FRANCISCO CAIO LIMA PAIVA

Assimilating sentiment analysis in reinforcement learning for intelligent trading

São Paulo
2023

FRANCISCO CAIO LIMA PAIVA

Assimilating sentiment analysis in reinforcement learning for intelligent trading

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Mestre em Ciências.

São Paulo
2023

FRANCISCO CAIO LIMA PAIVA

Assimilating sentiment analysis in reinforcement learning for intelligent trading

Versão Original

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Mestre em Ciências.

Área de Concentração:
Engenharia de Computação

Orientadora:
Profa. Dr.ª Anna Helena Reali Costa

São Paulo
2023

Catalogação-na-publicação

Paiva, Francisco Caio Lima
Assimilating sentiment analysis in reinforcement learning for intelligent trading / F. C. L. Paiva -- São Paulo, 2023.
92 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1.Inteligência artificial 2.Métodos estatísticos para aprendizagem 3.Redes neurais 4.Linguagem natural 5.Mercado financeiro I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais II.t.

Dedicatória

To my mother, whose tireless efforts and sacrifices allowed me to thrive. To my life partner, Glória Bolani Porteiro, for their unwavering support in all my endeavors. To all other members of my family for just being there. Through their chain of love and incentive, I have found the courage and strength to chase my dreams and accomplish my aspirations.

ACKNOWLEDGMENTS

I thank all the personnel at *Escola Politénica da Universidade de São Paulo* and particularly at the *Departamento de Engenharia de Computação e Sistemas Digitais* (PCS) for supporting my inquiries and technical issues. I would especially like to thank my advisor Prof.ª Dr.ª Anna Helena Reali Costa, who guided me through all the research and development processes in this work, which ended up making me a better researcher.

I want to thank *Itaú Unibanco S.A.* for supporting this work through the scholarship program of *Programa de Bolsas Itaú* (PBI), linked to the *Centro de Ciência de Dados* (C^{2}D) of *Escola Politécnica da USP*. Any views and opinions expressed in this dissertation are those of the authors and do not necessarily reflect the official policy or position of the funding companies.

I also thank Jardel Trento Alves Negrão for being supportive and understanding during all these years of effort.

Last but not least, I also thank Leonardo Felizardo, Roberto Fray da Silva, Angel F. Magnossão de Paula, Taynan Ferreira, Rodrigo Aquino, Andrey Ruschel, Waldyr Faustini, and all other colleagues with whom I partnered and worked together in research and classes. Our interactions made it possible to explore the unknown without the fear of getting lost.

“We can only see a short distance ahead, but -we- can see plenty there that needs to be done.”

-Alan M. Turing-

RESUMO

A viabilidade de obter lucro por meio de negociação em alta frequência de um único ativo financeiro é uma questão de pesquisa em aberto. O aprendizado por reforço (RL) e a análise de sentimentos textual (SA) são cada vez mais relevantes para esse problema financeiro. Notavelmente, apesar de sua proeminência, as técnicas de RL e SA raramente foram combinadas para aprender estratégias de negociação de ativos. Além disso, os tópicos não abordados incluem: capturar o impulso do sentimento do mercado por meio da extração explícita de características de sentimento que refletem a condição do mercado ao longo do tempo; e verificar se tal incorporação de informações aos algoritmos de RL não afeta negativamente a consistência e estabilidade em diferentes situações. O presente trabalho propõe que o Sentiment-Aware Reinforcement Learning Intelligent Trading System (ITS-SentARL) preencha esta lacuna. O ITS-SentARL melhora o lucro e a estabilidade ao alavancar o humor do mercado por meio de uma faixa ajustável de recursos de sentimentos obtidos de notícias textuais. Ao contrário de pesquisas anteriores, projetamos um extrator de sentimentos de acordo com o design vencedor da rede neural convolucional da competição de análise de sentimentos SemEval-2017 – o treinamento desse extrator de sentimentos foi feito com dados rotulados por especialistas de mercado. Depois de treinar o extrator de sentimento, ele pode ser usado para pontuar novos dados e usá-los como parte da representação de estado de um algoritmo Advantage Actor-Critic (A2C), uma abordagem RL. Tanto uma estratégia A2C sem sentimentos quanto a estratégia clássica de compra e retenção (BH) são usadas como linhas de base. A avaliação da arquitetura ITS-SentARL ocorre em vinte ativos financeiros, dois custos de transação e cinco diferentes períodos e inicializações. Notavelmente, os resultados mostram que o agente ITS-SentARL superou consistentemente o agente de negociação A2C de linha de base para diversas situações de mercado e, em alguns cenários, também a estratégia BH. Os resultados sugerem que a incorporação do sentimento de mercado é benéfica, mas depende da quantidade de notícias divulgadas e sua correlação com o preço.

Palavras-Chave – Aprendizado por reforço, Processamento de linguagem natural, Análise de sentimentos, Redes neurais profundas, Mercado de ações.

ABSTRACT

The viability of attaining profit through high-frequency active trading of a single asset is an open research question. Reinforcement learning (RL) and textual sentiment analysis (SA) are increasingly relevant for this financial task. Notably, despite their prominence, RL and SA techniques have rarely been combined for learning asset trading strategies. Furthermore, unaddressed topics include: capturing market sentiment momentum through the explicit extraction of sentiment features that reflect the market condition over time; and verifying that such information incorporation to RL algorithms does not negatively affect consistency and stability in different situations. The present work proposes that the Sentiment-Aware Reinforcement Learning Intelligent Trading System (ITS-SentARL) fills this gap. ITS-SentARL improves profit and stability by leveraging market mood through an adjustable range of past sentiment features obtained from textual news. Unlike previous research, a sentiment extractor was designed according to the convolutional neural network winning design of the renowned SemEval-2017 sentiment analysis competition – the training of this sentiment extractor was done with data labeled by market specialists. After training the sentiment extractor, it can be used to score new data and use it as part of the state representation of an Advantage Actor-Critic (A2C) algorithm, an RL approach. Both a sentiment-free A2C and the classical buy-and-hold (BH) strategy are used as baselines. The evaluation of ITS-SentARL architecture occurs over twenty assets, two transaction costs, and five different periods and initializations. Remarkably, the results show that the ITS-SentARL agent consistently outperformed the baseline sentiment-free A2C trading agent for diverse market situations and, in some scenarios, also the BH strategy. Results suggest that market sentiment incorporation is beneficial but depends on the amount of news released and its correlation to the price.

Keywords – Reinforcement Learning, Natural Language Processing, Sentiment Analysis, Deep Neural Networks, Stock Markets.

LIST OF FIGURES

LIST OF TABLES

LIST OF NOTATION SYMBOLS

t - Time instant during an experimental episode

w - Look-back window size for hour and price difference data

$\mathcal{S}$ - Set of all possible reachable s states

$\mathcal{A}$ - Set of all available actions at every given state

$\mathcal{T}$ - Transition probability function

$\mathcal{R}$ - Set of all possible rewards

S_t - State random variable at instant t

s - Particular value for a random variable S_t

A_t - Action random variable at instant t

a - Particular value for a random variable A_t

R_t - Numerical reward at instant t

s' - Set of all possible next states S_{t+1} for a given instant t

P - Probability value

T - Last time step of an experiment episode

μ - Deterministic policy

π - Current stochastic policy

G_t - expected accumulated reward from instant t onward

V - State value function

V^{π} - State value function for current stochastic policy π

$\mathbb{E}$ - Expected value

γ - Reward discount factor

Q - State-action value function

Q^{π} - State-action value function for current stochastic policy π

V^{*} - Optimal state value function

π^{*} - Optimal policy

$Q^{\pi^{*}}$ - State-action value function for optimal policy π

δ_t - Temporal Difference (TD) error

$\boldsymbol{\vartheta}$ - Parameter vector of the value function approximator (e.g., neural network weighs)

u - Vector dimension for $\boldsymbol{\vartheta}$

i - Index for a sequence of loss function iterations

L_i - Loss of the value function i

$\boldsymbol{\vartheta}_i$ - i_{th} iteration of parameter vector $\boldsymbol{\vartheta}$

$\boldsymbol{\theta}$ - Parameter Vector of the policy function approximator (e.g., neural network weighs)

d - Vector dimension for $\boldsymbol{\theta}$

J^{π} - Expected future cumulative reward for a policy π

$\boldsymbol{\theta}_t$ - Parameter vector at instant t

α - Gradient's step size or learning rate hyperparameter

∇ - Gradient of a function

b_t - Baseline function value at instant t

Λ - Advantage Function value

k - Number of steps before a gradient update to function parameters

p_t - Price at instant t

z_t - Price difference between current (t) and previous instant $(t-1)()$

h_t - Hour of the day at instant t

τ_t - Normalized hour of the day at instant t

S_t^M - Market-state vector representation at instant t

S_t^C - Agent-state vector representation at instant t

S_t^B - Base-state vector representation at instant t

$\rho_{t+1}^{Nominal}$ - Nominal (undiscounted) financial return at instant $t+1$

φ - Fixed amount of traded asset shares

$\rho_{t+1}^{Deduction}$ - Financial return deduction at instant $t+1$

ξ - Transaction cost

ρ_{t+1}^{Trader} - Trader or discounted financial return at instant $t+1$

j - Headline index

y_t^j - Score value for each news headline at given instant t

Ω - Sentiment weighting function

ψ - Initial net worth or cash amount available for investment

B - Bullishness index value

e_t - Weighted overall sentiment score at instant t

S_t^E - Sentiment-state vector representation at instant t

l - Market sentiment look-back window size

D - Last time series data point in all collected data

η - Index of a given time data point in D

z_η - Price difference between current and previous time data points η

e_η - Weighted overall sentiment score at time data point η

m - Size of word vector for news headlines

n - Size of word embedding dense vector

v - Word embedding dense vector of size n

σ - Time shift value between sentiment and price time series

χ - Number of trading days the model operated over during an episode

p_0^{asset} - Price of an asset at an initial instant $t = 0$

LIST OF ACRONYMS

A2C - Advantage Actor Critic

A3C - Asynchronous Advantage Actor Critic

AMH - Adaptive Market Hypothesis

ANN - Artificial Neural Network

BH - Buy-and-Hold Strategy

CNN - Convolutional Neural Network

CV - Cross-validation

DDPG - Deep Deterministic Policy Gradient

DQN - Deep Q-Network

EMH - Efficient Market Hypothesis

ETF - Exchange-Traded Fund

FDDR - Fuzzy Deep Direct Reinforcement

ITS-SentARL - Sentiment-Aware Reinforcement Learning Intelligent Trading System

MDP - Markov Decision Process

ML - Machine Learning

NLP - Natural Language Processing

OHLC - Open, High, Low, Close values

PM - Portfolio Management

PPO - Proximal Policy Approximation

RL - Reinforcement Learning

ResNet - Residual Neural Network

RRL - Recurrent Reinforcement Learning

SA - Sentiment Analysis

SARSA - State–Action–Reward–State–Action Algorithm

SAT - Single Asset Trading

S&P 500 - Standard & Poor top 500 companies index

SPY - S&P 500 ETF

SR - Sharpe Ratio

TC - Transaction Costs

TD - Temporal Difference

TD(0) - One-step Temporal-Difference Algorithm

CONTENTS

1 INTRODUCTION

The financial market is a rich and complex domain of study that attracts both investors and researchers. One of the most prominent problems in this domain is *active trading*, which comprises continuously operating (e.g., buying and selling) assets (e.g., stocks, currencies, and others) to profit over short-term price movements. When executing active trading, investors typically rely on the premise that customized strategies can leverage asset information such as historical price and volume as indicators of patterns that could forecast tendencies. Alternatively, according to some economics researchers, this financial market exploitability is unlikely because an asset's price might be exhibiting predominantly random walk behavior (KING; COOTNER, 1965; FAMA, 1965) and adjusting almost instantly to new information – *Adaptive Market Hypothesis* (AMH) (LO, 2004).

Notwithstanding, the market exploitability is an open economics discussion that, together with the repetitive nature of active trading, led the *machine learning* (ML) community to become interested in designing profitable intelligent trading systems. A systematic literature review Henrique, Sobreiro and Kimura (2019) identified the prevalence of ML works that approach active trading as a market forecasting problem and thus resort to supervised learning rule-based strategies. Although this research direction has led to positive results, active trading is, in its essence, a sequential decision-making problem. Unsurprisingly, the presence of typical trading operation costs may be enough to make a rule-based supervised approach less effective than a sequential decision-making solution (MOODY; WU, 1997; DENG et al., 2017; CARAPUÇO; NEVES; HORTA, 2018).

Reinforcement learning (RL) is an ML framework that defines a sequential decision-making structure for an agent to solve problems on a trial and error basis by interacting with the environment, taking actions, and receiving rewards (SUTTON; BARTO, 2018). Thus, unlike a supervised learning approach, RL does not require access to labeled data. Also, supervised techniques imply the design of customized rules to leverage predictions. Instead, an RL agent can directly learn a profitable trading strategy for a reasonable course of action that maximizes the long-term accumulated reward across market scenarios.

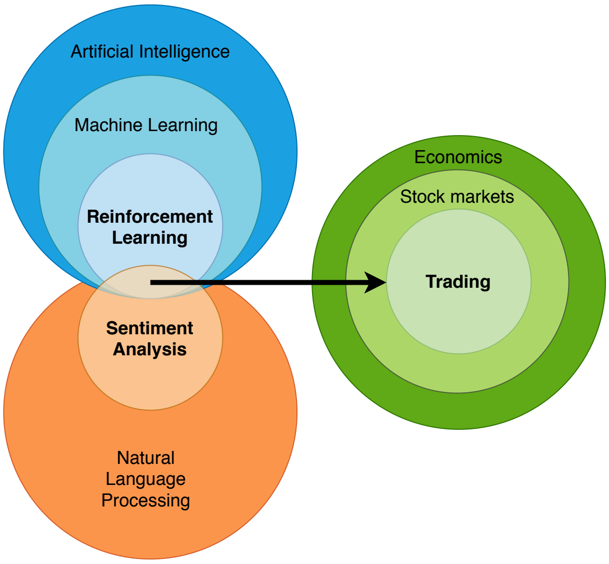
When trading, investors can employ information from two types of market indicators: *technical* or *fundamental*. It is usual among RL works to adopt predominantly technical indicators such as the normalized price time series (MOODY; WU, 1997; DENG et al., 2017; ALMAHDI; YANG, 2017) or moving average (GIACOMAZZI DANTAS; GUERREIRO E SILVA, 2018; ALIMORADI; HUSSEINZADEH KASHAN, 2018; ZHANG; MARINGER, 2014). Among RL works that employed fundamental indicators, they mostly use the information in companies' periodic balance reports, such as market capitalization or price-to-earnings ratio values (ZHANG; MARINGER, 2016; WANG et al., 2019; MA et al., 2021). However, these fundamental indicators rely on values reported on low frequencies, such as every quarter. Thus, for active trading scenarios of higher frequency, such as hourly or daily, the information in these types of fundamental indicators, after some period, may not be reliable enough. Alternatively, systematic literature reviews (KHADJEH-NASSIRTOUSSI et al., 2014; LOUGHRAN; MCDONALD, 2016; HENRIQUE; SOBREIRO; KIMURA, 2019) have shown that textual news is a well-adopted information source for high-frequency trading supervised approaches.

Notably, only recently have a few authors (FEUERRIEGEL; PRENDINGER, 2016; YANG; YU; ALMAHDI, 2018; YE et al., 2020) opened up this promising research path of improving RL solutions by incorporating features extracted through *natural language processing* (NLP). Nevertheless, despite these successful initial efforts, numerous issues are still available for exploration, from which there are two particular topics of interest, as follows. *Market sentiment momentum*: when extracting features from financial news for sequential decision-making, it is challenging to capture the prevailing market mood about a given asset instead of an eventful cause for price oscillation (KHADJEH-NASSIRTOUSSI et al., 2014). *Information incorporation vs. RL methods' instability*: RL techniques are well-documented to be unstable, generalize poorly (HENDERSON et al., 2018), and the stochasticity of the financial market environment (TSAY, 2010) amplifies these difficulties. Thus, introducing any new information entails a thorough examination to verify that it is not negatively affecting the algorithm's stability.

This work approaches these issues by proposing the *Sentiment-Aware RL intelligent trading system* (ITS-SentARL). ITS-SentARL presents a modular architecture that aids in incorporating past sentiment features such that it captures the persistent marked mood and the dissolution of the news impact over the period. In addition, this mechanism helps to deal with the issue of reduced *textual data coverage* (YE et al., 2020), which occurs because of the lack of news articles at every instant for each company. Moreover, ITS-

Sentarl presents the innovation of being the first RL architecture to adopt a sentiment extractor with the optimal configuration (FERREIRA et al., 2020) of the winner design (MANSAR et al., 2017) of the SemEval-2017 Task 5 challenge (CORTIS et al., 2017). This challenge employed the help of market specialists to label textual news headlines and thus provided the research community with a gold-standard (i.e., ground-truth) dataset. The present research also uses this gold-standard dataset to train the ITS-SentARL's sentiment extractor. Ultimately, as depicted in Figure 1, the research opportunity pursued in this present work lies in investigating the *sentiment analysis* (SA) subset of natural language processing techniques for improving reinforcement learning algorithms' performance over trading tasks.

Figure 1: The intersection of reinforcement learning and sentiment analysis for financial trading tasks outlines the research direction of the present work.



Source: Author's own production.

1.1 Contributions

As previously mentioned, the present work investigates the impact that feature incorporation may have on the stability of RL algorithms. So naturally, the methodology here adopted includes the suggestions by Henderson et al. (2018) regarding best practices about appropriate experimental design and evaluation of results of RL techniques. Hence, to meticulously examine ITS-SentARL capabilities, the following steps were crucial. First, data gathered encompassed twenty different assets from diverse segments. Second, for evaluation, the adoption of a rolling window setup comprised training and testing over five periods, including the COVID-19 pandemic crisis. Next, both a no-penalty and high-penalty environment were used as market transaction costs (TC). Then, to execute an ablation study, a similar but sentiment-free (i.e., no sentiment features) version of ITS-SentARL acts as one of two adopted baselines. The classic buy-and-hold (BH) strategy serves as the second baseline. Finally, ITS-SentARL and the sentiment-free baseline are subjected to five different model seed initializations for each combination of asset, period, and TC, adding up to a thousand trials for comparison between approaches. Valuation metrics included total return, annualized return, and Sharpe ratio.

The contributions in this research are twofold:

- **ITS-SentARL**, a novel Sentiment-Aware RL intelligent trading system for efficient incorporation of market sentiment momentum from news articles to a trading RL sequential decision process.
- First RL asset trading approach to verify the benefits of information introduction through a methodology that follows recent experimental design suggestions (HENDERSON et al., 2018) for RL methods' stability, generalization, and consistency proper evaluation.

1.2 Publications

In the course of this investigation, the following publications were made:

- Ferreira, T.; **Lima Paiva, F. C.**; Silva, R.; Paula, A.; Costa, A.; Cugnasca, C. *"Assessing Regression-Based Sentiment Analysis Techniques in Financial Texts"*. Proceedings of XVI National Meeting on Artificial and Computational Intelligence. Porto Alegre, RS, Brasil: SBC, 2020. p. 729–740.

  In this work, a thorough investigation of NLP techniques for extracting sentiment scores from news headlines was conducted. This examination aimed to refine the discoveries made in a renowned SA competition for extracting news headlines from the financial domain and determine the adequate ML architecture and configuration for such a task. Ultimately, the best-devised approach was reused in this present investigation to score the sentiment in financial news headlines. More implementation details and clarification of its incorporation into ITS-SentARL's design are discussed in the remainder of this manuscript. Among this article's outcomes is the sentiment extractor implementation and training data that are publicly available online[1].

- Felizardo, L. K.; **Lima Paiva, F. C.**; Costa, A. H. R.; Del-Moral-Hernandez, E. *"Reinforcement Learning Applied to Trading Systems: A Survey"*. arXiv, 2022.

  This work was essential in methodically identifying research gaps and, consequently, opportunities in the exploration of reinforcement learning techniques for the financial market trading problem. Achieving this level of fruition required searching for articles on the target subject published between 2014 and 2020. From the several identified studies, twenty-nine articles were selected. Then, using the seminal RL seminal literature as a basis (SUTTON; BARTO, 2018), this study devised a workflow pipeline that helps categorize and group studies in the field. Next, an in-depth review and comparison of these articles followed, which allowed the dissection and information extraction of these studies under the devised pipeline. Ultimately, this work observed the increase in the adoption of reinforcement learning for financial market trading and allowed the identification of state-of-the-art trends in the field, which led to insights and suggestions for future studies. Hence, this work was pivotal for identifying research venues investigated in the following two publications.

- **Lima Paiva, F. C.**; Felizardo, L. K.; Bianchi, R. A. d. C. B.; Costa, A. H. R. *"Intelligent Trading Systems: A Sentiment-Aware Reinforcement Learning Approach"*. Proceedings of the Second ACM International Conference on AI in Finance. New York, NY, USA: Association for Computing Machinery, 2021. (ICAIF '21).

  This article highlights the central research investigation described in the present manuscript. As such, it introduces ITS-SentARL's architecture and summarizes the presently discussed topics regarding the benefits of incorporating market sentiment momentum into an RL trading framework. Consequently, some of this manuscript's overall concepts, images, and tables were originally published at the Second ACM International Conference on AI in Finance (ICAIF 2021). Finally, as outcomes, this work favored providing the publicly available implementation of the source

[1] https://bit.ly/3kzau8G

code of a financial news web crawler[2] and ITS-SentARL algorithm and experimental setup[3].

- Felizardo, L. K.; **Lima Paiva, F. C.**; de Vita Graves, C.; Matsumoto, E. Y.; Costa, A. H. R.; Del-Moral-Hernandez, E.; Brandimarte, P. *"Outperforming algorithmic trading reinforcement learning systems: A supervised approach to the cryptocurrency market"*. Expert Systems with Applications, v. 202, p. 117259, 202.

  This research article investigated the adoption of several recent advancements in RL architectures and compared them to typical supervised learning methods for trading in the cryptocurrency market. As a result, this research showed alternative ways to frame the trading task under the different assumptions of the RL framework, which allowed for a better comparison between supervised and reinforcement learning approaches. Moreover, the state-of-the-art Residual Neural Network time series architecture (FAWAZ et al., 2019) displayed outstanding performance and allowed better investigation of the impact of features through the analysis of residual blocks. Thus, this publication study opens up a venue for future improvements to ITS-SentARL architecture. For instance, the promising ResNet could help enhance the proposed system's performance and further enhance our analysis of the impact of market sentiment momentum on the agent's behavior.

1.3 Organization of the Manuscript

The remainder of this manuscript is organized as follows.

- Chapter 2 reviews relevant articles and contains a comparative analysis of techniques and approaches related to the presented one.

- Chapter 3 includes financial concepts and the definition of financial trading as a sequential decision-making problem.

- Chapter 4 covers the RL approach and how it was adjusted to the trading problem. Moreover, the proposed ITS-SentARL architecture is introduced in this chapter, and the underlying mechanics of capturing sentiment momentum are also discussed.

[2] https://github.com/xicocaio/financial_web_crawler

[3] https://github.com/xicocaio/its-sentarl

- Chapter 5 includes details about the experimental data, design decisions, methodology, and evaluation of results.
- Chapter 6 concludes this study with final remarks and opportunities in this research path.

2 RELATED WORK

Researchers may focus either on the *single asset trading* (SAT) problem or its generalization, the *portfolio management* (PM). In SAT, an investor can only operate one asset at a time, while in PM, the objective is to balance and continuously redistribute assets in a wallet. Most RL studies favor the former (MOODY; WU, 1997; MARINGER; ZHANG, 2014; ZHANG; MARINGER, 2014; EILERS et al., 2014; GABRIELSSON; JOHANSSON, 2015; FEUERRIEGEL; PRENDINGER, 2016; DENG et al., 2017; SPOONER et al., 2018; CARAPUÇO; NEVES; HORTA, 2018; ALIMORADI; HUSSEINZADEH KASHAN, 2018; GIACOMAZZI DANTAS; GUERREIRO E SILVA, 2018; LI; ZHENG; ZHENG, 2019; WU et al., 2019; JEONG; KIM, 2019; PONOMAREV; OSELEDETS; CICHOCKI, 2019; MA et al., 2021; HIRCHOUA; OUHBI; FRIKH, 2021; TSANTEKIDIS et al., 2021; THÉATE; ERNST, 2021), while some others, the latter (ZHANG; MARINGER, 2016; ALMAHDI; YANG, 2017; KANG; ZHOU; KANG, 2018; PENDHARKAR; CUSATIS, 2018; YU et al., 2019; WANG et al., 2019; PARK; SIM; CHOI, 2020; ALMAHDI; YANG, 2019; ABOUSSALAH; LEE, 2020; YE et al., 2020). The present work will address the SAT problem while still appreciating distinguished solutions in PM.

