IMPERIAL

Imperial College London

Department of Mathematics

Developing and Backtesting a Trading Strategy Using Large Language Models, Macroeconomic and Technical Indicators

Author: Alireza Kargarzadeh (CID: 02092220)

A thesis submitted for the degree of

MSc in Mathematics and Finance, 2023-2024

Declaration

The work contained in this thesis is my own work unless otherwise stated.

Alireza Kargarzadeh

Acknowledgements

I would like to express my heartfelt gratitude to the MSc Mathematics and Finance program directors and staff, especially Ms. Rula Murtada, Professor Jack Jacquier, and Professor Damiano Brigo, for giving me the opportunity to study for this degree at Imperial College London. Their support made it possible for me to switch my field of study to something I truly enjoy.

I am deeply thankful to Dr. Eyal Neuman, my internal supervisor and personal tutor, for his invaluable guidance and support throughout my studies. Additionally, I would like to extend my appreciation to everyone at Zanista AI who assisted me with this project, particularly Dr. Arman Khaleidan and Dr. Nariman Khaledian, for their guidance, contribution, and the wonderful times we shared.

I would like to dedicate this work to the memory of my grandparents, who passed away during the course of my studies. May they rest in peace.

I am especially grateful to my family—my mother, father, and sister—for their unwavering support. Last but not least, I would like to thank my partner, Fatima, for standing by my side through every step of this journey.

Abstract

This thesis explores the development and backtesting of a trading strategy that integrates Large Language Models (LLMs) with macroeconomic and technical indicators. The primary objective is to enhance stock return predictions by leveraging LLMs to analyze vast amounts of textual data, particularly financial news. The research focuses on small-cap stocks from the Russell 2000 Index, where market inefficiencies are more pronounced, providing opportunities to capitalize on delays in price adjustments. We combined sentiment analysis derived from financial news using `GPT-4o` with macroeconomic and technical signals to inform trading decisions, effectively utilizing the synergy between qualitative and quantitative data. Our methodology introduces a novel approach to stock filtration using macroeconomic indicators and sentiment quantification, incorporating a decay function to model the diminishing impact of news over time. The strategy's performance was evaluated throughout 2022 and 2023, considering various holding periods and transaction costs. Our flagship "Pure Alpha" long-short strategy, which isolates stocks moving independently of their co-moving indicators, achieved Sharpe ratios of 3.64 and 5.10 in 2022 and 2023, respectively, for a one-day holding period. Notably, the strategy maintained profitability even after accounting for substantial transaction costs, consistently outperforming the Russell 2000 benchmark, highlighting its robustness and potential for real-world application. This study contributes to the growing body of research on LLM applications in finance, offering insights into integrating advanced language models with traditional financial analysis techniques.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In today's rapidly evolving financial landscape, the development and implementation of sophisticated trading strategies and algorithmic trading systems have become crucial for success in the markets. These advanced approaches are essential tools for both retail and institutional investors seeking to optimize returns and manage risk effectively. The effectiveness of these strategies relies on accurate signal generation, which enhances decision-making and the ability to respond swiftly.

The financial sector is characterized by an abundance of data, with one particularly valuable source being the vast amount of textual information, including real-time news about stocks. This information offers critical signals that can significantly enhance decision-making in trading strategies. Research has shown that news is often incorporated into stock prices with a certain delay, leading to inefficiencies. This delay is more pronounced in stocks with smaller market capitalizations, where market inefficiencies tend to be greater[40, 11]. Although these inefficiencies are limited by the constraints of arbitrage, they still present profitable opportunities when effectively exploited in trading strategies [11].

Historically, the vast amount of textual data, particularly from news sources, was analyzed using traditional methods of sentiment analysis[15, 66, 67, 44, 22]. These methods often fail to capture the nuances and complexities inherent in news, resulting in suboptimal outcomes. However, recent technological advancements have significantly transformed this landscape. Large Language Models (LLMs) have emerged as powerful tools, capable of capturing market sentiment and predicting movements by analyzing vast amounts of textual data.

Unlike previous studies on the use of LLMs in analyzing news for predicting returns[40, 11, 34], which primarily focused on the individual impact of news on returns, we have developed an innovative approach that considers the cumulative effects of all news leading up to an event. This is accomplished through the implementation of a decay function that models the diminishing impact of news over time, mirroring the real-world phenomenon where older news gradually loses its influence on market sentiment. By integrating the temporal aspect of news impact, our approach provides a more comprehensive and realistic model of how information flows affect market dynamics. This methodology not only captures the immediate effects of news but also accounts for the lingering influence of past events.

Despite these advancements, the task of analyzing news for all stocks remains computationally expensive and time-consuming. To address this challenge, a filtration process is necessary to narrow down the number of stocks subjected to sentiment analysis, thereby balancing the depth of insight with practical limitations. This study employed a novel quantitative approach to achieve this filtration, utilizing factors such as stock-specific data and macroeconomic indicators. These macroeconomic data were also utilized to dif-

ferentiate a stock's movement from the broader trends of associated indicators, such as sector-wide or market-wide shifts. This approach helps to distinguish stock-specific events (alpha) from broader market or sector movements (beta).

This thesis explores the integration of signals derived from sentiment analysis of news, provided by *NewsWitch*©[1], with signals from technical and macroeconomic indicators to develop reliable trading strategies. In this study, we devised an LLM-powered trading strategy that synergistically combines qualitative and quantitative signals, aiming to leverage the abundance of data in the financial market to outperform the market. We backtested the strategy over 2022 and 2023, evaluating performance metrics and comparing the results with the relevant benchmarks. By using the strengths of both qualitative and quantitative approaches, this strategy seeks to maximize the potential for market success. It should be noted that this thesis will be expanded for publication purposes.

Thesis Organization: Chapter 1 provides fundamental information about Large Language Models (LLMs) and their operational mechanisms. In Chapter 2, we present the detailed methodology of the study. This begins with the mathematical framework, incorporating news into our formal models, followed by an explanation of the signals used and the methods for the strategy's performance evaluation. Chapter 3 subsequently presents the results of the study. In Chapter 4, we discuss the findings, limitations, and potential directions for future work.

[1] *NewsWitch* is a service provided by *ZanistaAI* that delivers processed, clean news for any list of stocks over a particular timeframe.

Chapter 2

Background

2.1 Large Language Models

2.1.1 Foundation of Large Language Models

In this section, we will explain how Large Language Models (LLMs) operate. To begin, we will explore the level at which LLMs are positioned within the hierarchy of computational models.

As depicted in 2.1, LLMs are situated at the intersection of Natural Language Processing (NLP) and Deep Learning (DL). DL itself is a specialized branch of Machine Learning (ML), which is a subset of Artificial Intelligence (AI). This nested structure illustrates that LLMs employ DL and NLP techniques to process and generate language, leveraging the foundational principles of AI and ML.

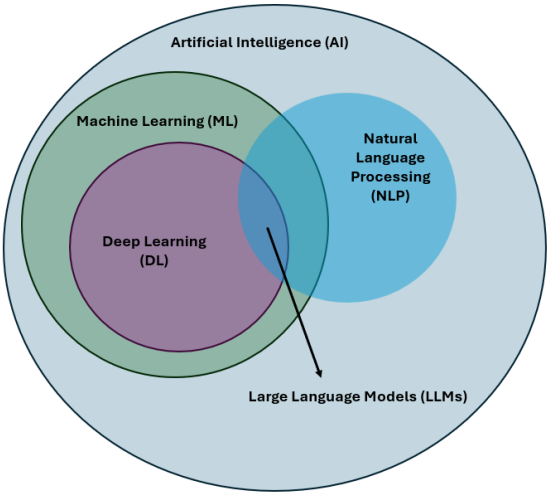

Figure 2.1: LLMs Hierarchy

LLMs are powerful models initially trained on extensive datasets to handle general language tasks, such as text generation and classification. These models are highly flexible and can be adapted for various industries, including finance. By fine-tuning with smaller, specialized datasets, they can address specific challenges in more targeted domains [33].

Training, Tuning, Fine-Tuning, and zero-shot: *Training* is the initial and most extensive phase where the model learns from a vast amount of data. During this phase, the model is exposed to large datasets, which could be diverse text from books, articles, websites, and other sources. The objective is for the model to learn the structure of the language, including grammar, syntax, and semantics. Training of complex models

is mainly self-supervised, a type of unsupervised learning during which the model itself generates labels for the data[33]. For example, the model can mask a word in a sentence and learn to predict it using the surrounding words in the sentence [11].

Tuning, sometimes referred to as hyperparameter tuning, involves adjusting the parameters of the model that govern the model architecture and the learning process. These parameters might include the learning rate, the number of layers in the neural network, batch size, and more. The tuning is crucial to optimize model performance and ensure it doesn't overfit on the training data[36].

Fine-tuning is a more targeted approach to training, where the pre-trained model is further trained (fine-tuned) on a smaller, specific dataset. This is done to adapt the general capabilities of the model to particular tasks or domains without the need to train from scratch. Fine-tuning adjusts the parameters of the pre-trained model slightly to specialize it for tasks like sentiment analysis, question answering, or legal document analysis[49]. While the initial training phase is conducted primarily using self-supervised learning, fine-tuning is performed using supervised learning, where the input data is labelled with both inputs and corresponding outputs [11].

Zero-shot (or few-shot) describes a model's capacity to accurately predict or execute tasks for which it has not been specifically trained, utilizing its pre-existing knowledge and ability to generalize[49].

Parameters and Hyperparameters: *Hyperparameters* are set before training and remain constant during the training process. Examples include learning rate, batch size, number of layers, and dropout rate. An example of hyperparameter is *temperature* that controls the randomness of the model's predictions. It is used during the text generation process to influence the creativity and diversity of the generated text. *Parameters* are learned during training and updated in each iteration to minimize the loss function[11].

It should be noted that "weights" and "parameters" are usually being used interchangeably in the context of LLM. However, to be more precise, weights are a subset of parameters. They specifically refer to the coefficients that are multiplied by the input features as they pass through the neurons in each layer of the neural network. LLMs are typically very large neural networks, with hundreds of millions to billions of parameters, which allow them to capture complex patterns in language data.

A key characteristic of LLMs is their number of parameters. The capacity of a model often correlates with its parameter count; for example, models like GPT-3 have billions of parameters. While this allows them to discern complex patterns in data, it also necessitates substantial computational resources[6].

A **token** refers to a piece of text that the model processes. It can be as small as a part of a word or as large as a word or sometimes even a small phrase, depending on the language and tokenizer used by the model. **Tokenization** is the process of breaking down text into smaller parts (tokens) that can be processed by the model. The tokenizer takes raw text input and splits it into these tokens according to specific rules. While traditional approaches often involve segmenting text into individual words based on predefined rules, LLMs employ more sophisticated tokenization techniques. These methods decompose uncommon words into smaller, semantically meaningful subword units. This strategy effectively mitigates the challenge of data sparsity by enabling the reuse of these subword tokens across various contexts, thus enhancing their frequency of occurrence within the dataset[11].

Embeddings are an important component of LLMs, serving as a form of representation learning where complex, high-dimensional data such as text is transformed into a lower-dimensional vector space. During the initial phase of training, through the process of tokenization, embeddings convert tokens—words and phrases—into vectors that capture the contextual nuances and relationships between different elements of the language.

In this vector space, distances between vectors reflect the similarity of their original data points, allowing similar items to be positioned closer together. For instance, synonyms, which are distinct tokens with related meanings, are typically found near each other, facilitating the model's ability to understand semantic similarities. This arrangement enables LLMs to perform operations on the text and grasp both its semantic and syntactic structures. As training progresses, these embeddings are refined, becoming increasingly sophisticated in representing linguistic features. This dynamic adjustment is crucial for the model's ability to handle complex language tasks effectively, laying the groundwork for all subsequent tasks that the model performs, including parsing sentence structure, recognizing the relevance of terms based on context, and generating coherent text outputs. This embedding process is foundational to the transformative capabilities of LLMs in interpreting and generating human-like text[18, 47].

When LLMs generate embeddings for pieces of text, **cosine similarity** can be used to compare these embeddings to determine how similar the texts are. For instance, finding similar sentences or paragraphs in a document. Cosine similarity measures the cosine of the angle between two vectors in a multi-dimensional space. It provides a similarity score that ranges from -1 to 1[47].

LLMs Architecture: LLMs are built on neural network architectures. Neural networks are a broad class of models designed to learn patterns from data through interconnected layers of neurons. LLMs are a specific type of neural network based on the transformer architecture, which excels in sequence processing tasks. **Transformers** are specifically designed to handle sequential data like text.

Introduced in the seminal paper "Attention is All You Need" in 2017 by Vaswani et al.[72], transformers employ a **Self-Attention** Mechanism. This mechanism enables the model to assess the relevance of each word within a sentence, irrespective of their positional relationship, thereby effectively capturing dependencies between words.

Transformers have **Encoder-Decoder Structure**, consisting of an encoder that processes the input sequence and a decoder that generates the output sequence. Each part consists of multiple layers of self-attention and feedforward networks[72].

In more detail, throughout the encoding phase, the input sequential data is *tokenized*. Each token is then converted into an *embedding vector*. This process happens in parallel for tokens, so the order of tokens is not necessarily preserved. To address this, **positional encodings** are added to embedding vectors to inject information about the position of tokens in the sequence. Each token, through the *self-attention mechanism* in the multi-head attention layer, has the opportunity to focus on other tokens in the sequence to better understand and capture the context and dependencies between them. The output of the self-attention mechanism, which is a sequence of the same length as the input but enriched with contextual information from other tokens, is then processed in a **position-wise feed-forward network (FFN)**. This network applies two linear transformations with a non-linear activation function in between. This step provides a refined representation for each token, enhancing complexity and incorporating higher-level features. To improve the training stability and efficiency of the deep neural network, **layer normalization** and **residual connections** are employed. In the residual connections, the input to each sub-layer (such as self-attention or FFN) is added to its output to preserve information from earlier layers and ensure smoother gradients. Meanwhile, layer normalization is applied to these summed vectors to ensure a stable distribution of the inputs to the next layers. The decoder stack has the same components as the encoder stack but is responsible for generating the output sequence from the enhanced and enriched representations processed during the encoder phase. The Decoder outputs a vector for each token in the input sequence. These enriched vectors are just dense numerical representations and are not interpretable as tokens or words. This vector, in fact, corresponds to the last

token in the input sequence, encapsulating all the context and information the model has learned through these layers. The output vectors are passed through a *linear layer* to be transformed into a format that can be mapped to tokens within the model's vocabulary. Specifically, the model maintains a set of all possible tokens it can generate. The output vector is multiplied by a weight matrix, converting it into a new vector whose size matches that of the model's vocabulary. The result of this linear transformation is a vector of logits, with each element corresponding to a specific token in the model's vocabulary. These logits represent unnormalized scores, indicating how likely the model considers each token to be the next one in the sequence. To convert these scores into probabilities, the *softmax function* is applied, ensuring that the values in the vector sum to 1 and each value represents the probability of the corresponding token being the next in the sequence. The final token is then selected based on a decoding strategy, such as greedy decoding, which chooses the token with the highest probability[72, 6, 49].

The illustration of the above explanation of the Transformer structure can be found in Figure 2.2[72].

In terms of structure and functionality, Transformers differ from traditional architectures like Recurrent Neural Networks (RNNs). Transformers are capable of handling long-range dependencies and allow for parallelized training, enabling them to process all elements of a sequence simultaneously. This parallel processing makes Transformers faster to train compared to RNNs, which process sequences step-by-step[49, 72].

It should be noted that some models use only the encoder or decoder part of this stack, which will be discussed in section 2.1.3.

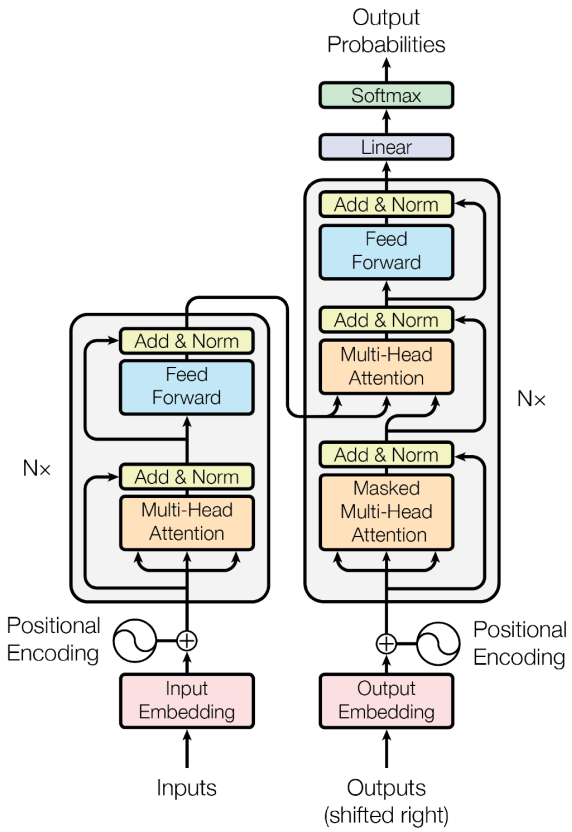

Figure 2.2: The architecture of Transformers: The left half represents the encoding layers, and the right half represents the decoding layers[72].

2.1.2 Advanced Techniques to Enhance LLM Performance

LLM performance can be further enhanced using advanced techniques to improve fine-tuning. **Instruction Tuning**[73] can complement fine-tuning by training models to follow specific instructions, which is particularly useful for applications like sentiment analysis. Additionally, the **Retrieval-Augmentation Module**[83] can improve the accuracy and relevance of LLM responses by fetching relevant information from large datasets and integrating it with the model's input. A specific implementation of this module is **Retrieval-Augmented Generation (RAG)**[37], which combines retrieved information with the generative capabilities of LLMs, enhancing the contextual accuracy of the output[49].

Fine-tuning typically requires a large amount of labelled, domain-specific data, which is not always available. **Low-Rank Adaptation (LoRA)**[30] offers a more efficient alternative by injecting low-rank matrices into the existing layers of the model, reducing computational costs and memory usage while still effectively adapting the model to specific tasks[49].

In environments with limited computational resources, LLMs can undergo **quantization**[42, 17], a process that reduces the precision of the model's weights, thereby decreasing the model's size and improving efficiency in terms of storage and speed.

High energy consumption, its environmental impact, and increased operational costs, along with the need for faster inference times for real-time applications like voice assistants, highlight the necessity for smaller models. These models provide sufficient performance for everyday applications without the overhead of larger models. Smaller models can be further optimized using advanced techniques such as **Knowledge Distillation**[29] and **Pruning**[27]. Knowledge distillation involves transferring the knowledge from a larger model (teacher) to a smaller model (student), effectively teaching the smaller model to mimic the output of the larger model. **Pruning** reduces the size of the neural network by removing less important weights and neurons, retaining only the most critical parts of the network that contribute the most to the model's performance.

2.1.3 Prominent LLMs and Their Evolution

In this section, a brief introduction to some well-known models is given.

GPT Models: GPT, a general-purpose LLM developed by OpenAI, stands for Generative Pre-trained Transformer. "*Generative*" refers to the model's ability to generate text based on the input it receives. "*Pre-trained*" refers to the initial phase where a language model is trained on a large amount of diverse data to learn the general patterns, structure, and nuances of a language. *Transformer* refers to the underlying neural network architecture used by the model[57].

GPT-1[57] was introduced in 2018 with 117 million parameters, pre-trained on the BookCorpus dataset. GPT-2[58] followed in 2019 with 1.5 billion parameters, trained on a large corpus of web texts. In 2020, GPT-3[6] was released with 175 billion parameters, utilizing a diverse dataset that included Common Crawl, books, Wikipedia, and other texts, marking a significant milestone for large language models (LLMs). In 2023, GPT-4[53] was launched with significant advancements over its predecessor GPT-3, including enhanced capabilities and a broader training dataset, though the exact number of parameters has not been officially disclosed.

In 2024, OpenAI introduced GPT-4o[51], a multimodal model integrating text, vision, and audio capabilities in a single network. GPT-4o advances natural and responsive human-computer interactions, enhancing processing speed and reducing costs while maintaining strong multimodal performance. Additionally, OpenAI released GPT-4o mini[52], a more compact and cost-effective version of GPT-4o. GPT-4o mini delivers solid performance in text and multimodal tasks at a lower cost, making it ideal for applications

demanding low latency and high efficiency.

Ploutos[69] is one of the latest financially fine-tuned LLMs based on GPT-4, excels in predicting stock movements with interpretability by combining multimodal data through its two key components, PloutosGen and PloutosGPT. PloutosGen integrates textual and numerical data using a variety of expert analyses, while PloutosGPT enhances clarity and accuracy through rearview-mirror prompting and dynamic token weighting. This method notably boosts prediction accuracy and interpretability in quantitative finance[49].

Bert: Bidirectional Encoder Representations from Transformers (BERT)[18] was released by Google in 2018. The base model of BERT has 110 million parameters, while the large model has 340 million parameters. BERT is built on the Transformer architecture, specifically using an encoder-only design. It was trained on the BookCorpus dataset and English Wikipedia, which together comprise approximately 3.3 billion words. Unlike unidirectional models like GPT, which predict the next word in a sequence based on the previous words, BERT's bidirectional models process text in both directions simultaneously, considering the context from both the left and the right sides of a word[49].

BERT's innovative approach allows it to understand the full context of a word by looking at the words before and after it, which enhances its performance in various NLP tasks. This bidirectional training enables BERT to achieve state-of-the-art results in tasks such as question answering and natural language inference. Additionally, variants of BERT, like FinBERT-19[4], FinBERT-20[79], FinBERT-21[38], and Mengzi-BERTbase-fin[86], have been fine-tuned for financial purposes, demonstrating the model's adaptability to different domains. FinBert models were fine-tuned using financial documents, such as earnings calls, news articles, financial reports, and other related textual data to finance[49].

T5: The Text-to-Text Transfer Transformer (T5)[31], developed by Google AI in 2019, uses a unified framework where all tasks are framed as text-to-text problems, making it highly versatile. The largest version, T5-11B, contains 11 billion parameters, showcasing its impressive ability to handle both language understanding and generation. This model follows an encoder-decoder structure and is pre-trained with a self-supervised task called "span corruption" [49].

Building on the T5 model, BBT (Big Bang Transformer)-FinT5 [41] was created specifically for the Chinese financial industry. This version uses knowledge-enhanced pre-training techniques and is based on the BBT-FinCorpus, a rich dataset that includes financial documents like corporate and analyst reports, social media posts, and news. These improvements make BBT-FinT5 highly effective for analyzing financial text and related tasks [49].

ELECTRA: ELECTRA (Efficiently Learning an Encoder that Classifies Token Replacements Accurately) is a pre-training method for language models. It employs two transformer models: a generator that replaces tokens in a sequence and a discriminator that detects which tokens were replaced. This replaced token detection task is more efficient than the traditional masked language modeling approach, enabling ELECTRA to achieve strong performance with fewer computational resources[12].

Building on ELECTRA, researchers developed FLANG[64], a specialized variant tailored for the financial domain. This adaptation leverages ELECTRA's efficient pre-training mechanism to enhance performance on financial text analysis tasks.

BLOOM: BLOOM [63] (BigScience Large Open-science Open-access Multilingual Language Model) is a large-scale language model released in 2022, featuring 176 billion parameters. It uses a decoder-only Transformer architecture and was trained on the ROOTS corpus, a multilingual dataset created as part of the BigScience project. BLOOM supports 46 natural languages and 13 programming languages, making it highly versatile [49].

Specialized versions of BLOOM have been developed for financial applications, such as BloombergGPT [75] and XuanYuan 2.0 [85]. BloombergGPT, a model with 50 billion

parameters, is designed specifically for the financial industry and trained on proprietary Bloomberg data, excelling in financial-specific tasks while remaining versatile. XuanYuan 2.0, focused on the Chinese financial sector, is a large open-source financial chatbot model designed for analyzing Chinese financial texts [49].

Llama-series: LLaMA [70] (Large Language Model Meta AI) is a set of language models introduced by Meta AI in February 2023, ranging from 7 billion to 65 billion parameters. The models, such as 7B, 13B, 33B, and 65B, were trained on trillions of tokens from open-access datasets, aiming for top-tier performance without relying on proprietary data. Impressively, the LLaMA-13B model outperforms GPT-3 (which has 175 billion parameters) on many benchmarks, while the LLaMA-65B competes with models like Chinchilla-70B and PaLM-540B.

Financial adaptations of LLaMA include FinMA[76], Fin-Llama[68], Cornucopia – Chinese[81], Instruct-FinGPT[82], and InvestLM[78]. InvestLM, built on the LLaMA-65B model and trained with a diverse investment-related dataset, provides investment recommendations on par with leading commercial models. Meta later introduced LLaMA 2[71], featuring enhancements such as a 40% larger pretraining corpus, doubled context length, and grouped-query attention for better inference scalability. Financial versions of LLaMA 2 include FinGPT, FinLlama, and GreedLlama. FinGPT, in particular, is an open-source model designed to offer accessible and transparent resources for developing financial language models, providing a flexible and cost-effective alternative to BloombergGPT[49].

