ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

UNIVERSITY OF PIRAEUS

UNIVERSITY OF PIRAEUS
DEPARTMENT OF DIGITAL SYSTEMS
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGIES

Design, Optimization, Development and Explanation of Artificial Intelligence algorithms, capable of running on distributed Big Data systems, based on multidimensional and complex datasets.

Georgios D. Fatouros

Doctoral Thesis

Piraeus, 2024

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

UNIVERSITY OF PIRAEUS

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ

Σχεδιασμός, Βελτιστοποίηση, Ανάπτυξη και Επεξήγηση αλγορίθμων Μηχανικής και Βαθιάς Μάθησης, με δυνατότητα εκτέλεσης σε κατανεμημένα συστήματα Μεγάλων Δεδομένων, βασισμένα σε πολυδιάστατα και σύνθετα σύνολα δεδομένων

Γεώργιος Δ. Φατούρος

Διδακτορική Διατριβή

Πειραιάς, 2024

UNIVERSITY OF PIRAEUS

DEPARTMENT OF DIGITAL SYSTEMS

Design, Optimization, Development and Explanation of Artificial Intelligence algorithms, capable of running on distributed Big Data systems, based on multidimensional and complex datasets.

Doctoral Thesis Presented

by **Georgios D. Fatouros**

in Fulfillment of the Requirements

for the Degree of Doctor of Philosophy

ADVISORY COMMITTEE:

Professor Dimosthenis Kyriazis

Professor Michail Filippakis

Professor Christos Doulkeridis

UNIVERSITY OF PIRAEUS

DEPARTMENT OF DIGITAL SYSTEMS

Design, Optimization, Development and Explanation of Artificial Intelligence algorithms, capable of running on distributed Big Data systems, based on multidimensional and complex datasets.

Doctoral Thesis Presented

by **Georgios D. Fatouros**

in Fulfillment of the Requirements

for the Degree of Doctor of Philosophy

APPROVED BY:

Dimosthenis Kyriazis, Professor, University of Piraeus

Michail Filippakis, Professor, University of Piraeus

Christos Doulkeridis, Professor, University of Piraeus

Nikitas-Marinos Sgouros, Professor, University of Piraeus

Andriana Prentza, Professor, University of Piraeus

Georgios Kousiouris, Associate Professor, Harokopio University of Athens

---

Andreas Menychtas, Assistant Professor, University of Piraeus

Piraeus, 2024

To my sons Dimitris and Vasilis.

Acknowledgments

The completion of this work would not have been possible without the support of some people. Therefore, this chapter is dedicated to them.

First and foremost, I would like to express my deepest gratitude to my supervisor, Prof. Dimosthenis Kyriazis, whose guidance, support, and encouragement have been invaluable throughout the years of my doctoral journey. His mentorship has been instrumental in shaping both my research and academic growth. I am also sincerely thankful to my advisory board members, Prof. Michael Filippakis and Prof. Christos Doulkeridis, for their insightful feedback and support over the years. Their expertise and advice have greatly contributed to the development and completion of this thesis.

I would also like to extend my heartfelt thanks to Prof. John Soldatos and Prof. George Kousiouris. Their mentorship and guidance during our collaboration on various research projects over the past years have been incredibly enriching. Their knowledge and support have significantly shaped the direction of my research.

Special thanks go to George Makridis, who, as a more senior PhD student during my early years, provided valuable guidance and support. His advice and assistance were crucial in helping me navigate the challenges during the initial, and most difficult, phase of my PhD journey.

I am also deeply grateful to Kostas Metaxas, whose domain expertise in finance played a significant role in my research related to financial analysis. His insights and collaboration were crucial in advancing my understanding and application of AI in this field.

Finally, I want to thank my wife, Kelly, for her unwavering support throughout these years. Her patience and understanding, despite the many late nights and long hours dedicated to

my studies, have been a source of strength for me. I also wish to thank my sons, Dimitirs and Vasilis, who had to sacrifice time with their father as I pursued this academic endeavor. Their love and the joy they bring have been a constant motivation for me.

Abstract

Keywords: artificial intelligence, large language models, chatgpt, financial analysis, retrieval augmented generation, knowledge mining, timeseries prediction, online learning, cloud computing, machine learning, deep learning

The exponential growth of digital data and advancements in Artificial Intelligence (AI) have redefined data-driven decision-making across industries. However, managing complex, high-dimensional, and multi-frequency datasets remains challenging, particularly in finance and cloud computing sectors where data variability, real-time adaptability, and predictive precision are crucial. This thesis introduces data-centric AI methodologies designed to create resilient and adaptable systems capable of processing diverse data across multiple formats and domains.

Specifically, the dissertation presents a Knowledge and Reasoning Framework that utilizes ontologies and retrieval-augmented generation (RAG) techniques for integrating, managing, and processing structured and unstructured data. This enables the combined utilization of different types of data originating from different sources. Furthermore, techniques are developed to improve the real-time adaptability of time-series analysis models, allowing continuous system updates to capture evolving trends and maintain robust performance in dynamic environments. Techniques are also presented for the optimized use of Large Language Models (LLMs) for text analysis and classification, comparing their effectiveness with traditional transfer learning techniques for sentiment analysis in financial articles. Finally, an innovative system is proposed that combines and extends these methodologies, managing text, numeri-

cal data, and time-series data from various sources, providing a comprehensive solution with applications in the financial sector.

The dissertation validates the proposed methodologies through real-world applications such as time-series prediction for cloud computing resource management, financial risk assessment, sentiment analysis, and systematic stock selection in finance. These contributions demonstrate significant improvements in handling complex datasets, ensuring real-time adaptability, and delivering interpretable and reliable AI-driven insights. By addressing the common challenges of data complexity and heterogeneity in real-world applications through multiple data-centric methodologies, this thesis offers scalable and resilient AI solutions for data-rich environments, effectively managing diverse and complex data formats across various domains.