Given the previously mentioned active trading circumstances, such as acting in a fast-paced changing environment, it is reasonable to view trading as a sequential decision-making problem. Not surprisingly, as early as 1997, researchers have viewed the RL framework as a straightforward approach to designing intelligent trading systems (MOODY; WU, 1997). RL algorithms aim at learning a policy (i.e., strategy) function – which is a mapping between a representation of the environment and allowed operations – that defines an agent's successful behavior under diverse situations. There are three major categories of RL solutions: *policy-based*, *value-based*, and *actor-critic*.

Among policy-based methods, the popular *policy gradient* is a technique for directly approximating parametric policies through the gradient ascent of rewards (e.g., financial return) over actions (e.g., trading operations). Policy gradient methods have been employed in trading since Moody and Wu (1997) proposed the *Recurrent RL* (RRL) approach, the first RL intelligent trading system to gain wide notoriety. RRL is a slightly modified version of the typical policy gradient algorithm. Despite its simplicity, RRL was a pioneering work that had such a profound influence over the community that, to this date, several studies (MARINGER; ZHANG, 2014; ZHANG; MARINGER, 2014; GABRIELSSON; JOHANSSON, 2015; ZHANG; MARINGER, 2016; ALMAHDI; YANG, 2017; DENG et al., 2017; ALMAHDI; YANG, 2019; ABOUSSALAH; LEE, 2020) continue to extend its most fundamental ideas and structure. For instance, Deng et al. (2017) proposed the FDDR, an updated version of RRL with deep neural networks. Alternatively, Almahdi and Yang (2019) leveraged the RRL architecture flexibility to combine it with particle swarm optimization meta-heuristic for PM tasks. Although the present work does not fully employ the entire RRL algorithm, two original RRL concepts were adopted. The first one is the concise financial return formulation, which accounts for the constant change in trading actions, price changes, amount of invested shares, and transaction costs. The second adopted concept is the market state representation, which contains observations from historical price differences and the last action. Representing the market state in such a way helps to ground the agent's interaction with the market situation.

Value-based techniques aim at learning optimal value functions, which assign scores for given situations and help the agent estimate future opportunities and, thus, derive optimal policies. Most RL financial studies that employed value-based approaches, opted for *Q-learning* (FEUERRIEGEL; PRENDINGER, 2016; GIACOMAZZI DANTAS; GUERREIRO E SILVA, 2018; CARAPUÇO; NEVES; HORTA, 2018), while others preferred SARSA (PENDHARKAR; CUSATIS, 2018; ALIMORADI; HUSSEINZADEH KASHAN, 2018). Interestingly, there is an increase in researchers (JEONG; KIM, 2019; PARK; SIM; CHOI, 2020; TSANTEKIDIS et al., 2021; THÉATE; ERNST, 2021) exploring the DQN algorithm (MNIH et al., 2015), a deep learning approach to Q-learning. Value-based methods address the issues that policy-based techniques present regarding local optima convergence and high variance. Unfortunately, while value-based techniques can lead to optimal solutions, they may suffer from bias and convergence issues on high dimensional feature spaces, i.e., *the curse of dimensionality* (RUSSELL; NORVIG, 2009).

Actor-critic methods emerged as a hybrid attempt to address the weaknesses of value-based and policy-based approaches. Learning policy functions (the actor) and value functions (the critic) allows the agent to balance bias and variance and achieve outstanding results. For instance, Mnih et al. (2016) proposed the Advantage Actor-Critic (A3C) algorithm and showed that it outperformed the, at the time, state-of-the-art DQN in the video game domain (MNIH et al., 2015). Unsurprisingly, there is a growing interest in the financial domain regarding the actor-critical model, with works using the Deep Deterministic Policy Gradient (DDPG) (KANG; ZHOU; KANG, 2018; YU et al., 2019; YE et al., 2020), proximal policy optimization (PPO) (HIRCHOUA; OUHBI; FRIKH, 2021), and A3C (LI; ZHENG; ZHENG, 2019; PONOMAREV; OSELEDETS; CICHOCKI, 2019). Notably, a few years later, after Mnih et al. (2016) compared the A3C model to the DQN algorithm in the game domain, Li, Zheng and Zheng (2019) did a similar comparative experiment for the active trading task and similarly observed the exceptional performance of the A3C algorithm. Thus, inspired by these remarkable results, the present study adopts a variation of A3C, the *Advantage Actor-Critical* (A2C) (MNIH et al., 2016). According to Wu et al. (2017), A2C is a less explored but equally effective variety of A3C. The main difference between A2C and A3C is that the former does not have the asynchronous part of the A3C model, which consists of several independent agents interacting with a different copy of the environment in parallel.

A crucial step in RL system design is selecting features to compose the market information (i.e., state) that the agent observes to take action. Most RL researchers favored information extraction through technical indicators generated from statistical preprocessing of the price time series of assets. These indicators include the moving average (EILERS et al., 2014; LI; ZHENG; ZHENG, 2019; WU et al., 2019), relative strength index (ZHANG; MARINGER, 2016; SPOONER et al., 2018), stochastic oscillator (ALIMORADI; HUSSEINZADEH KASHAN, 2018; GIACOMAZZI DANTAS; GUERREIRO E SILVA, 2018), and others. Nonetheless, a prevalent technical indicator is the normalization of price time series by taking the difference between consecutive price values. Moody and Wu (1997) first introduced a state composition built upon this normalized time series, which is still very popular among deep RL trading approaches (DENG et al., 2017; ALMAHDI; YANG, 2017; WANG et al., 2019; ABOUSSALAH; LEE, 2020). Deep learning allows researchers to reduce noise through neural networks' internal layers, diminishing the necessity for intense preprocessing. For instance, Deng et al. (2017) employed a fuzzy neural network internal layer for state representation encoding to reduce asset price noise.

In contrast to technical indicators, fundamental indicators comprise features extracted from sources external to the price series, broadening the spectrum of available information sources to include balance sheets, macroeconomic analysis, financial news, and others. Case in point, Eilers et al. (2014) adopted information regarding the trading period, which may anticipate market trends that can occur periodically next to events such as the turn-of-the-month (ARIEL, 1987), government announcements (LUCCA; MOENCH, 2015), and others (e.g., exchange holidays). On the other hand, some studies (WANG et al., 2019; MA et al., 2021) adopted company indicators such as market capitalization, price-earnings ratio, dividend, and others. Incorporating fundamental indicators generated from NLP techniques over textual sources, including SA methods is a much-explored approach for market forecasting (KHADJEH-NASSIRTOUSSI et al., 2014; LOUGHRAN; MCDONALD, 2016; HENRIQUE; SOBREIRO; KIMURA, 2019). Interestingly, only recently have RL authors explored such a research path (FEUERRIEGEL; PRENDINGER, 2016; YANG; YU; ALMAHDI, 2018; YE et al., 2020).

Choosing the source of textual data impacts the trading problem scope by restricting the frequency of operations. For instance, some SA researchers (REKABSAZ et al., 2017; FEUERRIEGEL; GORDON, 2019) selected companies' periodic exchange filings (e.g., Securities and Exchange Commission filings), which occur sparsely over months. On another side of the spectrum, social media allows high-frequency trading and facilitates capturing large amounts of data, which attracted considerable attention (BOLLEN; MAO; ZENG, 2011; NGUYEN; SHIRAI, 2015; JIANG; LAN; WU, 2017; XING; CAMBRIA; WELSCH, 2018). Unfortunately, preprocessing social media content is more challenging than company filings and may come from less credible writers (LOUGHRAN; MCDONALD, 2016). News headlines appear as an intermediary source of information that balances frequency of availability and reliability and, naturally, also attracted considerable awareness from the SA community (KHADJEH-NASSIRTOUSSI et al., 2015; DUAN et al., 2018; REN; WU; LIU, 2019; GLASSERMAN et al., 2020). The current work aims at designing an hourly frequency trading system, and thus the properties of news headlines favored its selection. Interestingly, all RL studies that employ NLP techniques (FEUERRIEGEL; PRENDINGER, 2016; YANG; YU; ALMAHDI, 2018; YE et al., 2020) also adopted news headlines as their sources.

According to Medhat, Hassan and Korashy (2014), textual information extraction can occur through *lexicons* (i.e., a dictionary with the corresponding sentiment of words), machine learning models, or a hybrid combination of both. Lexicons can be generic and support various tasks from various domains (HUTTO; GILBERT, 2014). Nonetheless, some researchers defend using domain-specific lexicons, such as Loughran and McDonald (2011), who proposed a financial lexicon with positive, negative, and neutral sentiment dimensions for financial words. One of the most influential studies that use NLP techniques for market prediction (BOLLEN; MAO; ZENG, 2011) devised a lexicon that categorized words according to six sentiment dimensions. Regarding machine learning approaches, there is a recent focus (PINHEIRO; DRAS, 2017; DUAN et al., 2018; XING; CAMBRIA;

WELSCH, 2018) on adopting deep neural networks with state-of-the-art latent feature representation, such as the GloVe word embedding (PENNINGTON; SOCHER; MANNING, 2014). Notably, some current state-of-the-art architectures for sentiment extraction from textual financial sources adopt the hybrid combination of lexicons and machine learning to extract sentiment for market prediction (MANSAR et al., 2017; JIANG; LAN; WU, 2017).

Amid RL studies in the financial domain to explore NLP techniques, Feuerriegel and Prendinger (2016) adopted a Q-learning algorithm and explored composing the market state representation with SA features and normalized price time series. To generate these fundamental indicators, Feuerriegel and Prendinger (2016) employed a lexicon-based approach to extract word-level sentiment to compose the overall sentiment score of news headlines. Alternatively, Yang, Yu and Almahdi (2018) aimed at first refining sentiment features by extracting the underlying relation between investors' sentiment and market trends. Therefore, Yang, Yu and Almahdi (2018) produced sentiment-charged reward features using the asset's price time series and news sentiment scores, from a third-party provider[1], through an inverse RL method. Nevertheless, Yang, Yu and Almahdi (2018) did not employ these sentiment reward features for learning a trading strategy and instead adopted them as input for a rule-based strategy that relied on supervised learning predictions. Finally, Ye et al. (2020) adopted a deep learning-based approach to extract word embeddings from financial news and subsequently generate market forecasts for a portfolio management problem. Then, these features and predictions served to augment the market's state representation. Noticeably, none of these RL works (FEUERRIEGEL; PRENDINGER, 2016; YANG; YU; ALMAHDI, 2018; YE et al., 2020) employed hybrid textual information extraction (i.e., lexicon and ML model), a missed opportunity that the present work approaches.

Like Feuerriegel and Prendinger (2016) and Ye et al. (2020), the present work focuses on improving market state representation with textual news features. However, explicit sentiment information is used instead of using implicit latent features (YE et al., 2020). Following this approach facilitates examining the influence of extracted features on the systems' behavior and, consequently, its impact on the financial return (GLASSERMAN et al., 2020). Nonetheless, developing a sentiment extractor can be challenging, and one contributing factor to this difficulty is the scarcity of labeled data for training ML models (KHADJEH-NASSIRTOUSSI et al., 2014). As a workaround, researchers might gather

[1] Thomson Reuters News Analytics (TRNA): https://fsc.stevens.edu/thomson-reuters-news-analytics-trna

sentiment data labeled by users in financial social media platforms (e.g., Stocktwits), which can be biased or unreliable (XING; CAMBRIA; WELSCH, 2018).

Fortunately, researchers in the SA and financial communities joined forces to address these issues by producing the labeled gold-standard dataset for the SemEval-2017 Task 5 competition (DAVIS et al., 2016; CORTIS et al., 2017). SemEval-2017 Task 5 encompassed two financial challenges, one with data composed of social network tweets and the other with news headlines. In both challenges, researchers would compete to devise sentiment extractors for accurately matching the sentiment scores of texts according to market specialists' labeling. Although researchers displayed several promising approaches regarding the news headlines challenge, Fortia-FBK (MANSAR et al., 2017) stood out as the state-of-the-art winner design. In their hybrid approach, Mansar et al. (2017) combined the GloVe pre-trained word vectors (PENNINGTON; SOCHER; MANNING, 2014) with the general-domain VADER lexicon (HUTTO; GILBERT, 2014) into a *convolutional neural network* (CNN) (KIM, 2014). Then, past exploratory work (FERREIRA et al., 2020) experimented with different Fortia-FBK configurations to examine improvements to the sentiment scoring according to the gold-standard reference data. The present work adopts a Fortia-FBK-based design as its sentiment extractor with its optimal configuration, as described by Ferreira et al. (2020).

Finally, Table 1 concludes this chapter by depicting a brief comparison between the proposed system (ITS-SentARL) and some of the RL studies discussed here. Characteristics depicted in Table 1 include the type of problem (e.g., either SAT or PM), data sampling frequency (e.g., Minute, Day), category of information used (e.g., Technical and Fundamental indicators), NLP technique employed (e.g., state-of-the-art ML-based sentiment extractor) and the RL algorithm employed (e.g., A2C, DQN).

Table 1: Overview of some RL works for easier comparison with the proposed architecture ITS-SentARL. First, among analyzed characteristics, the type of problem refers to either single asset trading (SAT) or portfolio management (PM). Second, data sampling reflects the availability of data which impacts trading frequency. Next, the variety of indicators and NLP techniques show the most common category of information. Finally, the last column informs the employed RL technique.

Authors	**Type**	**Data Sampling**	**Indicators**	**NLP method**	**RL algorithm**
Eilers et al. (2014)	SAT	Daily	Tech.+ Fund.	No	TD(0)
Gabrielsson and Johansson (2015)	SAT	Minute	Technical	No	Recurrent RL
Feuerriegel and Prendinger (2016)	SAT	Daily	Tech.+ Fund.	Sentiment Lexicon	Q-learning
Deng et al. (2017)	SAT	Minute, Daily	Technical	No	Fuzzy Deep Recurrent RL
Almahdi and Yang (2017)	PM	Weekly	Technical	No	Recurrent RL
Carapuço, Neves and Horta (2018)	SAT	Hour	Technical	No	DQN
Giacomazzi Dantas and Guerreiro e Silva (2018)	SAT	Daily	Technical	No	Q-Learning
Yang, Yu and Almahdi (2018)	SAT	15 min.	Tech.+ Fund.	Sentiment charged reward	Inverse RL
Li, Zheng and Zheng (2019)	SAT	Minute	Technical	No	A3C, DQN
Wang et al. (2019)	PM	Monthly	Tech.+ Fund.	No	Policy Gradient
Aboussalah and Lee (2020)	PM	Hour	Technical	No	Stacked Deep Dynamic Recurrent RL
Ye et al. (2020)	PM	Daily, 30 min.	Tech.+ Fund.	Word embedding	DDPG
Weng et al. (2020)	PM	30 min.	Technical	No	Policy Gradient
Ma et al. (2021)	SAT	Daily	Tech.+ Fund.	No	Double DQN
Théate and Ernst (2021)	SAT	Daily	Technical	No	DQN
ITS-SentARL	SAT	Hour	Tech.+ Fund.	SOTA Sent. ML Model	A2C

Source: Author's own production.

3 BACKGROUND AND PROBLEM DEFINITION

This chapter describes the theoretical background required to formulate trading activities as a sequential decision-making problem. The initial section introduces the essential financial market concepts for understating the complexity of market investment. Next are details about the trading scenario, available information, restrictions, and necessary simplifications. Ultimately, the last sections discuss the Markovian formulation or MDP.

3.1 Financial Concepts

There are several types of assets and ways to invest in financial markets. The most well-known type of asset is companies' shares or *stocks*, which represent a small fraction of company ownership. The negotiation of stocks occurs at marketplaces known as stock exchanges (e.g., New York Stock Exchange). Although the term *stock market* refers to the combination of all *stock exchanges*, these two designations are popularly used interchangeably. Curiously, although its name might suggest otherwise, participants might exchange several types of assets in stock markets, not only stocks. For instance, an *Exchange-Traded Fund* (ETF) is an asset that tracks the value of a *market index* and can be traded like a typical stock in stock markets. Market indexes represent the overall combined price of a portfolio of companies' stocks. For example, the S&P 500 (Standard & Poor 500) is an index of the combined price of the top five hundred companies with the highest stock price, traded volume, and that meets given selection criteria. Although an investor can trade some indexes directly, buying an index indirectly through an ETF is more usual.

Assets' prices follow the supply and demand law given by the number of shares available in the stock market and how much participants are willing to pay for them. Naturally, this willingness varies according to how stock market participants perceive companies' intrinsic value. As a company grows, its attractiveness to investors increases, driving its stock price up. To grow even further, companies can raise more capital for investments by releasing extra stocks, which increases the asset volume available for public investing. As a result, an asset's *liquidity* (i.e., the readiness of buying or selling) is directly affected by its attractiveness and shares' volume in the market. Hence, stocks price oscillate depending on the companies' current and expected future performance and growth. Assessing current value is relatively straightforward given the transparency that market exchanges require from companies when publicly displaying their financial reports and announcements to shareholders. However, estimating companies' expected future performance is a non-trivial effort based on subjective assumptions and speculation. Even so, the economy is expected to grow over time, and thus, the most traditional way of investing is to *buy-and-hold* (BH) stocks over extended periods (e.g., months, years, or decades), even though they could undergo unexpected price oscillations in the short term. Opposite to traditional BH investors, active traders leverage the uncertainty that arises from the market's speculative nature and how it affects participants' immediate perception and behavior. Thus, for active trading, investors focus on the short-term price fluctuation for increased gains even though it might incur a higher exposure to risk.

As already mentioned, there is an open discussion about whether it is feasible to attain profit regularly with active trading, given the seemingly random characteristics of the market (KING; COOTNER, 1965; FAMA, 1965). Also, Fama (1970) proposed the *Efficient Market Hypothesis* (EMH) that included three forms (i.e., weak, semi-strong, and strong) that stated that the asset price could encompass all information available to market participants. Moreover, market efficiency relates to overall companies' liquidity which subsequently relates to market size. In this sense, the EMH's strong form ensured that efficient markets would be unpredictable, and unobserved information would not exist for exploitation. Interestingly, even though the EMH supporters back BH investors' practices, not all BH investors defend the EMH premises. For instance, some BH investors (e.g., Warren Buffet, Peter Lynch, and others) – popularly known for attaining outstanding profit consistently over decades – employ fundamental indicators (e.g., earnings per share, price to earnings ratio and others) for estimating companies' intrinsic value and then selecting stocks deemed underpriced.

Furthermore, there is both conceptual and empirical criticism of EMH. For example, according to Behavioral economists – who study the psychology of how market participants make decisions – EMH does not account for the fact that investors measure and perceive companies' potential differently and may come up with different evaluations. Also, some market anomalies, such as the *calendar effect* (e.g., turn-of-the-month), are still hard to explain under EMH and can be exploited (VASILEIOU, 2013). Subsequently, the Adaptive Market Hypothesis (AMH) reconciled EMH with behavioral economics and helped justify anomalies. Essentially, Lo (2004) proposed that evolutionary models explain investors' behavior by describing how humans could shape market efficiency by iterating strategies through trial and error. In this sense, automated trading might be exploiting market inefficiencies but also, at the same time, filling the opportunity gaps and, consequently, increasing market efficiency.

Behavioral economics questions the EMH idea of the *rational investor*, and one of its particularly notorious fields explores market or investors' sentiment. In this regard, *bullish* market sentiment is equivalent to a general optimistic expectation that prices will increase. Oppositely, *bearish* sentiment is a pessimistic view that prices will go down. Market sentiment researchers have opposed EMH with examples where investors presented excessive bullish or bearish sentiment, which hypothetically led specific stock prices to get much higher (overpriced) or lower (underpriced) their intrinsic value. Barberis, Shleifer and Vishny (1998) were the first authors to show that the market can irrationally overreact or underreact to the news. These authors also argued that overpricing or underpricing caused by prolonged excess bullish or bearish sentiment can eventually cause a price *correction* in the form of a price mean reversion. Nonetheless, it is essential to notice that assets possess varying degrees of susceptibility to investors' sentiment (BAKER; WURGLER, 2007). Interestingly, the characteristics that define this asset's susceptibility are related to those proposed by EMH, such as liquidity or traded volume.

News traders are investors that embody behavioral economics research principles. This type of trader recognizes that rumors can build up expectations that cause assets to become mispriced. Then, the realization or not of the fact may trigger price corrections that lead to trends moving in an opposing direction. Such situations are typical and can be exploited by news traders (practice summarized by the adage "buy the rumor, sell the news" (KADAN; MICHAELY; MOULTON, 2014)). In particular, traders can profit from overreactions caused by disruptive events, especially the unexpected ones that are sometimes known as *black swan*. Taleb (2007) characterized the black swan as such a surprisingly rare event that: first, most observers could not even propose estimating its chance of occurrence; second, it inflicts a disastrous effect when it happens; third, it is justified as foreseeable in hindsight. Examples of such events include the 2000 *dot-com bubble* and the 2008 housing crisis. While the effects of the COVID-19 pandemic can arguably not be considered black swan events – even by the author that coined this term (TALEB, 2007) – the exact moment of the economic impact was unknown, the effects were catastrophic, and its duration uncertain.

It is also notable that the uncertainty in the effect duration of most events, either devastating or not, can also be exploited by news traders using a strategy that aims to seize the opportunity to invest contrary to the fading sentiment trend. Alternatively, Barberis and Thaler (2003) exposed that optimism or pessimism can be well-founded. Therefore, a well-founded sentiment can explain when asset prices do not oscillate sharply after the expected fact: the excitement was steadily incorporated into the price over time.

In essence, news traders examine the market mood to exploit asset mispricing opportunities. Eventually, an adequate market mood assessment can be used to secure a good position over an appropriately priced asset or profit from sharp price swings caused by sentiment and price mismatches.

3.2 Trading Scenario

Each market participant possesses a given *net worth* which is the term that describes the total amount of an investor's wallet value (e.g., cash, stocks). Naturally, investors aim to increase their net worth by appropriately selecting assets for investing with a higher potential of yielding good financial returns. The selection of assets an investor possesses is known as a *portfolio* or *wallet*. When examining trading tasks, researchers can focus on portfolio management (PM) tasks where an investor seeks higher profit by redistributing the amount of owned stocks in its wallet. The present work focus on single asset trading (SAT), which is the specific case of PM, where the trader operates assets in a separated manner, and thus the position over one asset does not affect others in the wallet. The most usual asset positions (i.e., operations) are *long*, *short*, and *neutral*. First, the long operation is equivalent to buying an asset so that profit occurs if prices increase. Next, the short operation borrows assets in the market with the expectation that prices will go down. Finally, the neutral operation is similar to selling or staying out of the market, thus, remaining unaffected by its oscillation. When performing these operations, traders can select a variable finite amount of shares.

After an investor decides on which operation to perform (e.g., long, short, neutral), it places an ask or bid *order* that reflects its decision. An order contains the number of shares to trade and its selling (i.e., ask) or buying (i.e., bid) the desired price. The *spread* is the difference between the ask and bid prices and can relate to the liquidity of a given asset. In this sense, big stable companies with high-liquidity stocks tend to present a low spread. On the other hand, low-liquidity stocks will usually exhibit a higher spread. Finally, an order is fulfilled, and assets are effectively traded after the same volume of shares of the bid and ask prices of different investors match or cross (i.e., the price the buyer is bidding or the seller is asking reaches or traverses each other). Ultimately, all fulfilled orders account for the historical aggregated data of an asset.