In April 2024, Meta launched LLaMA [2], which includes models with 8 billion and 70 billion parameters. These new models exhibit state-of-the-art performance and advanced reasoning capabilities, marking them as the most capable openly available LLMs to date. Following this, Meta released LLaMA 3.1[46], an extension of the series, with additional models at 8B, 70B, and a new 405B parameter variant. LLaMA 3.1 continues the trend of strong performance across text, vision, and multimodal tasks, with the 405B model excelling particularly in complex reasoning and multilingual applications.

OPT: The Open Pre-trained Transformer (OPT) models developed by Meta AI [84] are decoder-only transformers that have been made available as open-source tools, making them valuable for research purposes. These models come in various sizes, ranging from 125 million to 175 billion parameters. The OPT models were trained on a wide variety of publicly available datasets, comprising approximately 180 billion tokens[84].

2.2 Sentiment Analysis and Prompt

2.2.1 Evolution of Sentiment Analysis

At a later stage in this study, we will analyze the sentiments of news related to stocks selected by the initial filtration process of our strategy using indicators. In this section, we will discuss the evolution of sentiment analysis from the pre-LLM to the post-LLM era.

Sentiment analysis was originally based on **lexicon-based approaches**, where the sentiment of a sentence was identified by the presence of specific words linked to positive or negative emotions. While this method was simple and efficient in some cases, it struggled with more nuanced texts, such as those containing sarcasm or irony [49]. Several studies have explored the use of this method in finance, as highlighted in [49, 65, 80].

Following lexicon-based methods, **machine learning-based approaches** were introduced for sentiment analysis. These techniques surpass lexicon-based methods by identifying complex language patterns, though they need a large volume of data to perform effectively. Machine learning methods can be applied using labelled data (supervised learning) or through techniques such as clustering (unsupervised learning) [49].

The Bag of Words (BoW) approach, for example, can be categorized as a machine learning-based method, representing a step forward from lexicon-based approaches due

to its ability to work with a broader range of words without relying solely on predefined lists[11].

BoW represents text as an unordered set of words, disregarding grammar and word order while retaining frequency. It generates a high-dimensional vector where each dimension corresponds to a unique term in the corpus, with values representing term frequency in each document. This approach effectively captures the overall content of a document through term frequency, but it fails to perceive more nuanced information embedded in term relationships and word order [11].

While dimensional reduction techniques like Latent Dirichlet Allocation (LDA)[26, 5] can help mitigate the high dimensionality of BoW vectors, they don't address its fundamental limitations in capturing semantic and syntactic nuances. BoW cannot understand context, word relationships, or the subtle meanings that arise from specific word arrangements.

In summary, although the BoW method represents an advancement over lexicon-based methods in text analysis and sentiment classification, it still falls short in capturing the full semantic richness of language. These limitations paved the way for more sophisticated techniques like word embeddings and, eventually, LLMs, which can better understand and represent the complexities of natural language.

The introduction of word **embeddings** further improved sentiment analysis by representing each word as a vector in a high-dimensional space, maintaining semantic relationships between words. This method allowed for more nuanced understanding of text but still required a substantial amount of data, which is not always accessible in all subdomains [59, 49].

The recent advances in **LLMs** marked a significant milestone in sentiment analysis. LLMs enable a deeper understanding of complex text structures, informal expressions, and specialized language commonly used in specific fields and on social media[16, 10, 32]. They can capture context, idioms, and subtle linguistic cues more effectively than previous methods[75]. Additionally, LLMs have demonstrated the ability to detect nuanced sentiment and potentially deceptive information, making them even more reliable for comprehensive sentiment analysis[49]. Another advantage of using LLMs for sentiment analysis is that they are less susceptible to human biases resulting from psychological factors and personal incentives.[8].

This evolution from simple lexicon-based methods to sophisticated LLMs represents a significant advancement in our ability to accurately analyze and interpret sentiment in various types of text, including financial news and social media content related to stock markets.

2.2.2 Prompt and its Importance

A **prompt** is a text or query provided by the user to guide the LLM in generating a response. By inputting a prompt, the user instructs the model on the expected output. The prompt can vary in length, ranging from a single sentence to a full paragraph, depending on the complexity of the task. The model processes the prompt, generating multiple potential responses based on the input and then selecting the most relevant and coherent one to output [40].

In more detail, when a prompt is provided by the user, it first gets tokenized by the model. The resulting tokens are numerical representations of the input prompt that can be processed by the model. These tokens are then fed through the transformer layers, where the attention mechanism assigns different weights to each token based on their contextual relevance. Finally, during the decoding phase, a response is generated token by token, and these output tokens are converted back into text [9].

A **content prompt** is a term commonly used in the context of LLMs when executing code to generate results, often used interchangeably with prompt. A content prompt is a static input text that sets the context and background for the model, while a prompt is a dynamic input that can include more specific information about the task. Examples of these are shown below:

- **Content Prompt:** `You are an expert financial analyst in stock market analysis.`
- **Prompt:** `Given the following news article about` *XYZ*`, please analyze its content and label it as either "Positive" or "Negative" in terms of its effect on its price.`

The efficiency of LLMs can be enhanced by using **prompt engineering**, a process of structuring and refining the prompt to obtain more relevant and desired output from the model without altering the model's parameters. Along with model-specific parameters, such as the model's hyperparameters, the prompt significantly impacts the model's response. Prompt engineering is crucial due to its ability to make LLMs useful across different fields and sectors by increasing the model's versatility and adaptability[9, 62].

A list of good prompt features, based on the study carried out by Chen et al.[9], is provided below:

- The prompt should be clear and unambiguous.
- Instruct the model to act in a specific role, such as a financial analyst in the above example.
- Separate different parts of the prompt for more complex and longer prompts. For example, use triple quotes when the prompt itself contains a quote.

An exemplary case of an engineered prompt for sentiment analysis of financial news, complete with an elaborate explanation, can be found in the study by Lopez-Lira and Tang [40]. It is important to note, however, that as models become more complex and larger in scale, they can effectively process and respond to increasingly detailed prompts.

2.3 State-of-the-Art Related Literature

The use of news to predict stock returns has been a subject of extensive research, with many studies employing less sophisticated methods such as lexicon-based approaches [15, 66, 67, 44, 22]. These traditional methods have provided valuable insights into the relationship between news sentiment and market movements. However, recent breakthroughs in artificial intelligence, particularly in the development of Large Language Models (LLMs), have created new opportunities for researchers to incorporate more powerful tools into trading strategies. These advanced models offer the potential to capture nuanced information and complex relationships that may be missed by simpler lexicon-based methods. Among the many recent investigations [40, 11, 28], our research focuses on those most closely aligned with the use of LLMs for return prediction and trading strategy development.

Lopez-Lira and Tang [40] investigated the potential of general-purpose LLMs in predicting stock returns. Specifically, they utilized `ChatGPT`, a general-domain model not explicitly trained for financial tasks, to predict stock returns. Their findings were remarkable, reporting a cumulative return of 400% and a Sharpe Ratio of 3.8 over the period from 2021 to 2022. They compared `ChatGPT` with simpler language models with lower

complexity (fewer parameters), such as `GPT-1`, `GPT-2`, and `BERT`, for extracting sentiment from news headlines. Their results indicated that more complex models, such as `ChatGPT`, were more effective, yielding higher returns and Sharpe Ratios. Additionally, they demonstrated that their self-financing strategy, which involved buying and selling stocks based on positive and negative news, performed better for smaller-cap stocks and in response to negative news [40].

A study conducted by Chen et al. [11] compared the effectiveness of various language models in processing textual data for financial applications. The researchers examined LLMs such as OPT and RoBERTa, as well as word-based methods like Word2Vec and SESTM (which is based on the Bag of Words (BoW) approach). Their findings revealed that trading strategies utilizing LLMs for data processing yielded higher Sharpe ratios and returns compared to traditional word-based methods.

The authors also investigated different portfolio construction approaches, including equal-weighted and value-weighted strategies. Notably, the equal-weighted strategy demonstrated superior performance, achieving a Sharpe Ratio of 4.51, significantly higher than the 1.24 obtained by the value-weighted strategy. This result suggests better performance for smaller stocks, which aligns with the findings of Lopez-Lira and Tang [40], who also observed superior performance of LLM-driven strategies for small-cap stocks. Chen et al. proposed two potential explanations for this phenomenon. First, smaller stocks receive less attention, potentially leading to more delayed market reactions. Second, the lower liquidity of smaller stocks may require more time for news to be fully incorporated into their prices [11]. These findings highlight the potential of LLMs in enhancing financial analysis and decision-making, particularly for smaller, less-followed stocks.

Kirtac and Germano [34] conducted a comparative study on the performance of various models in predicting stock returns. Their research utilized news headlines as input for sentiment analysis. The findings demonstrated that the OPT model, when employed in a long-short portfolio strategy, outperformed other models, including BERT and FinBERT. Specifically, the OPT model achieved an impressive Sharpe ratio of 3.05 over the period from 2021 to 2023. In addition to comparing different LLMs, the authors benchmarked their results against the Loughran-McDonald dictionary model, a conventional method for sentiment analysis in finance. The study revealed a substantial outperformance by the more complex LLMs, with the OPT model showing particularly strong results. This performance gap underscores the potential advantages of advanced language models in capturing nuanced sentiment information from financial news headlines[34].

It is important to note that all of the above studies[34, 40, 11] focused exclusively on the immediate impact of individual news items, rather than considering the cumulative effect of multiple news events over time. Specifically, they analyzed the market's reaction to single, isolated news headlines on the day they were published, without taking into account any preceding news or how a series of related news items might collectively influence stock prices. This approach may overlook the potential compounding effects that news flow can have on investor sentiment and stock performance over time. Furthermore, they all limited their analysis to news headlines, excluding the snippets and full news bodies.

The collective findings from these studies[34, 40, 11] strongly suggest that LLMs can offer significant improvements over traditional sentiment analysis techniques in the context of financial prediction and trading strategies. Moreover, a pattern emerges when comparing different LLMs: those with more complex and sophisticated architectures tend to perform better. For instance, the OPT model consistently demonstrated superior performance in multiple studies. This suggests a positive correlation between model complexity and predictive accuracy in financial applications.

As a valid critique of all the above studies, one can argue that we may encounter biased results when backtesting a strategy using an LLM trained on data from that period, leading

to biased results. This issue has been extensively discussed in [23]. The authors categorize bias into two types: look-ahead bias and distraction effect. They provide an illustrative example where a news headline announces the earnings for company XYZ in 2019. The look-ahead bias occurs when an LLM, trained on data from 2019, already knows the earnings outcome for XYZ and, therefore, labels the news as positive or negative rather than neutral. On the other hand, if the LLM does not know the specific earnings report of XYZ but has general knowledge about the company, it may still label the news as non-neutral. This is referred to as the distraction effect[23].

Although look-ahead bias is not an issue when using an LLM out-of-sample (on data that the LLM has not been trained on), distraction can still occur. The authors explain that look-ahead bias tends to be positive since having some future information would likely generate more profit, while the impact of distraction (general knowledge of the company) could be either positive or negative. They used an innovative approach to anonymize company names or other identifiers, such as replacing "iPhone" and "iPad" with generic terms when referencing Apple, in news headlines to assess how this affects performance. Surprisingly, they found that anonymized headlines generated higher in-sample returns, concluding that distraction must have negative effects, since, as mentioned, look-ahead bias generally has a positive impact. They also noted that distraction mostly occurs for larger stocks, as there is more data available about them[23].

Chapter 3

Methodology

3.1 Mathematical Formulation of the Problem

In this section, we will provide a mathematical framework for the problem. The idea of this section is partially derived from papers [40, 25, 19] and lecture notes provided by Dr Johannes Muhle-Karbe for the Portfolio Management module at Imperial College London 2023-2024[48].

3.1.1 Mean-Variance Optimization

In this section, inspired by the works of [19, 56, 48], we investigate how an individual investor can make better trading decisions. The market under consideration consists of a risk-free asset and N risky assets. The risk-free asset delivers a return represented by R^f, while the returns from risky assets, denoted as $R_t^n = R^f + R_t^{n,e}$, are uncertain, where $R_t^{n,e}$ is the excess return.

If an investor has W_{t-1} dollars at time $t-1$ and allocates x_{t-1}^n to the n-th asset, the wealth at time t, following market changes and trade execution, is given as [48]:

$$\begin{aligned} W_t^{x_{t-1}} &= x_{t-1}^1(1+R_t^1) + \cdots + x_{t-1}^N(1+R_t^N) + (W_{t-1} - \sum_{n=1}^{N} x_{t-1}^n)(1+R^f) \\ &= W_{t-1}(1+R^f) + x_{t-1}^T R^e. \end{aligned}$$

In this scenario, the investor's goal is to maximize the expected wealth while applying a penalty for its variance, proportional to the risk aversion parameter γ [48, 56]. This leads to the following objective function:

$$\begin{aligned} J_{t-1}(x_{t-1}) &= \mathbb{E}_{t-1}[W_t^{x_{t-1}}] - \frac{\gamma}{2}\mathrm{Var}_{t-1}[W_t^{x_{t-1}}] \\ &= W_{t-1}(1+R^f) + x_{t-1}^T \mathbb{E}_{t-1}[R_t^e] - \frac{\gamma}{2} x_{t-1}^T \mathrm{Cov}_{t-1}[R_t^e, R_t^e] x_{t-1} \end{aligned} \tag{3.1.1}$$

The first-order condition for this optimization problem can be described as:

$$0 = \nabla J_{t-1}(x_{t-1}) = \mathbb{E}_{t-1}[R_t^e] - \gamma \mathrm{Cov}_{t-1}[R_t^e, R_t^e] x_{t-1}.$$

Assuming the covariance matrix of risky returns is invertible, the **optimal allocation** in the risky assets is [48]:

$$\mathbb{X}_{t-1} = (\gamma \mathrm{Cov}_{t-1}[R_t^e, R_t^e])^{-1} \mathbb{E}_{t-1}[R_t^e].$$

For simplicity, in this study, we assume $R^f = 0$. In the case of a single risky asset, the excess return R_t^e can be expressed as $R_t^e = \frac{p_t - p_{t-1}}{p_{t-1}}$, where p_t is the asset price at time t. Under this assumption, the **optimal holding** at time $t-1$, $x_{t-1} = \frac{\mathbb{X}_{t-1}}{p_{t-1}}$, can be derived from the following equation:

$$x_{t-1} = \frac{\mathbb{E}_{t-1}[p_t] - p_{t-1}}{\gamma \mathrm{Var}_{t-1}[p_t]}. \tag{3.1.2}$$

3.1.2 Introduction of Expected Utility Maximization

While mean-variance optimization provides a straightforward method of optimizing investment decisions using first-order conditions, it is not always monotonic with respect to wealth [48], as shown in 3.1.1.

Consequently, an investor using mean-variance methods might reject a free lottery ticket if its variance is too high compared to its expected return. To overcome this issue and avoid inconsistent decision-making, an alternative is to use an increasing concave utility function $U(\cdot)$ based on wealth, with the objective of maximizing expected utility [48, 21].

$$\sup_{x_{t-1}} \mathbb{E}_{t-1}[U(W_t^{x_{t-1}})].$$

For the exponential utility function $U(x) = -\exp(-\gamma x)$, along with conditionally normally distributed excess returns $R_t^e \sim \mathcal{N}(\mu_{t-1}, \Sigma_{t-1})$, the calculation becomes straightforward. Using the moment generating function of a multivariate normal distribution, we derived equation 3.1.3 [48].

$$\begin{aligned}\mathbb{E}_{t-1}[U(W_t^{x_{t-1}})] &= \mathbb{E}_{t-1}\left[-\exp\left(-\gamma\left((1+R^f)W_{t-1} + x_{t-1}^T R_t^e\right)\right)\right] \\ &= -\exp\left(-\gamma(1+R^f)W_{t-1}\right)\exp\left(-\gamma x_{t-1}^T \mu_{t-1} + \frac{\gamma^2}{2} x_{t-1}^T \Sigma_{t-1} x_{t-1}\right).\end{aligned} \tag{3.1.3}$$

3.1.3 Model Structure

Building on the concepts presented in[40], we are considering two types of agents, both with Constant Absolute Risk Aversion (CARA)[45] behaviour with risk aversion γ, meaning that the portion of the money they invest in risky assets is fixed and does not depend on their total wealth. In this model, we categorize agents into two types: **Attentive** and **Inattentive**, with proportions represented by $\pi_A \in (0, 1)$ for attentive agents and $\pi_I = 1 - \pi_A$ for inattentive ones. Later in this study, we will introduce an LLM agent to explore how it processes new information about the asset compared to these agents.

Three periods are defined for the model. In the first period, the relevant news about the asset is realised. While attentive agent has enough capacity to fully process and incorporate the news in their estimations, the inattentive agent fails to do so, only able to partially understand and incorporate this set of new information. From period one to two, the inattentive agent continues processing the information, reaching full incorporation in period two. Period three is the end of the economy for our model, when the dividend D is paid off.

We also assume that the asset is in zero net supply, meaning there is no shortage or surplus of the asset in the market [40].

In this model, noise traders are also included, resulting in the non-fundamental risks. The presence of this risk does not allow attentive agents to exploit their advantage over inattentive agents to execute riskless trades, which would be a non-real scenario.

We assume that the main source of asset uncertainty is the random liquidating dividend D_t scheduled to be paid at time t[48]. To avoid arbitrage, $p_t = D_t$, where p_t is the price just before dividend payment. The optimal holding of the asset can be found using 3.1.2, which gives equation 3.1.4.

$$x_{t-1} = \frac{\mathbb{E}_{t-1}[D_t] - p_{t-1}}{\gamma \mathrm{Var}_{t-1}[D_t]} \tag{3.1.4}$$

D_t is assumed to be random and given by

$$D_t = \mu_D + \sigma_D \epsilon_D, \quad \epsilon_D \sim N(0,1),$$

where μ_D is the expected value of the dividend, referred to as the asset fundamental, and ϵ_D is its standard deviation[48]. We will assume that agents have a correct prior of the dividend, which is

$$\mu_D \sim N(\hat{D}, \sigma_D^2). \tag{3.1.5}$$

Therefore $\mathbb{E}_{t-1}[D_t] = \mathbb{E}_{t-1}[\mu_D] = \hat{D}$, and $\mathrm{Var}_{t-1}[D_t] = \sigma_D^2$. Using these in Equation 3.1.4 gives

$$x_{j,t} = \frac{\hat{D} - p_t}{\gamma \sigma_D^2}. \tag{3.1.6}$$

In which $j \in A, I$, meaning that $x_{j,t}$ shows the optimal holding of the asset by each agent at time t.

Given the asset's zero net supply and let u represent the noise traders' demand for the asset, the equilibrium price at time $t-1$ (p_{t-1}) can be found using equation 3.1.8, which used the market clearing condition 3.1.7 [40].

$$\frac{\hat{D} - p_{t-1}}{\gamma \sigma_D^2} + u = 0 \tag{3.1.7}$$

$$p_{t-1} = \hat{D} + u\gamma\sigma_D^2 \tag{3.1.8}$$

3.1.4 Incorporation of News

In this section, based on the work of [40], we will incorporate the unexpectedly released new information. We assume that all agents observe the same signal but interpret it based on their individual processing capacities. The signal s is given by

$$s = \mu_D + \epsilon, \epsilon \sim N(0, \sigma_s^2), \tag{3.1.9}$$

in which $\sigma_s^2 = \frac{1}{\tau_s}$. The signal s is comprised of noise ϵ and μ_D, which is as before, the fundamental value. Parameter τ_s is the total precision of the released news. Precision can be defined as a measure of the certainty of the information, quantifying how accurately news reflects the true value of the company's fundamentals. Agents use the precision of news to update their beliefs on the asset's fundamental value [40].

As mentioned before, agents have different abilities to perceive and process new information. We reflect this with varying precision of information processing for agents, as can be seen in 3.1.10.

$$\tau_A = \alpha_A \tau_s, \quad \tau_I = \omega \tau_A, \quad \tau_L = \lambda(c,p)\tau_s \tag{3.1.10}$$

Parameter $\alpha_A \in (0,1]$ is the attentive agent's information processing capacity. An inattentive agent possesses precision of τ_I, which is the multiplication of the attentive agent's precision with constant $\omega \in (0,1]$. LLM agent's precision is proportional to $\lambda(c,p)$

which is a function of news complexity c and model size p. As news complexity c increases, the precision of LLM agent decreases, while it improves as the model size p increases [40].

Inattentive agents, with limited processing capacity, will perceive and process the new information (signal) with precision $\omega\tau_A$ in period one, and the remaining $(1-\omega)\tau_A$ until the second period. It should be noted that even an attentive agent might not be able to process the whole information, when $\alpha_A < 1$. In this case, an agent who has the ability to process the whole signal can exploit the signal to make a profit. This scenario was thoroughly studied in the works of Lopez-Lira and Tang [40] and will be mentioned in Section 3.1.5 of this thesis, assuming that LLM has a higher capacity, namely $\lambda(c,p) > \alpha_A$. For now, we assume $\tau_L < \tau_A$.

3.1.5 Using Bayesian Method to Update Agents' Beliefs

This section describes how agents update their beliefs on the expectation and variance of the asset's fundamentals upon receiving and processing the new information. Bayesian updating[54, 74] will be used to find these formulas. Bayesian updating is a statistical method to update the probability distribution once new information becomes available.

The prior distribution is the initial belief of the agent about the asset's fundamental value (μ_D) before the release of new information, which is $\mu_D \sim N(\hat{D}, \sigma_D)$, as shown before in equation 3.1.5. The news gets released, which can be seen as a signal representing new information about the asset's fundamental value, written as a summation of true fundamental value and some noise, as can be seen in 3.1.9. The updated belief after incorporation of the news, namely posterior mean and variance, can be found using Bayes' theorem.

Bayes' theorem for continuous variables can be expressed as:

$$f(\mu_D|s) = \frac{f(s|\mu_D)f(\mu_D)}{f(s)} \tag{3.1.11}$$

where $f(\mu_D|s)$ is the posterior distribution of the asset's fundamental value given the released signal s. $f(s|\mu_D)$ is the likelihood of the signal given the asset's fundamental value. $f(\mu_D)$ is the prior distribution of the μ_D, and $f(s)$ is the marginal likelihood of the signal.

We assumed μ_D is normally distributed $\mu_D \sim N(\hat{D}, \sigma_D)$, so the probability density function (PDF) of the prior can be written as

$$f(\mu_D) = \frac{1}{\sqrt{2\pi\sigma_D^2}} \exp\left(-\frac{1}{2}\frac{(\mu_D - \hat{D})^2}{\sigma_D^2}\right). \tag{3.1.12}$$

Similarly, the likelihood function, representing the probability of observing the signal given the asset's fundamental value, can be written as below:

$$s = \mu_D + \epsilon, \quad \epsilon \sim N(0, \sigma_s^2) \tag{3.1.13}$$

$$f(s_|\mu_D) = \frac{1}{\sqrt{2\pi\sigma_s^2}} \exp\left(-\frac{1}{2}\frac{(s - \mu_D)^2}{\sigma_s^2}\right) \tag{3.1.14}$$

Now, combining the PDFs from equations 3.1.12 and 3.1.14, and using them in equation 3.1.11, we can obtain the following by ignoring the marginal likelihood since it is a constant with respect to μ_d.

$$
\begin{aligned}
f(\mu_D|s) &\propto \exp\left(-\frac{1}{2}\frac{(s-\mu_D)^2}{\sigma_s^2}\right)\exp\left(-\frac{1}{2}\frac{(\mu_D-\hat{D})^2}{\sigma_D^2}\right) \\
&\propto \exp\left(-\frac{1}{2}\left[\frac{(s-\mu_D)^2}{\sigma_s^2}+\frac{(\mu_D-\hat{D})^2}{\sigma_D^2}\right]\right) \\
&\propto \exp\left(-\frac{1}{2}\left[\frac{s^2-2s\mu_D+\mu_D^2}{\sigma_s^2}+\frac{\mu_D^2-2\mu_D\hat{D}+\hat{D}^2}{\sigma_D^2}\right]\right) \\
&\propto \exp\left(-\frac{1}{2}\left[\left(\frac{1}{\sigma_s^2}+\frac{1}{\sigma_D^2}\right)\mu_D^2-2\left(\frac{s}{\sigma_s^2}+\frac{\hat{D}}{\sigma_D^2}\right)\mu_D+\left(\frac{s^2}{\sigma_s^2}+\frac{\hat{D}^2}{\sigma_D^2}\right)\right]\right)
\end{aligned}
$$

Noting that the posterior distribution must be of the same form as the normal distribution

$$
f(\mu_D|s_j) \propto \exp\left(-\frac{1}{2\sigma_{j|s}^2}\left(\mu_D-\mu_{j|s}\right)^2\right),
$$

we can match coefficients to find the variance and mean.

The coefficient of μ_D^2 in the exponent gives us the precision of the posterior distribution (inverse of the variance):

$$
\frac{1}{\sigma_{j|s}^2}=\frac{1}{\sigma_s^2}+\frac{1}{\sigma_D^2}
$$

Thus, the posterior variance is:

$$
\sigma_{j|s}^2=\frac{1}{\frac{1}{\sigma_s^2}+\frac{1}{\sigma_D^2}}=\frac{1}{\tau_s+\tau_D}
$$

where $\tau_s=\frac{1}{\sigma_s^2}$ and $\tau_D=\frac{1}{\sigma_D^2}$.