Περίληψη

Λέξεις κλειδιά: τεχνητή νοημοσύνη, μεγάλα γλωσσικά μοντέλα, chatgpt , χρηματοοικονομική ανάλυση, ανάκτηση βάσει περιεχομένου, εξόρυξη γνώσης, πρόβλεψη χρονοσειρών, υπολογιστικό νέφος, μηχανική μάθηση, βαθιά μάθηση

Η εκθετική αύξηση των ψηφιακών δεδομένων και οι εξελίξεις στην Τεχνητή Νοημοσύνη (AI) έχουν επαναπροσδιορίσει τη λήψη αποφάσεων που βασίζεται σε δεδομένα σε διάφορους κλάδους. Ωστόσο, η διαχείριση και εκμετάλλευση σύνθετων, υψηλής διάστασης και διαφορετικών τύπων συνόλων δεδομένων παραμένει πρόκληση, ιδιαίτερα στους τομείς της χρηματοοικονομικής και του cloud computing, όπου η ποικιλία των δεδομένων, η προσαρμοστικότητα σε πραγματικό χρόνο και η ακρίβεια πρόβλεψης είναι κρίσιμες. Η παρούσα διατριβή παρουσιάζει μεθοδολογίες AI με επίκεντρο τα δεδομένα, σχεδιασμένες να δημιουργούν ανθεκτικά και προσαρμοστικά συστήματα ικανά να επεξεργάζονται ποικίλα δεδομένα σε πολλαπλές μορφές και τομείς.

Συγκεκριμένα, η διατριβή παρουσιάζει ένα Πλαίσιο Συλλογισμού και Γνώσης που αξιοποιεί οντολογίες και τεχνικές ανάκτησης-ενισχυμένης δημιουργίας (RAG) για την ενσωμάτωση, διαχείριση και επεξεργασία δομημένων και μη δομημένων δεδομένων. Αυτό επιτρέπει τη συνδυαστική χρήση διαφορετικών τύπων δεδομένων προερχόμενων από διαφορετικές πηγές. Επιπλέον, προτείνονται τεχνικές για τη βελτίωση της προσαρμοστικότητας μοντέλων ανάλυσης χρονοσειρών σε πραγματικό χρόνο, επιτρέποντας συνεχή ενημέρωση του συ-

στήματος για την αποτύπωση εξελισσόμενων τάσεων και τη διατήρηση ισχυρής απόδοσης σε δυναμικά περιβάλλοντα. Παρουσιάζονται επίσης τεχνικές για βελτιστοποιημένη χρήση Μεγάλων Γλωσσικών Μοντέλων (LLMs) για ανάλυση και ταξινόμηση κειμένου, συγκρίνοντας την αποτελεσματικότητά τους με παραδοσιακές τεχνικές μεταφοράς μάθησης για ανάλυση συναισθήματος σε χρηματοοικονομικά άρθρα. Τέλος, προτείνεται ένα καινοτόμο σύστημα που συνδυάζει και επεκτείνει αυτές τις μεθοδολογίες, διαχειριζόμενο κείμενο, αριθμητικά δεδομένα και δεδομένα χρονοσειρών από ποικίλες πηγές, παρέχοντας μια ολοκληρωμένη λύση με εφαρμογές στον χρηματοπιστωτικό τομέα.

Η διατριβή επικυρώνει τις προτεινόμενες μεθοδολογίες μέσω εφαρμογών στον πραγματικό κόσμο, όπως η πρόβλεψη χρονοσειρών για τη διαχείριση πόρων στο cloud computing, η αξιολόγηση χρηματοοικονομικού κινδύνου, η ανάλυση συναισθήματος και η συστηματική επιλογή μετοχών στη χρηματοοικονομική. Αυτές οι συνεισφορές επιδεικνύουν σημαντικές βελτιώσεις στη διαχείριση σύνθετων συνόλων δεδομένων, εξασφαλίζοντας προσαρμοστικότητα σε πραγματικό χρόνο και παρέχοντας ερμηνεύσιμα και αξιόπιστα αποτελέσματα με τη χρήση AI. Αντιμετωπίζοντας τις κοινές προκλήσεις της πολυπλοκότητας και της ετερογένειας των δεδομένων σε εφαρμογές του πραγματικού κόσμου μέσω συνδυασμένης χρήσης των προτεινόμενων μεθοδολογιών με επίκεντρο τα δεδομένα, αυτή η διατριβή προτείνει καινοτόμες λύσεις για την αποτελεσματική εκμετάλλευση ποικίλων και σύνθετων δεδομένων, συμβάλλοντας στην ανάπτυξη εφαρμοσμένων συστημάτων Τεχνητής Νοημοσύνης σε διάφορους τομείς.

Table of Contents

List of Figures

List of Tables

List of Algorithms

Acronyms

AI Artificial Intelligence

ADF Augmented Dickey-Fuller

AMQP Advanced Message Queuing Protocol

API Application Programming Interface

ARIMA Autoregressive Integrated Moving Average

AWS Amazon Web Services

BiGAN Bidirectional Generative Adversarial Network

DA Directional Accuracy

DL Deep Learning

DNN Deep Neural Network

ECB European Central Bank

ETFs Exchange Traded Funds

ETS Exponential Smoothing

EVT Extreme Value Theory

FaaS Function as a Service

FED Federal Reserve

FIBO Financial Industry Business Ontology

FX Foreign Exchange

GANs Generative Adversarial Networks

GARCH Generalized AutoRegressive Conditional Heteroskedasticity

GPT Generative Pre-trained Transformer

JSON JavaScript Object Notation

HS Historical Simulation

IoT Internet of Things

LM Language Model

LLM Large Language Model

LLMs Large Language Models

LSTM Long Short-Term Memory

MAE Mean Absolute Error

MC Monte Carlo

ML Machine Learning

MMLU Massive Multitask Language Understanding

NLP Natural Language Processing

OWL Web Ontology Language

PCA Principal Components Analysis

QA Question Answering

RAG Retrieval-Augmented Generation

RDF Resource Description Framework

REST Representational State Transfer

RM RiskMetrics

RNNs Recursive Neural Networks

SEC Securities and Exchange Commission

SES Simple Exponential Smoothing

S-MAE Sentiment Mean Absolute Error

STD Standard Deviation

SVM Support Vector Machines

VaR Value at Risk

0DTE Zero-Day Expiry options

Chapter 1

Introduction

Chapter Structure

This Chapter is constructed as follows:

- **Section 1.1 - Motivation**, explains the reasons behind this research, and highlights its relevance and potential impact.
- **Section 1.2 - Objectives**, outlines the goals and objectives of the research.
- **Section 1.3 - Research Questions**, outlines the main questions that the study aims to answer.
- **Section 1.4 - Research Contributions**, explains the contributions made by the research.
- **Section 1.5 - Outline of Dissertation**, presents a structural overview of the dissertation.

1.1 Motivation

Over the past decade, advancements in Artificial Intelligence (AI) and Big Data have transformed various industries, improving existing practices and introduc-

ing new paradigms of efficiency and innovation [Siripurapu et al., 2023]. AI, including Machine Learning (ML) and Deep Learning (DL), has demonstrated significant potential in analyzing vast, complex, and heterogeneous datasets, automating intricate processes, and providing predictive insights that were previously unattainable. The integration of scalable software systems, advanced ML frameworks, and parallel hardware capabilities has enabled the processing and analysis of multi-frequency and multi-type data on an unprecedented scale [Elshawi et al., 2018].

AI and Big Data technologies have advanced significantly, driven by the exponential growth of digital data and the demand for data-driven decision-making [Langer and Mukherjee, 2023]. These technologies have been successfully applied in fields such as healthcare, retail, transportation, and finance, demonstrating their ability to extract meaningful insights from structured, unstructured, and semi-structured data, thereby enhancing operational efficiency, improving customer experiences, and supporting strategic planning [Yadav et al., 2023]. For instance, AI algorithms are now capable of diagnosing diseases with high accuracy [Al-Antari, 2023], predicting consumer behavior, optimizing supply chains [Toorajipour et al., 2021], and automating financial analysis and decisions [Soldatos and Kyriazis, 2022].

The transformative potential of AI lies in its ability to learn from historical data, adapt to new information, and provide actionable insights, even when dealing with highly dimensional and multi-frequency datasets. In finance, AI-driven models are capable of comprehensively analyzing market trends, forecasting risks, and significantly assisting in making informed investment decisions [Emmert-Streib, 2021]. Similarly, in cloud computing, AI can optimize resource allocation efficiently, enhance performance, and contribute to substantial cost reductions. These advancements highlight AI's wide-reaching impact across various sectors, ushering in a new era of innovation and effi-

ciency [Hameed et al., 2016].

Furthermore, the advent of Large Language Models (LLMs), especially following the introduction of OpenAI's GPT-3.5 model, has revolutionized the application and transformative potential of AI [Ye et al., 2023]. LLMs, with their advanced Natural Language Processing (NLP) capabilities, can understand, generate, and translate human language with impressive accuracy. By processing and extracting knowledge from unstructured data through context understanding, these models provide a novel approach to utilizing diverse data formats effectively. For example, in finance, LLMs can analyze large volumes of financial documents, news articles, and social media feeds, helping investors gauge market sentiment and make informed decisions [Fatouros et al., 2023d].

These recent advancements and the rapid pace of AI development present challenges that must be addressed to enable the effective integration of such models into real-world settings while leveraging all the knowledge available in complex data. These challenges include managing high-dimensionality and heterogeneous data sources, extracting meaningful information from multi-type and multi-frequency datasets, and ensuring real-time adaptation and performance.

More specifically, AI models often need to process and analyze data that is not only vast in volume but also highly complex and multidimensional. This complexity arises from the variety of data sources and types, such as structured data, unstructured text, sensor data, and data collected at different frequencies [Chen, 2022a]. Addressing these challenges requires robust data-centric approaches, knowledge modeling via ontologies, semantic interoperability among different data sources and systems, and datastores capable of managing such data [Munir and Anjum, 2018]. However, developing models that can effectively handle and extract meaningful insights from heterogeneous data sets, such as those used in financial applications and cloud resource management, remains a significant challenge [Jung, 2018].

Additionally, AI systems must be capable of adapting in real-time while maintaining high performance. In dynamic environments, where data and conditions change rapidly, AI models must continuously update and refine their parameters. This requires advanced techniques for real-time learning and the design of data-efficient models [Guan et al., 2023]. Balancing trade-offs between accuracy, speed, and resource consumption is crucial for developing effective real-time AI applications.

Moreover, LLMs such as ChatGPT have demonstrated remarkable capabilities in information retrieval from unstructured data while performing tasks such as text summarization, Question Answering (QA), and Massive Multitask Language Understanding (MMLU), often outperforming human experts [Ovadia et al., 2023]. With appropriate prompt engineering and agent-based techniques, LLMs can provide personalized explanations in natural language, assisting humans in decision-making [Najork, 2023]. Despite these strengths, LLMs face limitations related to the scope of their training data and the probabilistic nature of their outputs, which can lead to hallucinations [Ai et al., 2023]. Furthermore, these models often lack robustness in mathematical reasoning, limiting their utility in contexts involving complex numerical analysis [Xiong et al., 2024].