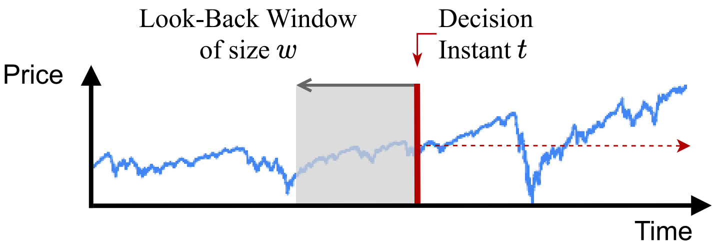
In essence, traders devise strategies to maximize their net worth over the long term and rely on different market information to define the supposedly best operation to take on a given instant. Traders can sample information with different frequencies according to the desired operation frequency (e.g., minute, hour, day). Moreover, the availability of price information allows for high-frequency trading (e.g., operations at each second). At any observed period that includes minutes, hours, days, and others, there is a starting price (i.e., open) that any investor paid for (i.e., bid) or sold (i.e., ask) a given asset. Moreover, during such a given period, both bid and ask prices achieved maximum (i.e., high) and minimum (i.e., low) values before reaching a final price (i.e., close) at the end of the observed period. Ultimately, volume is the number of asset shares traded in the market during any given period. Usually, active traders gather the historical bid and ask traded volumes and OHLC (i.e., open, high, low, close) prices as input to formulating strategies. Although, in practice, it is a popular convention among both ML works to only use an asset's bid or the ask prices as information sources.

In the case of behavioral economics adherents such as news traders, there could be a scarcity of information even for high-liquidity companies, where many hours could pass before news vehicles publish even a single news article. Notice, for example, that quarterly companies performance reports are still a common source of information for news traders (REKABSAZ et al., 2017; FEUERRIEGEL; GORDON, 2019). Thus, the sparsity between news releases regarding companies may hinder the maximum frequency news traders could operate. Not surprisingly, it is becoming more common for news traders to include social media (e.g., Twitter posts) as sources of information for supporting trading operation decisions (BOLLEN; MAO; ZENG, 2011; DAVIS et al., 2016; CORTIS et al., 2017).

The market environment assumes that operations can incur trading costs (TC) that can drastically affect the financial return and penalize traders for frequent position shifts. Consequently, due to the chaotic nature of the stock market (TSAY, 2010), trying to devise a system that accurately forecasts near-future trends may lead to a *prediction pitfall*. In this regard, every prediction error in a system that frequently changes positions on an asset can be extremely costly in a real-world scenario. Hence, as TC can drastically impact results, researchers should consider it essential in their experiments.

When traders operate, they constantly analyze historical data, such as the price time series, to obtain indicators for decision-making. Nonetheless, recent data points might be more informative than older ones. Thus, a trader or an intelligent trading system observes processes past w points to take a trading action at a trading decision instant t. The look-back window is this *time-window* $[t, ..., t - w + 1]$ that sequentially moves forward in time. Hence, Figure 2 portrays an example of a look-back window of size w over a price time series. The look-back window is an important concept that is employed throughout this manuscript.

Figure 2: Example of a look-back window of size w over a price time series. Given a decision instant t that is always moving forward in time (in red), the look-back window is an excerpt of the last w data points (a shadowed grey area) in the time series.



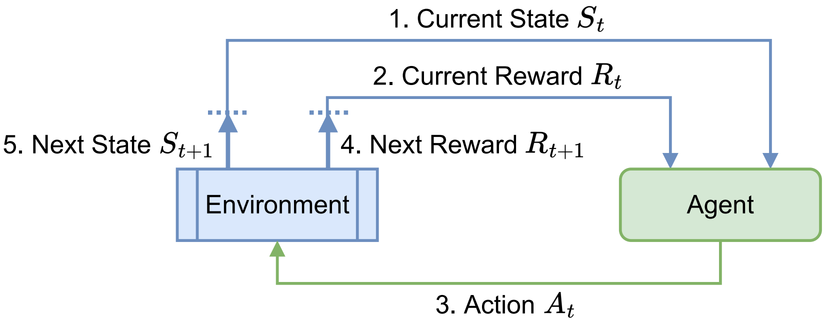
Source: Author's own production.

3.3 The Markov Decision Process Formulation

The present work formulates the single asset trading problem as a Markov decision process (MDP) given by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$ where $\mathcal{S}$ is the set of states (i.e., conditions) of the environment, and $\mathcal{A}$ is the set of actions (i.e., operations) available to the agent. The state transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A} \mapsto P(\mathcal{S})$ describes the probability of transitioning from states after a given action and describes the environment dynamics stochasticity. $\mathcal{R} : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ is the function of the reward received for taking actions at given states.

The interaction of an agent in the MDP formulation is represented in Figure 3. First, the agent, at an instant t, observes the current state $S_t = s \in \mathcal{S}$, reward $R_t \in \mathcal{R}$, and decides to perform an action $A_t = a \in \mathcal{A}$. Then, the agent receives a reward $R_{t+1} \in \mathcal{R}$ and transitions to the next state $S_{t+1} = s' \in \mathcal{S}$ with a given probability $P(S_{t+1})$. This process repeats during each episode that comprises a period with timesteps that start at $t = 0$ until $t = T$. There are several possible methods for solving the MDP, and Reinforcement Learning (RL) techniques, discussed in the next Section, are some of the most popular ones.

Figure 3: The MDP dynamic. Initially, the agent starts with information about the current environment state S_t and reward R_t. Next, it selects an action A_t to take. Then, it receives a reward R_{t+1} and transitions to the next state S_{t+1}.



Source: Author's own production.

Before proceeding, observe that this work draws upon the symbols and formulations of a seminal material in the field (SUTTON; BARTO, 2018). As such, instead of using R_t to represent the reward resulting from action A_t, it is better to use R_{t+1}, highlighting that the following reward and state, namely R_{t+1} and S_{t+1}, are determined together. Consequently, from now on, to guarantee the consistency of symbols and formulations, only the representation R_{t+1} is employed.

4 SOLVING THE PROBLEM: ITS-SENTARL

Recall from Section 3.2 the analysis of the prediction pitfall and the risks of predicting precisely every price change in an asset. For instance, this problem can arise when tracking a rule-based strategy that matches ML predictions causes a trading system to frequently change positions on an asset, which can be costly in a real-world scenario. Consequently, it is essential to investigate an agent's performance under both no-penalty and high-penalty scenarios. Under such conditions, Deng et al. (2017) showed that defining an architecture for learning a trading strategy that identifies market momentum can circumvent the prediction pitfall. Additionally, the market's high stochasticity causes the agent to experience many distinguished situations among training and testing environments, leading to poor performance due to generalization issues. Hence, under these circumstances, the present trading agent is an autonomous system that interacts with the environment with the ultimate goal of learning a trading strategy that consistently maximizes the accumulated total financial return over a period.

In essence, formulating the SAT formulation as an MDP allows an RL algorithm to learn a strategy for maximizing trading profits by interacting with the environment on a trial and error basis. Thus, this chapter describes the proposed Sentiment-Aware Reinforcement Learning Intelligent Trading System (ITS-SentARL). The first section shows how reinforcement learning techniques can solve the MDP. Then, the second section exposes the mapping of the trading task to the MDP formulation and defines the base sentiment-free architecture. Next, the following section shows how the market sentiment was extracted and subsequently incorporated into the market state representation. Also, in the last section, there are details about the state-of-the-art sentiment extractor employed.

4.1 A Reinforcement Leaning Approach

As mentioned in Section 3.3, the MDP describes a continuous process of interaction of an agent with a given environment. In this scenario, the agent employs a policy (i.e., strategy) to decide which action A_t to select when in state S_t. In essence, a policy is a mapping from states to actions that can be either deterministic $\mu : \mathcal{S} \mapsto \mathcal{A}$ or stochastic $\pi : \mathcal{S} \mapsto P(\mathcal{A})$, where $P(\mathcal{A})$ is the probability over the action set. The present work adopts a stochastic policy π. Ultimately, the agent's goal is to learn a policy, by trial and error, that maximizes the expected discounted accumulated reward[1] G_t from the current instant t onward given by

$$\begin{aligned} G_t &\doteq R_{t+1} + \gamma G_{t+1} \\ &= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots \\ &= \sum_{k=0}^{T-t} \gamma^k R_{t+k+1}, \end{aligned} \tag{4.1}$$

where $\gamma = \in [0, 1]$ is the discount factor that balances the impact between immediate and possible future rewards (SUTTON; BARTO, 2018).

MDPs can be solved by Reinforcement Learning (RL) techniques. However, there are several possible RL methods. In the present research, the RL algorithm employed is the A2C, the synchronous version of the Asynchronous Advantage Actor-Critic (A3C) (MNIH et al., 2016). Value-based techniques suffer from poor convergence, and policy-based techniques tend to converge to local maxima and suffer from high variance and sample inefficiency. Thus, the actor-critic methods aim to reduce the disadvantages of both. Hence, it is essential to understand how these methods work and how they can be combined to achieve better performance.

Value-based techniques find policies that solve the MDP indirectly through value functions that determine the best policy π to follow. For instance, given that an agent follows a policy π, the value of being in a state $S_t = s \in \mathcal{S}$ is given by the state value function

$$\begin{aligned} V^{\pi}(s) &\doteq \mathbb{E}[G_t | S_t = s] \\ &= \mathbb{E}[R_{t+1} + \gamma G_{t+1} | S_t = s] \\ &= \mathbb{E}[R_{t+1} + \gamma V^{\pi}(S_{t+1}) | S_t = s], \end{aligned} \tag{4.2}$$

with the expected cumulative future reward G_t given by Eq. 4.1. Similarly, the state-

[1] The expected discounted accumulated reward G_t is usually referred to in the canonical RL literature as *return*. However, the term *return* indicates the financial return in the present context.

action value function

$$\begin{aligned} Q^{\pi}(s,a) &\doteq \mathbb{E}[G_t|S_t=s,A_t=a] \\ &= \mathbb{E}[R_{t+1}+\gamma V^{\pi}(S_{t+1})|S_t=s,A_t=a], \end{aligned} \tag{4.3}$$

represents the value of selecting an action $A_t = a \in \mathcal{A}$ when in state $S_t = s \in \mathcal{S}$, given that a policy π is being followed. Moreover, from Eq. 4.3, it is possible to see how the state and state-action value functions are related. More importantly, the relation given by the *Bellman optimality equation* $V^*(s) = \max\limits_{a\in\mathcal{A}} Q^{\pi^*}(s,a)$ expresses that the optimal policy π^* is the one that selects the most rewarding action according to the optimal state value function $V^*(s)$. Hence, finding an optimal value function $V^*(s) \doteq \max\limits_{\pi} V^{\pi}(s)$ is equivalent to finding an optimal policy π^*.

Temporal difference (TD) algorithms allow finding optimal value functions. TD methods start with an initial estimate of the value function (i.e., *bootstrapping*) and then update its estimates towards the true value by interacting with the environment and collecting samples. Thus, TD methods learn the true value functions by using the TD error

$$\delta_t \doteq R_{t+1} + \gamma V(S_{t+1}) - V(S_t), \tag{4.4}$$

representing the difference between the observed value of transitioning to a new state $R_{t+1} + \gamma V(S_{t+1})$, and previous estimates $V(S_t)$. Furthermore, because of this bootstrapping that relies on initial guessed estimates of the true value function, TD methods are considered biased. Now, consider a parameterized state value function $V(s;\boldsymbol{\vartheta})$ with the parameter vector $\boldsymbol{\vartheta} \in \mathbb{R}^u$, where u is the vector's dimension. The TD algorithm can iteratively find the parameters $\boldsymbol{\vartheta}$ that approximates the optimal value function, $\max\limits_{\boldsymbol{\vartheta}} V(s;\boldsymbol{\vartheta}) \approx V^*(s)$, by minimizing the mean-squared TD error given by a sequence of i loss functions

$$L_i(\boldsymbol{\vartheta}_i) = \mathbb{E}\left[R_{t+1} + \gamma V\left(s';\boldsymbol{\vartheta}_{i-1}\right) - V\left(s;\boldsymbol{\vartheta}_i\right)\right]^2 \tag{4.5}$$

where $s' \in \mathcal{S}$ represents the set next state. In this equation, the estimated value of the following state s' is computed using the parameters from the previous iteration, $\boldsymbol{\vartheta}_{i-1}$, because the value of the state s' is not observed directly but instead is estimated using the reward received at the next time step R_{t+1} and the estimated value of the next state. On the other hand, the estimated value of the current state s is computed using the current set of parameters $\boldsymbol{\vartheta}_i$ since this ultimately is the value to improve through learning.

Unlike value-based, policy-based techniques learn policies directly by favoring actions that maximize the expected future accumulated reward $\mathbb{E}[G_t]$. Let $\pi(s;\boldsymbol{\theta})$ denote the parameterized policy function where the parameter vector $\boldsymbol{\theta} \in \mathbb{R}^d$, and d is the vector's dimension. Then, an agent that follows a policy π at a given time step t, while in state $S_t = s$, selects an action according to the following formulation, $A_t = a \sim \pi(S_t; \boldsymbol{\theta})$ with $a \in \mathcal{A}$. Afterward, a given policy's objective function or performance measure can be written as the expected future cumulative reward (SUTTON; BARTO, 2018)

$$J^{\pi}(\boldsymbol{\theta}_t) \propto Q^{\pi}(S_t, A_t) = \mathbb{E}[G_t | S_t, A_t]. \tag{4.6}$$

Eventually, to find an adequate solution, a policy gradient algorithm learns a policy by making small gradient ascent updates that adjust the parameters $\boldsymbol{\theta}$ in the direction that maximizes the performance measure

$$\boldsymbol{\theta}_{t+1} \doteq \boldsymbol{\theta}_t + \alpha \cdot \nabla J^{\pi}(\boldsymbol{\theta}_t),$$

where $\alpha \in \mathbb{R}$ is the gradient's step size known as the learning rate hyperparameter. Among these techniques, the Monte Carlo policy gradient or REINFORCE (WILLIAMS, 1992) is a well-known, established algorithm. REINFORCE solves this equation by executing the gradient over the action taken at a given instant with

$$\nabla J^{\pi}(\boldsymbol{\theta}) = \mathbb{E}\left[G_t \frac{\nabla \pi(A_t | S_t, \boldsymbol{\theta})}{\pi(A_t | S_t, \boldsymbol{\theta})} \right],$$

where $\pi(A_t | S_t, \boldsymbol{\theta})$ represents the policy inclination to select an action A_t at a state S_t given a parameter vector $\boldsymbol{\theta}$. Ultimately, combining previous formulations and instantiating $\boldsymbol{\theta}$ to instant t leads to the solution below

$$\begin{aligned} \boldsymbol{\theta}_{t+1} &\doteq \boldsymbol{\theta}_t + \alpha \cdot \nabla J^{\pi}(\boldsymbol{\theta}_t) \\ &= \boldsymbol{\theta}_t + \alpha \cdot G_t \frac{\nabla \pi(A_t | S_t, \boldsymbol{\theta}_t)}{\pi(A_t | S_t, \boldsymbol{\theta}_t)}. \end{aligned} \tag{4.7}$$

However, given that a typical Monte Carlo method such as REINFORCE is episodic, an episode must be completed before the accumulated reward can serve to update the gradients. Hence, these methods present high variance since immediate rewards can be very distinct, and different policies may present the same accumulated rewards.

When proposing REINFORCE, Williams (1992) suggested that a reduction in variance to this policy gradient method is achievable, while keeping it unbiased, by subtracting a value, given by the baseline function $b_t(S_t)$, from the expected future accumulated reward G_t as follows

$$\boldsymbol{\theta}_{t+1} \doteq \boldsymbol{\theta}_t + \alpha \left(G_t - b(S_t) \right) \frac{\nabla \pi(A_t | S_t, \boldsymbol{\theta}_t)}{\pi(A_t | S_t, \boldsymbol{\theta}_t)}. \tag{4.8}$$

Then, by adopting the state value function as the baseline function $b(S_t) = V^{\pi}(S_t = s; \boldsymbol{\vartheta})$, Mnih et al. (2016) introduced the *advantage function*

$$\begin{aligned}\Lambda(A_t, S_t; \boldsymbol{\vartheta}) &\doteq G_t - V(S_t; \boldsymbol{\vartheta}) \\ &= Q(S_t, A_t) - V(S_t; \boldsymbol{\vartheta}) \\ &= R_{t+1} + \gamma V(S_{t+1}; \boldsymbol{\vartheta}) - V(S_t; \boldsymbol{\vartheta}).\end{aligned} \tag{4.9}$$

Thus, given the identity $\nabla \ln x = \dfrac{\nabla x}{x}$ (SUTTON; BARTO, 2018), the parameter update function for the A2C can be written as

$$\begin{aligned}\boldsymbol{\theta}_{t+1} &\doteq \boldsymbol{\theta}_t + \alpha \cdot \Lambda(A_t, S_t; \boldsymbol{\vartheta}) \frac{\nabla \pi(A_t | S_t, \boldsymbol{\theta}_t)}{\pi(A_t | S_t, \boldsymbol{\theta}_t)} \\ &= \boldsymbol{\theta}_t + \alpha \cdot \Lambda(A_t, S_t; \boldsymbol{\vartheta}) \nabla \ln \pi(A_t | S_t, \boldsymbol{\theta}_t).\end{aligned} \tag{4.10}$$

Silver (2015) notices that taking the gradient of the loss functions in Eq. 4.5 leads to the TD error in Eq. 4.4, an unbiased estimate of the advantage function in Eq. 4.9. Hence, the advantage function can be used to update both the policy function and value function parameters which means (making $i = t$). As such, the value function parameter update gradient can be written as

$$\begin{aligned}\boldsymbol{\vartheta}_{t+1} &\doteq \boldsymbol{\vartheta}_t + \alpha \cdot \nabla L_t(\boldsymbol{\vartheta}_t) V(S_t; \boldsymbol{\vartheta}_t) \\ &= \boldsymbol{\vartheta}_t + \alpha \cdot \Lambda(A_t, S_t; \boldsymbol{\vartheta}_t) \nabla V(S_t; \boldsymbol{\vartheta}_t).\end{aligned} \tag{4.11}$$

Moreover, similarly to Mnih et al. (2016), the present work adopts deep neural networks to parameter functions vectors $\boldsymbol{\theta} \in \mathbb{R}^d$ and $\boldsymbol{\vartheta} \in \mathbb{R}^u$. Subsequently, the policy $\pi(A_t|S_t; \boldsymbol{\theta})$ uses a multilayer perceptron (MLP) with a softmax output while the value function $V(S_t; \boldsymbol{\vartheta})$ adopts a linear output.

It is essential to notice that the final A2C algorithm employs an n-step parameter update where a gradient ensues after a given number of k steps (MNIH et al., 2016). Thus, A2C collects rewards for a k number of steps, calculates the gradient for each step, and then updates the parameters with the sum of these step gradients. Ultimately, the pseudo code that describes the complete A2C procedure is presented with Algorithm 1.

In conclusion, combining these policy-based and value-based actions to compose the hybrid actor-critic methods, such as the A2C, favors balancing its strengths and weaknesses. For instance, as already mentioned, even though policy-based methods tend to converge to suboptimal solutions (i.e., local minima), they also have better overall convergence properties and are more effective in high-dimensional spaces (i.e., *the curse of dimensionality* (RUSSELL; NORVIG, 2009)). Besides, regarding the bias-variance tradeoff, by bootstrapping, the *critic* (i.e., state value function) introduces a slight bias to

Algorithm 1 A2C training algorithm

```
Initialize actor policy function $\pi(S_t; \boldsymbol{\theta})$ with parameter vectors $\boldsymbol{\theta} \in \mathbb{R}^d$
Initialize critic value function $V^\pi(S_t; \boldsymbol{\vartheta})$ with parameter vectors $\boldsymbol{\vartheta} \in \mathbb{R}^u$
Initialize hyperparameters $\alpha$ (gradient step size), $\gamma$ (reward discount factor), and
$k \in \mathbb{Z}^+$ (max number of steps before the gradient update)
Set the evaluation for the last step of the value function $V(S_{t=T}; \boldsymbol{\vartheta}) \leftarrow 0$
while not last episode do
    Initialize $t \leftarrow 0$ (first time step)
    Initialize $S_t$ (first state)
    Initialize $d\boldsymbol{\theta} \leftarrow 0$ and $d\boldsymbol{\vartheta} \leftarrow 0$
    for $t \leq T$ do                                   ▷ $T$ is the last time step of the episode
        $A_t \leftarrow a \sim \pi(S_t; \boldsymbol{\theta})$                                ▷ Sampling action
        Take $A_t$, receive $R_{t+1}$ and transition to $S_{t+1}$
        $\Lambda(A_t, S_t; \boldsymbol{\vartheta}) \leftarrow R_{t+1} + \gamma V(S_{t+1}; \boldsymbol{\vartheta}) - V(S_t; \boldsymbol{\vartheta})$     ▷ Calculate advantage
        $d\boldsymbol{\theta} \leftarrow d\boldsymbol{\theta} + \alpha \cdot \Lambda(A_t, S_t; \boldsymbol{\vartheta}) \nabla \ln \pi(A_t | S_t, \boldsymbol{\theta}_t)$     ▷ Accumulating step gradient
        $d\boldsymbol{\vartheta} \leftarrow d\boldsymbol{\vartheta} + \alpha \cdot \Lambda(A_t, S_t; \boldsymbol{\vartheta}) \nabla V(S_t; \boldsymbol{\vartheta})$
        if $t \bmod k = 0$ then                         ▷ Update parameters every $k$ steps
            $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \cdot d\boldsymbol{\theta}$                 ▷ Policy function parameters update
            $\boldsymbol{\vartheta} \leftarrow \boldsymbol{\vartheta} + \alpha \cdot d\boldsymbol{\vartheta}$           ▷ State value function parameters update
            $d\boldsymbol{\theta} \leftarrow 0$ and $d\boldsymbol{\vartheta} \leftarrow 0$             ▷ Resetting cumulative gradients
```

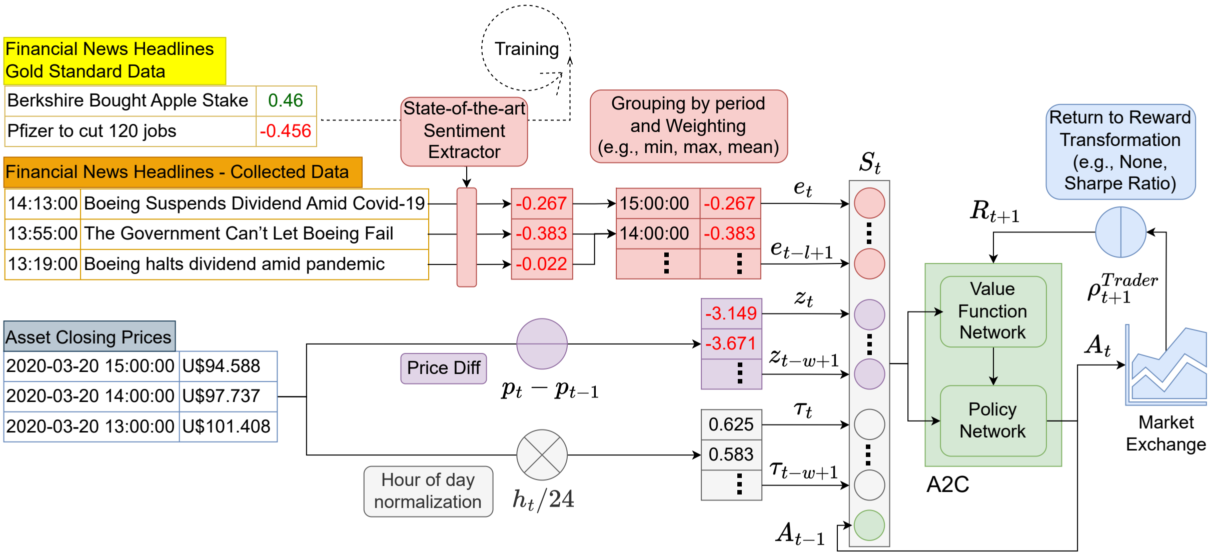
the *actor* (i.e., policy function), which helps address its high variance issues. In essence, the critic is doing policy evaluation by estimating a policy's potential given the current parameters. Subsequently, the actor follows the direction suggested by the critic. In this sense, the critic indicates how good the policy actions are compared to the expected state values. For these reasons, A2C was chosen in this work.