The coefficient of μ_D in the linear term gives us the weighted sum of the prior mean and the signal, weighted by their precision:

$$
\mu_{j|s}=\frac{\frac{s_j}{\sigma_s^2}+\frac{\hat{D}}{\sigma_D^2}}{\frac{1}{\sigma_s^2}+\frac{1}{\sigma_D^2}}=\frac{s_j\tau_s+\hat{D}\tau_D}{\tau_s+\tau_D}
$$

Therefore, the posterior mean and variance at period n can be computed using equations 3.1.15 and 3.1.16, respectively[40].

$$
\mu_{j|s}=\frac{s_j\tau_{n,j}+\hat{D}\tau_D}{\tau_{n,j}+\tau_D} \tag{3.1.15}
$$

$$
\sigma_{j|s}^2=\frac{1}{\tau_{n,j}+\tau_D} \tag{3.1.16}
$$

where $j\in\{A,I,LLM\}$

In the Absence of LLM

Following the methodology outlined in [40], we ignore the LLM agent for now. Since the attentive agent is aware of the inattentive agent's limited capacity, they exploit this in period 1 to make a profit. However, in period 2, the inattentive agent has completely updated its belief, and the attentive agent has no edge over the inattentive one.

In period two, both agents agree on the expectation and variance since there is no further news to be released, and also the inattentive agent becomes fully updated. Therefore, expectation and variance can be found using equations 3.1.15 and 3.1.16 with $n = 2$, respectively[40].

$$\mu_{j|s} = \frac{s_j \tau_{2,j} + \hat{D}\tau_D}{\tau_{2,j} + \tau_D}$$

$$\sigma^2_{j|s} = \frac{1}{\tau_{2,j} + \tau_D}$$

with $j \in \{A, I\}$.

CARA assumption enables us to find the optimal holding in the asset using the equation 3.1.4 with $\mathbb{E}_{t-1}[D_t] = \mu_j$:

$$x_{2,j} = \frac{\mu_j - p_2}{\gamma \sigma^2_{j|s}}, \quad j \in \{A, I\}. \tag{3.1.17}$$

Similarly, the price can be found using [40]

$$p_2 = \mu_A + u\gamma\sigma^2_{j|s}, \quad j \in \{A, I\}.$$

In period one, attentive traders have the ability to fully understand the news and try to exploit this to make a profit. Inattentive traders are also executing trades while they are not aware that they do not fully perceive the news. This framework is inspired by the model presented in [40]. In this period, the attentive agent will try to maximize the utility function of the wealth in period 2 (w_2), as described in section 2.1.2. The wealth in period 2 is $w_2 = w_1 + x_1(p_2 - p_1)$. While solving $\sup_{x_1} \mathbb{E}_1[V_2(w_2)]$, we need to find $V_2(w_2)$ [40].

$$\begin{aligned}
V_2(w_2) &= \sup_{x_2} E[U(w_3)] \\
&= \sup_{x_2} E[-\exp(-\gamma w_3)] \\
&= \sup_{x_2} -\exp\{-\gamma E[w_3] + \frac{\gamma^2}{2}\mathrm{Var}(w_3)\} \\
&= \sup_{x_2} -\exp\left\{-\gamma\left(w_2 + x_2(\mu_A - p_2)\right) + \frac{\gamma^2}{2}x_2^2\sigma^2_{A|s}\right\} \\
&= -\exp\left\{-\gamma w_2 - \frac{1}{2}\frac{(\mu_A - p_2)^2}{\sigma^2_{A|s}}\right\}
\end{aligned} \tag{3.1.18}$$

Note that in the steps involved in 3.1.18, we used the moment generating function of the multivariate normal and $w_3 = w_2 + x_2(\hat{D} - p_2)$.

Now we can find $\sup_{x_1} E_1[V_2(w_2)]$ as below [40]:

$$\begin{aligned}
\sup_{x_1} E_1[V_2(w_2)] &= \sup_x E_1\left[\exp\{-\gamma(w_2) - \frac{1}{2}\frac{(\mu_A - p_2)^2}{\sigma^2_{A|s}}\}\right] \\
&= \sup_x E_1\left[\exp\{a(x) + b(x)p_2 + c(x)p_2^2\}\right]
\end{aligned} \tag{3.1.19}$$

where $a = a(x) = \gamma p_1 x - \frac{\mu_A^2}{2\sigma^2_{A|s}} - \gamma w_1$, $b = b(x) = \frac{\mu_A}{\sigma^2_{A|s}} - \gamma x$, and $c = c(x) = -\frac{1}{2\sigma^2_{A|s}}$. In paper [40], noticing $p_2 \sim N(\mu_p \equiv \mu_A, \sigma_p^2 \equiv \alpha^2\sigma_u^2\sigma^4_{A|s})$ the closed form formula of the expectation is shown to be

$$E_1\left[\exp\{a(x)+b(x)p_2+c(x)p_2^2\}\right]=\frac{\exp\left(\frac{-4ac\sigma_p^2+2a+b^2\sigma_p^2+2b\mu_p+2c\mu_p^2}{2-4c\sigma_p^2}\right)}{\sqrt{1-2c\sigma_p^2}}.$$

Therefore, the optimal holding for the attentive agent in period one can be computed using 3.1.20 [40]

$$x_{1,A}=\frac{(\mu_A-p_1)\left(\gamma^2\sigma_{A|s}^2\sigma_u^2+1\right)}{\gamma^3\sigma_{A|s}^4\sigma_u^2} \tag{3.1.20}$$

This equation can be rearranged to 3.1.21.

$$x_{1,A}=\frac{\mu_A-p_1}{\gamma\sigma_{A|s}^2}+\frac{\mu_p-p_1}{\gamma\sigma_p^2} \tag{3.1.21}$$

In 3.1.21, the initial term reflects the conventional demand from CARA investors for a dividend-paying asset, while the latter term arises from the potential to exploit short-term pricing discrepancies. The first component occurs because, based on the expected demand in the second period, there is a chance to purchase the asset at a potentially reduced price and retain it until the dividend is received [40].

Using $\tau_p=\frac{1}{\sigma_p^2}$, equation 3.1.21 can be also written as

$$\begin{aligned}x_{1,A}&=(\tau_{A|s}+\tau_p)\frac{\mu_A-p_1}{\gamma}\\&=\tau_{p,d}\frac{\mu_A-p_1}{\gamma}\end{aligned} \tag{3.1.22}$$

where $\tau_{p,d}=\tau_{A|s}+\tau_p$ is the total precision.

An inattentive agent, however, does not have the chance to exploit informational advantage. Their optimal holding in period one can be found using equation 3.1.23 [40].

$$x_{1,I}=\frac{\mu_I-p_1}{\gamma\sigma_{I|s}^2}=\tau_{I|s}\frac{\mu_I-p_1}{\gamma} \tag{3.1.23}$$

Market clearing condition 3.1.24 can be again used to find the price.

$$\pi_A x_{1,A}+\pi_I x_{1,I}+u=0 \tag{3.1.24}$$

If we use (3.1.20) and (3.1.23) in (3.1.24), and define $\mu_E=\frac{\mu_A\tau_{\pi_A}+\mu_I\tau_{\pi_I}}{\tau_{\pi_A}+\tau_{\pi_I}}$ and $\sigma_E^2=\frac{1}{\tau_{\pi_A}+\tau_{\pi_I}}$, we can find the price in period 1 (p_1) to be [40]:

$$\begin{aligned}p_1&=\frac{\mu_A\pi_A\tau_{p,d}+\mu_I\pi_I\tau_I+\alpha u}{\pi_A\tau_{p,d}+\pi_I\tau_I}=\frac{\mu_A\tau_{\pi_A}+\mu_I\tau_{\pi_I}}{\tau_{\pi_A}+\tau_{\pi_I}}+\frac{\alpha u}{\tau_{\pi_A}+\tau_{\pi_I}}\\&=\mu_E+\alpha u\sigma_E^2\end{aligned} \tag{3.1.25}$$

In equation 3.1.25, μ_E and σ_E^2 can be seen as the economy-wide expectation and variance, respectively. Equation 3.1.25 can be seen as two parts: the first term is the weighted average of the dividend expectations from different types of agents, where each agent's expectation is adjusted by the precision of their information. The second term accounts for the influence of non-fundamental demand, stemming from noise traders [40].

The impact of new information about the asset on the price in period one can be seen as $\mathbb{E}[p_1]-\mathbb{E}[p_0]=\mu_E-\hat{D}$, where $\mathbb{E}[p_0]$ was found using 3.1.8. This is the dollar profit that can be exploited by high-frequency traders.

Similarly, the dollar profit in period two can be found $\mathbb{E}_1[p_2-p_1]=\mu_A-\mu_E+\alpha u\sigma_E^2$.

In the Presence of LLM

Guided by the principles in [40], the expectation of dividend and precision from the LLM point of view can be found using 3.1.15 and 3.1.16 with $\tau_{n,j} = \lambda\tau_s$, $j \in \{L\}$. Therefore, we will have below two equations for them [40]:

$$\begin{aligned} \mu_L &= \mathbb{E}_L[D|s] = \frac{\bar{d}\tau_D + s\lambda\tau_s}{\tau_D + \lambda\tau_s}, \\ \tau_{L|s} &= \frac{1}{\sigma^2_{L|s}} = \tau_D + \lambda\tau_s. \end{aligned} \tag{3.1.26}$$

The optimal holding for the LLM agent can be found using 3.1.17 which gives us:

$$x_{1,L} = \frac{\mu_L - p_1}{\gamma\sigma^2_{L|s}} = \tau_{L|s}\frac{\mu_L - p_1}{\gamma}. \tag{3.1.27}$$

The expected profit for the LLM agent from its own perspective is

$$x_{1,L}(\mu_A - p_1) = \tau_{L|s}\frac{(\mu_L - p_1)(\mu_A - p_1)}{\gamma}, \tag{3.1.28}$$

while, the profit from the attentive agent's point of view is [40]

$$x_{1,L}(\mu_L - p_1) = \tau_{L|s}\frac{(\mu_L - p_1)^2}{\gamma}. \tag{3.1.29}$$

The interpretation of equation 3.1.29 is that the profit, from the attentive agent's perspective, is affected by how well the two predictions align. Consistency between the LLM's and the attentive agent's expectations leads to higher profits. However, equation 3.1.28 shows that the profit depends on the magnitude of the difference between the predicted dividend and the current price, from the LLM agent's point of view.

The below propositions are achieved and proved in the work of Lopez-Lira and Tang [40]:

1. For a fixed set of parameters and news complexity, there is a unique threshold of model size, p^*, such that only larger Large Language Models (LLMs) with $p > p^*$ can predict returns profitably. p^* is:

    - Increasing in inattentive agents' information capacity (ω): Inattentive agents process information less effectively. If their capacity increases, they become better at understanding the market, making it harder for the model to outperform them. Thus, a larger model size (p) is required.
    - Increasing in attentive agents' information capacity: Attentive agents are already good at processing information. If their capacity increases, the model needs to be larger to extract more subtle patterns that these agents might miss.
    - Increasing in the proportion of attentive agents: A higher proportion of attentive agents means more agents are effectively processing information, increasing market efficiency. Thus, the model needs to be larger to find profitable opportunities.
    - Decreasing in agents' risk aversion (γ): Higher risk aversion means agents are less likely to take risky positions. A smaller model can be profitable as it requires less precision to make cautious predictions that align with the agents' risk preferences.

- Decreasing in noise trader variance (σ_u^2): More noise in the market means there is more randomness and less reliance on precise information. A smaller model can capitalize on the increased volatility without needing high precision.

2. the informativeness of the asset price improves with a higher proportion of agents using LLMs and with the size of these LLMs. Essentially, as more agents use LLMs that can better interpret and predict the impact of news on the asset's fundamental value, the overall market price becomes a more accurate reflection of true value.

3. When all inattentive agents adopt sufficiently large LLMs, return predictability in the market disappears. In fact, the absence of informational advantage among market participants eliminates predictable excess returns.

4. Smaller or illiquid markets with fewer attentive participants do not fully incorporate all available information into prices. In these markets, LLMs can provide a significant predictive edge, leading to higher return predictability.

5. If attentive agents start with a low information capacity ($\tau_A < \tau_A^*$) and LLMs have a relatively small capacity ($\lambda(c,p) < \tau_A^*$), the use of LLMs by these agents will increase return predictability by enhancing their ability to process information without making the market too efficient. However, if attentive agents already possess high information capacity ($\tau_A \geq \tau_A^*$) and use powerful LLMs ($\lambda(c,p) \geq \tau_A^*$), return predictability will decline as these LLMs make the market overly efficient by quickly assimilating all available information, thereby eliminating informational inefficiencies and reducing the scope for predictable returns.

3.2 Data

3.2.1 Initial Stocks Universe

Small-cap stocks with a market capitalization of less than $2 billion were selected from the Russell 2000 Index, resulting in an initial universe of 1,351 stocks for analysis by our strategy. To avoid any potential look-ahead bias, the market capitalization values used were those available at the time of selection, ensuring the integrity and accuracy of our analysis.

3.2.2 Time Frame

We tested and evaluated the performance of our trading strategy using a dataset from 2022 and 2023. This evaluation allowed us to assess the strategy's effectiveness in various market conditions over a two-year time frame.

3.2.3 News

NewsWitch©[1], a product developed by ZanistaAI, was used to obtain processed news articles published up to one day prior to the day when a stock was triggered according to our criteria. Due to the scope of this study and limitations on computational resources, we analyzed only the sentiment of news titles and snippets, and did not include the news body.

When searching for news related to a specific stock, the results are displayed as illustrated in Figure 3.1. The snippet, highlighted in the blue rectangle, is a brief excerpt from the beginning of the article's body, offering a quick summary or key information from the

[1]In obtaining data, we followed the terms and conditions of `ZanistaAI`, which can be found at this link.

article. The title, enclosed in the purple rectangle, provides a concise description of the news topic, allowing for an immediate understanding of the content's focus.

Figure 3.1: An example of news title and snippet

Since we utilized only the title and snippet of each news article, the exact date and time of the events were not available. To mitigate potential look-ahead bias, we excluded any news directly related to T_{event} itself.

Approximately 500,000 news articles in 2022 and 600,000 in 2023 were collected and processed using *NewsWitch*©. These data were subsequently analyzed as part of this thesis.

3.3 Trading Strategy Pipeline

3.3.1 Stocks Filtration Process

As mentioned in Section 3.2.1, we focused on small-cap stocks in the Russell 2000 index.

Overview of the Procedure

As previously discussed, monitoring news and extracting sentiment using LLMs for all stocks in the initial pool is computationally infeasible and cost-prohibitive. Therefore, a filtration step is necessary to reduce the size of this stock pool.

We devised an approach utilizing the stocks themselves and macroeconomic indicators to filter down the initial pool of stocks.

Our method focuses on observing news only for stocks which themselves or their co-moved macroeconomic indicators experienced a larger-than-average move. For this purpose, we employed two key statistical measures: Z-score and Beta. The Z-score is a statistical measure that quantifies how many standard deviations an observation or data point is from the mean of a distribution. It allows us to identify when a stock or indicator has moved significantly compared to its historical behavior. We also employed **Beta**, which can be computed using equation 3.3.1, an estimate of the co-movement between two variables, which measures the magnitude of this co-movement.

$$\beta = \frac{\text{Cov}(R_i, R_{ind})}{\text{Var}(R_{ind})} \tag{3.3.1}$$

In this formula:

- R_i represents the return of the individual stock.
- R_{ind} represents the return of the indicator.
- $\text{Cov}(R_i, R_{ind})$ is the covariance between the stock's return and the indicator's return.
- $\text{Var}(R_{ind})$ is the variance of the indicator's return.

Unlike the correlation coefficient, which only provides insight into the direction and strength of the relationship without considering the magnitude, Beta offers a more comprehensive understanding of the co-movement. Historical returns (standardization of prices) were analyzed to calculate the Beta of these indicators against stocks, identifying pairs of indicators and stocks that have exhibited historical co-movement.

Using these measures, we observed news only for stocks where they themselves had significant changes, assessed by their Z-score, or for stocks whose co-moved macroeconomic indicators (identified through historical beta analysis) saw considerable moves, as measured by the indicator's Z-score. We refer to stocks selected by any of the above analyses as triggered stocks. This approach eliminates the need to monitor news for all stocks in the initial universe.

For these triggered stocks, LLMs are employed to extract sentiment from the news, offering insight into both the direction and magnitude of the potential impact. This targeted approach enhances efficiency and allows for better management of computational resources.

The procedures for the filtration process and sentiment analysis will be discussed in greater detail in the following sections.

Selection of Key Macroeconomic Indicators

An initial exhaustive list of indicators was compiled, totalling 153. These indicators include macroeconomic indicators such as GDP, inflation rates, and unemployment rates; commodities; foreign exchange rates; bonds; and cryptocurrencies. Additionally, industry and sector Exchange Traded Funds (ETFs) were incorporated. For the selection of ETFs, we prioritized those with the highest relevance and the largest market capitalization, ensuring they were equity-only. A complete list of these indicators can be found in Appendix A.

We aimed to achieve a more concise list of indicators through a combination of **quantitative** and **qualitative** approaches. In our *qualitative approach*, we studied the relevance between these indicators. Among those with high relevance in terms of their holdings, we chose those with higher market capitalization (e.g., sector ETFs over industry ones). We also sought advice from industry practitioners to gain insights into the important indicators they usually monitor to obtain trading signals. Additionally, we conducted trial and error assessments, evaluating the number of triggered tickers for each indicator over different time frames. Indicators with lower exclusivity (those triggering tickers too frequently) were removed.

For the *quantitative approach* to filter out highly correlated indicators, we employed the Variance Inflation Factor (VIF) measure to address multicollinearity. Multicollinearity occurs when some of the independent variables (predictors) are highly correlated, thereby containing similar information about the variance of the dependent variable and causing inflation of standard errors. VIF quantifies the severity of multicollinearity between independent variables.

Assuming the price of indicators to be $X_1, X_2, \ldots, X_n$, we used VIF by creating an auxiliary regression for the price of each indicator against all the others. Specifically, the regression model is $X_i = \beta_0 + \sum_{j \neq i} \beta_j X_j + \epsilon$, where $i = 1, \ldots, n$ and $j = 1, \ldots, n$ with $j \neq i$. VIF is then calculated as $\text{VIF} = \frac{1}{1-R_i^2}$, where R_i^2 is the R-squared from each of these regressions.

It should be noted that at the beginning of the first year of backtesting (2022), we used price data from the previous four years. To address multicollinearity, we exclude any indicators with a VIF greater than 10, ensuring that the retained indicators provide distinct and independent information.

To conclude, we have refined our selection to 50 indicators, down from the initial 153. This process involved excluding indicators with high multicollinearity and those representing the entire market. We eliminated broad market indicators because significant movements in these would influence most stocks due to their comprehensive coverage and importance as benchmarks. Such a wide-ranging impact would trigger signals for many stocks simultaneously, reducing the specificity of our analysis.

Detailed Explanation

We categorized indicators into two groups: those with daily price data, such as ETFs and market indices, and those with quarterly published data, including macroeconomic indicators like GDP, inflation, and unemployment rates. We sourced historical prices for daily indicators from Yahoo Finance, and for quarterly indicators, we utilized the FRED series provided by the Federal Reserve Bank of St. Louis.

In addition to the traditional beta, we also examined the co-movement of stocks and indicators during more extreme market conditions, specifically in the tails of the return distribution. We focused on how these variables interact when returns deviate significantly from the mean, either by exceeding one standard deviation above the mean or falling more than one standard deviation below the mean.

This approach allows us to assess the co-movement between stocks and indicators during periods of extreme positive returns (when returns are greater than $\mu + \sigma$) and extreme negative returns (when returns are less than $\mu - \sigma$), where μ represents the mean return and σ the standard deviation. By analyzing these tail conditions, we gain insights into how stocks and indicators behave together under more volatile and extreme market scenarios.

We calculated the traditional beta, as well as the co-movement of each indicator with stocks during extreme market conditions, using historical price data. For indicators with daily price data, we used a moving window of 120 trading days. For indicators with quarterly published data, we employed a moving window of 240 trading days. This analysis allows us to identify which indicators have demonstrated historical co-movement with particular stocks. When indicators exhibit larger-than-average performance, they will trigger movements in stocks with which they have shown historical co-movement.

Now, we are observing the z-score of the returns of the indicators:

$$Z = \frac{R - \mu}{\sigma},$$

where:

- R is the return
- μ is the mean return
- σ is the standard deviation of returns

We are using this approach because we are interested in larger-than-average movements of indicators, which can be identified by observing the z-score. A $|Z_{indicator}| > Z_{\text{threshold}}$ indicates that the return of the indicators is far from the mean and closer to the tails, indicating noticeable deviations from the average performance.

Formally:

$$|Z_{indicator}| > Z_{\text{threshold}} \implies \text{significant deviation from mean}$$

where:

- $|Z_{indicator}|$ is an absolute value of the z-score of indicators

- $Z_{\text{threshold}}$ is a predetermined threshold value

This method allows us to identify unusually large movements in either positive or negative directions.

We first focus on the z-scores of the indicator returns. For each indicator, we examine its z-score (Z) relative to a predetermined threshold ($Z_{\text{threshold}}$). When $|Z_{indicator}| \geq Z_{\text{threshold}}$, we proceed to analyze stocks that have historically shown co-movements with this particular indicator. The selection process is as follows: a stock is called triggered if $|\beta| > 1$ and

$$\begin{cases} Z_{\text{indicator}} \geq Z_{\text{threshold}} & \Rightarrow \text{significant co-movement during extreme positive returns,} \\ Z_{\text{indicator}} < -Z_{\text{threshold}} & \Rightarrow \text{significant co-movement during extreme negative returns.} \end{cases}$$

This process identifies stocks with strong historical co-movements with indicators showing significant deviations from their mean returns.

The filtration process is designed to identify stocks that are likely to react strongly to significant movements in specific indicators. The intuition is that when an indicator shows an unusually large change (as measured by its z-score exceeding a threshold, $|Z_{\text{indicator}}| \geq Z_{\text{threshold}}$), it may signal important economic shifts. We then look for stocks that have historically been sensitive to this indicator. The requirement that beta's absolute value exceeds 1 ($|\beta| \geq 1$) ensures that the stock typically moves at least as significantly as the indicator. Additionally, we refine this by considering how the stock behaves during extreme movements of the indicator—whether the indicator experiences large positive or negative changes. This approach aims to capture not just general co-movement, but also the stock's tendency to move more significantly during these extreme conditions.

We refer to the stocks selected through the above process as "**stocks triggered by indicators**".

We also defined another set of triggered stocks, which we refer to as "**self-triggered stocks**". These are stocks that show significant returns, measured by the z-score of their own return. We select stocks whose z-scores exceed the threshold, specifically $|Z_{\text{stock}}| > Z_{\text{threshold}}$, where Z_{stock} is the z-score of the stock's return and $Z_{\text{threshold}}$ is the same predetermined threshold used for indicator z-scores. This approach allows us to identify stocks experiencing significant movements.

$Z_{\text{stock}} < -Z_{\text{threshold}}$

The two approaches outlined above—triggered by indicators or self-triggered—will assist in distinguishing stock-specific movements from broader market or sector trends.

Given these two sets of triggered stocks, namely "stocks triggered by indicators" and "self-triggered stocks", we defined three distinct strategies:

1. Stocks triggered by indicators but not self-triggered (**Pure Beta**): $S_I \setminus S_S$
2. Intersection of self-triggered and indicator-triggered stocks (**Beta**): $S_I \cap S_S$
3. Stocks self-triggered but not triggered by indicators (**Pure Alpha**): $S_S \setminus S_I$

Where S_I represents the set of stocks triggered by indicators, and S_S represents the set of self-triggered stocks.

We will refer to these three strategies as **Pure Beta** (for stocks triggered by indicators only), **Beta** (for stocks triggered by both criteria), and **Pure Alpha** (for stocks triggered solely by their own movements and not by indicators). In the Pure Alpha strategy, we specifically isolate stocks that have moved significantly independent of their broader sector. This categorization allows us to examine the performance of the trading strategy across each of these distinct scenarios.

Next, we will screen news related to stocks filtered by any of the above procedures and utilize LLMs to analyze the sentiment. This will provide insights into both the direction and magnitude of the sentiment.

To determine the $Z_{\text{threshold}}$, eight $Z_{\text{threshold}}$ values were initially chosen for examination:

$$Z_{\text{threshold}} \in \{1, 1.5, 1.75, 2, 2.25, 2.5, 2.75, 3\}$$

Considering the computational and cost resources available for this study, and after a series of trials and errors, we selected an optimal $Z_{\text{threshold}}$ based on observing different metrics of triggered tickers. These metrics included the average number of triggered tickers and their distribution. Based on this analysis, we determined:

$$Z_{\text{threshold}} = 2$$

The selected threshold corresponds to returns that deviate from the mean by two standard deviations.

3.3.2 Sentiment Analysis Procedure

A well-structured prompt, based on the features discussed in Section 2.3, was employed. While the exact prompt remains the proprietary intellectual property of *ZanistaAI* and cannot be disclosed, key elements of its design will be discussed.