The rapid advancements in AI and Big Data technologies hold great promise for revolutionizing various industries. However, addressing the challenges of handling multi-frequency, multi-type, high-dimensional data, ensuring real-time adaptation and performance, effectively bootstrapping LLMs with proprietary knowledge, and developing techniques that enhance their mathematical reasoning are essential for realizing the full potential of these technologies and the available real world data. By tackling these challenges, we can create more reliable and impactful AI solutions that drive innovation and improve outcomes across diverse sectors.

1.2 Objectives

The primary objective of this research is to develop advanced data-centric AI methodologies capable of handling and extracting meaningful insights from heterogeneous, high-dimensional, and complex data sets. This involves creating robust systems that can effectively process and analyze data from diverse sources, including structured, unstructured, and multi-frequency data. By addressing the challenges associated with data variety, complexity, and high-dimensionality, this research aims to enhance the accuracy, efficiency, and scalability of AI applications across various domains, such as finance and cloud computing.

Another critical objective is to enhance the real-time adaptability and performance of AI systems. In dynamic environments where data and conditions evolve rapidly, AI models must be capable of continuously updating their parameters. This research introduces advanced techniques for real-time learning, emphasizing data efficiency and resource optimization, to ensure that AI applications maintain high performance and adapt swiftly to new information.

Additionally, the research aims to enhance the utility and robustness of LLMs, such as ChatGPT, for specific applications by integrating domain-specific information and providing more reliable and interpretable outputs. This involves developing prompt engineering methods, techniques to mitigate LLM limitations like numerical reasoning, and approaches to improve reproducibility and structured outputs. By extending the capabilities of LLMs, this research seeks to improve their performance in tasks such as information retrieval, text summarization, question answering, and personalized explanations, thereby making these models more effective tools for decision-making and innovation.

To evaluate these methodologies, they applied in various real-world scenarios and across different ML tasks, ranging from classification to time-series analysis.

This research advances existing methods used for financial risk assessment, sentiment analysis, explainable stock selection, and reasoning across diverse data sources, including data from cloud computing and financial domains.

1.3 Research Questions

This thesis aims to answer the following research questions:

RQ1: **"How can advanced AI methodologies be developed to effectively handle and extract meaningful insights from heterogeneous, high-dimensional, and complex data sets, particularly in the domains of finance and cloud computing?"**

This question seeks to explore the development of robust systems that can manage diverse data sources and types, with a focus on improving the accuracy, efficiency, and scalability of AI applications. This is particularly relevant when dealing with the intricate nature of financial data and the vast, dynamic datasets typical of cloud computing environments. To address these points, this thesis evaluates the reasoning framework, RAG, and in-context learning techniques to assess their effectiveness in extracting insights from complex datasets.

RQ2: **"What techniques can be implemented to enhance the real-time adaptability and performance of AI systems in dynamic environments?"**

This question focuses on the development of advanced techniques for real-time learning and inference to ensure AI applications can maintain high performance and adapt swiftly to evolving information and conditions. This is critical in high-impact domains where model drift may have significant consequences. The thesis explores the implementation of adaptive algorithms on top of the reasoning framework from extbfRQ1, utilizing real-time monitoring and feedback loops to dynamically adjust models. Additionally, state-of-the-art DL time-series models are parameterized to facilitate efficient online learning. By

leveraging these techniques, AI systems can remain robust and efficient even as underlying data patterns evolve rapidly.

RQ3: "How can LLMs be optimized and compared to state-of-the-art transfer learning approaches in text classification?"

This question examines the optimization of LLMs and their comparative performance against traditional transfer learning methods in text classification tasks such as sentiment analysis. The study investigates various prompt engineering strategies to evaluate the efficacy of LLMs compared to domain-specific pretrained models. By comparing the performance of LLMs with existing transfer learning techniques, the thesis aims to determine the conditions under which LLMs provide superior results, particularly in handling complex and context-rich text data.

RQ4: "In what ways can LLMs be enhanced to improve their utility and relevance for specific applications, while mitigating their inherent limitations?"

This question explores methods for integrating domain-specific information into LLMs, leveraging the findings from **RQ1**, **RQ2**, and **RQ3** to improve their performance in tasks such as information retrieval, text classification, summarization, and question answering, particularly in the financial sector. The research delves into techniques such as prompt engineering, knowledge distillation, and the incorporation of proprietary data to overcome limitations like processing numbers, ensuring reproducibility, and generating structured outputs. By enhancing LLMs with targeted knowledge, this thesis aims to expand their applicability and reliability in specialized domains.

1.4 Research Contributions

The research conducted and presented in this thesis makes four key contributions, which are aligned with the formulated research questions. These contributions are divided into the following main areas: (*i*) the development of robust AI methodologies for handling heterogeneous data, (*ii*) the enhancement of real-time adaptability and performance of AI systems, (*iii*) the optimization and comparison of LLMs with traditional methods, and (*iv*) the bootstrapping of LLMs for specific applications while mitigating their limitations.

RC1: A Knowledge-Based Reasoning Framework

In order to answer **RQ1**, this thesis presents a novel Knowledge-Based Reasoning Framework. This framework leverages semantic matching, knowledge graphs, text embeddings, and ontology technologies to effectively manage and extract meaningful insights from diverse data sources. The framework addresses the challenges of data variety and complexity, thereby improving the accuracy, efficiency, and scalability of AI applications. The thesis evaluates two different versions of the Reasoning Framework: (*i*) managing application and resource data used in cloud computing services and (*ii*) enabling data mining from diverse financial reports and articles.

RC2: Real-Time Adaptability Enhancement Techniques

Addressing **RQ2**, this thesis introduces advanced techniques for enhancing the real-time adaptability and performance of AI systems via online learning. The research includes the development of adaptive algorithms, utilizing real-time monitoring and feedback loops to dynamically adjust models. These techniques ensure that AI applications maintain high performance and adapt quickly to new information and conditions, which is crucial in high-impact domains where model drift may have significant consequences. This work presents two systems that allow for real time time series prediction: (*i*) predict-

ing resource utilization in serverless cloud computing and (*ii*) predicting risk of financial assets in real time.