4.2 Base Architecture

ITS-SentARL presents an RL modular architecture that incorporates fundamental indicators from news headlines through an explicit market mood extraction preprocessing. This architecture depicted in Figure 4 achieves its sentiment-awareness by introducing past market sentiment features to the market state representation. Furthermore, this flexible architecture allows experimentation with several sentiment grouping methods (e.g., min, mean, max) and sentiment features' quantity. In addition, the other observable features that compose the market state include past assets' closing prices, hours of trading prices, and the immediate last action. Besides, the selection of the amount of each of these features can occur during experimentation. Lastly, ITS-SentARL employs an A2C algorithm that processes the state representation to select actions.

The present work adopts a popular convention among ML works (HENRIQUE; SO-

Figure 4: The Sentiment-Aware Reinforcement Learning Intelligent Trading System (ITS-SentARL) architecture. ITS-SentARL employs a news headline sentiment extractor trained with gold-standard data; the sentiment scores are then grouped by period to represent the market mood e_t in the last l instants. In this illustration, the grouping method is the minimum sentiment score among headlines in the period. Differences in previous asset's closing prices, z_t and the normalized time of each data point τ_t also compose the market state, along with the last action taken by the agent A_{t-1}. Based on S_t, A2C decides A_t and receives R_{t+1}, and the system transitions to state S_{t+1}.



Financial News Headlines
Gold Standard Data

Berkshire Bought Apple Stake	0.46
Pfizer to cut 120 jobs	-0.456

Financial News Headlines - Collected Data

14:13:00	Boeing Suspends Dividend Amid Covid-19
13:55:00	The Government Can't Let Boeing Fail
13:19:00	Boeing halts dividend amid pandemic

Asset Closing Prices

2020-03-20 15:00:00	U$94.588
2020-03-20 14:00:00	U$97.737
2020-03-20 13:00:00	U$101.408

Source: Lima Paiva et al. (2021).

BREIRO; KIMURA, 2019), where only an asset's bid closing prices are used as input information. This practice helps address correlation issues among input features and avoids overwhelming deep learning models with input features directly related to the assets' price. Besides, another crucial simplification assumes that given a selection of high-liquidity assets, trading operations will occur immediately in a decision instant t where $T > t >= 0 \in \mathbb{Z}^+$, and T is the last time step of an episode. Then, given the bid closing price p_t at instant t, the price difference z_t between two consecutive instants can be written as

$$z_t = p_t - p_{t-1}. \tag{4.12}$$

Moreover, much like a real-life trader, even though the agent starts trading at the beginning of an episode at instant $t = 0$, it observes features that represent the market scenario before the agent starts trading (i.e., $t < 0$). This procedure was described as look-back windows in Section 3.2, and it happens because of the necessity to observe past data time series (e.g., time-series vectors with indexes $[t, ..., t - w + 1]$). Consequently, even though $t \geq 0$, the agent, at the beginning of any episode, can receive observations regarding moments before it started trading (e.g., p_{-1}, z_{-1}). This type of circumstance appears throughout this section.

Regarding the input features, Spooner et al. (2018) proposed a state abstraction in which features are conceptually separated as they can represent either the *market-state* or the *agent-state*. Market-state features, such as price time series, technical or fundamental indicators, refer to the situation of the market environment and, thus, can not be affected by a single trading investor. Alternatively, agent-state features are the ones that the agent can have some degree of control over, including the amount of capital available, asset inventory size, or last assumed position over an asset. Subsequently, the complete state representation can include any combination of these two types of states.

Initially, the market-state composition, S_t^M, includes only technical indicator features given by the concatenation of the price difference z_t and its corresponding hour of the day h_t time series,

$$S_t^M = [z_t, ..., z_{t-w+1}] \parallel [\tau_t, ..., \tau_{t-w+1}] \in \mathbb{R}^{w+w}, \tag{4.13}$$

considering a look-back window of size $w \in \mathbb{Z}^+$. Also, $\tau_t = h_t/24$ is the normalized hour of the day at the instant t in the 24-hour format.

Next, the trading operations available to the agent are formalized as the action set $\mathcal{A} = \{Long, Neutral, Short\} \doteq \{1, 0, -1\}$. In the present architecture, the last operation assumed by the agent $A_{t-1} \in \mathcal{A}$ is the only information it can influence and thus compose the agent-state, S_t^C, and so

$$S_t^C = [A_{t-1}]. \tag{4.14}$$

The introduction of this information into the state is one factor that can help stabilize an agent's position shifting (MOODY; WU, 1997; DENG et al., 2017). It is crucial to notice that the agent only starts trading at instant $t = 0$. Hence, the agent was outside the market before $t < 1$ and could take no actions implying that $A_{-1,...,-w} = 0$ (i.e., *Neutral* position).

Finally, the base-state representation $S_t^B \in \mathcal{S}$ is given by concatenating Eq. 4.13 and Eq. 4.14 as such

$$\begin{aligned} S_t^B &= S_t^M \parallel S_t^C \\ &= [z_t, ..., z_{t-w+1}] \parallel [\tau_t, ..., \tau_{t-w+1}] \parallel [A_{t-1}]. \end{aligned} \tag{4.15}$$

This base-state representation S_t^B is the one used by the baseline sentiment-free version of ITS-SentARL named here with the acronym *No Sent. A2C*. In the next section this state receives the additional market information regarding market sentiment features.

Assessing an agent's performance requires that financial return calculations include the absolute nominal return from price movements and deductions from the TC. Moreover, in financial scenarios, a trader's position in the market during price oscillations, meaning both current A_t and previous action A_{t-1} are relevant. Subsequently, a nominal return (i.e., undiscounted return) which represents the absolute amount of capital earned after taking action A_t, is given by the multiplication

$$\rho_{t+1}^{Nominal} = \varphi z_{t+1} A_t,$$

where φ represents the fixed amount and z_{t+1} the price difference from t to $t+1$. Then, return deductions occur if the agent selects an action that is different from the previous one ($A_t \neq A_{t-1}$). This deduction is proportional to the number of shares traded and given by the product

$$\rho_{t+1}^{Deduction} = \varphi \xi |A_t - A_{t-1}|,$$

with the transaction cost $\xi \in \mathbb{R}$ as a percentage of the transaction value affected by the absolute value of the difference in trading position. Next, subtracting the transaction penalties from the nominal return produces the real trader return as

$$\begin{aligned} \rho_{t+1}^{Trader} &= \rho_{t+1}^{Nominal} - \rho_{t+1}^{Deduction} \\ &= \varphi z_{t+1} A_{t-1} - \varphi \xi |A_t - A_{t-1}| \\ &= \varphi \left[z_{t+1} A_{t-1} - \xi |A_t - A_{t-1}| \right]. \end{aligned} \tag{4.16}$$

Ultimately, the present research adopts one straightforward and effective way to calculate the immediate reward in trading problems by making it equal to the financial return in Eq. 4.16, as follows

$$R_{t+1} = \rho_{t+1}^{Trader}. \tag{4.17}$$

Ultimately, notice that to keep consistency with the adopted reward formulation R_{t+1} (SUTTON; BARTO, 2018) across this manuscript, the financial return representation also describes future steps.

4.3 Market's Mood Incorporation

The sentiment extracted from news headlines is a fundamental indicator that helps the agent perceive more information about the market environment. Therefore, introducing sentiment features to the agents' market-state representation turns it into a sentiment-aware system. However, to fully capture the market sentiment momentum, the agent must observe multiple past sentiment features regarding the past period's sentiment, similar to the price time series. In addition, this sentiment time series should facilitate the agent verifying discrepancies between the price movement and the market sentiment that indicates if assets could be underpriced, overpriced, or at the appropriate value.

Traders can sample information with different frequencies according to the desired operation frequency (e.g., minute, hour, day). Price information availability allows for very high-frequency trading at each second. However, even for high-liquidity companies, many hours could pass before news vehicles publish even a single news article. Consequently, trading operations were restricted to an hourly frequency to guarantee adequate news coverage. Even so, the formulation described here could be applied to other operation frequencies. In short, the system will observe an asset's past hourly price and financial news headlines. Also, the system will keep track of all its previously selected positions. Moreover, to help increase strategies' overall stability and generalization capacity between training and testing, the agent will process news headlines to extract market mood information – or sentiment momentum. Finally, all this information is deemed sufficient to guarantee the agent's adequate awareness of the market environment.

A diverse number of market events can occur during a given period, triggering the publication of more than one news article per period. Consequently, the sentiment extractor must score each of the released news headlines. In the following Section 4.3.1, details about the sentiment extractor design and implementation will be provided. Therefore, for now, it suffices to say that it produces a score $y_t^j \in [-1, 1]$ for each news headline $j = 1, 2, 3, ... \in \mathbb{Z}^+$ in the hour before the decision instant t. In this continuous scoring system, lowest values are the most pessimistic (i.e., -1), higher values are the most optimistic (i.e., 1), and intermediary values are neutral (i.e., 0).

In the unitary score format, this sentiment information can not be appropriately employed. Hence, sentiment for each data instance in the past period should be transformed into a single overall sentiment feature. Let Ω be the function that generates a single overall sentiment score by weighting the list of headlines scores that occurred between consecutive decision instants t and $t - 1$. Possible weighting methods for Ω include applying simple mathematical functions for taking the average $avg(\cdot)$, maximum $max(\cdot)$ or minimum $min(\cdot)$ values over all the unitary scores for each sentence. Lastly, the *bullishness index* is a worthy mentioning weighting method to consider, given its popularity in financial sentiment analysis studies (OLIVEIRA; CORTEZ; AREAL, 2017; LI; DALEN; REES, 2018). (ANTWEILER; FRANK, 2004) proposed the bullishness index as a metric for weighting the overall market sentiment from individual sentiment values of several text instances. Employing this weighting method requires the discretization of the sentiment score values: if the sentiment score is above zero, it assigns a positive polarity or negative otherwise. However, if the sentiment score is precisely zero, it assumes a neutral overall market sentiment. Conclusively, let *pos* denote the count of positive sentiment news and *neg* the equivalent for negative sentiment news; then, the bullishness index formulation is given by

$$B \doteq ln\left(\frac{1+pos}{1+neg}\right), \tag{4.18}$$

which compares the number of positive and negative news in a given period.

By applying any weighting method discussed, it is possible to produce a general market sentiment $e_t \in \mathbb{R}$ for the period between the current trading instant t and the previous one $t-1$. Therefore, the general market sentiment can assume different values depending on the selection of method for the weighting function Ω as such:

$$e_t = \begin{cases} min([y_t^j, y_t^{j+1}, ...]), & \text{if } \Omega = min(\cdot) \\ max([y_t^j, y_t^{j+1}, ...]), & \text{if } \Omega = max(\cdot) \\ avg([y_t^j, y_t^{j+1}, ...]), & \text{if } \Omega = avg(\cdot) \\ bullishness([y_t^j, y_t^{j+1}, ...]), & \text{if } \Omega = B(\cdot) \end{cases} \tag{4.19}$$

In resume, producing the general emotion $e_t \in \mathbb{R}$ requires grouping text instances according to the trading frequency and applying one weighting method such as the ones described above. In Chapter 5, weighting methods are compared experimentally to select the most appropriate one. It is necessary to notice that, given that overall emotion e_t is matched to the price time series difference z_t, hence, when grouping the headlines by period, market sentiment features that do not match their financial features counterparts are removed. Essentially, this implies the disposal of news articles published outside the trading hours for that asset.

With the conclusion of the formalization of the process to prepare the final hourly overall sentiment score e_t, the market mood formalization can resume. Finally, the market sentiment momentum can be written as the sentiment feature vector

$$S_t^E = [e_t, ..., e_{t-l+1}] \in \mathbb{R}^l, \tag{4.20}$$

representing the look-back sentiment hourly window of size $l \in \mathbb{Z}_0^+$. Finally, to arrive at the complete state representation for ITS-SentARL, Eq. 4.13, Eq. 4.14, and Eq. 4.20 are combined below

$$\begin{aligned} S_t &= S_t^E \parallel S_t^B \\ &= [e_t, ..., e_{t-l+1}] \parallel [z_t, ..., z_{t-w+1}] \parallel [\tau_t, ..., \tau_{t-w+1}] \parallel [A_{t-1}]. \end{aligned} \tag{4.21}$$

Moreover, the ITS-SentARL architecture design considered different look-back windows for each type of feature for guarantying independence and flexibility of experimental verification.

Notice that the notations employed throughout this manuscript reflected the boundaries of an episode. These notations regarded trading operations that could occur in a live real-time environment or an offline setting. Although ITS-SentARL can be used for online trading with minimal adjustments, an offline version was adopted for training and evaluation in the present work. Thus, since experimentation took place in a simulated offline environment, data collection occurred before any trial. Hence, consider that the data gathered – and after preparation – consists of sequential, not continuous (because of the market closing periods) hourly data points dataset with the last data point $D \in \mathbb{Z}^+$. Also, the time instant $\eta \in \mathbb{Z}_0^+$ for a data point from this complete dataset of size $D + 1$ is not equivalent to a time instant t for a given episode. Consequently, equal indexes for variables do not guarantee identical values (i.e., $e_{t=0} \neq e_{\eta=0}$. This distinction is crucial for comprehending some of the analyses elaborated in the remainder of this manuscript.

Algorithm 2 describes ITS-SentARL offline operation on a given episode where all data is preprocessed before trading starts. As such, in the initial part of Algorithm 2, there is an iteration through the news headlines for each hour, extracting the sentiment score associated with each headline and preparing the complete sentiment feature vector. Moreover, preprocessing of price and hour series occurs to compose respective feature vectors with size depending on the desired episode size and look-back window sizes (i.e., l, w). Now that the sentiment scores, price differences, and hour-day information have been organized, the episode for a training or testing set is ready for trading. Ultimately, the algorithm iterates through each trading instant t, observing the episode features available at each instant ($S_t = S_t^E \parallel S_t^B$), taking actions A_t defined by the A2C Algorithm 1, and receiving rewards R_{t+1}. Moreover, the sentiment-free baseline version of ITS-SentARL – the No Sent. A2C architecture – behaves similarly to Algorithm 2 but without the sentiment information S_t^E.

Algorithm 2 ITS-SentARL offline pseudocode

```
Initialize look-back windows $w$ and $l$
Initialize $\varphi, \xi, T, D$
Initialize empty vectors $SentVec, PriceDiffVec, HourDayVec$
$\eta \leftarrow 0$
for $\eta \leq D$ do
    for $j$ in news headlines in hour $\eta$ do
        $y_\eta^j \leftarrow$ SentimentExtractor($j$)
    $e_\eta \leftarrow \Omega([y_\eta^j, y_\eta^{j+1}, ...])$        ▷ Grouping and weighting hourly sentiment scores
    $SentVec.append(e_\eta)$
    $z_\eta \leftarrow p_\eta - p_{\eta-1}$
    $PriceDiffVec.append(z_\eta)$
    $\tau_\eta \leftarrow h_\eta/24$
    $HourDayVec.append(\tau_\eta)$
    $\eta \leftarrow \eta + 1$
$EpisodeSize \leftarrow T + 1$
$[e_{-l}, ..., e_0, ..., e_T] \leftarrow$ EpisodeSelector($SentVec$, $EpisodeSize$, $l$)
$[z_{-w}, ..., z_0, ..., z_T] \leftarrow$ EpisodeSelector($PriceDiffVec$, $EpisodeSize$, $w$)
$[\tau_{-w}, ..., \tau_0, ..., \tau_T] \leftarrow$ EpisodeSelector($HourDayVec$, $EpisodeSize$, $w$)
$t \leftarrow 0$
$A_{t-1} \leftarrow 0(Neutral)$        ▷ Initial trading position
for $t \leq T$ do        ▷ Starting trading
    $S_t^E \leftarrow [e_t, ..., e_{t-l+1}]$
    $S_t^B \leftarrow [z_t, ..., z_{t-w+1}] \parallel [\tau_t, ..., \tau_{t-w+1}] \parallel A_{t-1}$
    $S_t \leftarrow S_t^E \parallel S_t^B$
    $A_t \leftarrow$ A2C($S_t$)        ▷ A2C takes action
    $\rho_{t+1}^{Trader} \leftarrow \varphi\,[z_{t+1}A_{t-1} - \xi|A_t - A_{t-1}|]$        ▷ Financial return
    $R_{t+1} \leftarrow \rho_{t+1}^{Trader}$
    $t \leftarrow t + 1$
```

4.3.1 Sentiment Extractor

The sentiment extractor is responsible for attributing an individual sentiment score for each news headline. Recall from Chapter 2 that the extractor adopted in the present work is directly inspired by the winner design at the SemEval-2017 Task 5 challenge for financial news headlines (MANSAR et al., 2017). Moreover, the author of the present dissertation contributed with other researchers to determine a configuration that could improve this state-of-the-art sentiment extractor (FERREIRA et al., 2020). Thus, the publicly available source code[2] produced by (FERREIRA et al., 2020) of the complete sentiment extractor design with its optimal hyperparameters configuration is used here. This extractor encompasses two components, the preprocessing and the sentiment scoring machine learning model.

The preprocessing subcomponent transforms raw unstructured textual data into a standard format that machine learning models can handle. The first preprocessing step is to find and replace companies' names with a placeholder word with the help of a dictionary of names. This step reduces dimensionality and, paired with the final feature representation scheme, helps the machine learning model score the sentence according to the target entity. As some sentences contain references to more than one company, the target company name is replaced by 'TARGET-COMPANY' while the term 'COMPANY' is used instead for all other cases. Next, the tokenization technique used punctuation and blank spaces to identify word boundaries and help us extract individual textual elements, called tokens. Then, case-folding of all tokens occur – except for the companies' placeholders – for converting all word characters to their lowercase versions. Also, some term disposal involved punctuation and stop-word removal, which are notorious for not being informative and even contributing to noise in data. Before the final preprocessing step, padding is necessary so that all entries contain the same size $m \in \mathbb{Z}^+$. In Table 2, except for the padding, there is an example of the preprocessing steps.

After preprocessing, textual data is structured and ready for usage in the sentiment-scoring ML model. The purpose of this model is to determine a sentiment score in the numerical range $[-1, 1] \in \mathbb{R}$ for each sentence, where a higher value means a very positive sentence, and a lower number means a very negative one.

The sentiment score component employs a hybrid combination of ML model and lexicon approaches, as shown in Figure 5. Initially, an embedding layer transforms the m-sized word vector input into an improved word representation. *Word embedding* is

[2] https://bit.ly/3kzau8G

Table 2: Example of a news headline going through the textual preprocessing component. First, from the raw text, companies' names are replaced. Then words are tokenized and then lower-cased. Finally, the punctuation and stop words are removed. Finally, the unstructured text was converted into a structured word vector.

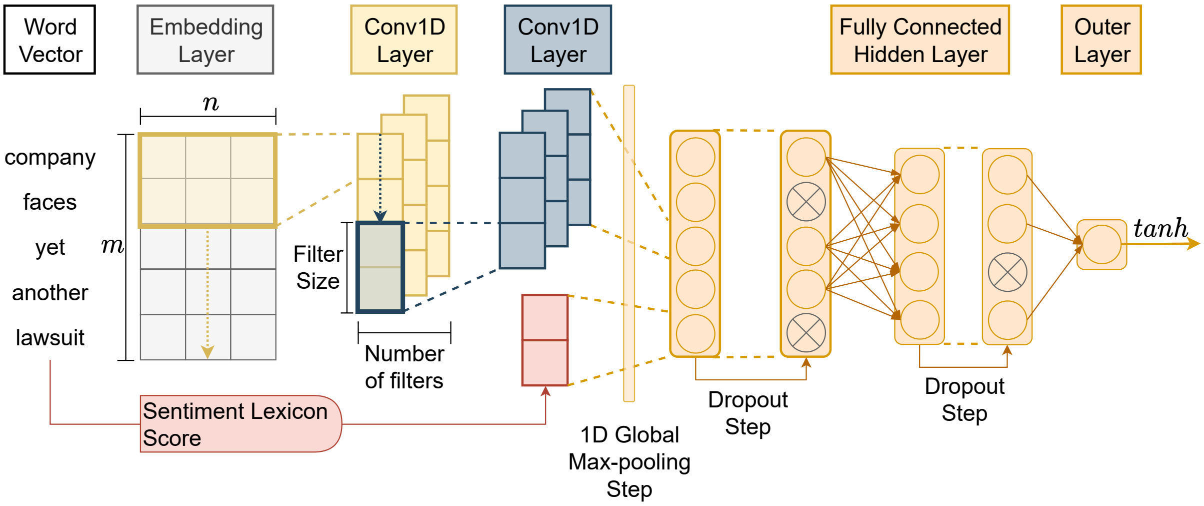
Step	**Output**
Raw Text	Stakeholders from Berkshire Hathaway buy stake in Apple
Placeholder replacement	Stakeholders from COMPANY buy stake in TARGET-COMPANY
Tokenization	[Stakeholders, from, COMPANY, buy, stake, in, TARGET-COMPANY]
Lowercasing	[stakeholders, from, COMPANY, buy, stake, in, TARGET-COMPANY]
Punct. and stop-words removal	[stakeholders, COMPANY, buy, stake, TARGET-COMPANY]

Source: Author's own production.

a technique for representing each token feature as an n-dimensional short, dense vector $v \in \mathbb{R}^n$. For faster convergence, the GloVe pre-trained word vectors serve to warm-start the word representation weights in the retrainable embedding layer of size $m \times n$. Then, features are processed by two one-dimension convolutional layers (Conv1D) displayed sequentially. Both these convolutional layers present the same *filter size* and *number of filters*. Next, the sentiment lexicon extracts initial sentiment scores combined with the features resulting from the second Conv1D before undergoing feature reduction by a one-dimension global max-polling layer, followed by a dropout step. From this point onward, a fully connected (FC) configuration is adopted, with a hidden dense layer of neurons, followed by another dropout step for reducing overfitting by eliminating internal neurons of the hidden layer. Also, all hidden layers mentioned so far use the ReLU activation function. Finally, the outer layer contains a neuron with an activation function *tanh* that outputs values in the desired range of $[-1, 1] \in \mathbb{R}$.

In the collaborative work by Ferreira et al. (2020), other ML models, including the second and third places in the SemEval-2017 Task 5, were implemented and compared using the same gold-standard dataset with other evaluation criteria. Still, the winner model (MANSAR et al., 2017) outperformed its challengers according to the mean squared error (MSE) metric. Moreover, changing the internal architecture, such as the number of layers, did not help to increase performance. Thus, the sentiment extractor presented in this section assumes the same design as the original winner of the SemEval-2017 news headlines task.

Figure 5: The CNN-based sentiment extractor for scoring news headlines, receiving a word vector of size m as input. This input is transformed to a word embedding feature representation of dimension size $m \times n$. Also, the sentiment value of the word vector according to the sentiment lexicon is generated. The word embedding representation passes through two convolutional layers for extracting the most relevant latent features. First, the convoluted features are combined with the ones from the lexicon. Next, these features undergo a global max-pooling step to reduce feature dimensionality and then a dropout step to reduce overfitting. Then, the resulting neurons enter a fully connected hidden layer. Finally, the outer layer with a *tanh* function that outputs a value in the range $[-1, 1] \in \mathbb{R}$.



Source: Author's own production.

On the other hand, the mentioned investigation configurations regarding hyperparameters such as lexicon, filter, and embedding sizes affected the overall performance. Consequently, the parameters in Table 3 represent the findings by (FERREIRA et al., 2020) regarding the optimal configuration of hyperparameters and lexicons. For instance, the domain-specific economic sentiment lexicon yielded slightly worse results than the general domain VADER lexicon. In the same way, high word embedding sizes (e.g., 300) led to increasingly better results than lower sizes (e.g., 50). Ultimately, after finishing the mentioned investigation, Ferreira et al. (2020) made the source code publicly available for reproducing the described evaluation and a ready-for-use sentiment extractor. Also, this sentiment extractor used in the present work was trained using all the news headlines gold-standard data instances.