The sentiment analysis of news articles was performed utilizing *GPT-4o*, made available through *Azure* by *Microsoft*, with support from the infrastructure provided by *ZanistaAI*. *Azure*'s robust and scalable cloud platform ensured reliable access to *GPT-4o*. We selected *GPT-4o* from three options: *GPT-4o*, *GPT-4o mini*, and *Llama 3.1 70B*. One of the primary reasons for this choice is that *GPT-4o* is the newest and most sophisticated model with a larger capacity. Additionally, we conducted an experiment, detailed in Section 3.7, to evaluate which of these three models produces the most consistent sentiment analysis output when processing news both with and without the body text. The results, as presented in Section 4.1, demonstrate that *GPT-4o* is the most consistent, with its output showing the least variation when the news body is included.

For each news-stock pair, the following steps were taken:

- **Input:** The models were provided with the title and snippet of each news article, the corresponding stock to which the news was associated during the news scraping process, and the description of the stock associated with the news.
- **Sentiment Direction:** The models categorized the sentiment direction into one of the following:
  - "Strongly Positive"
  - "Positive"
  - "Negative"
  - "Strongly Negative"
  - "Neutral"
  - "Irrelevant"
- **Sentiment Intensity:** The intensity of the sentiment, reflecting the anticipated duration of its impact, was classified as:
  - "Long Term" for impacts lasting more than one month
  - "Medium Term" for impacts lasting one week to one month

– "Short Term" for impacts lasting one day to one week

– "Immediate" for impacts expected within a day

To quantify the directional outputs from the LLM, we implemented a mapping system where strongly positive news was assigned a value of 2, positive news a value of 1, Neutral or Irrelevant news a value of 0, and negative news mirrored these values with corresponding negative signs (i.e., -1 for negative and -2 for strongly negative news).

To quantify the impact of news on stock performance, we developed a sophisticated scoring system that accounts for both the intensity of the news and its temporal relevance. We defined a net cumulative score for news by assigning each piece of news a lasting impact based on its intensity. Crucially, we recognized that the influence of news naturally diminishes over time, reflecting the evolving nature of market reactions. To model this temporal aspect, we implemented a discrete decay function for the impact of each news item.

The decay of news impact is modelled using the following decay equation:

$$\text{Impact}(t) = \text{Initial Impact} \times (\text{decay factor})^t \tag{3.3.2}$$

In this equation, the decay factor represents the rate at which the impact decreases over time—in our case, a decay factor of 0.9. The variable t denotes the time elapsed since the news was released, measured in discrete time steps. This approach ensures that while news can exert a strong immediate effect, its influence gradually lessens over time, accurately reflecting the typical fading of market reactions.

To calculate the net cumulative score at any given time for a particular stock, we sum the decayed impacts of all relevant news to this particular stock:

$$\text{Net Cumulative Score} = \sum_{i=1}^{n} \text{Impact}_i(t_i) \tag{3.3.3}$$

where n is the number of news related to our stock being considered, t_i is the time elapsed since the release of the i-th news item, and $\text{Impact}_i(t_i)$ represents the decayed impact of that news item over time for a particular stock i. This method aggregates the influence of multiple news events, accounting for the natural decay of their impact as time progresses.

3.3.3 Technical Indicator - Moving Average Convergence Divergence

In addition to macroeconomic indicators, stock-specific factors, and news analysis, we also employed the Moving Average Convergence Divergence (MACD) technical indicator to execute trades [3].

The MAC indicator is widely used in analyzing stock price trends to detect shifts in momentum [1]. It involves determining the difference, between a 26-period Exponential Moving Average (EMA) and a 12-period EMA.

$$\text{MACD} = \text{EMA}_{12} - \text{EMA}_{26}, \tag{3.3.4}$$

In this formula, EMA_{12} refers to the 12-period Exponential Moving Average, while EMA_{26} refers to the 26-period Exponential Moving Average. The resulting MACD line oscillates around a zero line, reflecting the momentum of the trend. A positive MACD value indicates that the short-term average is above the long-term average, signaling upward momentum, while a negative value indicates downward momentum [3].

The signal line, which typically uses a 9-period EMA of the MACD, is used to generate trading signals:

$$\text{Signal Line} = \text{EMA}_9(\text{MACD}) \tag{3.3.5}$$

When the MACD line crosses above the signal line, it is seen as a bullish signal, indicating a potential buying opportunity. Conversely, when the MACD line crosses below the signal line, it is viewed as a bearish signal, suggesting a potential selling opportunity [1].

The MACD histogram, which represents the difference between the MACD line and the signal line, provides additional insights [3]:

$$\text{MACD Histogram} = \text{MACD} - \text{Signal Line} \tag{3.3.6}$$

The histogram fluctuates around the zero line, with positive values suggesting that the MACD is above the signal line (indicating bullish momentum) and negative bars suggesting that the MACD is below the signal line (indicating bearish momentum) [3].

For our study, when the MACD signal line crossed above the signal line, we considered it a Buy signal. Conversely, when the MACD crossed below the signal line, we regarded it as a Sell signal. In cases where there was no crossover, we interpreted it as a Hold signal.

3.3.4 Requirements for Executing Trades

Up to this point, we have detailed the process by which stocks are triggered (selected) for news exploration and sentiment analysis using LLMs.

To briefly summarize, as illustrated in Figure 3.2, the initial pool of stocks was filtered to obtain a smaller subset for news exploration and sentiment analysis. The stock filtering process occurs through two distinct mechanisms: self-triggered and indicator-triggered. In the self-triggered case, individual stocks exhibit significant price movements. Conversely, in the indicator-triggered case, the co-movement of relevant macroeconomic indicators prompts stock selection. The parameters β_{σ^+} and β_{σ^-} represent co-movement in the positive and negative tails, respectively. In Figure 3.2, stocks triggered by macroeconomic indicators are represented by orange circles, whereas stocks that are self-triggered are represented by blue circles. The union of these two sets of stocks forms the basis for further news exploration, conducted via *NewsWitch*, and sentiment analysis using `GPT-4o`. As discussed previously, we have established distinct trading strategies based on this stock selection process, including the Pure Alpha Strategy, the Pure Beta Strategy, and the Beta Strategy.

Once a particular stock is triggered on a specific date by any of these three strategies, we evaluate net cumulative news score and the MACD signal for that stock-date pair. A "Long Position" is opened only if both the net cumulative news score and the MACD signal align with the triggered information.

For instance, as shown in Figure 3.2, if a stock is triggered by the Pure Alpha Strategy, exhibiting a positive Z-score above $Z_{threshold}$ for the stock, without significant corresponding movements in co-moved indicators, a long position will be opened only if the MACD signal indicates a Buy, and the net cumulative news score is positive. Conversely, a "Short Position" will be opened if the conditions are exactly reversed. Otherwise, any existing position will be maintained until a negative net cumulative news score is encountered.

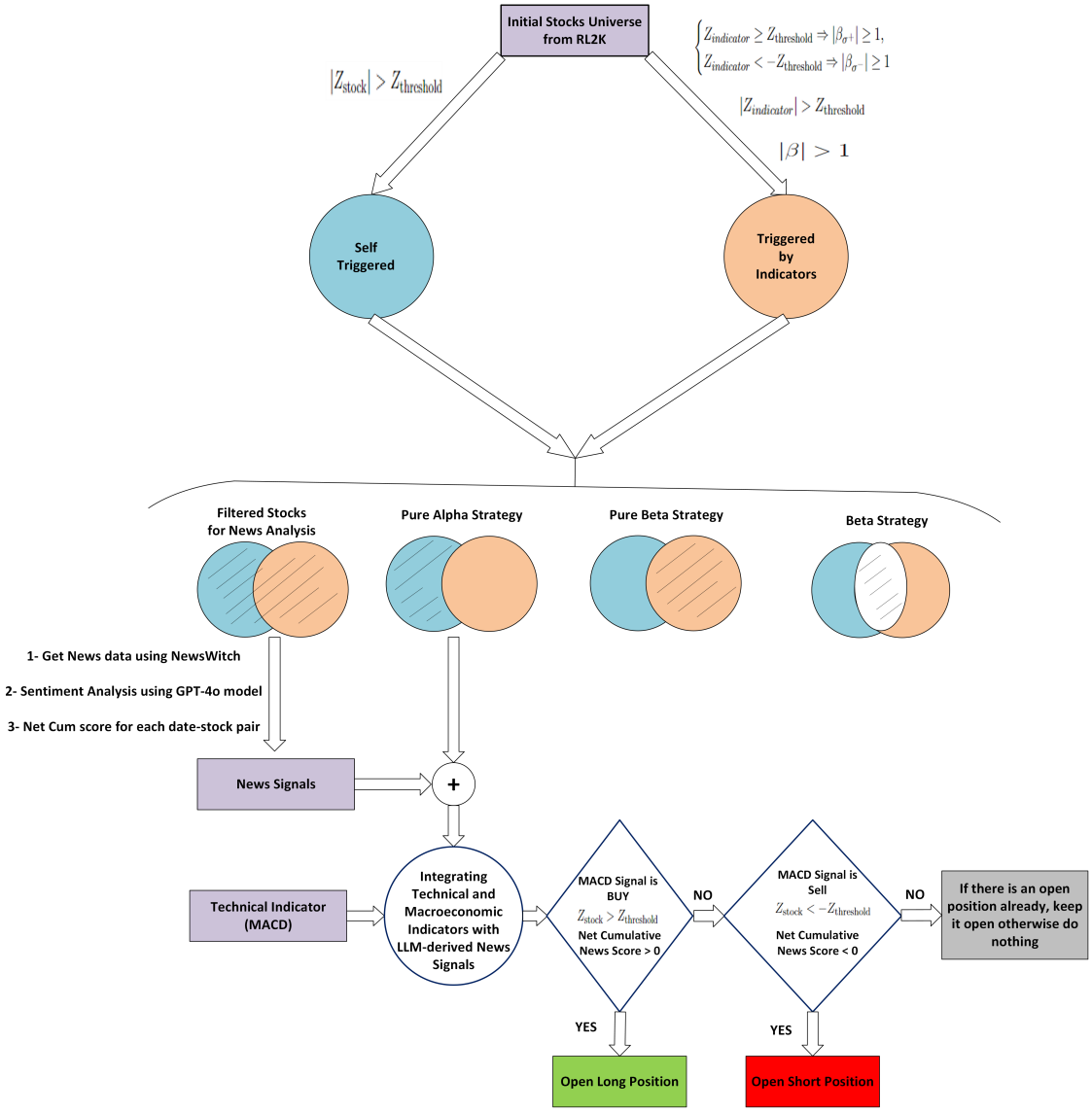

Figure 3.2: Flowchart of Stock Selection and Trading Strategy, integrating Technical and Macroeconomic Indicators with News Sentiment Analysis. Orange circles represent stocks triggered by indicators, and blue circles represent stocks triggered by themselves.

The same approach applies to opening a Short Position. For instance, in the Beta strategy, if the triggered information indicates a Sell signal—such as a stock exhibiting a negative Z-score below the $Z_{threshold}$ and corresponding indicators also showing negative Z-scores below the threshold—we will open the Short Position only if the net cumulative news score is also negative and the MACD signal indicates a Sell.

We employed a range of holding periods—1, 2, 3, 5, 10, 20, 40, and 60 days—to cover both short-term and long-term investment horizons. It should be noted that we continue holding positions as long as the news impact persists, even if it exceeds the maximum holding period. Positions are only closed when there is no positive net cumulative news score for long positions and no negative net cumulative news score for short positions.

As previously indicated, we do not have access to the exact time of news publication, and therefore, we cannot trade on the same day the signal arises. Specifically, we are unable to determine whether the news was released before the market opened, after it closed, or during trading hours. Consequently, we exploit the signal by opening and closing positions

using the next available opening price.

It is essential to highlight that our trading strategy relies exclusively on adjusted prices for all calculations and trade executions. Specifically, we initiate positions using the next available adjusted opening price and close positions using the next available adjusted closing price. Using adjusted prices allows us to automatically account for corporate actions such as stock splits, reverse splits, and dividend distributions. This ensures that our calculated returns are not distorted by these events, leading to a more accurate representation of the strategy's actual performance[35].

3.3.5 Portfolio Construction

The primary focus of our study was on the signal itself, so we employed two straightforward portfolio construction methods: equally weighted and risk-parity, with greater emphasis on the former. In the equally weighted approach, we invest an equal dollar amount in each position, allocating \$1M of Assets Under Management (AUM) equally between long and short positions. This method allows for clear attribution of performance to the quality of our sentiment signal.

The risk-parity[61, 43, 7] approach, in contrast, considers the volatility and correlation of stocks when determining portfolio weights. This method aims to equalize the risk contribution of each position to the overall portfolio risk, rather than equalizing the dollar amount. In practice, this often results in larger allocations to less volatile stocks and smaller allocations to more volatile ones.

The risk contribution of an asset i to the total portfolio risk is defined as:[60]

$$\text{Risk Contribution}_i = w_i \times \frac{\partial \sigma_p}{\partial w_i} = w_i \times \frac{(\Sigma w)_i}{\sigma_p} \tag{3.3.7}$$

where:

- w_i is the weight of asset i in the portfolio,
- $(\Sigma w)_i$ is the i-th element of the vector resulting from the multiplication of Σ and w,
- σ_p is the total portfolio volatility, defined as:

$$\sigma_p = \sqrt{w^T \Sigma w} \tag{3.3.8}$$

where:

- w is the vector of portfolio weights,
- Σ is the covariance matrix of asset returns.

The objective in risk parity is to find the portfolio weights w_i such that the risk contributions of all assets are equal, i.e.,

$$\text{Risk Contribution}_i = \text{Risk Contribution}_j \quad \forall i, j \tag{3.3.9}$$

To achieve this, we typically solve an optimization problem. A formulation for the risk budgeting problem, as presented by Bruder and Roncalli[7], can be expressed as:

$$w^* = \arg\min_w \sqrt{w^T \Sigma w}, \quad \text{s.t.} \quad \sum_{i=1}^{n} b_i \ln w_i \geq c, \quad \sum_{i=1}^{n} w_i^* = 1, \quad w_i > 0 \tag{3.3.10}$$

where:

- ln represents the natural logarithm,
- w_i represents the portfolio weights,
- Σ is the covariance matrix,
- b_i is the risk budget for asset i, with $\sum_{i=1}^{n} b_i = 1$,
- c is an arbitrary constant.

3.4 Transaction Costs

In this study, we incorporate transaction costs to evaluate their impact on the cumulative return and the Sharpe Ratio of our strategies. To simulate realistic trading conditions, we assume that transaction costs are constant and represent a fixed fraction of the asset price. Specifically, we examine transaction costs of 5, 10, 25, 50, and 100 basis points (bps).

Given the return without considering transaction costs, denoted by R, the return after accounting for constant transaction costs, denoted by R', can be computed as follows:

Let:

- P_1: Entry price (price at which the asset is bought)
- P_2: Exit price (price at which the asset is sold)
- R: Return without considering transaction costs
- R': Return after considering transaction costs
- x: Transaction cost as a percentage of the transaction amount

The return without transaction costs, denoted as R, is defined as:

$$R = \frac{P_2 - P_1}{P_1}$$

Transaction costs apply both when buying and selling the asset, which adjusts the entry and exit prices:

$$P_1' = P_1 \times (1 + x)$$

$$P_2' = P_2 \times (1 - x)$$

The return after considering transaction costs, R', is calculated using the adjusted prices:

$$R' = \frac{P_2' - P_1'}{P_1'}$$

Substituting the adjusted prices:

$$R' = \frac{P_2 \times (1 - x) - P_1 \times (1 + x)}{P_1 \times (1 + x)}$$
$$R' = \frac{(P_2 - P_1) - x(P_2 + P_1)}{P_1 \times (1 + x)}$$

Recognizing that $P_2 - P_1 = R \times P_1$ and substituting $P_2 = P_1 \times (1 + R)$:

$$R' = \frac{R \times P_1 - x\left[P_1 \times (1+R) + P_1\right]}{P_1 \times (1+x)}$$
$$R' = \frac{P_1 \times (R - x(2+R))}{P_1 \times (1+x)}$$

Finally, the return after considering transaction costs R' is:

$$R' = \frac{R(1-x) - 2x}{1+x} \tag{3.4.1}$$

3.5 Performance Metrics

After constructing the portfolio and running the strategy on historical data, we need to evaluate its performance using various commonly used ratios. This will enable us to compare our strategy's performance with relevant benchmarks and other trading strategies.

Price alone is insufficient as a reliable indicator to evaluate the performance of a trading strategy. Therefore, we must normalize the data to make meaningful comparisons. We primarily consider returns, which can be defined as

$$R_t = \frac{P_t - P_{t-1}}{P_{t-1}},$$

where P_t and P_{t-1} represent the current and previous prices, respectively. However, a positive return alone cannot indicate good performance, as almost any risk-free bond pays a nonzero rate. Therefore, we should consider the excess return, defined as

$$R_t^e = R_t - R^f,$$

where R^f is the risk-free rate. The excess return itself can be further disentangled into β and α, which represent the strategy's market exposure and its outperformance over the market, respectively, as shown in equation 3.5.1, where $R^e_{M,t}$ is the excess return of the market $R^e_{M,t} = R_t - R^f$[48, 55].

$$R_t = \alpha + \beta R^e_{M,t} + \epsilon_t \tag{3.5.1}$$

When it comes to α, often considered the "holy grail" of hedge funds, positive values are desired, while negative ones are usually seen as unfavorable. But is a higher α always better? This is where risk becomes important. Without factoring in risk, one could just use leverage to boost returns and increase profits, but this could lead to excessive risk. Therefore, it is crucial to consider risk-reward ratios [48].

One of the most well-known risk-reward ratios, is **Sharpe Ratio**, formulated as 3.5.2, which takes into account the risk and it does not depend on leverage[39].

$$SR_{t-1} = \frac{E_{t-1}\left[R_t - R_t^f\right]}{\mathrm{Std}_{t-1}\left(R_t - R_t^f\right)}. \tag{3.5.2}$$

The **Sortino Ratio** is a modification of the Sharpe Ratio that differentiates between downside risk and total volatility. While the Sharpe Ratio penalizes both upside and downside volatility, the Sortino Ratio only penalizes downside risk, providing a more accurate measure of a strategy's performance in terms of risk-adjusted returns[55]. Sortino Ratio can be found using equation 3.5.3.

$$\text{Sortino Ratio}_{t-1} = \frac{E_{t-1}\left[R_t - R_t^f\right]}{\text{Std}_{t-1}\left(\min(0, R_t - R_t^f)\right)} \tag{3.5.3}$$

Another important measure to evaluate the performance of a strategy is to assess how it performs relative to its past best performance. The high-water mark (HWM) is defined as the highest historical price of the fund [48]:

$$\text{HWM}_t = \max_{s \leq t} P_s,$$

and the drawdown (DD) is then defined as [55, 48]:

$$DD_t = \frac{\text{HWM}_t - P_t}{\text{HWM}_t}.$$

The maximum DD (**MAX DD**) is then reported as a percentage of the Asset Under Management (AUM), showing the largest peak-to-trough decline in the value of an investment or trading strategy during a specific period. It captures the worst possible loss an investor could have experienced if they had bought at the highest point and sold at the lowest point.

Another important metric is **Compound Return**, which represents the compounded return assuming all profits are reinvested. it captures the effect of earning returns on previous returns, aligning with the principle of the time value of money. This can be computed using the equation 3.5.4.

$$\text{Compound Return} = \left(\prod_{t=1}^{T}(1 + R_t)\right) - 1 \tag{3.5.4}$$

The **win ratio** is another important performance indicator that reflects the effectiveness of a trading strategy. It is defined as the percentage of profitable trades relative to the total number of trades executed. A higher win ratio indicates a greater proportion of successful trades.

Volatility, another crucial measure in the evaluation of strategies, quantifies the degree of fluctuation in a trading strategy's returns over time. It is typically expressed as the standard deviation of returns as a percentage. Volatility can be seen as an indicator of risk, with higher volatility suggesting greater uncertainty and the potential for both large gains and significant losses.

A common approach is to study the **correlation** of the trading strategy against a buy-and-hold strategy of the corresponding benchmark to assess how closely it tracks the benchmark. The Sharpe Ratio, Sortino Ratio, and volatility are typically reported as annualized values and can be compared to those of the benchmarks.

It should be noted that in the financial industry, most trading strategies are not self-financing (i.e., they involve cash inflows and outflows). Consequently, practitioners often use Profit and Loss (PnL) normalized by Assets Under Management (AUM) as the primary source of performance data, rather than returns. This approach provides a more accurate representation of strategy performance in real-world scenarios.

Turnover is an important metric that reflects the percentage of the portfolio that changes on each trading day. Although it is a time series metric, the average turnover is often reported to summarize the strategy's trading activity over a period. High turnover suggests a more active trading strategy, while low turnover indicates a more passive approach. The turnover for each day is defined as the sum of the absolute market values of the new positions opened on that day, divided by the gross market value of the portfolio from the previous day.

$$\text{Turnover}_t = \frac{\sum_{i=1}^{n} |MV_{i,t}|}{\text{GMV}_{t-1}}$$

where:

- $MV_{i,t}$ represents the market value of new positions (i) opened on day t,
- GMV_{t-1} is the total gross market value of the portfolio on the previous day.

For long positions, the market value is defined as the value of the assets held, calculated by multiplying the market price by the number of shares owned. In contrast, the market value of a short position represents the amount required to cover the short position, and it can be viewed as a liability. The **Gross Market Value** of the portfolio is the sum of the absolute market values of both long and short positions, while the **Net Market Value** of the portfolio is the market value of the long positions minus the market value of the short positions.

3.6 Stylized Facts

To evaluate a trading strategy, it is essential to assess whether the profit and loss (PnL) and returns generated by the strategy exhibit characteristics commonly observed in financial markets. This involves analyzing the PnL and returns to determine if they display key stylized facts[13]. For example, these facts include fat tails (which can be assessed through Tail Index analysis) and asymmetric distributions, which we will examine in the context of our strategy. By comparing these features with those of established financial returns, we can determine whether the strategy is realistic and well-aligned with typical market behavior.

Tail Index Analysis

The tail index is a key parameter in extreme value theory, used to quantify how heavy the tails of a distribution are. In the context of financial returns, it measures the likelihood of extreme events, such as significant gains or losses. A lower tail index indicates a heavier tail, meaning more frequent extreme events, which is typical of financial returns that exhibit fat tails [50].

The Hill estimator is a widely used method for estimating the tail index ξ. This method focuses on the largest observations within a dataset, as these observations provide the most insight into the behavior of the distribution's tail. The selection of the number of largest observations (denoted by k) to include in the estimation is critical, as it affects the balance between bias and variance in the estimator. As a straightforward, k was set to be $\sqrt{n}$, where n represents the sample size. This satisfies $K(n) \to \infty$ and $\frac{k(n)}{n} \to 0$ [50, 14].

Let $X_1, X_2, \ldots, X_n$ be our non-decreasing order of returns. The Hill estimator for the tail index ξ is given by[50]:

$$\hat{\xi} = \frac{1}{k} \sum_{i=1}^{k} \log X_{n-i+1} - \log X_{n-k}.$$

The tail index of financial returns was reported to be $2 < \xi \leq 5$[13].

Kurtosis

Kurtosis (κ) measures the "peakedness" of a distribution and specifically indicates the presence of extreme values in the tails. κ can be computed using Equation 3.6.1.

$$\kappa := \frac{\mathbb{E}[(X - \mathbb{E}[X])^4]}{\mathbb{E}[(X - \mathbb{E}[X])^2]^2} \tag{3.6.1}$$

For a normal distribution, the kurtosis (κ) is 3. If a symmetric, unimodal distribution has $\kappa > 3$, it is called leptokurtic, meaning it has a sharper peak and heavier tails. On the other hand, if $\kappa < 3$, it is called platykurtic, indicating a flatter peak and lighter tails [24].

Skewness

Financial return distributions are often not normal and tend to be asymmetric. This asymmetry can be measured by skewness, which is calculated using Equation 3.6.2. For a unimodal distribution, positive skewness means the right tail is longer or fatter, indicating more extreme positive returns. On the other hand, negative skewness implies the left tail is longer or fatter, suggesting a higher chance of extreme negative returns. In financial markets, skewness is a common characteristic, with negative skewness frequently seen in stock returns, reflecting a greater likelihood of large negative returns compared to positive ones [24].

$$\beta := \frac{\mathbb{E}[(X - \mathbb{E}[X])^3]}{\mathbb{E}[(X - \mathbb{E}[X])^2]^{3/2}} \tag{3.6.2}$$

3.7 Evaluating LLM Performance Without the News Body: Which Model is More Consistent?

This test was designed to determine which LLM model among our options performs better when the news body is excluded and to assess the consistency of each model. Additionally, we aimed to understand the impact of including the news body on sentiment analysis.

A dataset of 11,000 news articles was obtained using *NewsWitch*©[2], a service provided by *Zanista AI*. Each article included the title, snippet, body, and corresponding publication date. We employed three different LLMs from two companies, each varying in size (number of parameters):

- OpenAI's `GPT-4o` and `GPT-4o mini`
- Meta's `Llama 3.1 70B`

For each news article, sentiment was derived using all three models under two distinct approaches:

1. Sentiment based on "Title + Snippet"
2. Sentiment based on "Title + Body" (note that body includes snippet)

The sentiment analysis procedure described in Section 3.7 was applied uniformly across all news articles. Various metrics were then computed separately for each method ("Title + Snippet" and "Title + Body"). A detailed discussion of the results, including an assessment of model consistency and the impact of including the news body, will be presented in Section 4.1.