RC3: Optimization and Evaluation of LLMs in Text Classification

In response to **RQ3**, this thesis investigates the optimization of LLMs and their comparative performance against state-of-the-art transfer learning approaches in text classification tasks. Various prompt engineering strategies are explored to enhance the efficacy of LLMs. The study provides a comprehensive comparison, highlighting the conditions under which LLMs outperform traditional methods, thereby contributing to the understanding of their advantages and limitations in handling complex text data.

RC4: Bootstrapping LLMs for Enhanced Utility and Specific Applications

To address **RQ4**, this thesis explores methods for bootstrapping LLMs with domain-specific knowledge to enhance their utility and relevance for specific applications. Techniques such as prompt engineering, knowledge distillation, and the incorporation of proprietary data are developed to mitigate limitations like processing numbers, ensuring reproducibility, and generating structured outputs. The research leverages findings from **RQ1**, **RQ2**, and **RQ3** to improve the performance of LLMs in tasks such as information retrieval, text classification, summarization, and question answering, particularly in the financial sector.

By making these contributions, this thesis advances the state of the art in AI methodologies, particularly in the application of information retrieval, providing significant insights and practical solutions for handling complex, heterogeneous data to make timely predictions in dynamic environments.

Table 1.1 demonstrates how the publications are mapped to any of the aforementioned research contributions.

Table 1.1: List of publications related to the contributions of this thesis.

Title	Authors	Venue	Contribution
DeepVaR: a framework for portfolio risk assessment leveraging probabilistic deep neural networks	**G. Fatouros**, G. Makridis, D. Kotios J. Soldatos, M. Filippakis, D. Kyriazis	Springer, Digital Finance, 5(1), 29-56, 2023	C2
Transforming sentiment analysis in the financial domain with ChatGPT	**G. Fatouros**, J. Soldatos, K. Kouroumali, G. Makridis, D. Kyriazis	Elsevier, Machine Learning with Applications, 14, 100508, 2023	C3, C4
Can Large Language Models Beat Wall Street? Evaluating GPT-4's Impact on Financial Decision-Making with MarketSenseAI	**G. Fatouros**, K. Metaxas, J. Soldatos, D. Kyriazis	Springer, Neural Computing and Applications 1433-3058, 2024	C3, C4
Deep learning enhancing banking services: a hybrid transaction classification and cash flow prediction approach	D. Kotios, G. Makridis, **G. Fatouros**, D. Kyriazis	Springer, Journal of Big Data, 2023, 9(1), 100	C2
Embedding Automated Function Performance Benchmarking, Profiling and Resource Usage Categorization in Function as a Service DevOps Pipelines	V. Katevas, **G. Fatouros**, D. Kyriazis, G. Kousiouris	Elsevier, Future Generation Computer Systems, 2024	C1, C2
Knowledge graphs and interoperability techniques for hybrid-cloud deployment of faas applications	**G. Fatouros**, Y. Poulakis, A. Polyviou, S. Tsarsitalidis, G. Makridis, J. Soldatos, G. Kousiouris, M. Filippakis, D. Kyriazis	2022 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)	C1, C2, C4
Enhanced Runtime-Adaptable Routing for Serverless Functions based on Performance and Cost Tradeoffs in Hybrid Cloud Settings	**G. Fatouros**, G. Kousiouris, G. Makridis, J. Soldatos, M. Filippakis, D. Kyriazis	2023 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)	C1, C2
Hocc: an ontology for holistic description of cluster settings	Y. Poulakis, **G. Fatouros**, G. Kousiouris, D. Kyriazis	2022 International Conference on the Economics of Grids, Clouds, Systems, and Services (GECON).	C1, C2
Enhancing smart agriculture scenarios with low-code, pattern-oriented functionalities for cloud/edge collaboration	**G Fatouros**, G. Kousiouris, T. Lohier, G. Makridis, A. Polyviou, J. Soldatos, D. Kyriazis	2023 19th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT)	C1, C2
Enhancing Explainability in Mobility Data Science through a combination of methods.	G. Makridis, V. Koukos, **G. Fatouros**, M. Separdani, D. Kyriazis	2024 Science and Information Conference	C4
Towards a Unified Multidimensional Explainability Metric: Evaluating Trustworthiness in AI Models	G. Makridis, **G. Fatouros**, A. Kiourtis, D. Kotios, V. Koukos, D. Kyriazis, J. Soldatos	2023 19th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT)	C4
XAI for time-series classification leveraging image highlight methods	G. Makridis, **G. Fatouros**, V. Koukos, D. Kotios, D. Kyriazis, J. Soldatos	2023 International Conference on Management of Digital EcoSystems (MEDES)	C4
Addressing Risk Assessments in Real-Time for Forex Trading	**G. Fatouros**, G. Makridis, J. Soldatos, P. Ristau, V. Monferrino	Springer, Big Data and Artificial Intelligence in Digital Finance, 2022	C2
Large Language Models and Extended Reality Convergence for Personalized Training at the Industrial Metaverse	**G. Fatouros**, M. Touloupos, J. Soldatos	2024 32nd European Conference on Information Systems (ECIS)	C1, C4

1.5 Outline of Dissertation

The remainder of this dissertation is organized in the following way:

- **Chapter 2** presents the Reasoning Framework for semantic matching. The chapter initially performs a literature review covering knowledge bases, vector databases, RAG methods, and ontologies. It then details the

architecture of the proposed Reasoning Framework, which facilitates the semantic matching of diverse data. Furthermore, this chapter evaluates the proposed framework through case studies in cloud computing and finance, demonstrating its effectiveness in handling complex, heterogeneous data sets and optimizing resource allocation.