The present work exhibited the reasoning for adopting this particular architecture and the characteristics of the adopted sentiment extractor relevant to fully understanding ITS-SentARL. For further details, readers should resort to the complete research material (MANSAR et al., 2017; FERREIRA et al., 2020) that led to the construction of this

Table 3: Selected configuration according to Ferreira et al. (2020) for the sentiment extractor component. For instance, findings showed that word embeddings size of 300 and a higher number of filters of 384 led to better performance.

Configuration	**Value**
Lexicon	VADER
Input size m	21
Embedding dimension n	300
Number of filters	384
Filter size	2
Neurons in dense layer	150
Dropout rate	0.4
Objective Function	MSE
Training Optimizer	ADAM
Training Batch Size	32
Training Epochs	30

Source: Author's own production.

module.

5 EXPERIMENTAL VERIFICATION

This chapter describes the training and evaluation methods of the trading system, showing general results. Initial sections describe the data characteristics, followed by the configuration for training and evaluation of ITS-SentARL and others baselines. Moreover, the system used to perform data gathering, preprocessing, and training presents technical specifications described below:

- **Brand/Model Notebook:** Dell© Inspiron i7559-2512BLK
- **CPU:** 2.6 GHz Intel© Core™ i7
- **Memory:** 16 GB DDR3L SDRAM
- **Language:** python 3.6.8
- **OS:** Ubuntu 16.04 LTS

The analysis of the results compares the proposed ITS-SentARL architecture with its No Sent. A2C counterpart (ablation study) and the BH strategy. For such analysis, a thorough statistical examination of results from both implementations, for all metrics (e.g., total profit, average annual return, SR), in the similar characteristics of assets, trading frequency, and trading costs. Ultimately, there is a discussion about the implications of this statistical examination.

5.1 Data Characteristics

According to Henderson et al. (2018), RL techniques require extensive and diverse experimentation to confirm performance results consistency. In this regard, according to Théate and Ernst (2021), it is vital to adopt a high number of assets (e.g., greater than ten) for experimentation. Hence, as depicted in Table 4, twenty assets from different market sectors were employed for the present experimental trading simulation. In the financial market, assets are typically referred to by their tickers (i.e., financial identifiers). So then, the employed assets and market sector description follow AAPL, AMZN, FB, GOOGL, INTC, MSFT, NFLX, BA (High tech), JPM, MA, V (Financial), DIS, HD, JNJ, KO, PFE, PG (Consumer discretionary), XOM (Energy), BA (Industrial), T (Communication Services) and SPY (i.e., an S&P500 index ETF). Moreover, the selection criteria of the 19 stocks included price, traded volume, brand value, and popularity (i.e., measured as the number of mentions in news articles).

Table 4: General asset information regarding the ticker (i.e., financial identifier) of each of the 20 companies selected for simulation and also its sector details.

Asset ticker	Company	Sector
AAPL	Apple	High Tech
AMZN	Amazon	High Tech
BA	Boeing	Industrial
DIS	Disney	Consumer Discretionary
FB	Facebook	High Tech
GOOGL	Google	High Tech
HD	Home Depot	Consumer Discretionary
INTC	Intel	High Tech
JNJ	Johnson & Johnson	Consumer Discretionary
JPM	JPMorgan	Financial
KO	Coca-Cola	Consumer Discretionary
MA	Mastercard	Financial
MSFT	Microsoft	High Tech
NFLX	Netflix	High Tech
PFE	Pfizer	Consumer Discretionary
PG	P&G	Consumer Discretionary
SPY	SP500	ETF
T	AT&T	Communication Services
V	Visa	Financial
XOM	Exxon	Energy

Source: Author's own production.

Note, however, that the data collection progressed under an agreement that allows its use only for non-commercial purposes and does not permit distribution. As such, it was not possible to make the preprocessed news headlines or price data available due to the restrictions imposed by the original data sources (i.e., The Wall Street Journal, MarketWatch, and Duskacopy websites). It is paramount to make data available to the research community when feasible, but sadly, it could not be achieved in this case. Fortunately, this manuscript supplies all the tools and knowledge necessary for collecting and using all the data to reproduce and achieve the same results presented here.

As this present work aims at guaranteeing reproducibility of results for future researchers and, thus, adopted exclusively data gathered from sources that provide free usage for scientific purposes[1]. Consequently, only three years of hourly price data (from 2018 to 2019) was obtained, given by the data available for download at the Duskacopy website[2] database for price time series. Additionally, given the limiting factor that all the selected asset operations follow stock exchange trading hours (from 9:30 to 16:00), a total of 5,267 price data points could be collected for each asset.

For the textual data, recall from Section 4.3.1 that both a labeled gold-standard dataset (i.e., ground-truth) and newly collected unlabeled data from relevant financial news portals are employed. The gold-standard serves for the training of the sentiment extractor, while the sentiment extracted from data of news portals will support the actual trading operation. These sentences usually address one company but can also address multiple businesses. All textual data are mostly single sentences written by financial journalists that adopt relatively proper use of the English language. Moreover, a critical aspect of using data based on news outlets is that the journalists who wrote these news articles possess high knowledge of the market dynamics. Similarly, the gold-standard was labeled by market specialists who understand the impact of news on stock prices and are themselves actors in this market. Thus, a hypothetical advantage of training the sentiment extractor on such data is that it might lead to better acquisition of the mood of the agents who have some impact on the stock market. Altogether this selection of data for training and actual trading aligns with the objective of the present research in capturing the market mood.

The promoters of SemEval-2017 Task 5 (CORTIS et al., 2017) published the competition's gold-standard data into a public repository[3]. This dataset contains news headlines about various companies from diverse market segments carefully selected from financial news outlets. Also, each of these sentences was labeled with a sentiment score in the range $[-1, 1] \in \mathbb{R}$. The procedure for producing these scores involved asking market specialists to attribute a score between -1 (most pessimistic) and 1 (most optimistic) to the sentiment about the target company in the news headline. Then, the final score for each headline was the average over the scores given by the specialists.

[1] Until the publication of the present manuscript, all web portals employed as sources (for both price and textual data) grant non-commercial usage of their information. Nevertheless, conditions may have changed since publication.

[2] https://www.dukascopy.com/

[3] https://bitbucket.org/ssix-project/semeval-2017-task-5-subtask-2/

Finding public textual news datasets ready for active trading tasks can be challenging. Thus, providing textual data for ITS-SentARL required constructing a web crawler for gathering news headlines from relevant financial news portals such as The Wall Street Journal[4] and MarketWatch[5]. Even though designing and implementing a financial web crawler was an essential step of the present research, giving in-depth details about this crawler is out of this manuscript's scope. Hence, it suffices to mention that this crawler goes through the mentioned web portals, seeking headlines about any of the twenty target companies employed here. Moreover, the source code for the implemented financial web crawler was made publicly available and free for use[6].

Table 5 exhibits some of the news headlines present in the textual data and their corresponding sentiment. For the gold-standard dataset, the sentiment score displayed corresponds to the provided label for the SemEval-2017 Task 5. For the data gathered through web crawling, the headline examples display the sentiment scores according to the sentiment extractor described in Section 4.3.1. Ultimately, Table 5 exemplifies the perceived sentiment score for a given sentence. For instance, exaggerated wording (e.g., strong, soar, plunged) tends to lead to the higher extreme positive (e.g., AAPL) or negative (e.g., BA) scores. However, even when the words may represent strong sentiment, the score might be neutral if they do not appear to bring conclusive benefits or concerns to the target company (e.g., JPM). Notably, even though these are just small samples of data, they are representative of the complete news data in the sense that they mainly report on past situations. Therefore, it is noticeable that only a few headlines attempt to speculate about future events (e.g., MA) or suggest potential stocks for buying (e.g., XOM). It is also apparent how the COVID-19 pandemic can harm the market (e.g., SPY, AMZN, MA).

Table 6 depicts the overall characteristics of the textual data by each asset. Displayed characteristics include the number of instances, the median and the maximum number of words, and the sentiment score average and standard deviation for each dataset. For instance, except for the SPY (50,719 headlines) that combines a plethora of companies, no asset presents more than ten thousand news instances, with AAPL (8,740 headlines) and AMZN (8,629 headlines) being the greater ones. On the other hand, it is also noticeable that some stocks are less popular since few journalists mention these assets (e.g., KO, MA, PG), and thus, there are considerably fewer instances (less than a thousand) for use in trading. Also, it is possible to perceive that the sentiment extractor had a small

[4] https://www.wsj.com/

[5] https://www.marketwatch.com/

[6] https://github.com/xicocaio/financial_web_crawler

Table 5: Examples of news headlines from the gold-standard (i.e., ground-truth dataset) and some of the gathered datasets, with their corresponding sentiment scores.

Dataset	News headlines	Sentiment
Gold-standard	"Google Fiber to buy Webpass for big city Internet service"	0.329
SPY	"Dow, S&P Slip as Covid Shutdowns Weigh on Investors"	-0.3619
AAPL	"Apple earnings soar on strong iPhone, Mac sales"	0.8065
AMZN	"California subpoenas Amazon over worker safety amid pandemic"	-0.2719
BA	"Durable-goods orders plunge 14% in March as autos, Boeing take big hit"	-0.8267
JPM	"JPMorgan Just Killed the Bitcoin Dream"	0.0109
MA	"Collapse of Travel Will Hurt Mastercard"	-0.2478
XOM	"Large-Cap Buys: AT&T, GE, Intel, Exxon, BofA"	0.1594

Source: Author's own production.

amount of gold-standard headlines (1,633) for training, which can be prejudicial when using deep neural networks. Even so, given by the example samples in Table 5, there is evidence that the sentiment extractor can be reasonably accurate in scoring sentiment.

As expected, given the inherent nature of this data, Table 6 shows that news headlines tend to contain a small number of words with a median of eight words per sentence. Also, although various assets present headlines with a higher number of words (more than thirty), they seem to be the exception rather than the norm, given the much lower median word count. Notably, when producing the gold-standard dataset, the SemEval-2017 Task 5 researchers limited sentences to a maximum of eighteen words. Moreover, a piece of information not displayed in Table 6 is that all headlines present at least three words. At last, the comparison of the overall sentiment scores for each asset shows that the average sentiment datasets are positive and very close to zero, although the standard deviation is slightly higher in the gold-standard dataset.

Although looking at the characteristics in Table 6 might lead to the idea that all the adopted textual data is reasonably similar, the comparison of data distribution of each dataset according to the sentiment score values might reveal some meaningful differences. Such a comparison in Figure 6 helps to identify particularities in the available information.

Table 6: Characteristics of textual data by asset, including the total count of news headlines, the median and maximum words over headlines, and the average and standard deviation of the sentiment across data instances.

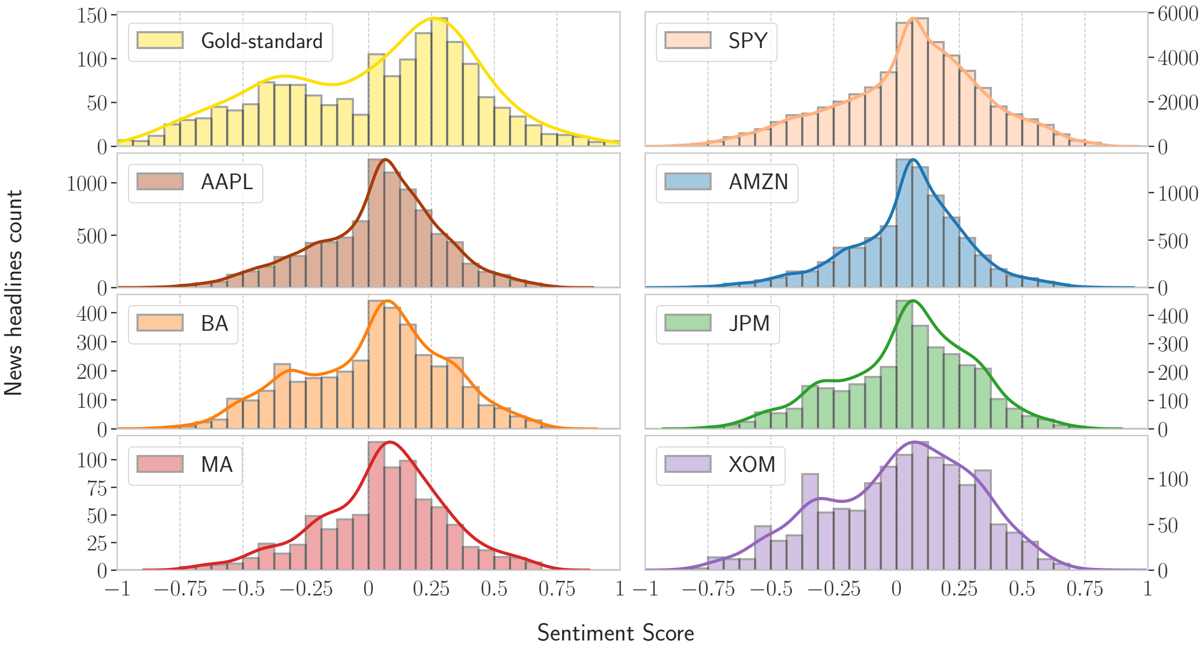
Dataset	News Count	Word Median	Max Words	Sentiment Avg. ± Std.
Gold-standard	1633	10	18	0.0262 ± 0.3998
AAPL	8740	8	41	0.0380 ± 0.2525
AMZN	8629	9	33	0.0475 ± 0.2327
BA	3882	8	31	0.0198 ± 0.2866
DIS	2924	8	31	0.0523 ± 0.2377
FB	6359	10	37	0.0292 ± 0.2247
GOOGL	5270	8	34	0.0272 ± 0.2302
HD	1178	8	28	0.0274 ± 0.2659
INTC	2267	8	35	0.0390 ± 0.2744
JNJ	1285	8	36	0.0688 ± 0.2702
JPM	3292	8	35	0.0408 ± 0.2629
KO	969	8	30	0.0767 ± 0.2638
MA	812	8	30	0.0588 ± 0.2501
MSFT	4149	8	40	0.0420 ± 0.2478
NFLX	3488	9	33	0.0419 ± 0.2344
PFE	2192	8	36	0.0833 ± 0.2722
PG	877	8	32	0.0771 ± 0.2684
SPY	50719	10	36	0.0579 ± 0.2928
T	2031	8	32	0.0675 ± 0.2261
V	1112	9	30	0.0340 ± 0.2795
XOM	1515	8	30	0.0133 ± 0.2949

Source: Author's own production.

This comparison involves the binned count of news headlines (y-axis) and their distribution over the sentiment scores (x-axis) in the range $[-1, 1] \in \mathbb{R}$. Therefore, it is possible to highlight similarities and differences between the gold-standard (i.e., ground-truth sentiment labels) and some of the assets news data scored by the present sentiment extractor. For instance, the gold-standard plot (top-left) indicates a bimodal concentration of data, with one prominent Gaussian distribution peak on the positive side and a smaller peak on the negative. These distributions show that, even though this dataset contains a moderately higher number of positive instances, the SA component was exposed to a diversified set of positive and negative instances during training.

Remember from Section 3.1 that the SPY data (top right) is an ETF that comprises stocks from 500 companies. Thus, it contains much more news headlines, and it helps to reference the overall market sentiment across companies. Not surprisingly, even though

Figure 6: The distribution of news headlines by asset regarding observed sentiment score compares the ground-truth (i.e., Gold-standard) data to other assets. Hence, the y-axis shows the binned count of news headlines, and the x-axis depicts the sentiment score in a continuous −1 (most pessimistic) to +1 (most optimistic) scale. Also, the line over the bins is the kernel density estimation that normalizes the distribution over an estimated probability density function.



Source: Author’s own production.

the SPY dataset exhibits a slightly skewed curve to the positive side, it still presents a compact distribution centered close to zero on the positive side and with almost no instances on the extremes of the sentiment axis. Hence, differently from the gold-standard data, the SPY distribution presents a typical unimodal Gaussian curve. Although, interestingly, some assets (AAPL, AMZN, MA) present a distribution similar to the SPY reference, but others resemble the gold-standard distribution. For example, although less pronounced, some data (BA, JPM, XOM) present a slight distortion similar to a bimodal distribution, most evident by the kernel density estimation drawn on top of the distribution bins. Nevertheless, the distribution for these assets is less sparse than the gold-standard and more concentrated over the neutral (i.e., 0) sentiment value. Even so, it is particularly noticeable that some of these data (BA, JPM, XOM) offer more data instances in the $[-0.5, -0.25]$ range than other assets (AAPL, AMZN, MA). Finally, differently from the gold-standard dataset, data gathered through web crawling seem to lack extremely positive (i.e., greater than 0.75) or negative (i.e., lower than −0.75) headlines.

5.1.1 News Analysis

As discussed in previous Chapter 4, all input data requires preprocessing before composing the market state representation for the agent to use. For instance, the hourly price of assets is transformed into a price time series given by the difference between consecutive closing prices z_η. Similarly, news data undergoes a preprocessing for generating a sentiment time series that represents the general emotion of the market in a given period. Therefore, after extracting the sentiment score from each headline, the market's emotion for a particular hour or day e_η is given by combining these scores according to a weighting function Ω. However, there might not be news about an asset to score during a given period, and, thus, some instants are not covered by headlines. Hence, *news coverage* represents the ratio of data points, from a total of 5,267, with at least one headline. Another insightful way to analyze the input data is to take the Pearson correlation between each asset's price difference $[z_{\eta=0}, ..., z_{\eta=D}]$ and sentiment time series $[e_{\eta=0}, ..., e_{\eta=D}]$, where D is the last data point of the dataset. Ultimately, Table 7 presents information for each asset regarding the news coverage and correlation considering each of the four sentiment weighting approaches given by Eq. 4.19: minimum, maximum, average, bullishness index.

When looking at the asset information in Table 7, it is straightforward to verify that, despite the popularity of some stocks (e.g., AMZN, AAPL), news articles might cover at most half of the available trading instants. Notably, even the SPY ETF, which encompasses 500 companies and is commonly mentioned by journalists, achieves 95.71% of news coverage (5,041 out of 5,267). Conversely, a quarter of the assets have news coverage lower than 10% (HD, JNJ, KO, MA, PG). Besides, the average news coverage among assets is 24.81%. Moving on to the correlation between the price and sentiment time series of an asset, values change only slightly across weighting methods. However, it is noticeable that the bullishness index presents a slightly lower correlation than other weighting approaches for most assets, the exception being the SPY. Distinctly, BA is the asset with the overall highest correlation, particularly for the maximum weighting approach (0.2221). Not surprisingly, MA, the asset with the lowest news coverage (5.60%), also presents the lowest and the only negative correlation values. The impact of correlation will be discussed in further evaluation sections, but for now, suffice to mention that it is crucial to verify that correlation is not exceedingly high or low. For instance, an exceedingly high correlation might indicate that most market information is already assimilated into the price, so adding sentiment information may be redundant and even prejudicial (causing multicollinearity of input variables). On the other hand, a very low or even negative correlation can imply that sentiment information is too detached from

Table 7: News coverage and sentiment correlation between different weighting methods and the price difference of each asset. The news coverage is a proportion of price data points with at least a news headline to compose a sentiment score for that time instant. The values on the remaining columns correspond to the Pearson correlation between the time series of the price difference of consecutive instants and the news sentiment scores with different weighting methods (minimum, maximum, average, and bullishness index).

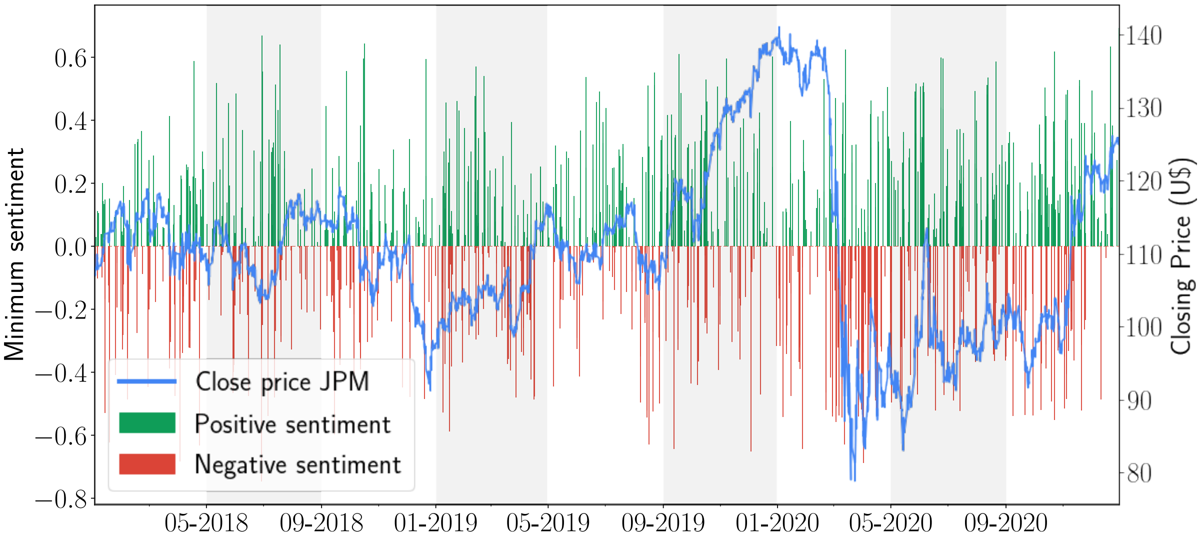
Asset	**News coverage**	**Sent. corr. (Min.)**	**Sent. corr. (Max.)**	**Sent. corr. (Avg.)**	**Sent. corr. (Bull. index)**
AAPL	2673 (50.75%)	0.0379	0.0353	0.0399	0.0139
AMZN	2617 (49.69%)	0.0551	0.0464	0.0527	0.0252
BA	1498 (28.44%)	0.2115	0.2058	0.2221	0.1942
DIS	1102 (20.92%)	0.1364	0.1468	0.1478	0.1224
FB	2002 (38.01%)	0.0561	0.0609	0.0624	0.0278
GOOGL	1762 (33.45%)	0.0273	0.0399	0.0388	0.0199
HD	481 (9.13%)	0.1131	0.1079	0.1130	0.0771
INTC	893 (16.95%)	0.1327	0.0978	0.1247	0.1028
JNJ	472 (8.96%)	0.0759	0.0815	0.0810	0.0855
JPM	1281 (24.32%)	0.1723	0.1879	0.1871	0.1417
KO	375 (7.12%)	0.0955	0.0916	0.0957	0.0892
MA	295 (5.60%)	-0.0231	-0.0137	-0.0189	-0.0282
MSFT	1471 (27.93%)	0.1151	0.1192	0.1245	0.1039
NFLX	1263 (23.98%)	0.0709	0.0746	0.0785	0.0389
PFE	684 (12.99%)	0.0493	0.0545	0.0576	0.0498
PG	320 (6.08%)	0.1006	0.1308	0.1202	0.0939
SPY	5041 (95.71%)	0.0867	0.0746	0.1272	0.1412
T	701 (13.31%)	0.0485	0.0201	0.0354	0.0347
V	535 (10.16%)	0.1524	0.1435	0.1501	0.1065
XOM	666 (12.65%)	0.1492	0.1531	0.1539	0.1289

Source: Author's own production.

the price series to be helpful and could be providing noise to the system.

In figure 7, there is an example of a market sentiment time series produced with the sentiment minimum weighting method and daily grouping, in parallel to the JPM close price time series. This plot helps identify periods of upward and downward price tendencies that seem to relate to a higher concentration of positive (green) and negative (red) sentiment averages. Furthermore, the devastating effect of the market crash caused by the Covid-19 pandemic in the 2020 first semester is quite impressive, where assets prices could drop to almost half their previous price. In this sense, it is interesting how even though the minimum weighting method is being used to group hourly sentiment, in periods of price recovery, there is also a notable decrease or absence of bearish sentiment news (e.g., from March to September of 2019). In particular, starting at around November 2020, there is an evident diminishing number of negative news, which might be related to the optimism about economic recovery given the prospect of vaccines being applied in the U.S. in early December of 2020. Ultimately, there is no daily sentiment minimum higher than 0.7 or lower than -0.8. In the further sections, there will be details about the reasoning for selecting the weighting method.