[2]In obtaining data, we followed the terms and conditions of `ZanistaAI`, which can be found at this link.

Chapter 4

Results

4.1 Assessing the Impact of News Body Exclusion on Sentiment Analysis: An Evaluation Across LLMs

Three different LLMs, namely `GPT-4o`, `GPT-4o mini`, and `Llama 3.1 70B`, were used to analyze the sentiment of 11,000 news articles, labelling their direction and intensity using two different methods: "Title+Body" and "Title+Snippet." We examined the impact of including the body of news on the direction and intensity of the sentiment as categorized by the different LLMs.

Our studies showed that 12% of news articles labelled as "Irrelevant" became relevant after adding the news body to the prompt using `GPT-4o`. This figure was higher for the other two models, with 18.5% of news for `GPT-4o mini` and 20.25% for `Llama 3.1 70B`. This suggests that `GPT-4o` is more effective at detecting the relevance of news from the "Title+Snippet" alone compared to the other two models. Additionally, the results from all models indicate a tendency for news to have a "Negative" direction. This tendency could be influenced by the timeframe and the specific stocks to which the news is associated.

For the rest of the study, news articles (including their body) labelled as "Irrelevant" by `GPT-4o`, accounting for 30% of all the news, were excluded.

As can be seen in Table 4.1, for `GPT-4o`, the intensity of sentiment changed for 42.12% of the news when the body was included, while this figure was higher for `GPT-4o mini` and `Llama 3.1 70B`, with changes in 55.63% and 53.42% of the news, respectively. However, `Llama 3.1 70B` appeared to be more consistent in terms of direction, with only 40.06% of the news showing a change in direction after including the body, compared to a higher change rate for the other two models. In terms of changes in both direction and intensity, the models were somewhat similar, with `GPT-4o` showing slightly better performance. We also analyzed the percentage of news articles where adding the body resulted in a change of view (i.e., from positive to negative or vice versa), revealing that `GPT-4o` and `GPT-4o mini` performed better than `Llama 3.1 70B`.

	GPT-4o	**GPT-4o mini**	**Llama3.1-70B**
Change in Intensity	42.12%	55.63%	53.42%
Change in Direction	50.12%	51.36%	40.06%
Change in (Direction, Intensity) pair	60.42%	64.61%	62.69%
Change in View (Direction)	65.07%	63.46%	69.28%

Table 4.1: Comparison of LLMs Consistency in Sentiment Analysis by Adding News Body in the Prompt (All numbers are percentages of the total relevant news.)

As shown in Figure 4.1, the percentage of irrelevant news is higher for all models when

the news body is excluded. Additionally, the figure illustrates how the inclusion of the body affects each model's assessment of the news direction.

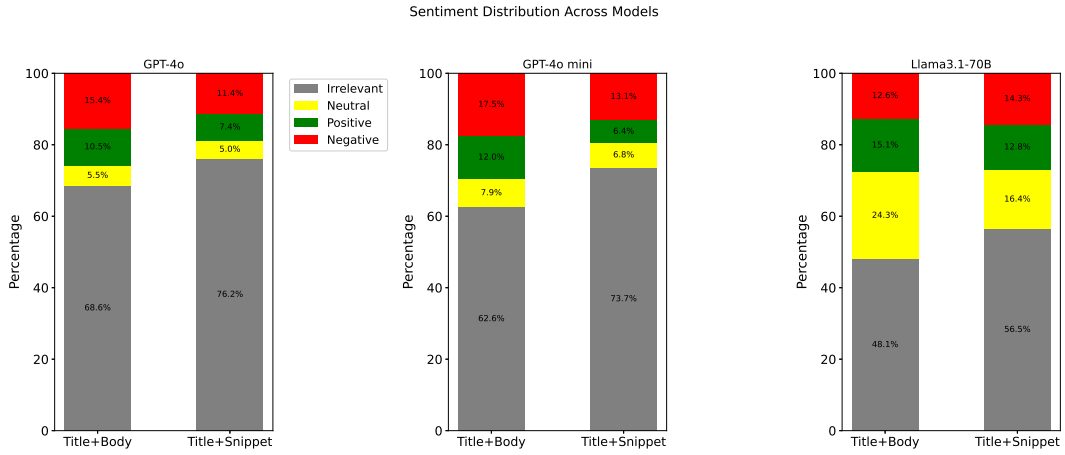

Figure 4.1: Distribution of news sentiment across different LLMs using two methods: "Title+Body" and "Title+Snippet"

We further studied the performance of the models and how sentiments vary across these three LLMs. As illustrated in 4.2, all three models agreed on the intensity of 54.85% of news when only the snippet was provided, without the body. However, when the body was included, their agreement significantly decreased by almost 40%. The direction of the news showed more consistency across models, with agreement on direction decreasing by only 2% after including the news body. When considering the direction-intensity pair predicted by the models, there is higher agreement when excluding the news body, 30% in comparison with 9.38% when including the news body.

The disagreement between models is higher when the body is included, at 6.6% compared to 4.15% when only the snippet is used. Agreement across models tends to be higher for news with a negative direction for both "Title+Body" and "Title+Snippet" approaches.

	Title+Body	**Title+Snippet**
Same (Intensity, Direction) pair	9.38%	30%
Same Intensity	15.89%	54.85%
Same Direction	44.43%	46.52%
Same view when Negative	34.43%	29.47%
Same view when Positive	21.30%	14.25%
Disagreement on the Direction	6.60%	4.15%

Table 4.2: Comparison of sentiment consistency between Title+Body and Title+Snippet approaches (All figures are percentages of the total relevant news).

In summary, among these three LLMs, we selected `GPT-4o` for the sentiment analysis of news because it shows the least variation in sentiment output compared to the other two models, as demonstrated in Table 4.1.

4.2 Evaluation of Strategy Performance (No Transaction cost)

In this section, we present the results of our trading strategies across different holding periods, analyzed year-over-year to demonstrate their performance consistency and facilitate inter-annual comparisons. We begin by providing a comprehensive overview of all three strategies—Beta, Pure Beta, and Pure Alpha—for various holding periods during 2022 and 2023. This initial presentation is followed by an in-depth analysis of the strategies that yielded the most promising outcomes.

Table 4.3 compares the performance of all three strategies across various holding periods in 2022 and 2023. The Pure Alpha strategy, which focuses on trading stocks experiencing significant returns independent of their associated co-moved indicators, consistently outperformed the other strategies in both 2022 and 2023. It achieved a notable reduction in Maximum Drawdown (Max DD %) from 4.69% in 2022 to 3.18% in 2023 for the 1-day holding period. Furthermore, the Sharpe Ratio improved from 3.64 to 5.10, and the Sortino Ratio from 6.56 to 8.92. Notably, the Sharpe Ratio for Pure Alpha remained above 5, and the Sortino Ratio above 6, up to the 5-day holding period in 2023, indicating superior risk-adjusted returns and effective downside risk management.

The Beta strategy, which involves trading stocks when both they and their associated co-moved indicators experienced larger-than-average moves, and the Pure Beta strategy, which trades stocks based on significant movements in their associated co-moved indicators without corresponding stock movements, also improved in 2023. For example, the Beta strategy's Sharpe Ratio rose from 1.32 to 2.69, and its Sortino Ratio from 1.66 to 5.75 for the 1-day holding period. However, these improvements were more moderate compared to Pure Alpha.

Additionally, these statistics, particularly the Sharpe Ratio, can be compared to those of the Russell 2000 Index (RL2K), which we used as a benchmark. The RL2K recorded much lower Sharpe Ratios of -0.77 in 2022 and 0.84 in 2023, further highlighting the superior performance of our strategies. The Sharpe Ratios of our strategies showed statistically significant superiority in both years ($p < 0.01$) in comparison with RL2K.

Overall, Pure Alpha demonstrated the most robust risk management and return generation. A comparison between the Risk Parity (RP) portfolio construction and the equally weighted approach (presented in the table) revealed that RP resulted in slightly lower Sharpe and Sortino Ratios while reducing overall risk, making it a more conservative approach. For instance, the Sharpe Ratio for Pure Alpha's 1-day holding period under RP was 2.3 in 2022 and 3.8 in 2023, still strong but lower than the equally weighted results. From this point forward, we will focus on the results of the equally weighted portfolio.

Strategy	Holding Days	Max DD %		Sharpe Ratio		Sortino Ratio	
		2022	**2023**	**2022**	**2023**	**2022**	**2023**
Beta	1	4.66	3.75	1.32	2.69	1.66	5.75
	2	4.66	3.75	1.34	2.67	1.70	5.68
	3	4.64	4.23	1.33	2.61	1.70	5.28
	5	5.11	4.40	1.27	2.59	1.51	5.17
	10	5.60	2.24	1.20	2.62	1.41	5.42
	20	6.08	2.24	1.07	2.71	1.24	5.59
	40	7.78	2.24	0.77	2.64	0.87	5.50
	60	7.68	2.22	0.79	2.69	0.91	5.68
Pure Beta	1	3.43	2.80	0.70	2.38	0.97	4.36
	2	3.39	2.81	0.70	2.39	0.97	4.37
	3	3.32	2.84	0.74	2.36	1.03	4.36
	5	3.50	2.91	0.69	2.41	0.99	4.37
	10	3.77	3.36	0.60	2.17	0.83	3.60
	20	3.90	3.22	0.63	2.15	0.90	3.56
	40	4.85	4.02	0.38	1.95	0.52	3.33
	60	5.11	4.14	0.24	1.95	0.33	3.38
Pure Alpha	**1**	**4.69**	**3.18**	**3.64**	**5.10**	**6.56**	**8.92**
	2	4.67	3.27	3.56	5.07	6.34	8.86
	3	4.64	3.33	3.55	5.06	6.24	8.85
	5	4.62	3.58	3.46	5.04	6.09	8.82
	10	5.07	3.65	3.12	4.88	5.47	8.45
	20	6.32	3.75	2.57	4.48	4.16	7.79
	40	7.79	3.95	2.05	4.19	3.19	7.34
	60	8.26	4.92	1.73	3.98	2.64	6.90

Table 4.3: Performance Comparison of Beta, Pure Beta, and Pure Alpha Strategies Across Different Holding Periods for the years 2022 and 2023.

Table 4.4 presents the trade statistics for the Beta, Pure Beta, and Pure Alpha strategies across various holding periods in 2022 and 2023. These statistics include the number of trades executed, the turnover percentage, and the percentage of winning trades, offering a comprehensive view of the trading activity and effectiveness of each strategy.

The Pure Alpha strategy, which generated the highest number of trades, showed a significant increase in trading volume from 468 trades in 2022 to 748 trades in 2023 for the 1-day holding period. This strategy also maintained a relatively high winning trade percentage, improving from 60.04% in 2022 to 66.71% in 2023. Turnover for Pure Alpha decreased slightly, indicating more efficient trading practices while maintaining strong performance.

In comparison, the Beta and Pure Beta strategies showed stable trading volumes and improved win percentages in 2023, with decreases in turnover. For example, the Beta strategy saw its turnover decrease from 6.78% to 5.73% for the 1-day holding period, while the winning trade percentage increased from 61.90% to 73.19%. Similarly, Pure Beta also exhibited a decrease in turnover alongside an increase in the percentage of winning trades. While these changes resulted in higher Sharpe Ratios in 2023, they still did not reach the effectiveness of Pure Alpha.

Strategy	Holding Days	Number of Trades		Winning Trades (%)		Turnover (%)	
		2022	2023	2022	2023	2022	2023
Beta	1	63	138	61.90	73.19	6.78	5.73
	2	63	138	61.90	73.19	6.78	5.73
	3	63	138	61.90	72.46	6.72	5.73
	5	63	138	60.32	71.74	8.71	5.70
	10	63	138	58.73	70.29	7.72	5.50
	20	63	138	53.97	69.57	7.83	4.85
	40	63	138	49.21	69.57	6.68	8.97
	60	63	138	50.79	69.57	6.37	8.41
Pure Beta	1	196	268	44.90	54.10	15.32	7.00
	2	196	268	44.39	53.73	15.32	6.97
	3	196	268	43.88	52.49	14.96	6.95
	5	196	268	41.33	51.49	17.19	6.90
	10	195	268	40.51	50.37	14.24	6.30
	20	195	268	35.90	49.25	12.27	6.09
	40	195	268	33.85	47.01	11.35	5.76
	60	195	268	33.33	47.01	9.30	5.57
Pure Alpha	1	468	748	60.04	66.71	11.42	8.79
	2	468	748	60.04	67.11	11.36	8.74
	3	468	748	59.62	66.18	11.18	8.65
	5	468	748	58.62	65.64	10.86	8.55
	10	467	748	56.32	63.64	9.57	7.97
	20	465	742	50.11	60.32	7.07	7.29
	40	465	742	44.52	58.22	6.51	6.69
	60	463	741	42.12	56.68	6.24	6.40

Table 4.4: Trade Statistics for Beta, Pure Beta, and Pure Alpha Strategies Across Different Holding Periods in 2022 and 2023.

Now, we present the figures for the **most desired strategy** and holding day, specifically **Pure Alpha Strategy with a holding period of 1 day**. Additional figures can be found in Appendix B.

Figure 4.2 presents the compound returns of the Pure Alpha 1-Day Holding Strategy alongside the Russell 2000 Index (RL2K) for the years 2022 and 2023. The results clearly demonstrate the superior performance of the Pure Alpha strategy over the benchmark across both years.

In 2022, the Pure Alpha strategy consistently outperformed the RL2K, with the compound returns reaching nearly 38% by the end of the year, while the RL2K exhibited a significant decline, ending the year with a return of approximately -22%. This contrast underscores the effectiveness of the Pure Alpha strategy in generating positive returns even during periods of broader market decline.

The trend continued into 2023, where the Pure Alpha strategy not only maintained its outperformance but also accelerated its growth, achieving compound returns exceeding 60% by the year's end. In contrast, the RL2K showed some recovery but remained volatile, with returns fluctuating and ending the year around 16%. This further emphasizes the resilience and strong return generation capability of the Pure Alpha strategy compared to the RL2K.

The consistent upward trajectory of the Pure Alpha strategy across both years, particularly in the face of market volatility as represented by the RL2K, highlights its robustness and effectiveness as a trading strategy.

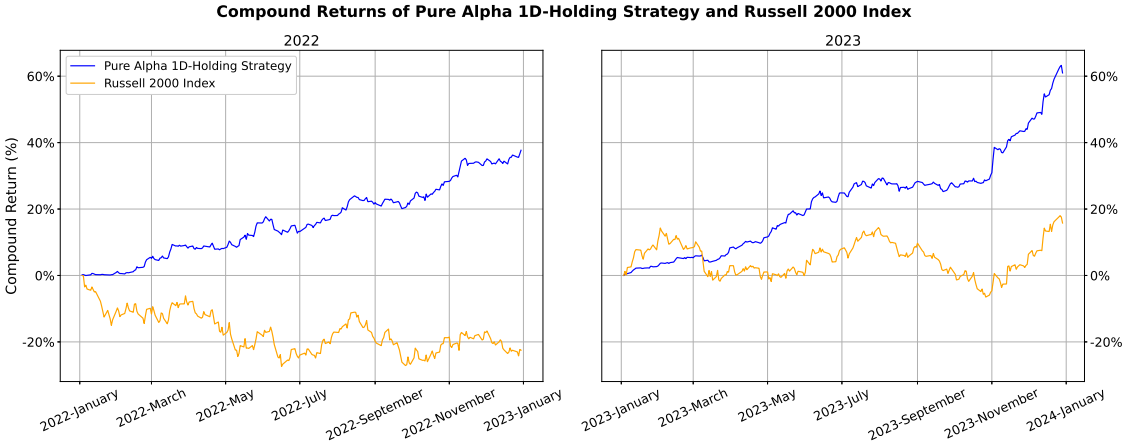

Figure 4.2: Compound Returns of the Pure Alpha 1-Day Holding Strategy and Russell 2000 Index (RL2K) in 2022 and 2023.

Figure 4.3 compares the 60-day rolling Sharpe Ratios of the Pure Alpha 1-Day Holding Strategy to those of the Russell 2000 Index (RL2K) during the years 2022 and 2023. In 2022, the Pure Alpha strategy consistently maintained a rolling Sharpe Ratio significantly above that of the RL2K. The strategy's Sharpe Ratio fluctuated around a solid average of 3.64, whereas the RL2K exhibited much lower and more volatile Sharpe Ratios, averaging -0.77 for the year. This considerable contrast highlights the superior risk-adjusted performance of the Pure Alpha strategy.

The trend of outperformance continued into 2023, where the Pure Alpha strategy's rolling Sharpe Ratio increased, peaking at values above 10, and averaged 5.10 over the year. Meanwhile, the RL2K showed some recovery, with its Sharpe Ratio improving to an average of 0.84. Despite this, the Pure Alpha strategy's Sharpe Ratio remained well above that of the RL2K throughout the year, further underscoring its superior risk-adjusted return profile.

While the overall trend mirrors that of the benchmark (RL2K), our strategy clearly outperforms it. This similarity in trend is likely due to the fact that we are trading stocks that are constituents of the RL2K.

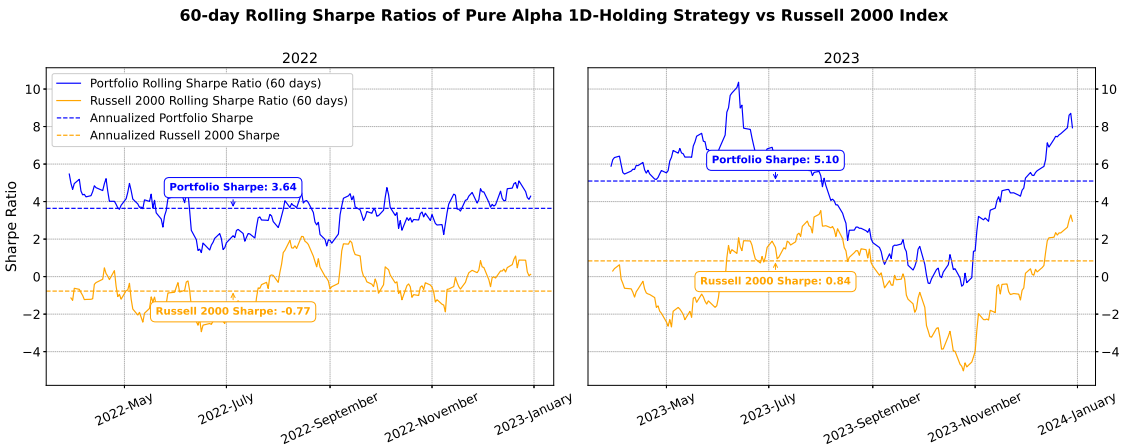

Figure 4.3: 60-day Rolling Sharpe Ratios of the Pure Alpha 1-Day Holding Strategy vs. Russell 2000 Index (RL2K) in 2022 and 2023.

Figure 4.4 presents the cumulative profit and loss (PnL) and drawdowns for the Pure Alpha 1-Day Holding Strategy across the years 2022 and 2023. The cumulative PnL, shown in green, reflects the strategy's ability to generate consistent profits over time, while the drawdowns, depicted in red as a percentage of assets under management (AUM), indicate the maximum observed losses from a peak to a trough within the period. Additionally, the High Water Mark (HWM), shown in blue, represents the highest point in cumulative PnL that the strategy achieved, providing a reference for drawdown calculations.

In 2022, the strategy experienced a maximum drawdown (Max DD) of 4.69% on June 16, as highlighted in the figure. Despite this, the strategy quickly recovered and continued to accumulate profits, ending the year with a strong cumulative PnL. Similarly, in 2023, the maximum drawdown was significantly lower, at 3.18% on September 22, demonstrating the strategy's improved risk management and resilience. The strategy maintained an upward trajectory throughout the year, with the cumulative PnL reaching new highs and consistently setting new HWMs by the end of the period.

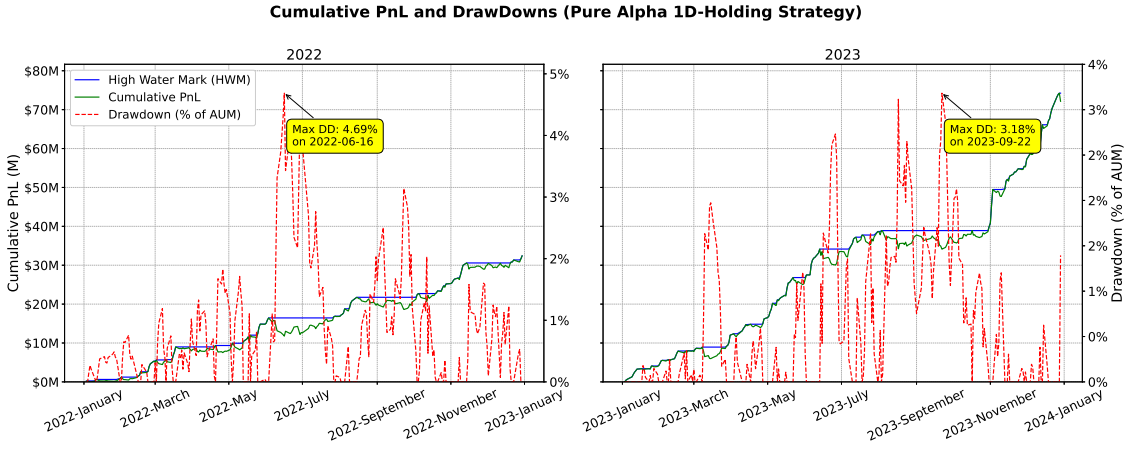

Figure 4.4: Cumulative PnL and Drawdowns for the Pure Alpha 1-Day Holding Strategy in 2022 and 2023. The figure illustrates the cumulative profit and loss (PnL) on the left y-axis alongside drawdown percentages on the right y-axis. AUM was $100 million in 2022 and $150 million in 2023.

Figure 4.5 illustrates the rolling 60-day realized volatility of the Pure Alpha 1-Day Holding Strategy, alongside the volatility of the Russell 2000 Index and the Russell 2000 Volatility Index (RVX) for the years 2022 and 2023. The figure provides a comprehensive view of the strategy's risk profile in relation to market benchmarks, with volatility expressed both in dollar terms and as a percentage (% of AUM).

In 2022, the Pure Alpha strategy exhibited moderate volatility levels, with whole-year volatility amounting to $8.94M, representing 8.94% of AUM. This relatively stable volatility is reflected in the strategy's consistent upward trajectory, even as the Russell 2000 Index and RVX experienced more pronounced fluctuations. Notably, there is a significant difference between the RVX, which represents forward-looking market expectations of volatility, and the realized volatility of the Russell 2000 Index, with the RVX generally showing higher volatility levels. This difference highlights the unpredictability and increased market uncertainty during this time.

Moving into 2023, the strategy's volatility increased, with the whole-year figure rising to $14.27M, equivalent to 9.51% of AUM. Despite this rise in volatility, the Pure Alpha strategy continued to perform well, maintaining a solid risk-adjusted return profile. Interestingly, the rolling volatility of the Pure Alpha strategy exhibits a trend somewhat

similar to the realized volatility of the Russell 2000 Index (indicated in yellow), reflecting the fact that the strategy trades stocks that are constituents of the Russell 2000 Index. However, the Pure Alpha strategy's volatility remains more controlled compared to the broader market, demonstrating its effective risk management.

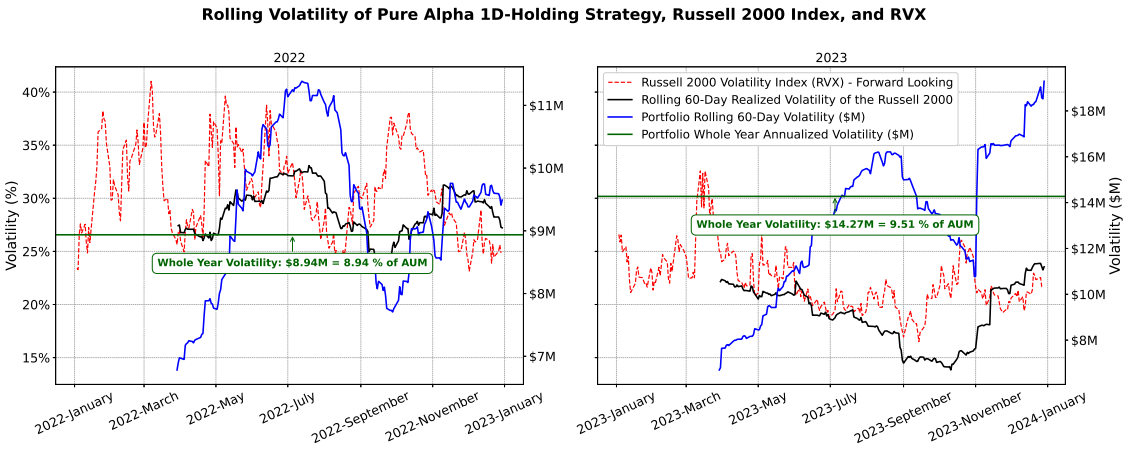

Figure 4.5: Rolling Volatility of the Pure Alpha 1-Day Holding Strategy, Russell 2000 Index, and RVX in 2022 and 2023. The right y-axis represents the volatility of the Pure Alpha strategy in millions of dollars (M$), while the left y-axis shows the volatility of the RVX and Russell 2000 Index in percentage terms. AUM was $100 million in 2022 and $150 million in 2023.