- **Chapter 3** focuses on online learning in time series prediction. The chapter presents an AI-based system designed to update model parameters in real-time, enabling the efficient processing of time-series data. The chapter evaluates this approach through two distinct use cases: the first predicts function latency in serverless cloud computing, supporting the Reasoning Framework discussed in Chapter 2; the second applies online learning to financial risk prediction by adapting a state-of-the-art deep learning algorithm. The chapter begins by reviewing the necessity of online learning in dynamic environments like cloud computing and finance, followed by a detailed explanation of the proposed system's architecture. It concludes with performance evaluations that assess the efficiency and accuracy of the online learning approach in these applications.

- **Chapter 4** focuses on textual data and NLP, examining the efficiency of LLMs, particularly ChatGPT, in text classification of financial data. The chapter begins with an introduction to text classification and sentiment analysis within the financial domain, followed by a review of prior work on sentiment analysis in finance and the role of AI tools in this area. It then details the methodology for using and evaluating ChatGPT for sentiment classification in the Foreign Exchange (FX) market. The chapter concludes with the research findings and their implications for real-world use cases, providing insights into the effectiveness of LLMs in financial sentiment analysis.

- **Chapter 5** presents a holistic LLM-based system for explainable stock

analysis, selection, and ranking, called MarketSenseAI. This system integrates the techniques discussed in previous chapters, leveraging knowledge bases, text analytics, LLM reasoning, and serverless processing to deliver comprehensive financial analysis. The chapter begins by highlighting the need for such a service and provides an analysis of the current state of the art. The proposed system and its processes are then described in detail, followed by an explanation of the evaluation methodology, including the data used, comparison methods, and empirical findings. The chapter concludes with a summary of the primary contributions and insights provided by MarketSenseAI.

- **Chapter 6** summarizes the doctoral dissertation and its main contributions. It reflects on the research questions posed at the beginning, outlining how each was addressed through the work presented. The chapter also discusses the open research topics and future goals that have emerged from this research, providing a roadmap for subsequent investigations and potential developments in the field.

Chapter 2

Knowledge Modeling for Semantic Matching of Heterogeneous Data Types

Chapter Structure

This Chapter is constructed as follows:

- **Section 2.1 - Background on Knowledge Systems**, analyzes the State of the Art on Knowledge and Vector databases, ontologies, RAG methods, and semantic reasoning with a focus on cloud computing and finance.
- **Section 2.2 - Overview and Design of the Reasoning Framework**, presents an overview of the proposed Reasoning Framework including its architecture, internal components and their relationships.
- **Section 2.3 - Use Cases in Cloud Computing and Finance, and Evaluation**, evaluates the proposed Reasoning Framework under two distinct uce cases in cloud computing and finance and presents the outcomes of this evaluation.
- **Section 2.4 - Discussion**, provides a summary of the chapter's contributions, highlights the challenges encountered, and outlines future work directions.

In today's data-driven world, the ability to semantically match and integrate heterogeneous data types has become a critical challenge across various domains, including cloud computing and finance. As organizations increasingly rely on diverse data sources—ranging from structured databases to unstructured text—the need for robust knowledge systems that can effectively manage and leverage this data has never been greater [Nguyen et al., 2022]. This chapter introduces the proposed Reasoning Framework, an architecture designed to facilitate semantic matching and reasoning across different types of data. By integrating techniques such as knowledge graphs, ontologies, and RAG methods, this framework aims to enhance decision-making processes and optimize resource allocation in complex environments [Wang et al., 2024]. The following sections will explore the state of the art in knowledge systems, outline the architecture of the Reasoning Framework, and evaluate its application in both cloud computing and financial use cases.

2.1 Background on Knowledge Systems

2.1.1 Knowledge Bases and Ontologies

2.1.1.1 Knowledge Bases

Knowledge bases are structured repositories that store information in a way that supports the retrieval and reasoning processes necessary for complex decision-making tasks. They are central to various AI applications, particularly those requiring the management and integration of large amounts of structured data [Krzywicki et al., 2016]. Knowledge bases typically consist of entities, attributes, and relationships, and they are designed to facilitate easy querying and manipulation of data. These systems are foundational in fields like natural language processing, semantic web technologies, and expert systems [Kingston, 2001]. Table 2.1 presents the most prominent graph databases

supporting Resource Description Framework (RDF) which is required in the development of knowledge systems.

Table 2.1: Comparison of Graph Databases

Name	Paper	License	Features
AlegroGraph	[Fernandes and Bernardino, 2018]	Commercial / OSS	Supports RDF, SPARQL, and graph analytics
Neo4j	[Miller, 2013]	GPL	ACID transactions, native graph storage, and real-time queries
GraphDB	[Lopez-Veyna et al., 2022]	Commercial	Supports RDF, SPARQL, and semantic graph processing
Virtuoso	[Erling, 2012]	Commercial	Multi-model data storage, SPARQL support, and linked data capabilities
Apache Jena	[Siemer, 2019]	Apache License 2.0	Framework for building semantic web applications, supports RDF and SPARQL

Historically, knowledge bases have been employed in a variety of domains, from medical diagnosis systems to customer service chatbots, where they support the retrieval of relevant information based on specific queries. The structure of knowledge bases allows them to answer complex queries by leveraging predefined rules and relationships between data points [Tran et al., 2022]. This makes them particularly valuable in environments where the precise organization of data is crucial for accurate and efficient retrieval.

2.1.1.2 Ontologies for Semantic Reasoning

Ontologies build upon the concept of knowledge bases by introducing a formal representation of a set of concepts within a domain and the relationships between those concepts. An ontology is essentially a model for describing the world that consists of a vocabulary of terms, and the specification of their meanings and relationships [Chalupsky et al., 2002]. Ontologies are used to create shared understanding among people or software agents, facilitating interoperability between systems by providing a common framework that all

components understand.

