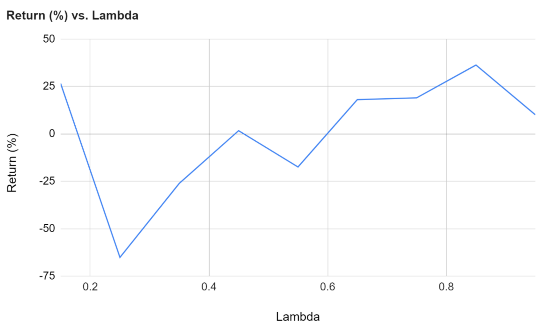

Figure 5: Return (%) of PPO_FEM_PT_Senti for different values of λ

The hyperparameters for training the DQN-based and PPO-based trading models are shown in Tables 11 and 12, respectively.

	CNN layer for prices		Layer for combining prices data and news data
Text Embedding Model	**CNN**	**Layer 1**	**Layer 1**
Gemma 2B	14×14	14×14	16×128
Gemma 7B			16×16
Llama 2 7B			16×128
Mistral 7B			16×16
Llama 3 8B			16×128

Table 8: Configuration of FEM for encoding the prices and combining news data and price data in context-aware approach

	Q Network and Target Network		
Trading Models	**Neural Layer 1**	**Neural Layer 2**	**Neural Layer 3**
DQN_P	14×64	64×64	64×1
DQN_FEM_P	14×64	64×64	64×1

Table 9: Configuration of Q network and target network in DQN-based RL models (price data only approach)

	Policy Network and Value Network		
Trading Models	**Neural Layer 1**	**Neural Layer 2**	**Neural Layer 3**
PPO_P	14×64	64×64	64×1
PPO_FEM_P	14×64	64×64	64×1
PPO_PT_Senti	15×64	64×64	64×1
PPO_FEM_PT_Senti	16×16	16×16	16×1
PPO_FEM_PT_Context	$dim \times 64$	64×16	16×1

Table 10: Configuration of value network and policy network in PPO-based RL models (sentiment-aware and context-aware approach).

E Additional Results

The year-wise return (%), MDD (%) and duration for price data only approach and sentiment-aware approach are shown in Tables 13, 14 and 15, respectively. The year-wise return (%), MDD (%) and duration for context-aware approach are shown in Tables 16, 17 and 18, respectively.

	Hyperparameters						
Trading Models	**Batch Size**	**Learning Rate**	**Buffer Size**	**Learning Starts**	**Train Frequency**	**Gradient Steps**	**Target Update Interval**
DQN_P	128	0.002	200000	10000	1 episode	30000	10000
DQN_FEM_P	128	0.0005	200000	200000	1 episode	20000	9500

Table 11: Hyperparameters of DQN-based trading models (price data only approach)

	Hyperparameters				
Trading Models	**Batch Size**	**Learning Rate**	**Entropy Co-efficient**	**Epochs**	**Steps**
PPO_P	128	0.002	0.02	5	200
PPO_FEM_P	128	0.0005	0.02	10	200
PPO_PT_Senti	128	0.002	0.02	5	50
PPO_FEM_PT_Senti	128	0.0002	0.02	9	50
PPO_FEM_PT_Context (Gemma 2B)	128	0.0002	0.02	7	1500
PPO_FEM_PT_Context (Gemma 7B)	64	0.0002	0.02	7	2000
PPO_FEM_PT_Context (Llama 2 7B)	128	0.00019	0.02	7	1500
PPO_FEM_PT_Context (Mistral 7B)	128	0.00019	0.02	6	1500
PPO_FEM_PT_Context (Llama 3 8B)	128	0.0002	0.02	6	2000

Table 12: Hyperparameters of PPO-based trading models (price data only, sentiment-aware and context-aware approach)

Return (%)			
Years	**PPO_P**	**PPO_PT_Senti**	**PPO_FEM_PT_Senti**
2017	24.04	-7.02	18.5
2018	10.78	38.74	68.43
2019	10.05	46.23	66.2
2020	70.52	46.07	64.01
2021	13.35	38.22	46.92
Avg. Return (%)	25.75	32.44	**52.81**

Table 13: The year-wise return (%) of models in price data only approach and sentiment-aware approach

MDD (%)			
Years	**PPO_P**	**PPO_PT_Senti**	**PPO_FEM_PT_Senti**
2017	26.14	35.09	40.8
2018	26.47	27.25	25.79
2019	28.04	28.94	26.5
2020	26.72	27.88	26.2
2021	26.66	32.39	29.11
Avg. MDD (%)	26.81	30.31	29.68

Table 14: The year-wise MDD (%) of models in price data only approach and sentiment-aware approach

MDD Duration (Days)			
Years	**PPO_P**	**PPO_PT_Senti**	**PPO_FEM_PT_Senti**
2017	2	144	135
2018	34	17	14
2019	86	102	9
2020	42	35	15
2021	74	27	35
Avg. MDD Duration (Days)	47.6	65	**41.6**

Table 15: The year-wise MDD duration (days) of models in price data only approach and sentiment-aware approach

Return (%)					
Years	**Llama 3 8B**	**Gemma 7B**	**Mistral 7B**	**Gemma 2B**	**Llama 2 7B**
2017	21.54	53.07	70.37	26.64	24.69
2018	12.12	42.01	55.49	88.1	52.15
2019	7.91	47.55	38.26	82.34	67.53
2020	71.61	135.36	131.24	90.85	142.56
2021	18.14	62.05	71.96	89.35	104.68
Avg. Return (%)	26.27	68.01	73.47	75.46	**78.33**

Table 16: The year-wise return (%) of PPO_FEM_PT_Context while using different text representation schemes to represent the news titles in the news data

MDD (%)					
Years	**Llama 3 8B**	**Gemma 7B**	**Mistral 7B**	**Gemma 2B**	**Llama 2 7B**
2017	30.44	33.7	30.24	27.44	33.62
2018	31.59	28.32	26.09	26.64	27.6
2019	26.76	26.06	27.17	26.51	26.97
2020	31.69	24.48	25.81	26.17	25.88
2021	24.99	25.51	27	31.4	24.96
Avg. MDD (%)	29.1	27.62	**27.27**	27.64	27.81

Table 17: The year-wise MDD (%) of PPO_FEM_PT_Context while using different text representation schemes to represent the news titles in the news data

MDD Duration (Days)					
Years	**Llama 3 8B**	**Gemma 7B**	**Mistral 7B**	**Gemma 2B**	**Llama 2 7B**
2017	61	62	50	132	147
2018	199	28	25	21	9
2019	32	34	29	9	24
2020	49	20	5	5	9
2021	135	17	39	23	5
Avg. MDD Duration (Days)	95.2	32.2	**29.6**	38	38.8

Table 18: The year-wise MDD duration (days) of PPO_FEM_PT_Context while using different text representation schemes to represent the news titles in the news data