Figure 4.6 shows the long, short, gross, and net market values of the Pure Alpha 1-Day Holding Strategy for 2022 and 2023, with all values in millions of dollars (M$). The green line represents the total value of long positions, the red line shows the absolute value of short positions, the blue line illustrates the gross market value (sum of long and short positions), and the purple line indicates the net market value (difference between long and short positions).

In 2022, the strategy maintained stable long and gross market values, with minimal short exposure, leading to net market values closely mirroring long values. This reflects a predominantly long-biased approach, contributing to the strategy's positive performance.

In 2023, similar market value dynamics persisted, with gross values generally tracking close to long values. However, the latter half of 2023 saw a noticeable increase in both gross and net market values, indicating greater exposure.

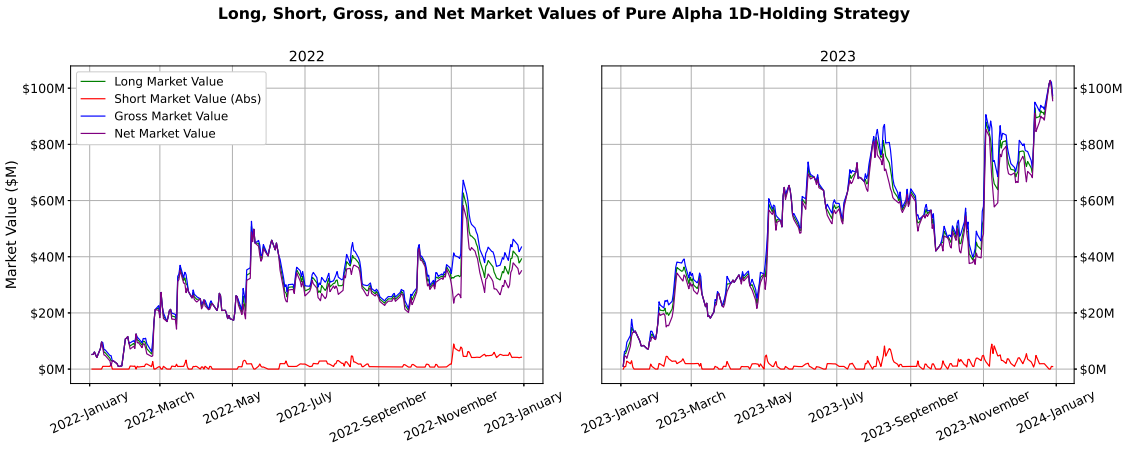

Figure 4.6: Long, Short, Gross, and Net Market Values of the Pure Alpha 1-Day Holding Strategy in 2022 and 2023, with all values expressed in millions of dollars (M$). AUM was $100 million in 2022 and $150 million in 2023.

Figure 4.7 presents boxplots of daily returns for the Pure Alpha Strategy with different holding periods in 2022 and 2023, compared to the Russell 2000 Index. In both 2022 and 2023, the Russell 2000 Index (represented by the first boxplot in each panel) exhibited a wider distribution of returns compared to the Pure Alpha strategy, indicating greater volatility in the index. The median return for the Russell 2000 was slightly negative in both years, reflecting the challenges faced by the broader market.

The Pure Alpha strategy, across all holding periods, showed a more compact distribution of returns, with fewer extreme outliers, especially in 2023. This suggests that the strategy was effective in controlling the daily return volatility, regardless of the holding period. The median returns for the Pure Alpha strategy were consistently closer to zero, with slight positive biases, indicating a stable performance across different market conditions.

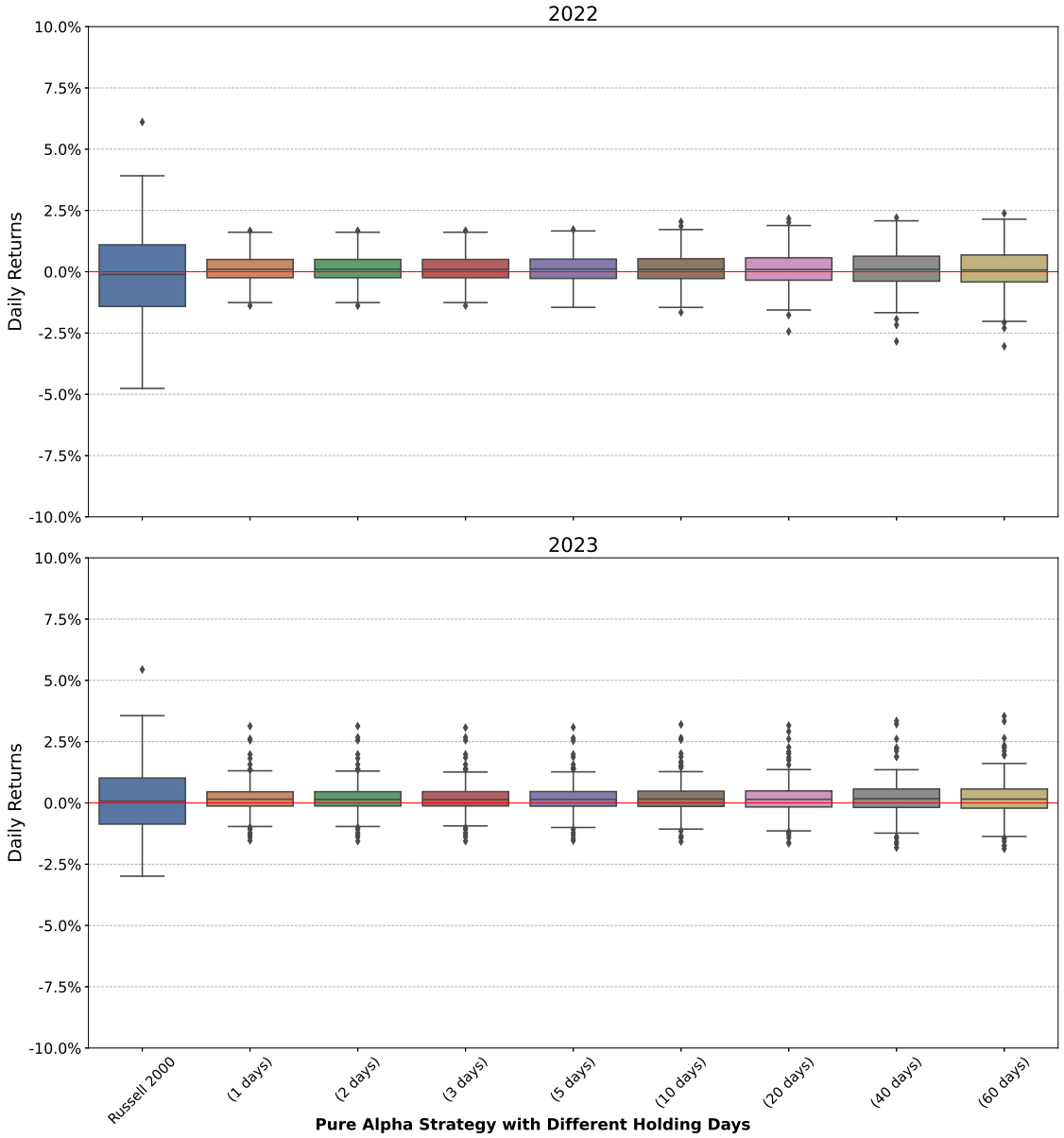

Figure 4.7: Boxplots of Daily Returns for the Pure Alpha Strategy with Different Holding Periods in 2022 and 2023 Compared to the Russell 2000 Index.

Furthermore, we analyzed the performance of our trading strategy across the 11 sectors and 171 industries to which our stocks universe belong.

Sector and Industry-wise Performance Analysis

Table 4.5 highlights the performance of various sectors in 2022 and 2023. The Financials sector led in 2023, showing a substantial increase in Sharpe Ratio from 2.59 in 2022 to 4.03, coupled with a low Max Drawdown of 0.91% and an increase in winning trades from 67.57% to 85.00%. Consumer Discretionary and Health Care also performed strongly, with Sharpe Ratios of 3.84 and 3.41 in 2023, respectively, although Health Care experienced increased volatility as indicated by a higher Max Drawdown. The Real Estate sector demonstrated a significant recovery, with its Sharpe Ratio improving from -0.36 in 2022 to 3.72 in 2023, and a notable rise in winning trades to 92.00%.

Other sectors like Information Technology and Industrials also saw considerable improvements in 2023, reflecting better market conditions. However, sectors such as Energy and Consumer Staples, despite improvements, had relatively lower Sharpe Ratios and higher drawdowns, indicating ongoing challenges.

Sectors	Sharpe Ratio		MaxDD %		Trades Count		Win Ratio %	
	2022	2023	2022	2023	2022	2023	2022	2023
Financials	2.59	4.03	0.81	0.91	37	60	67.57	85.00
Consumer Discretionary	2.19	3.84	1.53	2.07	82	100	59.76	65.00
Health Care	3.43	3.41	1.93	7.33	123	170	64.23	60.00
Real Estate	-0.36	3.72	0.76	0.45	11	25	54.55	92.00
Information Technology	1.13	3.37	2.31	3.66	47	95	53.19	60.00
Industrials	2.00	3.29	2.94	3.23	73	136	52.05	66.18
Communication	0.70	3.05	1.84	1.47	29	52	55.17	69.23
Materials	1.67	2.64	0.96	1.24	16	35	56.25	71.43
Consumer Staples	2.14	2.13	1.08	0.61	18	37	72.22	70.27
Energy	0.60	1.69	4.39	2.20	21	33	61.90	60.61

Table 4.5: Performance Statistics Across Sectors for 2022 and 2023, ordered by their Sharp Ratios in 2023.

Figure 4.8 illustrates the 60-day rolling Sharpe Ratios for the top-performing sectors, the overall strategy, and the Russell 2000 Index across 2022 and 2023. In 2022, the Health Care sector consistently maintained a high rolling Sharpe Ratio, often outperforming both the overall strategy and other sectors. The Financials and Consumer Discretionary sectors also demonstrated strong performance, though with more volatility as reflected in their fluctuating Sharpe Ratios. Notably, the Russell 2000 Index exhibited much lower and more volatile Sharpe Ratios throughout the year.

Both the Financials and Consumer Discretionary sectors recorded steady gains in their Sharpe Ratios, reinforcing their status as key drivers of the strategy's success. Meanwhile, the Russell 2000 Index continued to lag behind, with its rolling Sharpe Ratio remaining below that of the strategy and its leading sectors.

The performance of the top sectors varies each year, with different sectors leading in 2022 and 2023. This variability highlights the strategy's adaptability in leveraging the strengths of the best-performing sectors annually.

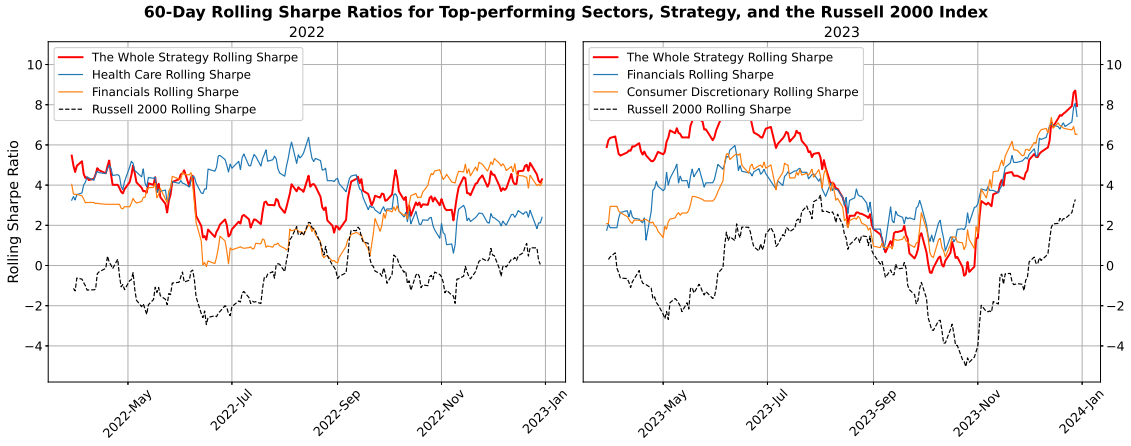

Figure 4.8: 60-Day Rolling Sharpe Ratios for the Top-Performing Sectors, the Whole Strategy, and the Russell 2000 Index in 2022 and 2023.

The performance of the Biotechnology industry was notable in both 2022 and 2023. In 2022, Biotechnology led with an impressive Sharpe Ratio of 3.19, supported by a Winning Trades Percentage of 65.28% across 72 trades. The sector continued to perform strongly in 2023, achieving a Sharpe Ratio of 2.84 despite an increase in the number of trades to 102. The Winning Trades Percentage for Biotechnology in 2023 was slightly lower at 60.78%, yet the industry remained one of the top performers throughout the year.

4.3 Incorporating Transaction Costs

We incorporated transaction costs as a constant percentage of the price.

Figure 4.9 illustrates the impact of transaction costs on the performance of the Pure Alpha 1-Day Holding Strategy across the years 2022 and 2023. The rolling 60-day Sharpe Ratio is used as a performance metric, with each line representing a different level of transaction cost, ranging from 0 bps to 150 bps. In both years, it is evident that as transaction costs increase, the Sharpe Ratio tends to decrease, indicating a deterioration in strategy performance. This trend is consistent across both years, though the magnitude of the Sharpe Ratio and its sensitivity to transaction costs vary. The Russell 2000 Index's rolling Sharpe Ratio is included as a benchmark, represented by the yellow line, demonstrating that even after incorporating transaction costs, the Pure Alpha 1-Day Holding Strategy outperforms the benchmark.

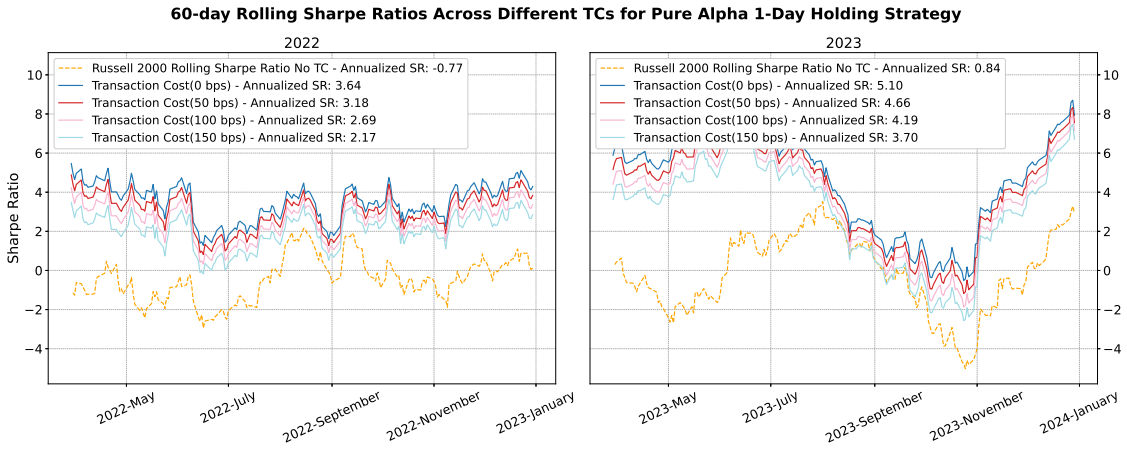

Figure 4.9: 60-day Rolling Sharpe Ratios Across Different Transaction Costs for the Pure Alpha 1-Day Holding Strategy in 2022 and 2023. Annualized Sharpe Ratios are shown in the legend for both years.

Figure 4.10 illustrates the compound returns of the Pure Alpha 1-Day Holding Strategy under varying levels of transaction costs (TC) across the years 2022 and 2023. The cumulative returns are displayed for transaction costs ranging from 0 to 150 basis points (bps). In 2022, the strategy shows positive returns across all levels of transaction costs, although the returns decrease as transaction costs increase. Despite this, the strategy consistently outperforms the Russell 2000 Index, which experienced a negative return over the same period.

In 2023, the Pure Alpha strategy continues to generate substantial positive returns, with the highest returns observed in the absence of transaction costs. As expected, the final cumulative returns decline as transaction costs increase. However, even with the highest transaction costs considered (150 bps), the strategy still outperforms the Russell 2000 Index, which shows modest gains.

This analysis highlights the resilience of the Pure Alpha strategy in generating positive returns, even when accounting for transaction costs, and its consistent outperformance relative to the market benchmark.

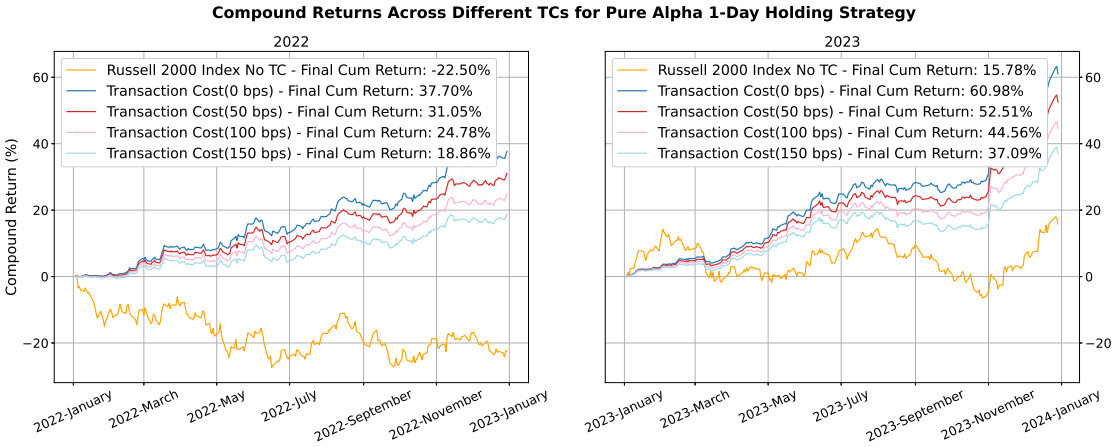

Figure 4.10: Compound Returns Across Different Transaction Costs for the Pure Alpha 1-Day Holding Strategy in 2022 and 2023. The final cumulative returns are shown in the legend for both years, comparing the impact of transaction costs ranging from 0 to 150 basis points (bps). The Russell 2000 Index is included as a benchmark.

Figure 4.11 illustrates the effect of varying transaction costs on the Sharpe Ratio for the Pure Alpha 1-Day Holding Strategy in the years 2022 and 2023. The x-axis represents the transaction costs as a percentage, while the y-axis shows the corresponding Sharpe Ratio. In both years, a clear downward trend is observed, where the Sharpe Ratio decreases as transaction costs increase. This trend indicates that the profitability and risk-adjusted performance of the strategy are negatively impacted by higher transaction costs.

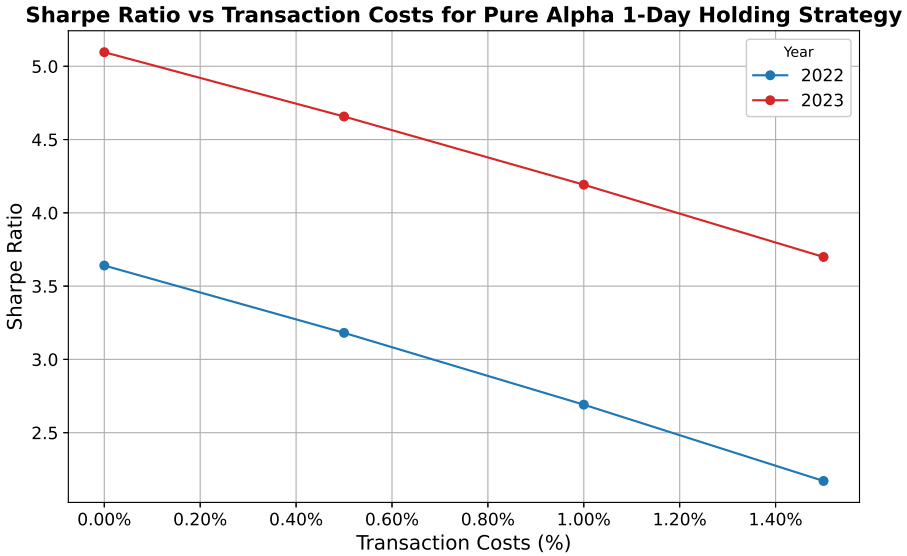

Figure 4.11: Sharpe Ratio vs Transaction Costs for the Pure Alpha 1-Day Holding Strategy in 2022 and 2023.

4.4 Stylized Facts

We conducted **tail index analysis** on PnLs generated by our strategy to see if they follow similar patterns as other established financial instruments.

Here we present these results for our best-performing strategy as an example. For the Pure Alpha 1D-Holding Strategy in the year 2023, we calculate the tail index to be 2.47, which is in line with the tail index commonly observed for financial returns, reported to be between 2 and 5 [13].

Figure 4.12 shows the distribution of Daily PnL for the Pure Alpha 1D-Holding Strategy in 2023. The blue bars represent the frequency distribution, with the green curve overlaying a Normal Distribution Kernel Density Estimate (KDE) for a smooth probability density. The red dashed line marks the maximum daily loss of -$2.31M. This distribution reflects key stylized facts, such as return asymmetry and the presence of fat tails, highlighting its alignment with typical market behavior.

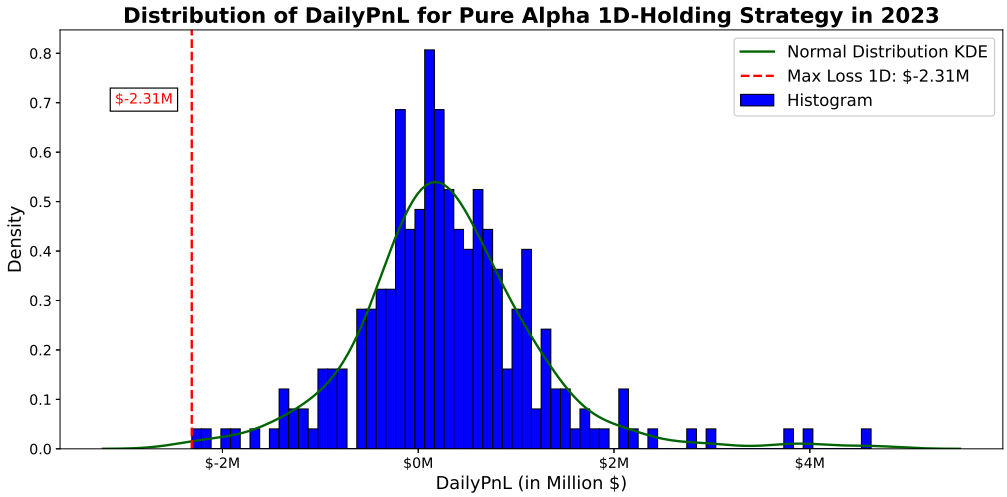

Figure 4.12: Distribution of Daily PnL for the Pure Alpha 1D-Holding Strategy in 2023.

For the Pure Alpha 1D-Holding Strategy in 2023, the **skewness** and **kurtosis** were calculated to be 0.6 and 6.5, respectively. The positive skewness indicates a distribution with a longer right tail, suggesting a tendency for more extreme positive PnLs. The kurtosis value of 6.5 is significantly higher than that of a normal distribution (3), indicating leptokurtic behavior with heavier tails and a higher peak. These statistics, along with the tail index of 2.47, align with typical financial return characteristics[13], demonstrating the strategy's consistency with established stylized facts.

Chapter 5

Discussion

5.1 Further Elaboration on Results

Choice of LLM: Due to time constraints and limited resources, we focused on processing the news title and snippet as input to the LLM, excluding the full news body. To select the most effective and consistent LLM, we compared three models: `GPT-4o`, `GPT-4o mini`, and `Llama 3.1 70B`. As detailed in Section 4.1, `GPT-4o` outperformed the others in efficiently detecting the relevance of news from titles and snippets. Moreover, `GPT-4o` provided the most consistent sentiment analysis results, particularly in determining the direction and intensity of sentiment, with minimal variation when the full news body was included. Additionally, previous studies have shown that more complex LLMs with a higher number of parameters generally perform better in similar tasks [40]. For these reasons, we chose to use `GPT-4o`.

The Role of Macroeconomic Indicators in our Study: Macroeconomic indicators played a crucial role in our study, serving two primary purposes. Firstly, we employed these indicators to streamline our news analysis process. Rather than examining news for all stocks in the initial pool, we focused our attention on stocks whose associated co-moving indicators or the stocks themselves exhibit significant price changes. This approach allowed us to concentrate our resources on potentially more informative news events. Secondly, we utilized these macroeconomic indicators to define a strategy we termed "Pure Alpha". This strategy aimed to isolate stocks that demonstrated significant price movements independent of their co-moving indicators. Specifically, we identified instances where individual stocks experienced considerable price changes while their associated co-moving indicators remained relatively stable. This approach sought to capture alpha-generating opportunities that were potentially driven by stock-specific factors rather than broader market or sector movements.