In the context of semantic reasoning, ontologies play a critical role by enabling machines to infer new knowledge from existing data. This is done through logical rules that define how different concepts relate to one another, allowing systems to draw conclusions that are not explicitly stated in the data. Ontologies are particularly effective in environments that require the integration of data from multiple sources, such as cloud computing environments where different services and resources must be coordinated [Bajraktari et al., 2018]. They enable the consistent interpretation of data across different systems, ensuring that semantic discrepancies do not hinder interoperability.

2.1.1.3 Applications in Cloud Computing and Finance

The application of knowledge bases and ontologies in cloud computing has gained significant traction, particularly with the advent of Function as a Service (FaaS) platforms. FaaS offers a novel cloud computing model that abstracts the complexity of managing infrastructure, allowing developers to focus solely on writing and deploying functions as part of an application workflow [Van Eyk et al., 2018]. These workflows, which consist of linked functions, services, or actions, can have varying requirements for computational resources, latency, and performance [Van Eyk et al., 2017].

In cloud computing, ontologies and knowledge bases are critical for enabling semantic matching between the specific requirements of these workflows and the available cloud resources. For instance, in hybrid cloud environments where workflows may span across public, private, and edge resources, ontologies can model the relationships between various components, such as virtual machines, containers, and network configurations. This modeling allows the system to optimize resource allocation by automatically matching the application's requirements—such as locality, latency, and hardware needs—with the

most suitable resources available in the cloud or at the edge [Rausch et al., 2021].

For example, in IoT-driven applications in agriculture, functions may need to be deployed on edge devices located in greenhouses to interact directly with sensors, while other functions, like simulation models, require more powerful cloud resources. Ontologies facilitate this by providing a semantic understanding of the application requirements and the capabilities of the available infrastructure, enabling efficient and context-aware deployment of the entire workflow. This semantic reasoning is crucial in overcoming challenges such as vendor lock-in and ensuring interoperability between different cloud environments, as noted in studies focusing on multi-cloud and hybrid cloud FaaS deployments [Kousiouris and Kyriazis, 2021, Hellerstein et al., 2018].

In the financial sector, the use of knowledge bases and ontologies, such as the Financial Industry Business Ontology (FIBO), is particularly prominent [Petrova et al., 2017]. FIBO provides a standardized framework for representing complex financial concepts, relationships, and processes, facilitating the integration and analysis of diverse financial data sources. Ontologies in finance enable automated systems to perform tasks such as risk assessment, fraud detection, and regulatory compliance more accurately by providing a shared understanding of financial terms and their interrelations.

2.1.1.4 Challenges of Knowledge Systems

While knowledge bases and ontologies offer substantial benefits in cloud computing and finance, several challenges remain, particularly in the dynamic and heterogeneous environments typical of these domains. In cloud computing, one of the key challenges is enabling the seamless deployment of FaaS applications across hybrid cloud environments. The lack of standardization and interoperability between different cloud platforms often results in vendor

lock-in, limiting the transferability of workflows from one cloud provider to another [Van Eyk et al., 2017]. Ontologies can address these issues by providing a common framework for describing and matching the requirements of application functions with available resources [Agbaegbu et al., 2021]. However, maintaining and updating these ontologies to keep pace with the rapidly evolving cloud landscape remains a significant challenge.

Another challenge is the integration of unstructured data within these knowledge systems. Traditional knowledge bases and ontologies are primarily designed to handle structured data, which limits their effectiveness in domains where unstructured data is prevalent, such as in financial markets where news articles, social media, and other textual data play a crucial role in decision-making. Additionally, the integration of ontologies with ML models poses another challenge, as these models often rely on large volumes of unstructured data that are not easily captured by traditional ontological frameworks.

2.1.2 Vector Databases and Embedding-Based Retrieval

2.1.2.1 Vector Databases

Vector databases have emerged as a crucial technology in managing and querying high-dimensional data, particularly in applications that involve unstructured or semi-structured data such as text, images, and multimedia [Adnan and Akbar, 2019]. Unlike traditional databases that rely on structured data and relational queries, vector databases store data in the form of vectors—numeric representations of data points in a high-dimensional space. This allows for efficient similarity searches, where the goal is to find data points (vectors) that are closest to a given query vector according to some distance metric (e.g., Euclidean distance, cosine similarity). Table 2.2 presents the most prominent vector databases along with their features.

Table 2.2: Comparison of Vector Databases

Name	Paper	License	Features
Manu	[Guo et al., 2022]	Commercial	Cloud-native, horizontally scalable, high performance, tunable consistency
Pinecone	[Touya and Lokhat, 2020]	Commercial	Real-time vector search, automatic scaling, and integration with ML models
Weaviate	[Singh et al., 2023]	Open Source	GraphQL interface, hybrid search, and support for contextionary embeddings
Faiss	[Douze et al., 2024]	Open Source	Efficient similarity search, high-dimensional vectors, and GPU support
Milvus	[Wang et al., 2021]	Open Source	High availability, distributed architecture, and support for various indexing methods
Chroma	[Ramprasad and Sivakumar, 2024]	Open Source	Utilizes chroma features for music information retrieval and singer identification

The key advantage of vector databases is their ability to handle large-scale, unstructured data, which is increasingly common in various domains such as NLP, computer vision, and recommendation systems [Kukreja et al., 2023]. By converting complex data types into vectors, these databases enable fast and scalable retrieval of similar items, making them indispensable for applications like image recognition, document search, and recommendation engines.