Figure 7: Example of an asset's minimum positive (green bars) or negative (red bars) sentiment and closing price in U$ (blue line). The y-axis represents the minimum hourly sentiment scores on the left, while on the right, there are the closing prices for the given asset (JPM), and in the x-axis, the date values for the examined period (2018 to 2020).



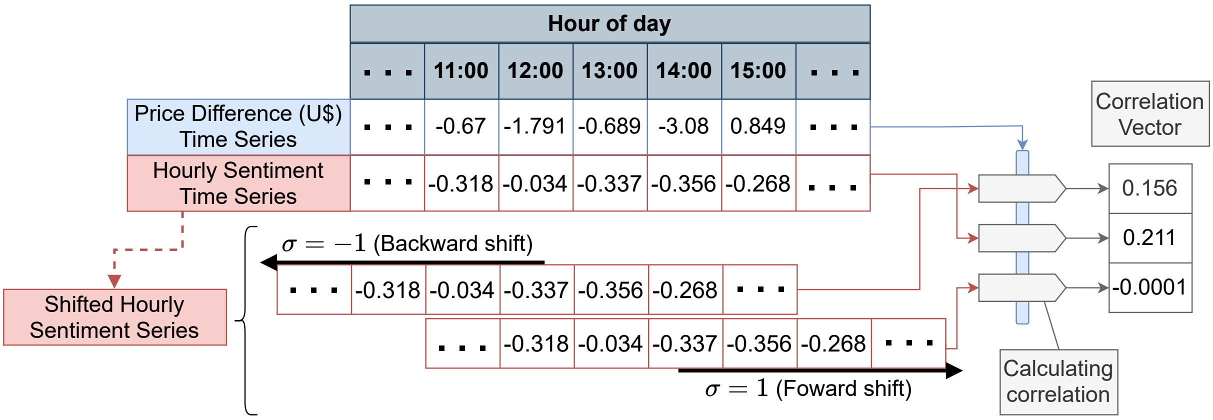
Source: Author's own production.

Despite being useful to analyze the correlation as just described, imagine that an event concerning a given company occurred between consecutive trading hours $[\eta - 1, \eta]$ and triggered the release of several news articles in less than an hour. Naturally, both overall sentiment e_η and difference in closing price z_η at η for that asset oscillate similarly. Thus, the correlation between the series of sentiment and prices, as presented in Table 7, serves to compare these series at equivalent moments.

However, depending on the magnitude of an event, news articles are still written hours after events occur, making the sentiment about it reverberate over time. Consequently, the overall sentiment at the posterior hour $(\eta + 1)$ could still be pessimistic and, thus, it is relevant to bring the sentiment series backward (i.e., negative shift) so that a later sentiment can be compared to a current price change. Hence, applying a negative sentiment shift may be desirable to verify the duration of the impact of a given event and how long its effects take to fade away. Essentially, a negative shift might help to identify the market sentiment momentum. Oppositely, pushing the sentiment series forward (i.e., positive shift) favors observing the market emotion at a previous hour and how it impacted the current price. Therefore, a positive sentiment shift can help identify the predictability capacity of the sentiment.

After shifting the sentiment time series towards the price difference and then taking the correlation between these series, it is possible to evaluate the impact of news on the price over time. Hence, the idea is to apply a negative (backward) or positive (forward) time shift $\sigma \in \mathbb{Z}_0^+$ to the sentiment time series $[e_{\eta=0+\sigma}, ..., e_{\eta=D+\sigma}]$, with $e_{\eta>D} = 0$ and $e_{\eta<0} = 0$, and subsequently get the correlation to the price series $[z_{\eta=0}, ..., z_{\eta=D}]$. In resume, a negative shift $\sigma < 0$ brings future sentiments to the past prices while a positive shift $\sigma > 0$ pushes the sentiment series to the future. Figure 8 depicts this shifting concept and shows the *correlation vector*, which is the outcome of making various shifts and extracting its correlation with the price series. Selecting different time shifts σ makes it possible to generate the correlation vector that helps verify the progression by asset of the correlation over time, as depicted in Figure 9. The sentiment-to-price correlation appears on the y-axis, while the shift σ values are on the x-axis.

Figure 8: The correlation shift procedure encompasses applying a time shift σ to the hourly sentiment series to move it backward ($\sigma < 0$) or forward ($\sigma > 0$) towards the price difference time series and then taking the correlation. Hence, the top part of the image shows the hour of the day and its corresponding price difference time series and hourly sentiment, while the bottom shows the shifted sentiment series. The correlation vector contains the values over different shifts and can help examine the duration of the impact of news over time and its predictive power.

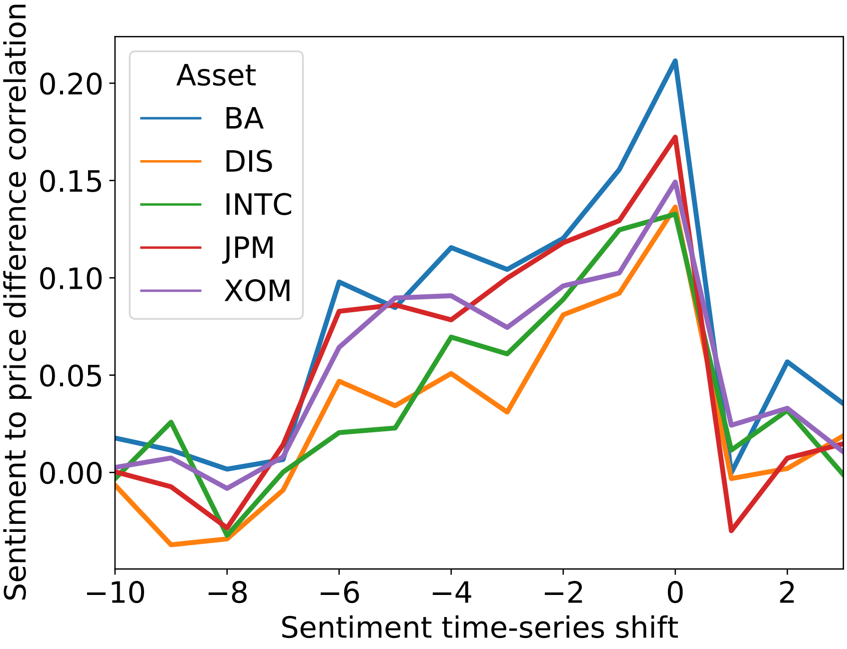


	Hour of day						
	...	11:00	12:00	13:00	14:00	15:00	...
Price Difference (U$) Time Series	...	-0.67	-1.791	-0.689	-3.08	0.849	...
Hourly Sentiment Time Series	...	-0.318	-0.034	-0.337	-0.356	-0.268	...

Source: Author's own production.

Thus, by examining Figure 9, it is noticeable that the correlation progression shows the lowest values for most assets at $\sigma \leq -8$ and $\sigma \geq 1$. Interestingly, this progression indicates a pulse that starts with an increase in correlation around a shift of -7 until achieving its

Figure 9: The evolution of the correlation between sentiment and price difference over time for some of the employed assets. The shift σ to the sentiment time series $[e_{\eta=0+\sigma}, ..., e_{\eta=D+\sigma}]$ is shown in the x-axis, while the Pearson correlation with the price difference $[z_{\eta=0}, ..., z_{\eta=D}]$ appears in the y-axis. For instance, the rising correlation from the negative shift to zero $-7 \geq \sigma \leq 0$ shows that news sentiment is mostly about past events, and there is a poor correlation to the immediate next trading hour $\sigma = 1$.



Source: Author's own production inspired by Lima Paiva et al. (2021).

maximum value at $\sigma = 0$ and then falling back to a very low correlation at $\sigma = 1$. Hence, this correlation progression and this sudden change in value are two particularities about the correlation pulse that could be very insightful. In fact, recall the discussion from Section 3.1 about how news traders examine market sentiment to exploit asset mispricing opportunities. The correlation pulse suggests the present approach for extracting market sentiment information to be compatible with behavioral economic expectations about market mood.

In this regard, the correlation pulse could be capturing the speculative behavior that happens before an anticipated event. Therefore, the lower correlation at a posterior hour $\sigma = 1$ might represent the asset getting corrected by a price mean reversion. Alternatively, the correlation profile could also be capturing that events have a lasting effect that might take up to 7 hours to dissipate completely. This enduring influence could result from remarkable events prompting journalists to continuously produce news articles about the subject, leading to a cyclical influence between the market mood and the asset price. Still, the market sentiment's direct linear predictive power for the posterior hour to an event $t + 1$ appears nonexistent. Furthermore, the correlation between series reaches its lowest value with a shift $\sigma = 1$ before increasing for most assets when $\sigma > 2$. Hence, it seems there is a lower alignment between price and sentiment series in the hour posterior $t + 1$ to events. This low correlation could be caused by the investors failing to fully comprehend the situation and then overreacting very arbitrarily, causing the market to become unstable while digesting the contents of the event. Ultimately, investors digest the information, and the market starts to operate more consistently again ($t > 2$).

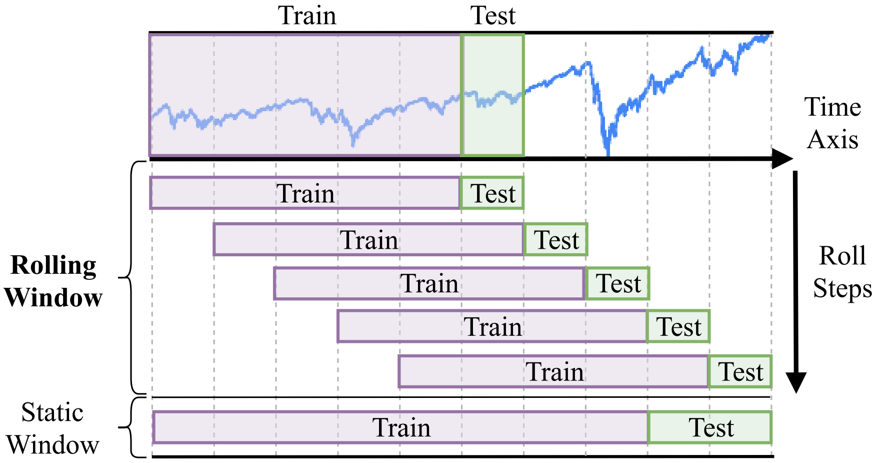
In resume, the correlation pulse pattern, observed in Figure 9, is not a surprise for two main reasons. First, textual data examples in Table 5 showed that news headlines predominantly discuss past events instead of speculating about future trends. Second, the present sentiment data analysis confirms the patterns that Barberis and Thaler (2003) – in their survey – found out to be shared among behavioral economic works.

The correlation pulse shows that the collected textual data might adequately convey the prevailing sentiment market momentum the present works aims at capturing. In this sense, much like a human news trader, the proposed intelligent system should be capable of evaluating if either excitement is being adequately assimilated into the price or if overreaction or underreaction is causing asset mispricing. Ultimately, the correlation pulse, together with the experimental results discussed in the next section, serve as pieces of evidence that corroborate the hypothesis that observing market sentiment momentum is a promising research direction. Essentially, having both a sentiment and price time series should allow the intelligent system to observe discrepancies between these series that affect its decisions to maintain or change positions, focusing on the future decision and not the immediate next one (i.e., prediction scenario). Thus, this sentiment information should stabilize the intelligent system decisions to avoid unnecessary action change and focus on the longer-term benefits of maintaining a particular position. Also, the following section discusses how the correlation pulse served as a guide for selecting the look-back sentiment window of size l.

5.2 Simulation details

When devising a trading simulation for comparing systems and strategies, it is crucial to define which experimental setup to use. For example, Figure 10 compares the most typical setups for training and evaluates the performance of a system that uses time series data. The static window setup resembles typical setups for non-sequential data. In this case, all the data is completely used in either the train or test sets with static periods of fixed proportion. Alternatively, as the name suggests, all sets roll forward with a given number of steps for the rolling window setup. Eventually, all the data is processed for this later setup but in a segmented format that allows system exposition to different market situations. This approach helps expand the ways to evaluate a system's consistency and thus helps address the issues regarding the comparison of RL techniques discussed by Henderson et al. (2018).

Figure 10: The training and evaluation of systems can occur either in a rolling window or a static setup. In a static window setup (bottom), the complete dataset is divided into two parts, one used for training and the other for testing. In the present work, the rolling window setup is used (top), with the training and test sets covering part of the data and, at each rolling step, they move forward to cover a different portion of the data.



Source: Author's own production.

The present work aimed at providing enough variability in the testing sets for properly evaluating consistency over distinct scenarios, and thus a rolling window setup that enacts five roll steps was selected. Furthermore, the selection of the number of rolling steps also considered the number of data points available for training, which is essential when using deep neural networks as function approximators, given their data-hungry characteristics (RUSSELL; NORVIG, 2009). Subsequently, this number of rolling steps implies dividing each window into training and testing sets on the ratio of 0.9 (3,377 data points) for training and 0.1 (374 data points) for testing. This configuration allows testing each model for each asset without superposition or look-ahead issues regarding the testing sets. Additionally, the selection of periods for training and testing should reflect the goal of the rolling window in sampling from different periods with distinct market climates. In consequence, the first test window covers, for the most part, a period of optimism and market growth (from 2019-12-09 to 2020-02-26), while the second one (from 2020-02-27 to 2020-05-13) includes a market crash and its fallout caused by the Covid-19 pandemic.

The proposed ablation study compares ITS-SentARL against a sentiment-free baseline version of the proposed architecture, No Sent. A2C, to evaluate the benefits of market sentiment momentum incorporation. Additionally, as discussed in financial background Section 3.1, the traditional buy-and-hold (BH) is a passive strategy widely adopted in stock markets, and thus, it is helpful to consider it as a reference baseline. Furthermore, although not all RL works use BH as a baseline (CARAPUÇO; NEVES; HORTA, 2018; HIRCHOUA; OUHBI; FRIKH, 2021), it is relatively usual to observe researchers (DENG et al., 2017; MA et al., 2021) comparing their approaches to BH. Ultimately, ITS-SentARL, No Sent. A2C and BH are subjected to the same simulation scenarios whenever possible. Nonetheless, notice that initialization diversification does not apply to the BH strategy, and also, the TC has no significant effect (given there are no position changes in the period).

Regarding the metrics for comparing methods, the present work adopts the standard metrics in the financial domain that RL researchers most frequently adopt (MOODY et al., 1998; DENG et al., 2017; YE et al., 2020). Recall from Section 3.2 that a trader's ultimate goal is to increase their net worth by exploiting market inefficiencies. Hence, one obvious choice for evaluating net worth growth is the total return (TR). Total return is given by taking the relation between the accumulated return from each instant ρ_{t+1}^{Trader} (Eq. 4.16) in a period and the initial net worth ψ (cash amount available for investment)

$$TR = \frac{\sum_{t=0}^{T} \rho_{t+1}^{Trader}}{\psi}, \tag{5.1}$$

where ψ and ρ_{t+1}^{Trader} are in U\$. Therefore, a positive return indicates an increase in net worth (profit), while a negative return indicates the opposite (loss). In addition, TR makes it possible to derive the annualized return (AR), a more convenient metric for comparing results over different periods and studies. The measure of the annualized return (AR) is given by

$$AR = (1 + TR)^{\frac{\chi}{365}} - 1, \tag{5.2}$$

where χ is the number of trading days each model operated over.

The financial return alone may not recognize the risk taken to reach profitability and, thus, the risk-adjusted *Sharpe ratio* (SHARPE, 1966) evaluation metric – the return normalized by the risk (volatility) – helps address this issue. The Sharpe ratio (SR) function adjusts the risk by comparing the average and the standard deviation of an asset's total return over different circumstances, given by the following formulation

$$SR \doteq \frac{Average(TR)}{StandardDeviation(TR)}. \tag{5.3}$$

As such, the SR of a model or strategy over a given asset accounts for TR covering all trials (i.e., different periods, initializations, and trading costs) of that specific asset. Moreover, given the formulation just described, it is straightforward that SR also evaluates a strategy's financial return dispersion (volatility).

All simulation trials assume that the trading agent can only conduct operations over a fixed amount of shares φ. Hence, independent of the taken action A_t (e.g., Long, Short, Neutral), the agent will permanently be shifting a predetermined number of shares $\varphi = 1$. Moreover, the simulations assume an initial net worth of $\psi = \varphi * p_0^{asset}$, which is equivalent to the initial amount of cash required to buy φ shares of the given asset at a price p_0^{asset} at the instant $t = 0$. Sequentially, simulations ran over two penalties scenarios: no penalty and high penalty. The former scenario does not penalize transactions with a TC of zero, while the latter applies a TC of 0.25% to each change in position.

As previously mentioned, Henderson et al. (2018) point out reproducibility and stability issues in RL research results concerning the initialization seed of neural networks based algorithms. These researchers concluded that RL results might vary widely depending on the initialization seed used to define initial training characteristics, such as the internal weights of neural networks. This difference in initialization can lead to very different trained models that lead to huge differences when testing models in an unseen environment (i.e., generalization issues). Hence, following recommendations by Henderson et al. (2018), five different internal weights initialization seeds are selected to verify the reliability of each system under the same conditions. Ultimately, ITS-SentARL and No Sent. A2C architecture run over 1000 trials each (20 assets × 5 window rolls × 2 TCs × 5 initialization seeds). Also, for each trial, the training extends for 100 episodes (i.e., epochs). In essence, the high number of trials covering distinct situations, including extended training analysis, allows a thorough and robust verification of results, compatible with recently suggested guidelines (HENDERSON et al., 2018; THÉATE; ERNST, 2021). In Table 8, there is a summary of the adopted simulation characteristics discussed in this section.

Considering the rigor employed in the experimental setup that verifies the presented research hypotheses, applying the same procedure for model hyperparameter selection would have been prohibitively time-consuming. Therefore, it was necessary to run hyperparameter selection with a reduced scope of 10% of the training set for validation, with fewer assets, initializations and only in a high-penalty environment. Consequently, future investigation is required to explore a possibly better configuration setup for ITS-SentARL. In resume, the final hyperparameter selection for all models (i.e., ITS-SentARL and No

Table 8: Selected parameters for the simulations and models hyperparameters. The first part of the table shows the simulation parameters that impact the evaluated strategies, such as initialization seeds and training episodes. Next, the second part shows the selected characteristics of the adopted base A2C architecture for ITS-SentARL and No Sent. A2C. For instance, both models adopt the same look-back window for the price difference and hour time series and a deep neural network (DNN) with two hidden layers, each with 64 nodes. Ultimately, the final part of the table shows information about ITS-SentARL regarding the sentiment look-back window and the sentiment weighting function that selects the minimum sentiment in a given hour.

Parameter	**Type**	**Value**
Fixed amount of traded asset shares (φ)	Simulation	1
Initial net worth in U$	Simulation	$\psi = \varphi * p_1^{asset}$
Number of rolling window steps	Simulation	5
Rolling window Train/Test division ratio	Simulation	0.9/0.1
Number of initialization seeds	Simulation	5
No-penalty and high penalty TCs (ξ)	Simulation	0% and 0.25%
Number of training episodes	Simulation	100
Trading days in the testing set(χ)	Simulation	77
Price Difference and Hour look-back window size (w)	Base A2C	20
DNN architecture (Value and policy functions)	Base A2C	MLP
DNN hidden layers (Value and policy functions)	Base A2C	2
DNN neurons (Value and policy functions)	Base A2C	64
DNN learning rate	Base A2C	0.99
A2C batch update steps	Base A2C	5
Market sentiment look-back window size (l)	ITS-SentARL	5
Sentiment weighting function (Ω)	ITS-SentARL	Minimum

Source: Author's own production.

Sent. A2C) is also present in Table 8. Finally, discussion and justification regarding the selection of some of these configurations present in Table 8 include:

- **Sentiment weighting.** For most assets, taking the minimum sentiment score across all headlines in an hour showed slight improvements over selecting the average, maximum, or Bullishness index weighting methods. Although it was noticeable that other weighting techniques worked better for some assets. Thus, there might be an opportunity to monitor the correlation and news coverage to better select this method. Eventually, due to the scarcity of news in some periods, different weighting methods presented similar time series values. Hence, combining more than one weighting method led to poor results, probably due to collinearity among input features. Future work could address this issue by some data preprocessing mechanism that verifies and selects when to use multiple weighting methods. Another alternative might include adopting some model architecture with built-in mechanisms for proper feature selection (e.g., CNN or ResNET).

- **Look-back windows.** Even though price and hour-of-the-day features are essential, defining high-sized look-back windows for these features can limit the importance of the sentiment features. On the other hand, selecting a small window might hinder the trading agent's ability to perceive the market environment correctly. Ultimately, the value of $w = 20$ led to better results. Furthermore, during hyperparameter selection among the experimented values (e.g., 1, 5, 10), a look-back sentiment window of size $l = 5$ appeared to more adequately capture the market sentiment momentum to improve ITS-SentARL performance. Although this value appears to be following what is expected from the sentiment correlation pulse previously observed in Figure 9, a more thorough investigation regarding other look-back windows should be performed in the future.

- **A2C policy and value networks.** The A2C algorithm requires approximating both policy and value functions, and during validation, it was observed that the use of separate ANNs with similar configurations for each function provided better performance than a single ANN for both functions. In this sense, each function adopts a deep ANN multilayer perceptron MLP with two hidden layers with 64 nodes each. Experimenting with deeper networks proved to be less efficient due to overfitting. However, this aspect also relates to the number of input features. Hence, slightly deeper networks might work better if look-back window sizes increase. The learning rate of 0.99 provided better performance among other values.

- **A2C steps before update.** After a given number of steps, the A2C algorithm executes batch updates to its policy and value functions. Therefore, performing these updates after every five steps showed to be the best option. However, this parameter value performance could relate to the sizes of the look-back windows and, thus, require further investigation.

The simulation scenarios adopted the OpenAI Gym environments library (BROCKMAN et al., 2016), while the A2C implementation came Stable-Baselines (RAFFIN et al., 2019). The final implementation adapted these libraries to the present stock trading problem and is publicly available together with ITS-SentARL source code[7].

[7] https://github.com/xicocaio/its-sentarl

5.3 Results Analysis

This section examines various aspects of SentARL's performance and its comparison against the sentiment-free baseline (or No Sent. A2C) and the BH strategy. Thus, to support the present research hypotheses, the following characteristics are explored: model generalization, consistency, the impact of market sentiment momentum, and overall return and risk.