Optimizing Signal Analysis and Decision-Making Framework: Unlike the majority of previous studies[40, 34, 11], which employed a straightforward strategy of buying stocks on positive news and selling those on negative news, we implemented a novel, multi-faceted approach. Our methodology is distinguished by several key features that aim to provide a more comprehensive and nuanced analysis of market dynamics.

Firstly, when conducting sentiment analysis of news, we extracted both the intensity and direction of each news item. We categorized each of these outputs to differentiate between various levels of impact. For example, we distinguished strong positive news from moderately positive news, and highlighted the difference between news with potential long-term effects and those with shorter-term impacts. Next, we considered all pre-event news rather than relying on single news items. This comprehensive news analysis provided a more holistic view of market sentiment, allowing us to capture the cumulative effect of multiple news sources over time. To reflect the dynamic nature of market information, we

implemented a decay function to account for the diminishing impact of older news. This temporal weighting ensures that recent developments are given appropriate significance while still considering the lingering effects of past events.

Our model goes beyond stock-specific news by incorporating broader economic factors. We also included macroeconomic indicators, enabling us to disentangle stock-specific movements (alpha) from movements stemming from the broader market, sector, or index (beta). To complement this fundamental analysis, we utilize MACD as a technical indicator, providing insights into momentum and potential trend reversals. The integration of these diverse analytical tools—fundamental, macroeconomic, and technical—offers a more comprehensive framework for evaluating trading opportunities and understanding market dynamics.

Rationale for Profitability of our Trading Strategy: We demonstrated that our trading strategy is profitable and outperforms the corresponding benchmark, even after accounting for transaction costs. The profitability of our strategy aligns with concepts such as Delayed Information Diffusion and limited investor capacity for processing information, as discussed in [40]. Delayed Information Diffusion suggests that news information is not immediately incorporated into stock prices, but rather with a delay that can be exploited by investors. This delay can vary from stock to stock depending on factors such as their market capitalization. Moreover, investors do not have unlimited capacity for processing information. When faced with vast amounts of data, they may struggle to process all of it correctly and quickly, leading to potential underreaction. Investors utilizing LLMs can exploit this limited human capacity by capitalizing on the initial underreaction to new information.

Profitability of our Trading Strategy: Our flagship strategy, termed the Pure Alpha strategy, demonstrated compelling results over both 2022 and 2023, consistently outperforming its benchmark, the Russell 2000 index. Notably, even after accounting for a substantial round-trip transaction cost of 300 basis points, the strategy maintained its profitability. The strategy achieved annualized Sharpe ratios of 2.17 and 3.7 in 2022 and 2023, respectively, for a holding period of one day. Our approach to evaluating the strategy's performance was more comprehensive than those found in previous related studies [40, 11, 34]. While these earlier works primarily focused on Sharpe ratios and cumulative returns, we conducted an extensive analysis of additional performance metrics. These included maximum drawdown, volatilities, Sortino ratio, percentage of winning trades, and portfolio turnover. This multi-faceted evaluation allowed us to better assess not only the strategy's return potential but also its risk characteristics and practical implementation challenges.

Cross-Sectional Analysis - Sector and Industry Performance: Our study included a comprehensive analysis of the strategy's performance across various sectors and industries, revealing significant variations and interesting patterns. This cross-sectional approach provided valuable insights into the strategy's behavior in different market segments. Among the 11 sectors analyzed, Healthcare demonstrated consistently high performance, with Sharpe ratios of 3.43 and 3.41 in 2022 and 2023, respectively. This strong performance may be attributed to the prevalence of positive news surrounding healthcare-related companies during and after the COVID-19 pandemic, as well as increased focus on the healthcare sector during this period. Conversely, the Real Estate sector presented a more volatile performance profile. In 2022, it reported a negative Sharpe ratio, likely due to the challenging market conditions in the post-pandemic and ongoing pandemic era. However, the sector showed a recovery in 2023, achieving a Sharpe ratio of 3.72 and ranking as the third-best-performing sector for that year. At the industry level, our analysis revealed that Biotechnology exhibited consistent profitability over both years. This aligns with the strong performance observed in the broader Healthcare sector, further supporting

the rationale behind the sector's success.

5.2 Limitations and Future Work

Our research was constrained by limited access to news data, as discussed in the Methodology chapter. We only had access to news titles and snippets, rather than full articles, due to the time-intensive nature of parsing complete news content. This limitation had several important implications for our study. Without access to the full articles, we **lacked precise publication timestamps**, making it impossible to determine whether news was released during trading hours, pre-market, or post-market. As a result of this timing uncertainty, we had to execute trades using the next available opening price whenever a buy or sell signal was generated. This approach was consistently applied to both opening and closing positions, introducing a **significant delay between signal generation and trade execution**, potentially impacting the strategy's effectiveness.

To overcome these limitations in future research, it would be essential to develop a system capable of parsing the full news content and accurately capturing the exact time of publication. This enhancement would enable us to **react more swiftly** to market-moving news, potentially improving the precision and profitability of the trading strategy. We plan to explore this in further studies, aiming to analyze the impact of incorporating full news articles and their exact publication times on trading decisions, with the goal of refining the strategy accordingly.

Additionally, the absence of the full news body could affect sentiment analysis and, consequently, trading decisions. By excluding the news body and relying solely on the title and snippet, approximately 90% of the news was labelled irrelevant by the language model. This percentage could decrease if the full news body were included. **Incorporating the full news content might provide more information and lead to more accurate signals**. However, it could also **introduce noise** and be potentially misleading. Therefore, further studies are necessary to investigate this, which we plan to conduct as part of our future research.

In our study, we encountered a significant challenge: multiple news articles often covered the same event but were published by different journalists and websites, resulting in variations in titles and content. This **news article repetition** had the effect of amplifying the perceived impact of certain events due to their widespread coverage across different sources. This phenomenon presents a double-edged sword in news analysis. On the one hand, widespread reporting can indeed indicate the significance of an event and suggest that it merits greater attention in our analysis. On the other hand, this repetition may have led to an overrepresentation of certain events in our dataset, potentially skewing our results.

Although we attempted to exclude repeated news by checking the number of exact matching words in article titles and snippets for each unique stock and date pair, we were unable to implement a more comprehensive method to address these duplicates due to time constraints in the current study. This limitation is acknowledged as a potential source of bias in our findings. However, we have identified this as a critical area for improvement in future research. Our proposed solution involves leveraging advanced natural language processing techniques to refine our dataset and analysis.

For future studies, we plan to implement a sophisticated approach using **embedding techniques to capture the semantic meaning of news articles, allowing us to exclude repeated news**. This method will involve generating vector representations of each article's content using state-of-the-art language models. We will then apply similarity measures, such as cosine similarity, to these embedding vectors to identify and remove highly similar or duplicate articles. By setting an appropriate similarity threshold, we can

ensure that only one version of each unique event is included in our dataset.

To quantify the sentiments of news, we employed a novel approach that includes all news relevant to the triggered stock up to one day prior to the triggering event. We applied a decay method, as formulated in 3.3.2 and 3.3.3, to account for the **lasting impact of news** with higher intensity and longer-term effects. This approach could be further enhanced by incorporating a more sophisticated technique that assigns greater weight to news published by more reliable sources. This **source-weighted sentiment scoring** is currently an ongoing research project at *ZanistaAI* , and we plan to implement it to evaluate its impact on the performance of the trading strategy, with the intention of publishing our findings in future work.

As mentioned before, we used the next available opening price for both opening and closing our positions. These **open prices are not most liquid**, especially for small-cap stocks. This approach introduces potential pricing inefficiencies and may not accurately reflect the immediate market reaction to the news, particularly for less frequently traded stocks. Having **access to the exact time of news publication** would enable us to use more liquid prices, such as the closing price if the news is released within trading hours. This would provide a more accurate representation of the market's response to new information and potentially improve the performance of our trading strategies. For instance, if we knew a piece of news was released during trading hours, we could execute trades at the same-day closing price, which typically offers better liquidity and more closely reflects the market's digestion of the news. This improvement in trade execution timing could significantly enhance the accuracy of our backtesting results and provide a more realistic simulation of real-world trading conditions.

In our backtesting, we utilized data from 2022 and 2023 for the `GPT-4o` model, which was trained on data from the same years. This approach **potentially introduces a future look-ahead bias**, which can negatively impact the accuracy of the backtesting results. This bias arises from data leakage, where the model inadvertently uses future information during training, leading to misleading backtesting outcomes. **To mitigate this issue, we plan to validate our results using 2024 data (out-of-sample data), which is unseen by** `GPT-4o`. However, due to time constraints, this validation was not conducted for this study. Additionally, one could utilize **point-in-time LLMs**, such as TiMaGPT (TimeMachineGPT). These models are specifically designed to avoid incorporating future information by being trained exclusively on data that follows a chronological timeline. Ensuring that the data is labelled with precise dates is crucial to maintaining temporal integrity. Date labels help preserve the sequence of events, ensuring the model only accesses information available at that specific time, thereby preventing future data from influencing the training process. This method would greatly enhance the reliability of our backtesting and provide a more accurate assessment of the model's performance in real-time market conditions.

Another approach to address this issue could involve anonymizing news titles and snippets, as discussed in [23]. The idea behind anonymization is to remove specific identifiers, such as company names or product names (e.g., replacing "Apple" with "Company A" or "iPhone" with "Product X"), to prevent the model from leveraging prior knowledge about the company that could introduce bias into the predictions. By anonymizing the text, the model is forced to rely solely on the sentiment or context of the news without being influenced by preconceived notions or historical data related to the company. However, it is important to note that Glasserman and Lin conducted a study[23] that revealed an intriguing outcome: after applying anonymization, the reported returns were actually higher. This finding suggests that the negative effects of distraction—where the model's predictions are influenced by its existing knowledge about the company—can outweigh the positive bias introduced by look-ahead bias when the news is not anonymized.

This observation underscores a critical point: while look-ahead bias tends to introduce a positive bias by allowing the model to "peek" into future information, the distraction effect caused by the model's familiarity with certain companies or products can introduce significant inaccuracies in predictions. These inaccuracies may stem from the model overestimating or underestimating the impact of news based on its prior knowledge. Thus, in some cases, anonymizing the data can lead to more accurate backtesting results, as it eliminates the distraction effect and forces the model to make predictions based solely on the sentiment expressed in the news rather than any external knowledge about the entities involved[23]. This finding highlights the complexity of dealing with biases in LLM-based financial models and the need for careful consideration of how information is presented to these models during training and testing.

There is some scepticism regarding the use of LLMs for real-time trading, especially with models of higher complexity and larger sizes. **The process of obtaining sentiment analysis in real-time might be too slow** to facilitate high-speed trading activities. Although recent models with smaller sizes, such as `GPT-4o mini`, yet comparable efficiency to many larger models have already decreased processing time, there is still a need for faster models with sufficiently good performance.

In terms of LLM selection for sentiment analysis, one approach to enhance sentiment predictions is to **fine-tune the model with extensive labelled news data**. Although this might initially seem like an effective strategy, some research suggests that general-purpose LLMs could actually outperform models fine-tuned with domain-specific knowledge [77, 20]. Another promising approach is to use **instruction-tuned models**, which adapt general-purpose LLMs with specific instructions and can perform well without the need for fine-tuning [83]. This opens a potential area for future research, where we plan to assess the impact of financially fine-tuned LLMs or instruction-tuned models on the performance of our strategy. This will be explored in further studies.

Due to time constraints, our study focused exclusively on small-cap tickers. To enhance the scope and generalizability of our findings, we plan to extend this strategy to **all stocks in the Russell 3000 Index** in future research. This broader analysis will allow us to examine the **effects of market capitalization and liquidity on our results**. By including a wider range of stocks, from small to large-cap companies, we can assess how our strategy performs across different market segments. This comprehensive approach will provide valuable insights into whether the effectiveness of our news-based trading strategy varies with company size and stock liquidity, potentially revealing new opportunities or challenges in applying our strategy to a more diverse set of stocks.

The expansion to include large-cap stocks presents both opportunities and challenges. **Large-cap** stocks offer a wealth of news and textual data, making them ideal candidates for LLMs' advanced analytical capabilities. These models excel at processing and interpreting vast amounts of information, potentially uncovering insights that might elude human analysts. However, the efficiency of large-cap markets means news is **rapidly incorporated into prices**, necessitating extremely fast execution of trading strategies to capitalize on fleeting opportunities. Moreover, the macroeconomic indicators in our strategy demonstrated higher historical betas with larger-cap stocks, potentially reducing the exclusivity of this approach. Conversely, **small-cap stocks**, while generating less news and data, often exhibit a more **pronounced delay between news release and price adjustments**. This lag creates a wider window for traders to exploit market inefficiencies. Additionally, the macroeconomic indicators used in our strategy showed better exclusivity and niche relevance for smaller-cap stocks, potentially offering a competitive edge in this market segment.

To address these challenges and capitalize on the unique characteristics of both small and large-cap stocks, a promising approach is the **LLM+Human model**, as explored

in [8]. This hybrid strategy **leverages the strengths of both artificial and human intelligence**. For small-cap stocks, where specialized industry knowledge and nuanced understanding are crucial, human analysts can provide invaluable insights. Their expertise can help interpret the limited news and data available, providing context that might be missed by LLMs alone. Meanwhile, LLMs can offer a significant advantage in analyzing the vast data landscapes of large-cap stocks, processing information at a scale and speed impossible for human analysts [8]. This combination of human expertise and machine efficiency could potentially optimize our strategy across the entire spectrum of market capitalization, allowing for more robust and adaptable trading approaches.

Conclusion

This thesis has demonstrated the efficacy of integrating Large Language Models, specifically GPT-4o, with macroeconomic and technical indicators to develop a profitable trading strategy for small-cap stocks. Our flagship "Pure Alpha" strategy, which focuses on stocks with significant moves independent of their co-moved indicators, consistently outperformed the Russell 2000 benchmark. The strategy demonstrated robust performance, with Sharpe ratios of 3.64 and 5.10, and Sortino ratios of 6.56 and 8.92 in 2022 and 2023, respectively. Maximum drawdowns decreased from 4.69% to 3.18%. These metrics significantly outperformed the Russell 2000 benchmark, which had Sharpe ratios of -0.77 and 0.84 in the same periods, highlighting our strategy's superior risk-adjusted returns.

The strategy's success across various sectors, particularly in "Healthcare" and "Financials", highlights the potential of our approach to capture alpha-generating opportunities driven by stock-specific factors rather than broader market movements. Our innovative approach to news sentiment analysis, which incorporates the lasting effects of all news prior to the event date through a decay function modelling the diminishing impact of news over time, significantly contributes to the growing field of financial text analysis using LLMs. However, we acknowledge limitations, such as potential look-ahead bias due to the LLM's training data and the exclusion of full news articles in our analysis. Future research should address these limitations by potentially incorporating point-in-time LLMs, employing anonymization techniques, and exploring the impact of including complete news content.

This work paves the way for further research, including expanding the strategy to encompass a wider range of stocks with varying liquidity levels, developing a source-weighted sentiment system that prioritizes news from more reliable sources, and leveraging precise publication dates to enable faster and more responsive trading decisions.

Appendix A

List of All Indicators

Table A.1: Macroeconomic Indicators, Tickers, and Their Asset Classes

Asset Class	**Name**	**Ticker/Series ID**
Bond	US Short-Term Treasury (1-3 Year)	SHY
Bond	US 10-Year Treasury	IEF
Bond	UK Short-Term Gilt (1-3 Year)	ISHG
Bond	UK 10-Year Gilt	GLTL.L
Bond	Euro Short-Term Government Bond (1-3 Year)	IBGS.AS
Bond	Euro 10-Year Government Bond	IEGA.AS
Bond	Japan Short-Term Government Bond (1-3 Year)	JT13.MI
Bond	Japan 10-Year Government Bond	JPXN
Commodity	Brent Oil	BNO
Commodity	WTI Oil	USO
Commodity	TTF Gas	UNG
Commodity	Corn	CORN
Commodity	Wheat	WEAT
Commodity	Soybeans	SOYB
Commodity	Coffee	JO
Commodity	Cotton	BAL
Commodity	Sugar	SGG
Commodity	Gold	GLD
Commodity	Silver	SLV
Commodity	Platinum	PPLT
Commodity	Palladium	PALL
Commodity	Crude Oil	USO
Commodity	Natural Gas	UNG
Commodity	Gasoline	UGA
Commodity	Copper	CPER
Commodity	BCOM Index	CMDY
Commodity	Live Cattle Futures	LE=F
Cryptocurrency	Bitcoin	BTC-USD
Cryptocurrency	Ethereum	ETH-USD
Equity	Latin America	ILF
Equity Index	S&P500	ĜSPC

Continued on next page

Asset Class	Name	Ticker/Series ID
Equity Index	Russell 1000	^RUI
Equity Index	Russell 2000	^RUT
Equity Index	FTSE 100	^FTSE
Equity Index	CAC40 France	^FCHI
Equity Index	DAX Germany	^GDAXI
Equity Index	Nikkei225 Japan	^N225
Equity Index	S&PTSX Canada	^GSPTSE
Equity Index	Euro Stoxx 50 (Eurozone)	^STOXX50E
Equity Index	Stoxx Europe 600	^STOXX
Equity Index	China (Shanghai Composite)	000001.SS
Equity Index	Australia (ASX 200)	^AXJO
Equity Index	European Union (EU)	^STOXX
Equity Index	Istanbul Bursa (BIST 100)	XU100.IS
Equity Index	Hang Seng Index (Hong Kong)	^HSI
Equity Index	KOSPI (South Korea)	^KS11
Equity Index	Sensex (India)	^BSESN
Equity Index	Tadawul All Share Index (Saudi Arabia)	^TASI.SR
Equity Index	BOVESPA (Brazil)	^BVSP
Equity Index	FTSE MIB Italy	^ITLMS.MI
Equity Index	Nifty 50	^NSEI
FX	GBP/USD	GBPUSD=X
FX	AUD/USD	AUDUSD=X
FX	CAD/USD	CADUSD=X
FX	EUR/USD	EURUSD=X
FX	JPY/USD	JPYUSD=X
FX	NZD/USD	NZDUSD=X
FX	NOK/USD	NOKUSD=X
FX	SEK/USD	SEKUSD=X
FX	CHF/USD	CHFUSD=X
FX	BRL/USD	BRLUSD=X
FX	RUB/USD	RUBUSD=X
FX	INR/USD	INRUSD=X
FX	CNY/USD	CNYUSD=X
FX	ZAR/USD	ZARUSD=X
FX	MXN/USD	MXNUSD=X
FX	IDR/USD	IDRUSD=X
FX	TRY/USD	TRYUSD=X
FX	KRW/USD	KRWUSD=X
FX	PLN/USD	PLNUSD=X
FX	Dollar Index (DXY)	DX-Y.NYB
Volatility Index	VIX	^VIX
Volatility Index	Russell 2000 VIX	^RVX
Macro - FRED Indicators		
Asset Class	**Name**	**Ticker/Series ID**
Economic Indicator	GDP	GDP
Economic Indicator	Inflation	CPIAUCSL

Continued on next page

Asset Class	Name	Ticker/Series ID
Economic Indicator	Unemployment	UNRATE
Economic Indicator	Capacity Utilisation	TCU
Economic Indicator	Consumer Confidence	UMCSENT
Economic Indicator	Housing Starts	HOUST
Economic Indicator	Building Permits	PERMIT
Economic Indicator	Federal Funds Rate	FEDFUNDS
Economic Indicator	10-Year Treasury Yield	DGS10

Table A.2: Sector/Industry and Their Corresponding ETFs

Sector/Industry	ETF Ticker
Industrials	XLI
Building Products & Equipment	PKB
Consumer Cyclical	XLY
Auto & Truck Dealerships	TSLL
Financial Services	XLF
Insurance - Life	KIE
Healthcare	XLV
Apparel Retail	XRT
Basic Materials	XLB
Industrial Distribution	XLI
Technology	XLK
Drug Manufacturers - Specialty & Generic	IHE
Utilities	XLU
Specialty Chemicals	XLB
Consumer Defensive	XLP
Software - Infrastructure	VGT
Energy	XLE
Coking Coal	XME
Real Estate	VNQ
Engineering & Construction	PKB
Communication Services	XLC
Software - Application	IGV
Information Technology Services	VGT
Specialty Retail	XRT
Metal Fabrication	XME
Electrical Equipment & Parts	GRID
Building Materials	PAVE
Utilities - Regulated Gas	XLU
Scientific & Technical Instruments	IYW
Biotechnology	IBB
Packaged Foods	XLP
Banks - Regional	KRE
Oil & Gas E&P	XOP
Oil & Gas Equipment & Services	VDE
Steel	SLX
Mortgage Finance	IYG

Continued on next page

Sector/Industry	**ETF Ticker**
Waste Management	EVX
Household & Personal Products	XLP
Medical Care Facilities	XLV
Insurance - Specialty	EUFN
Oil & Gas Midstream	AMLP
Credit Services	XLF
Specialty Industrial Machinery	XLI
Electronic Components	SOXX
Pollution & Treatment Controls	PHO
Rental & Leasing Services	IYR
Medical Devices	IHI
Asset Management	IYG
Health Information Services	XLV
Residential Construction	ITB
REIT - Retail	XLRE
Gambling	BETZ
Capital Markets	IYG
Marine Shipping	BOAT
Specialty Business Services	IYJ
Medical Instruments & Supplies	IHI
Aerospace & Defense	ITA
Semiconductors	SOXX
Oil & Gas Drilling	XES
Electronics & Computer Distribution	XLK
Solar	TAN
Semiconductor Equipment & Materials	SOXX
Oil & Gas Refining & Marketing	XLE
Utilities - Regulated Electric	XLU
REIT - Hotel & Motel	XLRE
Grocery Stores	XLP
Luxury Goods	XLY
Insurance - Property & Casualty	KIE
Computer Hardware	XLK
Lumber & Wood Production	WOOD
REIT - Industrial	XLRE

Appendix B

Strategy Results - Figures

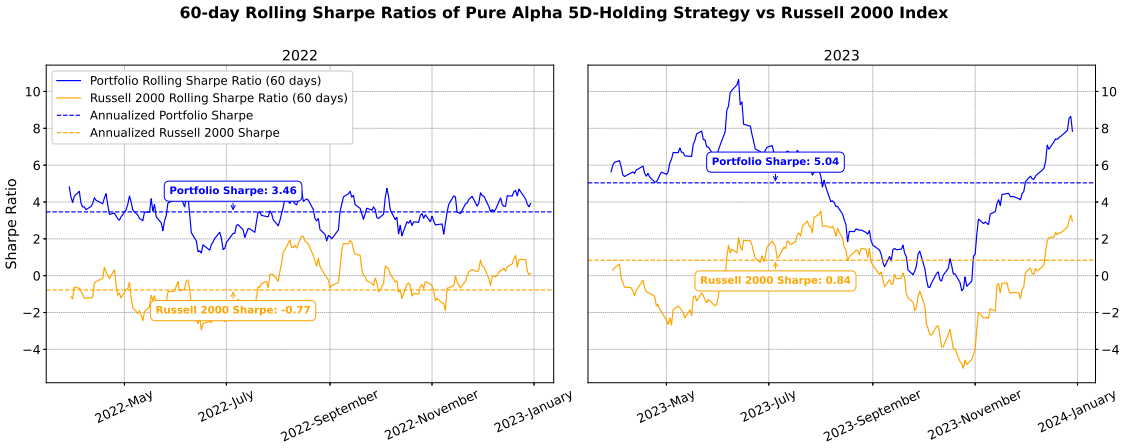

Figure B.1: 60-day Rolling Sharpe Ratios of the Pure Alpha 5-Day Holding Strategy vs. Russell 2000 Index (RL2K) in 2022 and 2023.

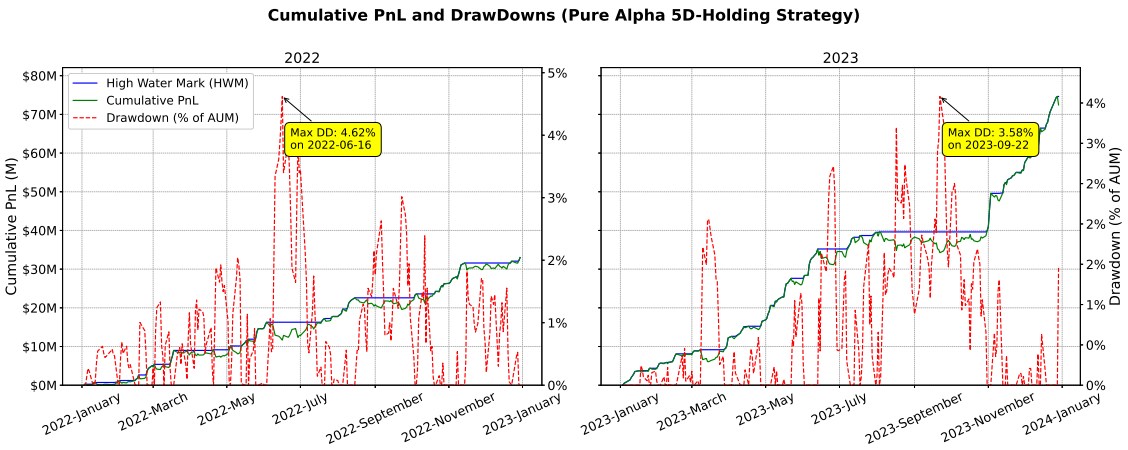

Figure B.2: Cumulative PnL and Drawdowns for the Pure Alpha 5-Day Holding Strategy in 2022 and 2023. The figure illustrates the cumulative profit and loss (PnL) on the left y-axis alongside drawdown percentages on the right y-axis. AUM was $100 million in 2022 and $150 million in 2023.