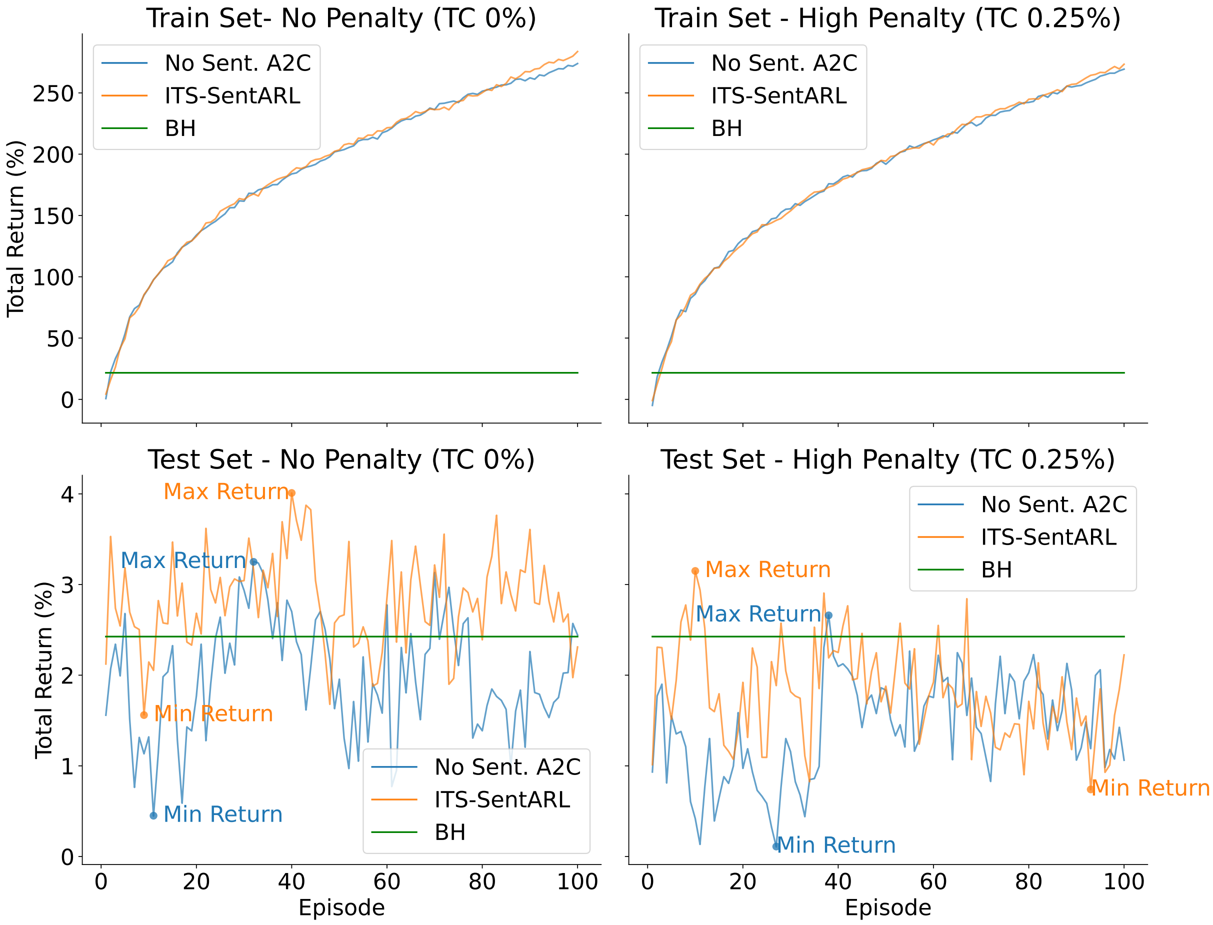
5.3.1 Model Performance Progression

As mentioned in Section 5.2, one of the purposes of the present work was to follow strong modern experimental RL guidelines (HENDERSON et al., 2018; THÉATE; ERNST, 2021). As such, it was essential to analyze aspects that indicate model stability over time. Therefore, Figure 11 focused on reporting the overall financial return for each episode in training and testing, considering all combinations of assets, window rolls, and initialization seeds for both TCs. As a result, the evolution of average total return (y-axis) according to the number of evaluated episodes (x-axis) can be seen in Figure 11. Hence, the total return progression for ITS-SentARL (orange line) and No Sent. A2C (blue line) are displayed for both training (top part) and test sets (bottom part) and both no-penalty (left part) and high penalty (right part) scenarios. Alternatively, as the number of episodes only affects ML models, the BH strategy (green line) presents the same constant average total return values of 21.67% in training and 2.43% in the test period. Thus, BH serves as a reference for the models' performance.

Looking at the training profile, independent of penalty, both ITS-SentARL and No Sent. A2C present a steady and consistent increase in average total return until the max number of a hundred episodes. This progression from neutral or negative returns in the first episode to more than 250% total return in the final episodes indicates that independent of transaction costs, both models continuously learn ways to exploit the market trends for profit. For instance, models achieve better performance than BH in only three episodes, and in just ten episodes, they achieve more than four times greater total returns than BH. Still, between ten and fifteen episodes, it appears to exist an inflection point at the rate at which the total return increases per training episode. Ultimately, this inflection could indicate that both models found more outstanding exploitation opportunities at the beginning of training.

For the progression of financial return on the test set, none of the models presented a

Figure 11: Comparison between ITS-SentARL and No Sent. A2C model performance in the training and test sets according to average total return overall assets by episode, with BH as reference. Top: Training Set; Bottom: Test Set; Left: No-penalty scenario (TC 0.0%); Right: High-penalty (TC 0.25%).



Source: Lima Paiva et al. (2021).

well-behaved curve profile such as the one observed during the training phase. Nevertheless, this outcome is not surprising since trading in the financial market is well known to be a highly chaotic scenario (TSAY, 2010). Thus, to clarify how to identify the number of episodes each model takes to overfit, it is paramount to do more than compare the total return profile between training and testing sets. In this sense, the high dissociation between consecutive periods, given by the chaotic stock markets' nature, could be causing this inconsistent performance in an unseen environment. Consequently, expecting to observe satisfactory model generalization by traditional ML approaches that compare progress curves of return over episodes might not be adequate for the trading scenario.

In this regard, looking for evidence of a diminished variation of the financial return across episodes can help better assess that a model is less prone to performance oscillation and, thus, more consistent and reliable. Hence, the present work suggests investigating the three following aspects for each model or benchmark, considering the 100 episodes of examination: maximum and minimum values of total return achieved during the; difference between max and min returns considering all episodes; the number of episodes with better performance.

Initially, the maximum and minimum average total returns in the no-penalty scenario (down left in Figure 11) are 4.01% and 1.56%, respectively, for ITS-SentARL and 3.25% and 0.45% for No Sent. A2C. Hence, the difference between the max and min total returns for ITS-SentARL (2.45%) was 13% lower than No Sent. A2C baseline (2.81%). Then, considering the high-penalty scenario, ITS-SentARL presented a max return of 3.15% and a min of 0.74%, while for the no sentiment model, these values were 2.66% and 0.11%, respectively. Consequently, ITS-SentARL showed a 6% lower difference between max and min returns (2.41%) than No Sent. A2C (2.56%). These results reveal that ITS-SentARL provided better maximum and minimum total returns in both penalty scenarios and better consistency demonstrated by exhibiting narrower differences between its best and worst performance.

Next, notice in Figure 11 that ITS-SentARL presented higher episode total return maximums and minimums for both transaction cost scenarios. For instance, in the high-penalty scenario, ITS-SentARL displayed a return at its maximum performance (episode 10) 18% greater than No Sent. A2C maximum return (episode 38). Furthermore, in the same scenario, ITS-SentARL achieved an almost six times greater return at its worst (episode 93) than the No Sent. A2C model's lowest performance (episode 27). Finally, the difference is also noteworthy in the no-penalty scenario, with ITS-SentARL achieving a 23% greater maximum return and 250% better minimum return than its sentiment-free counterpart. In conclusion, ITS-SentARL improved total return considering models' highest and lowest performances, is another aspect that helps support the benefits gains against No Sent. A2C.

The third meaningful aspect to observe is that despite significant oscillation in performance, ITS-SentARL consistently displays better total return across episodes under most circumstances. In particular, for the no-penalty scenario, IT-SentARL was more profitable than its sentiment-free counterpart in 89 out of 100 episodes. Also, ITS-SentARL managed to be more profitable than BH in 79 episodes, while No Sent. A2C only outperformed BH in 25% of the episodes. Next, in the high-penalty scenario, introducing transaction costs caused performance declines in both models' profitability, such that outperforming the BH occurred rarely. For instance, ITS-SentARL presented better returns than the BH in 14 episodes, while No Sent. A2C did the same in only one episode. Still, even though

ITS-SentARL prevalence reduced, it kept outperforming its no sentiment counterpart in most episodes (66%). Notably, in this high-penalty scenario, while ITS-SentARL performance is higher in the initial 50 episodes and declines afterward, the opposite is true for No Sent. A2C. Hence, this aspect favors ITS-SentARL, as it seems to take fewer episodes to reach a more profitable stage.

5.3.2 Overall Analysis

Table 9 presents the general performance of both models and the BH benchmark when considering all assets, test periods, initializations, and episodes according to each transaction cost and evaluation metric. In this sense, these results summarize the behaviors observed in the test profile in Figure 11. For instance, the mean TR takes the average over the TR of the 100 test episodes. Hence, ITS-SentARL achieved 43% and 27% higher mean TR for the high-penalty and the no-penalty scenarios, respectively, than No Sent. A2C. Still, by comparing the change in models' performance between the no-penalty and high-penalty scenarios, it is clear that ITS-SentARL presented the highest decrease in TR from 2.83% to 1.79% (36.8% reduction), whereas No Sent. A2C goes from 1.98% to 1.41% (28.8% decline). Alternatively, while the no sentiment A2C was unable to achieve better mean TR than the BH strategy in neither TCs, ITS-SentARL outperformed the BH by 16.5% in the no-penalty scenario. The AR metric in Table 9 is an alternative representation of the TR values that mainly facilitates researchers and real-world investors to compare their results with ours. Subsequently, previous assessments regarding the TR results remain the same for the AR metric.

Table 9: Overall results of ITS-SentARL, No Sent. A2C, and BH according to total return TR, annualized return AR, and Sharpe ratio SR. Values in bold indicate the strategy that performed better considering the high-penalty scenario (TC 0.25%), while underlined value indicates which models reached more promising results for each transaction cost.

TC	Strategy	Mean TR	Mean AR	SR
-	BH	**2.43%**	**12.03%**	**0.64**
0.0%	No Sent. A2C	1.98%	9.73%	0.44
	ITS-SentARL	2.83%	14.12%	0.62
0.25%	No Sent. A2C	1.41%	6.86%	0.19
	ITS-SentARL	1.79%	8.80%	0.51

Source: Lima Paiva et al. (2021).

Moving on to the SR results in Table 9, most of the trends observed for TR are also present in this metric. For example, ITS-SentARL reached 41% and 169% higher SR values than No Sent. A2C in the no-penalty and high-penalty scenarios, respectively. Therefore, according to this metric and independent of penalty, ITS-SentARL presented a steadier performance and was more efficient in balancing risk and return to achieve its purposes. In addition, comparing the different TC scenarios shows that introducing operational costs causes No Sent. A2C to degrade its SR by 56.8% against a one-third lower reduction (17.7%) for ITS-SentARL. Hence, these SR values reinforce its stability qualities and suggest that ITS-SentARL is much less susceptible to the introduction of penalty dynamics. Still, ITS-SentARL displayed a 3.9% lower SR than the BH, which indicates this model is still slightly riskier than a passive strategy (i.e., BH).

There is a compelling aspect to dissect when observing the decrease in SR and TR that the introduction of TC caused in both models. For instance, the decrease in SR that ITS-SentARL presented was half of its TR reduction. Meanwhile, oppositely, No Sent. A2C displayed an SR degradation twice its TR decline. In resume, ITS-SentARL sustained a higher impact on its profitability than risk management capacity, while No Sent. A2C situation was the opposite. Considering all these aspects, ITS-SentARL seems to be a more risk-aware system. Thus, the market sentiment momentum information might be helping ITS-SentARL better assess each situation and thus grounding its balance of profitability and risk. Subsequently, ITS-SentARL can exploit higher profitability actions in no-penalty scenarios and retain a superior risk-management balance in high-penalty circumstances. Besides, ITS-SentARL stayed considerably more lucrative than its sentiment-free counterpart even with the profit reduction of the high-penalty scenario.

Then moving on to the results presented in Table 10, it is possible to observe the overall models' performance and the BH strategy according to the twenty different assets and two penalty scenarios in terms of SR. For example, in the case of the BA asset high-penalty scenario, ITS-SentARL achieved an SR (0.665) several times greater than BH (-0.056) and No Sent. A2C (0.083). Also, observe that ITS-SentARL outperformed No Sent. A2C according to the SR metric in 12 and 14 assets for the no-penalty and high-penalty scenarios, respectively. Furthermore, ITS-SentARL surpassed the BH for 10 (no-penalty) and 9 (high-penalty) assets, while the sentiment-free counterpart achieved the same for 9 and 7 cases. These results reinforce previous analysis and show the higher deterioration of No Sent. A2C model's performance when trading costs are introduced. Moreover, this analysis demonstrates that the overall sound performance of ITS-SentARL can not be attributed to outstanding results in some cases and instead is a consequence of consistent decision-making taken across various assets and circumstances.

Table 10: Sharpe ratio by asset and TC for BH, ITS-SentARL, and No Sent. A2C. Underlined values identify the best performance between models for a given transaction cost. Bold values identify the best performance considering high penalty scenario and BH.

		TC 0.0%		TC 0.25%	
Asset	BH	ITS-SentARL	No Sent. A2C	ITS-SentARL	No Sent. A2C
AAPL	-0.125	0.423	1.439	0.082	**1.299**
AMZN	**1.120**	0.497	0.263	0.454	0.190
BA	-0.056	1.738	-0.136	**0.665**	0.083
DIS	**0.252**	-0.489	-0.361	-0.291	-0.484
FB	**0.704**	0.042	-0.025	-0.102	0.156
GOOGL	**0.742**	-0.280	0.226	-0.287	-0.059
HD	**0.490**	1.273	1.047	0.412	0.000
INTC	-0.205	-0.175	-0.662	**0.554**	-0.351
JNJ	**0.912**	0.529	-0.209	0.189	-0.428
JPM	**0.037**	-0.195	-0.369	-0.150	-0.435
KO	**0.056**	-0.192	-0.171	-0.073	-0.109
MA	0.414	1.556	0.807	**1.379**	0.762
MSFT	**1.792**	1.112	0.778	0.915	0.629
NFLX	**1.249**	0.501	0.194	0.128	0.554
PFE	-0.177	-0.284	-0.072	-0.159	**-0.155**
PG	0.264	0.674	0.523	**0.572**	0.469
SPY	0.384	0.940	0.857	0.517	**0.535**
T	**-0.433**	-0.336	-0.289	-0.472	-0.770
V	0.594	1.059	1.835	**1.242**	0.692
XOM	-0.455	0.009	0.080	**-0.273**	-0.555

Source: Lima Paiva et al. (2021).

There are, however, two curious phenomenons to observe in Table 10. First, ITS-SentARL and No Sent. A2C improved SR results after introducing trading costs in six (DIS, INTC, JPM, KO, PFE, and V) and five assets (BA, FB, INTC, KO, and NFLX), respectively. Notably, two improvements occurred for the same assets (INTC and KO). These events could indicate that adding penalties might cause models to become more risk-averse and thus improve their risk balancing capacities for some assets. Nonetheless, even though they are the minority cases for each model, there is no clear explanation for this unexpected occurrence. Second, it is notable that for seven assets, after the introduction of trading costs, the better performing model changed from ITS-SentARL to No Sent. A2C (FB, NFLX, and SPY), or vice-versa (DIS, KO, T, and V). This phenomenon indicates that the market sentiment information can influence how a model reacts in a more adverse scenario and that, in most cases, it can be beneficial.

Table 11 depicts results regarding the TR metric instead of the previous one that reported SR values (Table 10). For instance, in the case of BA with the high-penalty scenario, ITS-SentARL achieved much greater TR (6.71%) than the BH (-2.14%) and No Sent. A2C (0.58%). Moreover, considering all twenty assets, ITS-SentARL achieved higher TR than No Sent. A2C in 10 and 13 assets for the no-penalty and high-penalty scenarios, respectively. At the same time, ITS-SentARL achieved better TR than the BH for 11 (no-penalty) and 10 (high-penalty) assets, while No Sent. A2C achieved the same for 10 and 9 cases. Similar to previous analysis regarding SR, these comparisons for TR show that the addition of trading costs causes No Sent. A2C to take a performance decrease greater than ITS-SentARL.

Table 11: Total average financial Return (%) by asset and TC for BH, ITS-SentARL, and No Sent. A2C. Underlined values identify the best performance between models for a given transaction cost. Bold values identify the best performance considering the high penalty scenario and BH.

		TC 0.0%		TC 0.25%	
Asset	BH	ITS-SentARL	No Sent. A2C	ITS-SentARL	No Sent. A2C
AAPL	-4.53	7.85	11.27	2.31	**11.32**
AMZN	**13.00**	8.09	5.51	7.96	4.36
BA	-2.14	10.64	-2.11	**6.71**	0.58
DIS	**6.27**	-6.14	-5.64	-3.32	-5.72
FB	**6.12**	0.50	-0.27	-1.76	1.93
GOOGL	**5.07**	-2.01	2.12	-1.91	-0.37
HD	**4.30**	4.38	6.18	2.29	0.00
INTC	-2.27	-1.59	-2.51	**1.66**	-1.38
JNJ	**2.21**	1.12	-0.73	0.44	-1.02
JPM	**0.83**	-2.26	-4.86	-1.24	-7.44
KO	**0.74**	-2.12	-2.04	-0.71	-1.01
MA	3.85	6.98	8.10	9.67	**11.15**
MSFT	**7.53**	11.16	7.07	6.55	6.12
NFLX	**12.47**	7.65	2.36	1.52	8.41
PFE	-0.90	-1.12	-0.33	**-0.60**	-0.71
PG	2.46	4.48	3.53	2.58	**2.69**
SPY	3.56	7.91	7.80	4.69	**4.78**
T	-5.01	-3.11	-2.12	**-2.97**	-3.28
V	3.50	4.05	5.24	**4.15**	2.54
XOM	-8.52	0.09	1.01	**-2.11**	-4.77

Source: Author's own production.

The previous phenomenons observed for the SR metric are also present when looking at the TR values. Initially, the performance improvement for assets when costs are introduced for ITS-SentARL (DIS, GOOGL, INTC, JPM, KO, MA, PFE, T, and V) and No Sent. A2C (AAPL, BA, FB, INTC, KO, MA, and NFLX). Again both models improved simultaneously for some assets (INTC, KO, and MA). Next, the change in the better-performing model by asset after the introduction of trading costs, where ITS-SentARL surpassed No Sent. A2C for seven assets (DIS, HD, KO, PFE, T, V, and XOM) and the other way around for four assets (FB, NFLX, PG, and SPY). Although similar conclusions can be drawn as before when looking at the SR metric, it is pretty noticeable that these phenomena seem more frequent for TR. This fact could be attributed to SR being a risk-management and consistency metric, whereas TR evaluates pure profitability. Thus these results again point out that the sentiment information can drastically impact a model's profitability while allowing it to be more consistent and stable.

By looking at the BH results, it is possible to observe which assets presented an average price appreciation or depreciation over time. For instance, only six assets (AAPL, BA, INTC, PFE, T, and XOM) presented an overall depreciation in the observed periods. Interestingly, among these six depreciating assets, ITS-SentARL outperformed No Sent. A2C and the BH in five cases (BA, INTC, PFE, T, XOM) for the high-penalty scenario. On the other hand considering the four instances (AAPL, MA, PG, SPY) where No Sent. A2C achieved higher TR than ITS-SentARL and the BH, only one was a depreciating asset (AAPL). These unique outcomes seem to indicate a tendency of ITS-SentARL to exhibit striking predominance against all benchmarks, primarily when assets underwent depreciation over time. Ultimately, it is not possible to know beforehand which assets will appreciate or not. However, depending on macroeconomic factors or long-lasting events (e.g., post-2008 market crash), it is conceivable to selectively employ ITS-SentARL over certain assets that could be more susceptible to devaluation in a given period. For instance, at the beginning of the Covid-19 pandemic, it was not far-fetched to correctly suppose that the aerospace industry would take a massive hit in sales due to uncertainty of flight restrictions. Thus, given the observed results, adopting ITS-SentARL for assets in this aerospace industry (e.g., BA) would be rather appropriate.

However, it is notable that these observed developments are particularly intriguing when recalling from Figure 6 that most assets exhibited a sentiment score distribution of their text instances that were slightly positively skewed. Nonetheless, previous studies (TETLOCK, 2007; TETLOCK; SAAR-TSECHANSKY; MACSKASSY, 2008) have shown that negative sentiment on the news can have more substantial power in estimating future market trends than positive texts. Thus, even though the amount of positive news is higher in most assets, negative news might better indicate future patterns. Therefore, analyzing the sentiment aspect of assets and the observed metrics for each model is very insightful. Hence, in Section 5.3.3, there is an in-depth investigation of such characteristics.

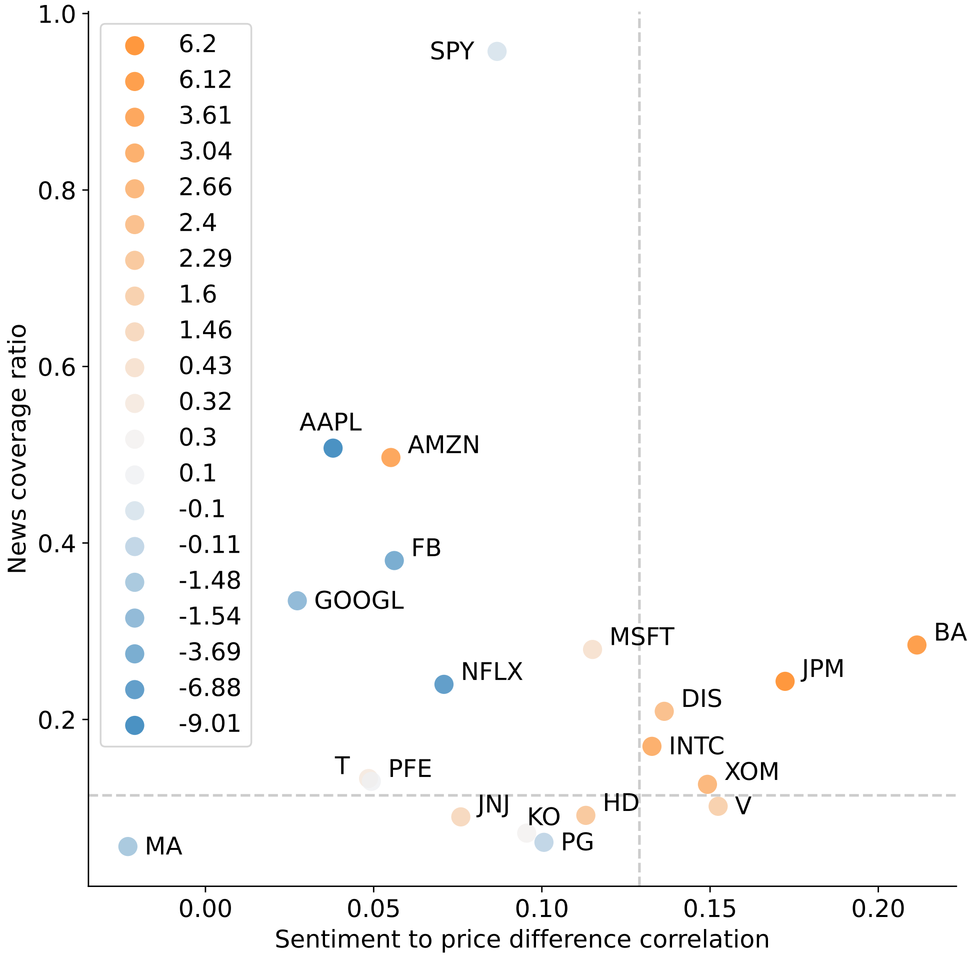
5.3.3 News Coverage, Sentiment Correlation and Total Return

When examining the impact of the news sentiment, it is essential to compare the performance improvement of ITS-SentARL over its sentiment-free counterpart under the different conditions of news coverage and correlation (without a shift) of sentiment-to-price time series. Thus, in Figure 12, the x-axis marks the correlation between news sentiment and price difference, the y-axis displays the news coverage, and the intensity of the color defines the TR difference between ITS-SentARL and No Sent. A2C for each asset, represented by a dot. For instance, very intense orange dots (BA, JPM, XOM, and others) mean an asset with a higher difference in TR between models favoring ITS-SentARL, while very intense blue dots mean the opposite (AAPL, NFLX, FB, and others).

At first, when examining the correlation axis (shown on the horizontal axis of Figure 12), it is noticeable that a higher correlation seems to correspond to an increased advantage for ITS-SentARL, which is in line with expectations. Nonetheless, it is worth noting that even a correlation as low as 0.1 can be sufficient to shift the balance in favor of ITS-SentARL. Furthermore, upon scrutinizing the news coverage axis (displayed on the vertical axis), we can confirm that news coverage can amplify ITS-SentARL's advantage for assets with similar correlations (such as XOM and V). However, when the correlation falls below a specific value (e.g., AAPL, FB, GOOGL, NFLX), an excessive amount of news appears to hinder ITS-SentARL's performance. A noteworthy exception is AMZN, where this phenomenon is not observed and could be considered an outlier. Additionally, it should be noted that capturing sentiment for the SPY asset may present difficulties as it consists of the price of 500 assets, and any given news item may only impact a small fraction of the index.