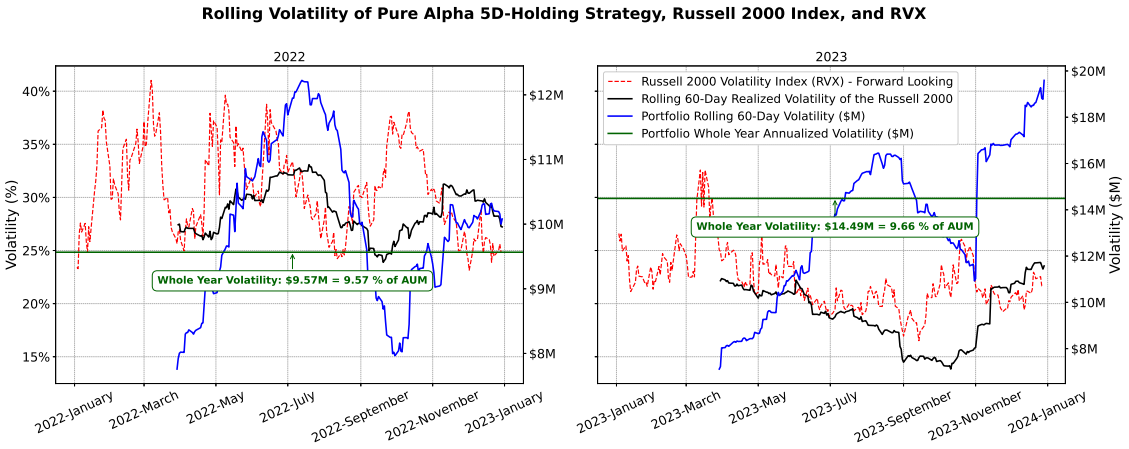

Figure B.3: Rolling Volatility of the Pure Alpha 5-Day Holding Strategy, Russell 2000 Index, and RVX in 2022 and 2023. The right y-axis represents the volatility of the Pure Alpha strategy in millions of dollars (M$), while the left y-axis shows the volatility of the RVX and Russell 2000 Index in percentage terms. AUM was $100 million in 2022 and $150 million in 2023.

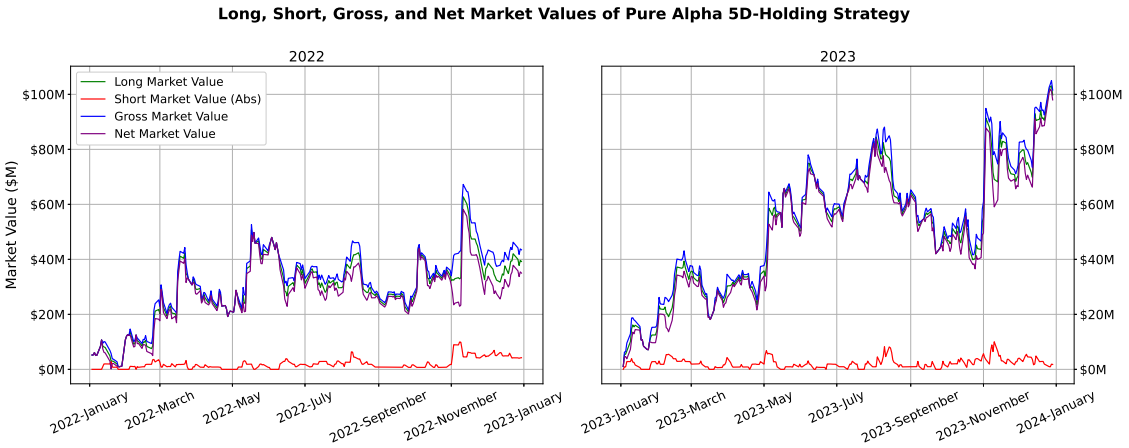

Figure B.4: Long, Short, Gross, and Net Market Values of the Pure Alpha 5-Day Holding Strategy in 2022 and 2023, with all values expressed in millions of dollars (M$). AUM was $100 million in 2022 and $150 million in 2023.

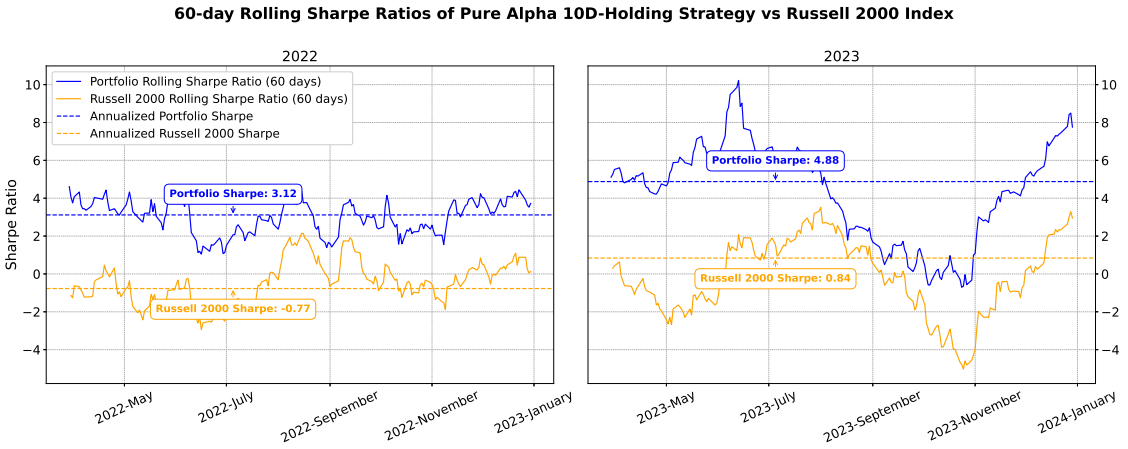

Figure B.5: 60-day Rolling Sharpe Ratios of the Pure Alpha 10-Day Holding Strategy vs. Russell 2000 Index (RL2K) in 2022 and 2023.

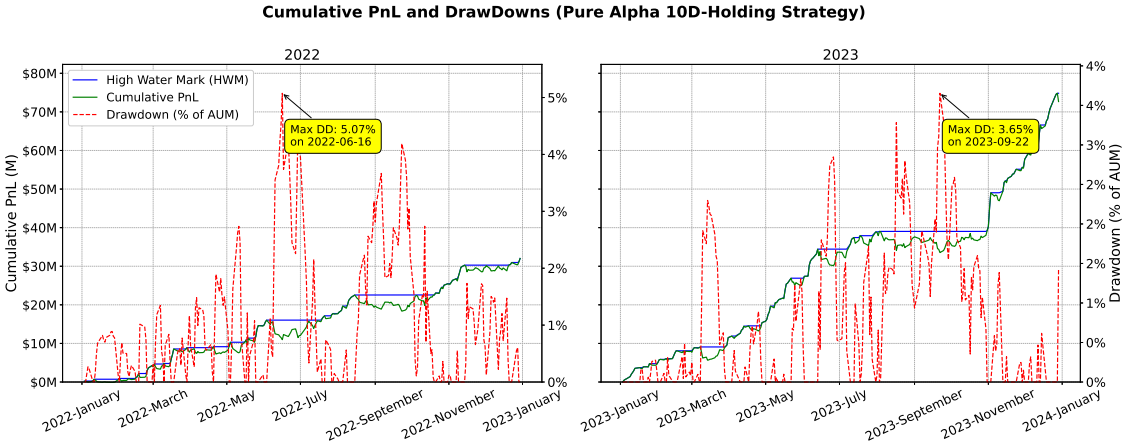

Figure B.6: Cumulative PnL and Drawdowns for the Pure Alpha 10-Day Holding Strategy in 2022 and 2023. The figure illustrates the cumulative profit and loss (PnL) on the left y-axis alongside drawdown percentages on the right y-axis. AUM was $100 million in 2022 and $150 million in 2023.

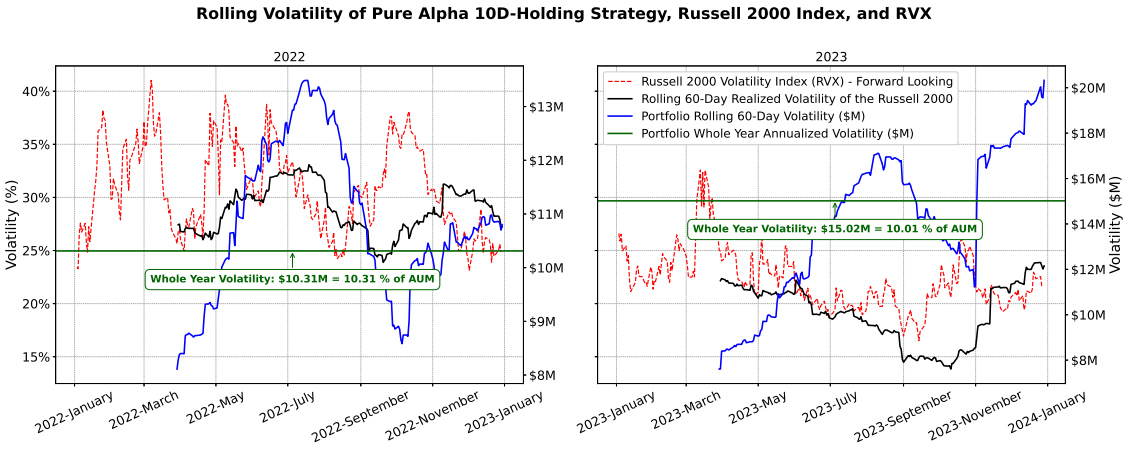

Figure B.7: Rolling Volatility of the Pure Alpha 10-Day Holding Strategy, Russell 2000 Index, and RVX in 2022 and 2023. The right y-axis represents the volatility of the Pure Alpha strategy in millions of dollars (M$), while the left y-axis shows the volatility of the RVX and Russell 2000 Index in percentage terms. AUM was $100 million in 2022 and $150 million in 2023.

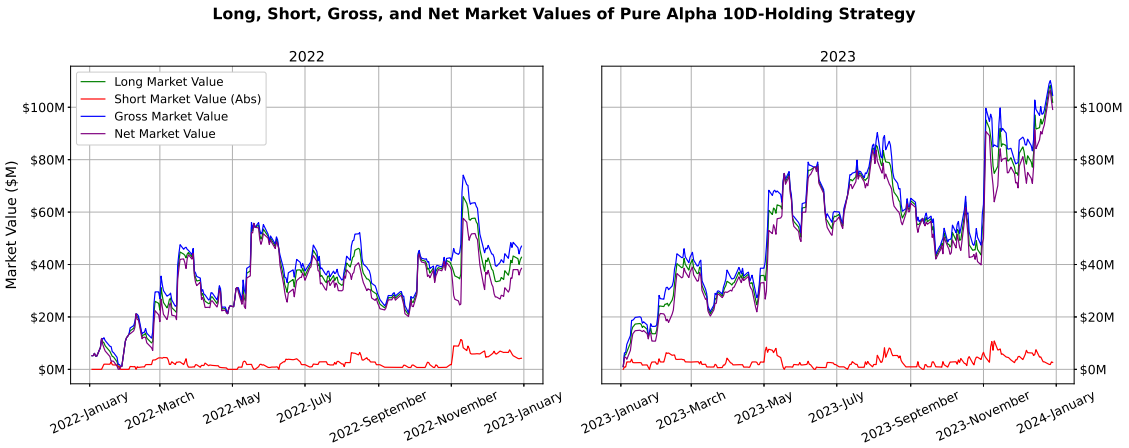

Figure B.8: Long, Short, Gross, and Net Market Values of the Pure Alpha 10-Day Holding Strategy in 2022 and 2023, with all values expressed in millions of dollars (M$). AUM was $100 million in 2022 and $150 million in 2023.

Bibliography

[1] S. B. Achelis, *Technical Analysis from A to Z*, McGraw Hill Professional, New York, 2nd ed., 2001.

[2] M. AI, *Introducing meta llama 3: The most capable openly available llm to date*, 2024. Accessed: 2024-07-16.

[3] G. Appel and F. Hitschler, *Technical Analysis: Power Tools for Active Investors*, Financial Times Prentice Hall, Upper Saddle River, NJ, 2005.

[4] D. Araci, *Finbert: Financial sentiment analysis with pre-trained language models*, 2019.

[5] D. Blei, A. Ng, and M. Jordan, *Latent dirichlet allocation*, vol. 3, 01 2001, pp. 601–608.

[6] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, and et al., *Language models are few-shot learners*, 2020.

[7] B. Bruder and T. Roncalli, *Managing risk exposures using the risk budgeting approach*, tech. rep., Lyxor Asset Management and Amundi Asset Management, January 20 2012. 33 Pages Posted: 23 Feb 2012 Last revised: 10 Apr 2012.

[8] S. Cao, W. Jiang, J. Wang, and B. Yang, *From man vs. machine to man + machine: The art and ai of stock analyses*, Journal of Financial Economics, 160 (2024), p. 103910.

[9] B. Chen, Z. Zhang, N. Langrené, and S. Zhu, *Unleashing the potential of prompt engineering in large language models: a comprehensive review*, 2024.

[10] S. Chen and F. Xing, *Understanding emojis for financial sentiment analysis*, 12 2023.

[11] Y. Chen, B. T. Kelly, and D. Xiu, *Expected returns and large language models*, (2022). Available at SSRN: `https://ssrn.com/abstract=4416687`.

[12] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, *Electra: Pre-training text encoders as discriminators rather than generators*, 2020.

[13] R. Cont, *Empirical properties of asset returns: stylized facts and statistical issues*, Quantitative Finance, 1 (2001), pp. 223–236.

[14] J. Danielsson, L. de Haan, L. Peng, and C. de Vries, *Using a bootstrap method to choose the sample fraction in tail index estimation*, Journal of Multivariate Analysis, 76 (2001), pp. 226–248.

[15] S. Das and M. Chen, *Yahoo! for amazon: Sentiment extraction from small talk on the web*, Management Science, 53 (2007), pp. 1375–1388.

[16] X. Deng, V. Bashlovkina, F. Han, S. Baumgartner, and M. Bendersky, *What do llms know about financial markets? a case study on reddit market sentiment analysis*, 2022.

[17] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, *Qlora: Efficient finetuning of quantized llms*, 2023.

[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, 2019.

[19] F. Fabozzi, H. Markowitz, and F. Gupta, *Portfolio Selection*, 09 2008.

[20] S. Fatemi and Y. Hu, *A comparative analysis of fine-tuned llms and few-shot learning of llms for financial sentiment analysis*, 2023.

[21] G. Frankfurter, *John h. cochrane, asset pricing (revised edition)*, Journal of Economic Behavior Organization - J ECON BEHAV ORGAN, 60 (2006), pp. 603–608.

[22] P. Glasserman, F. Li, and H. Mamaysky, *Time variation in the news-returns relationship*, SSRN Electronic Journal, (2019).

[23] P. Glasserman and C. Lin, *Assessing look-ahead bias in stock return predictions generated by gpt sentiment analysis*, 2023.

[24] L. Gonon, *Quantitative risk management lecture slides*, 2023-2024. Lecture slides.

[25] D. Gromb and D. Vayanos, *Limits of arbitrage*, Annual Review of Financial Economics, 2 (2010), pp. 251–275.

[26] Z. A. Güven, B. Diri, and T. Çakaloğlu, *Comparison of topic modeling methods for type detection of turkish news*, in 2019 4th International Conference on Computer Science and Engineering (UBMK), IEEE, Sept. 2019.

[27] S. Han, J. Pool, J. Tran, and W. J. Dally, *Learning both weights and connections for efficient neural networks*, 2015.

[28] A. Hansen and S. Kazinnik, *Can chatgpt decipher fedspeak?*, SSRN Electronic Journal, (2023).

[29] G. Hinton, O. Vinyals, and J. Dean, *Distilling the knowledge in a neural network*, 2015.

[30] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, *Lora: Low-rank adaptation of large language models*, 2021.

[31] M.-H. Hwang, J. Shin, H. Seo, J.-S. Im, H. Cho, and C.-K. Lee, *Ensemble-nqg-t5: Ensemble neural question generation model based on text-to-text transfer transformer*, Applied Sciences, 13 (2023), pp. 1–12.

[32] C. Jeong, *Domain-specialized llm: Financial fine-tuning and utilization method using mistral 7b*, Journal of Intelligence and Information Systems, 30 (2024), p. 93–120.

[33] K. S. Kalyan, *A survey of gpt-3 family large language models including chatgpt and gpt-4*, Natural Language Processing Journal, 6 (2024), p. 100048.

[34] K. Kirtac and G. Germano, *Sentiment trading with large language models*, Finance Research Letters, 62 (2024), p. 105227.

[35] Y. Kroll, H. Levy, and A. Rapoport, *Experimental tests of the mean-variance model for portfolio selection*, Organizational Behavior and Human Decision Processes, 42 (1988), pp. 388–410.

[36] H. Kunze, D. La Torre, A. Riccoboni, and M. R. Galán, *Engineering Mathematics and Artificial Intelligence: Foundations, Methods, and Applications*, CRC Press, 1st ed., 2023.

[37] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, and D. Kiela, *Retrieval-augmented generation for knowledge-intensive nlp tasks*, 2021.

[38] Z. Liu, D. Huang, H. Kaiyu, Z. Li, and J. Zhao, *Finbert: A pre-trained financial language representation model for financial text mining*, 07 2020, pp. 4463–4469.

[39] A. Lo, *The statistics of sharpe ratios*, Financial Analysts Journal, 58 (2003).

[40] A. Lopez-Lira and Y. Tang, *Can chatgpt forecast stock price movements? return predictability and large language models*, 2023.

[41] D. Lu, H. Wu, J. Liang, Y. Xu, Q. He, Y. Geng, M. Han, Y. Xin, and Y. Xiao, *Bbt-fin: Comprehensive construction of chinese financial domain pre-trained language model, corpus and benchmark*, 2023.

[42] S. Ma, H. Wang, L. Ma, L. Wang, W. Wang, S. Huang, L. Dong, R. Wang, J. Xue, and F. Wei, *The era of 1-bit llms: All large language models are in 1.58 bits*, 2024.

[43] S. Maillard, T. Roncalli, and J. Teiletche, *The properties of equally weighted risk contribution portfolios*, The Journal of Portfolio Management, 36 (2010), pp. 60–70.

[44] H. Mamaysky and C. Calomiris, *How news and its context drive risk and returns around the world*, Journal of Financial Economics, 133 (2018).

[45] R. C. Merton, *Optimum consumption and portfolio rules in a continuous-time model*, Journal of Economic Theory, 3 (1971), pp. 373–413.

[46] Meta AI, *Llama 3.1: The open source ai model*. `https://ai.meta.com`, 2024. Accessed: 2024-08-01.

[47] T. Mikolov, K. Chen, G. Corrado, and J. Dean, *Efficient estimation of word representations in vector space*, 2013.

[48] J. Muhle-Karbe, *Portfolio management, lecture notes*. Unpublished lecture notes, 2023-2024. Imperial College London.

[49] Y. Nie, Y. Kong, X. Dong, J. M. Mulvey, H. V. Poor, Q. Wen, and S. Zohren, *A survey of large language models for financial applications: Progress, prospects and challenges*, 2024.

[50] L. Németh and A. Zempléni, *Regression estimator for the tail index*, 2018.

[51] OpenAI, *Gpt-4o: A multi-modal ai model integrating text, vision, and audio*, 2024. Accessed: 2024-05-13.

[52] OpenAI, *Gpt-4o mini: Advancing cost-efficient intelligence*, 2024. Accessed: 2024-07-18.

[53] OpenAI, J. Achiam, S. Adler, S. Agarwal, and et al., *Gpt-4 technical report*, 2024.

[54] R. Paap, *Contemporary bayesian econometrics and statistics, john geweke wiley, new jersey (2005), (hardcover, 300 pages) isbn: 0-471-67932-1*, International Journal of Forecasting, 23 (2007), pp. 529–531.

[55] L. H. PEDERSEN, *Efficiently Inefficient: How Smart Money Invests and Market Prices Are Determined*, Princeton University Press, 2015.

[56] J. W. Pratt, *Risk aversion in the small and in the large*, Econometrica, 32 (1964), pp. 122–136.

[57] A. Radford and K. Narasimhan, *Improving language understanding by generative pre-training*, 2018.

[58] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, *Language models are unsupervised multitask learners*, 2019.

[59] T. Renault, *Sentiment analysis and machine learning in finance: a comparison of methods and models on one million messages*, Digital Finance, 2 (2020).

[60] T. Roncalli, *Introduction to Risk Parity and Budgeting*, Chapman and Hall/CRC, 2013.

[61] T. Roncalli and G. Weisang, *Risk parity portfolios with risk factors*, SSRN Electronic Journal, 16 (2012).

[62] P. Sahoo, A. K. Singh, S. Saha, V. Jain, S. Mondal, and A. Chadha, *A systematic survey of prompt engineering in large language models: Techniques and applications*, 2024.

[63] T. L. Scao, A. Fan, C. Akiki, E. Pavlick, and et al., *Bloom: A 176b-parameter open-access multilingual language model*, ArXiv, abs/2211.05100 (2022).

[64] R. S. Shah, K. Chawla, D. Eidnani, A. Shah, W. Du, S. Chava, N. Raman, C. Smiley, J. Chen, and D. Yang, *When flue meets flang: Benchmarks and large pre-trained language model for financial domain*, 2022.

[65] S. Sohangir, N. Petty, and D. Wang, *Financial sentiment lexicon analysis*, 01 2018, pp. 286–289.

[66] P. TETLOCK, *Giving content to investor sentiment: The role of media in the stock market*, Journal of Finance, 62 (2007), pp. 1139–1168.

[67] P. C. TETLOCK, M. SAAR-TSECHANSKY, and S. MACSKASSY, *More than words: Quantifying language to measure firms' fundamentals*, The Journal of Finance, 63 (2008), pp. 1437–1467.

[68] P. B. W. Todt and R. Babaei, *Fin-llama: Efficient finetuning of quantized llms for finance*. `https://github.com/Bavest/fin-llama`, 2023. Accessed: 2023.

[69] H. Tong, J. Li, N. Wu, M. Gong, D. Zhang, and Q. Zhang, *Ploutos: Towards interpretable stock movement prediction with financial large language model*, 2024.

[70] H. Touvron, T. Lavril, G. Izacard, X. Martinet, and et al., *Llama: Open and efficient foundation language models*, 2023.

[71] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, and et al., *Llama 2: Open foundation and fine-tuned chat models*, 2023.

[72] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, *Attention is all you need*, 2023.

[73] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le, *Finetuned language models are zero-shot learners*, 2022.

[74] M. West and J. Harrison, *Bayesian Forecasting and Dynamic Models (Springer Series in Statistics)*, Springer-Verlag, February 1997.

[75] S. Wu, O. Irsoy, S. Lu, V. Dabravolski, M. Dredze, S. Gehrmann, P. Kambadur, D. Rosenberg, and G. Mann, *Bloomberggpt: A large language model for finance*, 2023.

[76] Q. Xie, W. Han, X. Zhang, Y. Lai, M. Peng, A. Lopez-Lira, and J. Huang, *Pixiu: A large language model, instruction data and evaluation benchmark for finance*, 2023.

[77] H. Yang, Y. Zhang, J. Xu, H. Lu, P. Heng, and W. Lam, *Unveiling the generalization power of fine-tuned large language models*, CoRR, abs/2403.09162 (2024).

[78] Y. Yang, Y. Tang, and K. Y. Tam, *Investlm: A large language model for investment using financial domain instruction tuning*, 2023.

[79] Y. Yang, M. C. S. UY, and A. Huang, *Finbert: A pretrained language model for financial communications*, 2020.

[80] M. Yekrangi and N. Abdolvand, *Financial markets sentiment analysis: Developing a specialized lexicon*, Journal of Intelligent Information Systems, (2021).

[81] Y. Yu, *Cornucopia-llama-fin-chinese*. `https://github.com/jerry1993-tech/Cornucopia-LLaMA-Fin-Chinese`, 2023. Accessed: 2023.

[82] B. Zhang, H. Yang, and X.-Y. Liu, *Instruct-fingpt: Financial sentiment analysis by instruction tuning of general-purpose large language models*, 2023.

[83] B. Zhang, H. Yang, T. Zhou, A. Babar, and X.-Y. Liu, *Enhancing financial sentiment analysis via retrieval augmented large language models*, 2023.

[84] S. Zhang, S. Roller, N. Goyal, M. Artetxe, and et al., *Opt: Open pre-trained transformer language models*, 2022.

[85] X. Zhang, Q. Yang, and D. Xu, *Xuanyuan 2.0: A large chinese financial chat model with hundreds of billions parameters*, 2023.

[86] Z. Zhang, H. Zhang, K. Chen, Y. Guo, J. Hua, Y. Wang, and M. Zhou, *Mengzi: Towards lightweight yet ingenious pre-trained models for chinese*, 2021.