It is worth highlighting that several assets from the tech industry had poor correlations that resulted in unfavorable performance for ITS-SentARL. This trend implies that news related to tech firms, although more popular, can be more speculative and create a misleading impression of the overall market sentiment. In general, there appears to be a potential threshold above 11% news coverage and a correlation of 0.128, where ITS-SentARL demonstrates some of its most noteworthy performance gains over No Sent. A2C, indicating a higher likelihood of achieving better ITS-SentARL results. Hence, there

Figure 12: Difference in total return (%) between ITS-SentARL and No Sent. A2C by asset according to news coverage and sentiment-to-price time series correlation in the test set with TC 0.25%. Orange dots indicate situations where ITS-SentARL outperformed No Sent. A2C base model, while blue dots are the opposite. The intensity of colors indicates the magnitude of the percentage difference between models.



Source: Lima Paiva et al. (2021).

is a possibility of achieving better ITS-SentARL results beyond this threshold. However, these assumptions need further empirical validation or rigorous testing with more assets to establish a more robust and reliable general rule for the broader market.

6 CONCLUSION AND FUTURE WORK

This work presented ITS-SentARL for the single asset trading task, an effective architecture in identifying market sentiment momentum to achieve higher profit consistency than No Sent. A2C. In the presented experiments, considering the total return and Sharpe Ratio, ITS-SentARL outperformed No Sent. A2C for both transactions costs and the Buy&Hold strategy when there are no costs associated with the transactions. Moreover, the proposed architecture had reduced performance variation considering all the transaction costs, model parameters initialization, and periods. Finally, a lower bound requirement for textual data coverage and necessary correlation was identified to benefit from news information. In that sense, it would be necessary to monitor recent news coverage and sentiment correlation in new data to turn this architecture into a live application.

Results show that ITS-SentARL increased the profitability and stability of 14 assets out of 20 compared to No Sent. A2C. Then, looking at all assets together, ITS-SentARL outperforms No Sent. A2C in average total return in the no-penalty and high-penalty scenarios by 43% and 27%, respectively. Moreover, ITS-SentARL exhibited considerably improved stability with a Sharpe Ratio 141% better in the no-penalty TC and 270% better in the high-penalty case. The increase in TC negatively impacted No Sent. A2C with a reduction of 57% in Sharpe Ratio, while the ITS-SentARL suffered a much smaller performance reduction of 18%. ITS-SentARL also presented better results in three of the five initializations and periods (including the high volatility COVID-19 Pandemic). Ultimately, ITS-SentARL outperformed the BH strategy for 11 assets in the no-penalty scenario with an overall 17% higher total average return.

This study also concluded that:

- ITS-SentARL's performance depends on minimum textual data coverage and correlation between price and sentiment, which may guide the selection of assets.
- Market mood information helps shape and stabilize the system strategy by influencing the frequency of action shifts that could harm long-term cumulative reward.

Although there are assuring results to improve performance over the BH baseline, there are still plenty of promising options for examining the market sentiment incorporation as an additional state feature to RL methods. Furthermore, the NLP community has long adopted social networks as textual sources for trading as it provides a vast amount of data and can bring fresher information. Thus, even though this data source might entail additional data preprocessing to reduce noise, it might support increased trading frequency, better news coverage, and sentiment correlation.

From the technical analysis perspective, employing technical indicators, such as moving averages, could help smooth out the price time series. Furthermore, as Carapuço, Neves and Horta (2018) indicated, it might be worth including a feature to the agent-state that represents the net worth of the agent at each given instant. Finally, there might be room for improvement in adopting a stop-loss mechanism (ALMAHDI; YANG, 2017, 2019) so that ITS-SentARL avoids significant losses.

REFERENCES

ABOUSSALAH, A. M.; LEE, C.-G. Continuous control with Stacked Deep Dynamic Recurrent Reinforcement Learning for portfolio optimization. *Expert Systems with Applications*, Elsevier Ltd, v. 140, p. 112891, feb 2020. ISSN 09574174.

ALIMORADI, M. R.; HUSSEINZADEH KASHAN, A. A league championship algorithm equipped with network structure and backward Q-learning for extracting stock trading rules. *Applied Soft Computing*, Elsevier B.V., v. 68, p. 478–493, jul 2018. ISSN 15684946.

ALMAHDI, S.; YANG, S. Y. An adaptive portfolio trading system: A risk-return portfolio optimization using recurrent reinforcement learning with expected maximum drawdown. *Expert Systems with Applications*, Elsevier Ltd, v. 87, p. 267–279, nov 2017. ISSN 09574174.

ALMAHDI, S.; YANG, S. Y. A constrained portfolio trading system using particle swarm algorithm and recurrent reinforcement learning. *Expert Systems with Applications*, Elsevier Ltd, v. 130, p. 145–156, 2019. ISSN 09574174.

ANTWEILER, W.; FRANK, M. Z. Is All That Talk Just Noise? The Information Content of Internet Stock Message Boards. *The Journal of Finance*, [American Finance Association, Wiley], v. 59, n. 3, p. 1259–1294, jun 2004. ISSN 00221082.

ARIEL, R. A. A monthly effect in stock returns. *Journal of Financial Economics*, v. 18, n. 1, p. 161–174, mar 1987. ISSN 0304405X.

BAKER, M.; WURGLER, J. Investor Sentiment in the Stock Market. *Journal of Economic Perspectives*, v. 21, n. 2, p. 129–151, apr 2007. ISSN 0895-3309.

BARBERIS, N.; SHLEIFER, A.; VISHNY, R. A model of investor sentiment. *Journal of Financial Economics*, v. 49, n. 3, p. 307–343, sep 1998. ISSN 0304405X.

BARBERIS, N.; THALER, R. Chapter 18 A survey of behavioral finance. In: . [S.l.: s.n.], 2003. p. 1053–1128.

BOLLEN, J.; MAO, H.; ZENG, X. Twitter mood predicts the stock market. *Journal of Computational Science*, v. 2, n. 1, p. 1–8, mar 2011. ISSN 18777503.

BROCKMAN, G.; CHEUNG, V.; PETTERSSON, L.; SCHNEIDER, J.; SCHULMAN, J.; TANG, J.; ZAREMBA, W. *OpenAI Gym*. 2016.

CARAPUÇO, J.; NEVES, R.; HORTA, N. Reinforcement learning applied to Forex trading. *Applied Soft Computing*, Elsevier B.V., v. 73, p. 783–794, dec 2018. ISSN 15684946.

CORTIS, K.; FREITAS, A.; DAUDERT, T.; HUERLIMANN, M.; ZARROUK, M.; HANDSCHUH, S.; DAVIS, B. SemEval-2017 Task 5: Fine-Grained Sentiment Analysis on Financial Microblogs and News. In: *Proceedings of the 11th International Workshop* *on Semantic Evaluation (SemEval-2017)*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2017. p. 519–535.

DAVIS, B.; CORTIS, K.; VASILIU, L.; KOUMPIS, A.; MCDERMOTT, R.; HANDSCHUH, S. Social Sentiment Indices Powered by X-Scores. In: *ALLDATA 2016, The Second International Conference on Big Data, Small Data, Linked Data and Open Data*. Lisbon, Portugal: [s.n.], 2016.

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*, v. 28, n. 3, p. 653–664, mar 2017. ISSN 2162-237X.

DUAN, J.; ZHANG, Y.; DING, X.; CHANG, C.-Y.; LIU, T. Learning target-specific representations of financial news documents for cumulative abnormal return prediction. In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, 2018. p. 2823–2833.

EILERS, D.; DUNIS, C. L.; METTENHEIM, H.-J. von; BREITNER, M. H. Intelligent trading of seasonal effects: A decision support algorithm based on reinforcement learning. *Decision Support Systems*, Elsevier B.V., v. 64, p. 100–108, aug 2014. ISSN 01679236.

FAMA, E. F. Random Walks in Stock Market Prices. *Financial Analysts Journal*, 1965. ISSN 0015-198X.

FAMA, E. F. Efficient Capital Markets: A Review of Theory and Empirical Work. *The Journal of Finance*, v. 25, n. 2, p. 383, may 1970. ISSN 00221082.

FAWAZ, H. I.; FORESTIER, G.; WEBER, J.; IDOUMGHAR, L.; MULLER, P.-A. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, Springer Science and Business Media LLC, v. 33, n. 4, p. 917–963, Mar 2019. ISSN 1573-756X.

FERREIRA, T.; Lima Paiva, F. C.; SILVA, R.; PAULA, A.; COSTA, A.; CUGNASCA, C. Assessing regression-based sentiment analysis techniques in financial texts. In: *Proceedings of XVI National Meeting on Artificial and Computational Intelligence*. Porto Alegre, RS, Brasil: SBC, 2020. p. 729–740. ISSN 0000-0000.

FEUERRIEGEL, S.; GORDON, J. News-based forecasts of macroeconomic indicators: A semantic path model for interpretable predictions. *European Journal of Operational Research*, Elsevier B.V., v. 272, n. 1, p. 162–175, jan 2019. ISSN 03772217.

FEUERRIEGEL, S.; PRENDINGER, H. News-based trading strategies. *Decision Support Systems*, v. 90, p. 65–74, oct 2016. ISSN 01679236.

GABRIELSSON, P.; JOHANSSON, U. High-Frequency Equity Index Futures Trading Using Recurrent Reinforcement Learning with Candlesticks. In: *2015 IEEE Symposium Series on Computational Intelligence*. [S.l.]: IEEE, 2015. p. 734–741. ISBN 978-1-4799-7560-0.

GIACOMAZZI DANTAS, S.; GUERREIRO E SILVA, D. Equity Trading at the Brazilian Stock Market Using a Q-Learning Based System. In: *2018 7th Brazilian Conference on Intelligent Systems (BRACIS)*. [S.l.]: IEEE, 2018. p. 133–138. ISBN 978-1-5386-8023-0.

GLASSERMAN, P.; KRSTOVSKI, K.; LALIBERTE, P.; MAMAYSKY, H. Choosing news topics to explain stock market returns. In: *Proceedings of the First ACM International Conference on AI in Finance*. New York, NY, USA: ACM, 2020. p. 1–8. ISBN 9781450375849.

HENDERSON, P.; ISLAM, R.; BACHMAN, P.; PINEAU, J.; PRECUP, D.; MEGER, D. Deep Reinforcement Learning That Matters. In: *Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*. [S.l.: s.n.], 2018.

HENRIQUE, B. M.; SOBREIRO, V. A.; KIMURA, H. Literature review: Machine learning techniques applied to financial market prediction. *Expert Systems with Applications*, v. 124, p. 226–251, jun 2019. ISSN 09574174.

HIRCHOUA, B.; OUHBI, B.; FRIKH, B. Deep reinforcement learning based trading agents: Risk curiosity driven learning for financial rules-based policy. *Expert Systems with Applications*, v. 170, p. 114553, may 2021. ISSN 09574174.

HUTTO, C.; GILBERT, E. VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. In: *Eighth International AAAI Conference on Weblogs and Social Media*. [S.l.: s.n.], 2014. p. 18.

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*, Elsevier Ltd, v. 117, p. 125–138, mar 2019. ISSN 09574174.

JIANG, M.; LAN, M.; WU, Y. ECNU at SemEval-2017 Task 5: An Ensemble of Regression Algorithms with Effective Features for Fine-Grained Sentiment Analysis in Financial Domain. In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2017. p. 888–893.

KADAN, O.; MICHAELY, R.; MOULTON, P. C. Speculating on Private Information: Buy the Rumor, Sell the Fact. *SSRN Electronic Journal*, 2014. ISSN 1556-5068.

KANG, Q.; ZHOU, H.; KANG, Y. An Asynchronous Advantage Actor-Critic Reinforcement Learning Method for Stock Selection and Portfolio Management. In: *Proceedings of the 2nd International Conference on Big Data Research - ICBDR 2018*. New York, New York, USA: ACM Press, 2018. p. 141–145. ISBN 9781450364768.

KHADJEH-NASSIRTOUSSI, A.; AGHABOZORGI, S.; Ying Wah, T.; NGO, D. C. L. Text mining for market prediction: A systematic review. *Expert Systems with Applications*, v. 41, n. 16, p. 7653–7670, nov 2014.

KHADJEH-NASSIRTOUSSI, A.; AGHABOZORGI, S.; Ying Wah, T.; NGO, D. C. L. Text mining of news-headlines for FOREX market prediction: A Multi-layer Dimension Reduction Algorithm with semantics and sentiment. *Expert Systems with Applications*, Elsevier Ltd, v. 42, n. 1, p. 306–324, jan 2015. ISSN 09574174.

KIM, Y. Convolutional Neural Networks for Sentence Classification. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. [S.l.: s.n.], 2014. p. 1746–1751.

KING, B.; COOTNER, P. H. The Random Character of Stock Market Prices. *The Journal of Finance*, 1965. ISSN 00221082.

LI, T.; DALEN, J. van; REES, P. J. van. More than just Noise? Examining the Information Content of Stock Microblogs on Financial Markets. *Journal of Information Technology*, v. 33, n. 1, p. 50–69, mar 2018. ISSN 0268-3962.

LI, Y.; ZHENG, W.; ZHENG, Z. Deep Robust Reinforcement Learning for Practical Algorithmic Trading. *IEEE Access*, v. 7, p. 108014–108022, 2019. ISSN 2169-3536.

Lima Paiva, F. C.; FELIZARDO, L. K.; BIANCHI, R. A. d. C. B.; COSTA, A. H. R. Intelligent trading systems: A sentiment-aware reinforcement learning approach. In: *Proceedings of the Second ACM International Conference on AI in Finance*. New York, NY, USA: Association for Computing Machinery, 2021. (ICAIF '21).

LO, A. W. The Adaptive Markets Hypothesis. *The Journal of Portfolio Management*, v. 30, n. 5, p. 15–29, jan 2004. ISSN 0095-4918.

LOUGHRAN, T.; MCDONALD, B. When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks. *The Journal of Finance*, John Wiley & Sons, Ltd (10.1111), v. 66, n. 1, p. 35–65, feb 2011. ISSN 00221082.

LOUGHRAN, T.; MCDONALD, B. Textual Analysis in Accounting and Finance: A Survey. *Journal of Accounting Research*, v. 54, n. 4, p. 1187–1230, sep 2016. ISSN 00218456.

LUCCA, D. O.; MOENCH, E. The Pre-FOMC Announcement Drift. *The Journal of Finance*, v. 70, n. 1, p. 329–371, feb 2015. ISSN 00221082.

MA, C.; ZHANG, J.; LIU, J.; JI, L.; GAO, F. A parallel multi-module deep reinforcement learning algorithm for stock trading. *Neurocomputing*, v. 449, p. 290–302, aug 2021. ISSN 09252312.

MANSAR, Y.; GATTI, L.; FERRADANS, S.; GUERINI, M.; STAIANO, J. Fortia-FBK at SemEval-2017 Task 5: Bullish or Bearish? Inferring Sentiment towards Brands from Financial News Headlines. In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2017. p. 817–822.

MARINGER, D.; ZHANG, J. Transition variable selection for regime switching recurrent reinforcement learning. In: . [S.l.]: IEEE, 2014. p. 407–413. ISBN 9781479923809.

MEDHAT, W.; HASSAN, A.; KORASHY, H. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, v. 5, n. 4, p. 1093–1113, dec 2014. ISSN 20904479.

MNIH, V.; BADIA, A. P.; MIRZA, L.; GRAVES, A.; HARLEY, T.; LILLICRAP, T. P.; SILVER, D.; KAVUKCUOGLU, K. Asynchronous methods for deep reinforcement learning. In: *33rd International Conference on Machine Learning, ICML 2016*. [S.l.: s.n.], 2016. ISBN 9781510829008.

MNIH, V.; KAVUKCUOGLU, K.; SILVER, D.; RUSU, A. A.; VENESS, J.; BELLEMARE, M. G.; GRAVES, A.; RIEDMILLER, M.; FIDJELAND, A. K.; OSTROVSKI, G.; PETERSEN, S.; BEATTIE, C.; SADIK, A.; ANTONOGLOU, I.; KING, H.; KUMARAN, D.; WIERSTRA, D.; LEGG, S.; HASSABIS, D. Human-level control through deep reinforcement learning. *Nature*, v. 518, n. 7540, p. 529–533, feb 2015. ISSN 0028-0836.

MOODY, J.; SAFFELL, M.; LIAO, Y.; WU, L. Reinforcement learning for trading systems and portfolios: Immediate vs future rewards. In: REFENES, A.-P. N.; BURGESS, A. N.; MOODY, J. E. (Ed.). *Decision Technologies for Computational Finance: Proceedings of the fifth International Conference Computational Finance*. Boston, MA: Springer US, 1998. p. 129–140. ISBN 978-1-4615-5625-1.

MOODY, J.; WU, L. Optimization of trading systems and portfolios. In: *Proceedings of the IEEE/IAFE 1997 Computational Intelligence for Financial Engineering (CIFEr)*. [S.l.]: IEEE, 1997. p. 300–307. ISBN 0-7803-4133-3.

NGUYEN, T. H.; SHIRAI, K. Topic Modeling based Sentiment Analysis on Social Media for Stock Market Prediction. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2015.

OLIVEIRA, N.; CORTEZ, P.; AREAL, N. The impact of microblogging data for stock market prediction: Using Twitter to predict returns, volatility, trading volume and survey sentiment indices. *Expert Systems with Applications*, Elsevier Ltd, v. 73, p. 125–144, 2017. ISSN 09574174.

PARK, H.; SIM, M. K.; CHOI, D. G. An intelligent financial portfolio trading strategy using deep Q-learning. *Expert Systems with Applications*, Elsevier Ltd, v. 158, p. 113573, nov 2020. ISSN 09574174.

PENDHARKAR, P. C.; CUSATIS, P. Trading financial indices with reinforcement learning agents. *Expert Systems with Applications*, Elsevier Ltd, v. 103, p. 1–13, aug 2018. ISSN 09574174.

PENNINGTON, J.; SOCHER, R.; MANNING, C. Glove: Global Vectors for Word Representation. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2014. p. 1532–1543.

PINHEIRO, L. dos S.; DRAS, M. Stock Market Prediction with Deep Learning: A Character-based Neural Language Model for Event-based Trading. In: *Proceedings of the Australasian Language Technology Association Workshop 2017*. Brisbane, BNE, Australia: [s.n.], 2017. p. 6–15.

PONOMAREV, E. S.; OSELEDETS, I. V.; CICHOCKI, A. S. Using Reinforcement Learning in the Algorithmic Trading Problem. *Journal of Communications Technology and Electronics*, v. 64, n. 12, p. 1450–1457, dec 2019. ISSN 1064-2269.

RAFFIN, A.; HILL, A.; ERNESTUS, M.; GLEAVE, A.; KANERVISTO, A.; DORMANN, N. *Stable Baselines3*. [S.l.]: GitHub, 2019.

REKABSAZ, N.; LUPU, M.; BAKLANOV, A.; DÜR, A.; ANDERSSON, L.; HANBURY, A. Volatility Prediction using Financial Disclosures Sentiments with Word Embedding-based IR Models. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2017. p. 1712–1721.

REN, R.; WU, D. D.; LIU, T. Forecasting Stock Market Movement Direction Using Sentiment Analysis and Support Vector Machine. *IEEE Systems Journal*, v. 13, n. 1, p. 760–770, mar 2019. ISSN 1932-8184.

RUSSELL, S.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. 3rd. ed. [S.l.]: Pearson, 2009.

SHARPE, W. F. Mutual Fund Performance. *The Journal of Business*, v. 39, n. S1, p. 119, jan 1966. ISSN 0021-9398.

SILVER, D. *Lectures on Reinforcement Learning*. 2015. URL: ⟨https://www.davidsilver.uk/teaching/⟩.

SPOONER, T.; FEARNLEY, J.; SAVANI, R.; KOUKORINIS, A. Market Making via Reinforcement Learning. In: *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. Stockholm, Sweden: International Foundation for Autonomous Agents and Multiagent Systems, 2018. v. 1, n. AAMAS '18, p. 434–442.

SUTTON, R. S.; BARTO, A. G. *Reinforcement Learning: An Introduction*. 2nd. ed. Cambridge, MA, USA: A Bradford Book, 2018. ISBN 0262039249.

TALEB, N. N. *The Black Swan: The Impact of the Highly Improbable*. [S.l.]: Random House Group, 2007. ISBN 1400063515.

TETLOCK, P. C. Giving Content to Investor Sentiment: The Role of Media in the Stock Market. *The Journal of Finance*, v. 62, n. 3, p. 1139–1168, jun 2007. ISSN 00221082.

TETLOCK, P. C.; SAAR-TSECHANSKY, M.; MACSKASSY, S. More Than Words: Quantifying Language to Measure Firms' Fundamentals. *The Journal of Finance*, v. 63, n. 3, p. 1437–1467, jun 2008. ISSN 00221082.

THÉATE, T.; ERNST, D. An application of deep reinforcement learning to algorithmic trading. *Expert Systems with Applications*, v. 173, p. 114632, jul 2021. ISSN 09574174.

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*, v. 32, n. 7, p. 2837–2846, jul 2021. ISSN 2162-237X.

TSAY, R. S. *Analysis of financial time series*. [S.l.: s.n.], 2010. 1–677 p. ISBN 9781118017098.

VASILEIOU, E. Long Live Day of the Week Patterns and the Financial Trends' Role. Evidence from the Greek Stock Market during the Euro Era (2002-12). *Ssrn*, v. 12, n. 3, p. 19–32, 2013.

WANG, J.; ZHANG, Y.; TANG, K.; WU, J.; XIONG, Z. AlphaStock. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining - KDD '19*. New York, New York, USA: ACM Press, 2019. p. 1900–1908. ISBN 9781450362016.

WENG, L.; SUN, X.; XIA, M.; LIU, J.; XU, Y. Portfolio trading system of digital currencies: A deep reinforcement learning with multidimensional attention gating mechanism. *Neurocomputing*, Elsevier B.V., v. 402, p. 171–182, aug 2020. ISSN 09252312.

WILLIAMS, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 1992. ISSN 0885-6125.

WU, J.; WANG, C.; XIONG, L.; SUN, H. Quantitative Trading on Stock Market Based on Deep Reinforcement Learning. *2019 International Joint Conference on Neural Networks (IJCNN)*, IEEE, n. July, p. 1–8, 2019.

WU, Y.; MANSIMOV, E.; LIAO, S.; GROSSE, R.; BA, J. Scalable Trust-Region Method for Deep Reinforcement Learning Using Kronecker-Factored Approximation. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2017. (NIPS'17), p. 5285–5294. ISBN 9781510860964.

XING, F. Z.; CAMBRIA, E.; WELSCH, R. E. Intelligent Asset Allocation via Market Sentiment Views. *IEEE Computational Intelligence Magazine*, IEEE, v. 13, n. 4, p. 25–34, nov 2018. ISSN 1556-603X.

YANG, S. Y.; YU, Y.; ALMAHDI, S. An investor sentiment reward-based trading system using Gaussian inverse reinforcement learning algorithm. *Expert Systems with Applications*, Elsevier Ltd, v. 114, p. 388–401, dec 2018. ISSN 09574174.

YE, Y.; PEI, H.; WANG, B.; CHEN, P.-Y.; ZHU, Y.; XIAO, J.; LI, B. Reinforcement-Learning Based Portfolio Management with Augmented Asset Movement Prediction States. *Proceedings of the AAAI Conference on Artificial Intelligence*, v. 34, n. 01, p. 1112–1119, apr 2020. ISSN 2374-3468.

YU, P.; LEE, J. S.; KULYATIN, I.; SHI, Z.; DASGUPTA, S. Model-based Deep Reinforcement Learning for Dynamic Portfolio Optimization. jan 2019.

ZHANG, J.; MARINGER, D. Two parameter update schemes for recurrent reinforcement learning. In: *2014 IEEE Congress on Evolutionary Computation (CEC)*. [S.l.]: IEEE, 2014. p. 1449–1453. ISBN 978-1-4799-1488-3.

ZHANG, J.; MARINGER, D. Using a Genetic Algorithm to Improve Recurrent Reinforcement Learning for Equity Trading. *Computational Economics*, Springer US, v. 47, n. 4, p. 551–567, apr 2016. ISSN 0927-7099.