In the context of AI, vector databases are often used in conjunction with embedding models, which convert raw data into vector representations. These embeddings capture the semantic relationships between data points, allowing for more accurate searches. For example, in NLP, word embeddings can capture the meaning of words based on their usage in large text corpora, enabling the retrieval of semantically similar terms even if they are not exact matches [Touya and Lokhat, 2020].

2.1.2.2 Embedding Techniques for Semantic Matching

Embedding-based retrieval relies on the creation of vector representations (embeddings) of data points that preserve the semantic relationships inherent in the original data. Embeddings are typically generated using ML models trained on large datasets, which learn to map data points into a continuous vector space where semantically similar items are located closer together [Kukreja et al., 2023].

In the realm of textual data, embeddings such as Word2Vec [Church, 2017], GloVe [Pennington et al., 2014], and BERT [Devlin et al., 2018] have revolutionized how semantic similarity is measured. These models generate dense vector representations of words, sentences, or documents that capture their meanings based on context. For instance, Word2Vec generates embeddings by predicting a word based on its surrounding words in a sentence, effectively capturing the context in which the word appears. BERT, on the other hand, uses a transformer-based architecture to generate embeddings that consider the bidirectional context of words in a sentence, resulting in even more accurate representations.

When these embeddings are stored in a vector database, they enable powerful semantic matching capabilities. For example, a query for a particular concept can retrieve documents or items that are semantically similar, even if they do not contain the exact words or phrases used in the query. This is particularly valuable in applications such as document search, where the goal is to retrieve relevant information based on the meaning rather than the exact wording of the query.

2.1.2.3 Use Cases in Data Processing

Vector databases and embedding-based retrieval are emerging technologies that have shown significant promise in improving the efficiency and effectiveness of

information retrieval across various domains. Although these technologies are relatively new, they are being actively explored in several key areas, and there are some early-stage implementations.

In the financial sector, vector databases are being investigated for their potential to enhance the analysis and retrieval of unstructured text data, such as news articles, earnings reports, and Securities and Exchange Commission (SEC) fillings [LlamaIndex, 2024]. The ability to embed textual data into vectors allows financial institutions to explore more sophisticated methods of identifying trends, sentiments, and potential risks based on the semantic content of the data [CNBC, 2023]. However, real-world applications in this sector are still limited, and much of the current work is focused on research and development rather than widespread deployment.

In cloud computing, vector databases are being used experimentally to manage and retrieve information from logs, monitoring data, and other unstructured sources critical for maintaining the performance and security of cloud infrastructure [Naveen, 2023]. The embedding of these data points into vectors enables more effective detection of anomalies and retrieval of related events or issues. However, comprehensive implementations are not yet widespread, and these technologies are primarily being tested in controlled environments.

One of the more established use cases for embedding-based retrieval is in recommendation systems [Shi et al., 2023], where companies like Netflix and Amazon employ vector databases to store and analyze user preferences and item characteristics. These embeddings enable the recommendation of products, movies, or services that align with the user's interests. This application is relatively mature compared to other potential uses of vector databases.

Overall, while the use of vector databases and embedding-based retrieval in large-scale data processing is still in its early stages, the initial applications in finance, cloud computing, and recommendation systems illustrate the technol-

ogy's potential. As these technologies continue to evolve, further real-world implementations are expected to emerge.

2.1.2.4 Challenges in Vector Search

While vector databases and embedding-based retrieval offer significant advantages, they also present several challenges, particularly in scaling these systems to handle the ever-increasing volume of data. One of the primary challenges is the computational cost associated with generating and storing embeddings for large datasets [Kukreja et al., 2023].

Another challenge is the interpretability of embeddings [Khoshraftar et al., 2021]. Unlike traditional keyword-based search, where the relationship between a query and the retrieved results is relatively straightforward, the relationship between vectors in a high-dimensional space is more abstract. This can make it difficult for users to understand why certain results were retrieved, which is particularly problematic in sensitive domains like finance.

To address these challenges, ongoing research is focused on improving the efficiency of embedding generation and storage, as well as developing techniques for more interpretable vector-based retrieval. Additionally, there is growing interest in hybrid approaches that combine the strengths of vector databases with traditional relational databases or graph databases, allowing for more flexible and powerful data retrieval capabilities [Friedman and Broeck, 2020].

2.1.3 Retrieval-Augmented Generation Methods

2.1.3.1 Overview

RAG is an emerging technique in the field of NLP that combines the strengths of retrieval-based models with generative AI models by incorporating real-time retrieval of relevant information from external knowledge bases. By combining

retrieval mechanisms with generative models, RAG enables the production of more accurate and contextually relevant responses, especially when dealing with complex or specialized queries [Hu and Lu, 2024]. This hybrid approach leverages the strengths of both retrieval-based systems, which excel at accessing specific information, and generative models, which are adept at constructing coherent, human-like responses [Fatouros et al., 2024b].

2.1.3.2 Enhancing LLMs with RAG

Generative models, such as Generative Pre-trained Transformer (GPT) variants, rely solely on their training data, which can limit their effectiveness when responding to queries that require up-to-date or domain-specific knowledge. RAG addresses this limitation by integrating a retrieval component that searches an external vector database for relevant documents or information snippets [Shao et al., 2023]. The retrieved content is then used to inform the generative model, allowing it to produce responses that are not only coherent but also grounded and enriched with relevant details [Gao et al., 2023].

As shown in Figure 2.1, when a user submits a query, the RAG system first identifies and retrieves the most pertinent information from a pre-indexed knowledge base. This information is then passed to the generative model, which uses it as context to generate a more informed and accurate response. This process enhances the model's ability to provide precise answers, especially in scenarios where the original training data may not cover the query's context.

2.1.3.3 Applications in NLP

RAG methods are being actively explored in various NLP applications, particularly those that require accurate information retrieval combined with generative capabilities. One of the primary applications of RAG is in QA systems, where users ask questions, and the system provides answers based on a combination

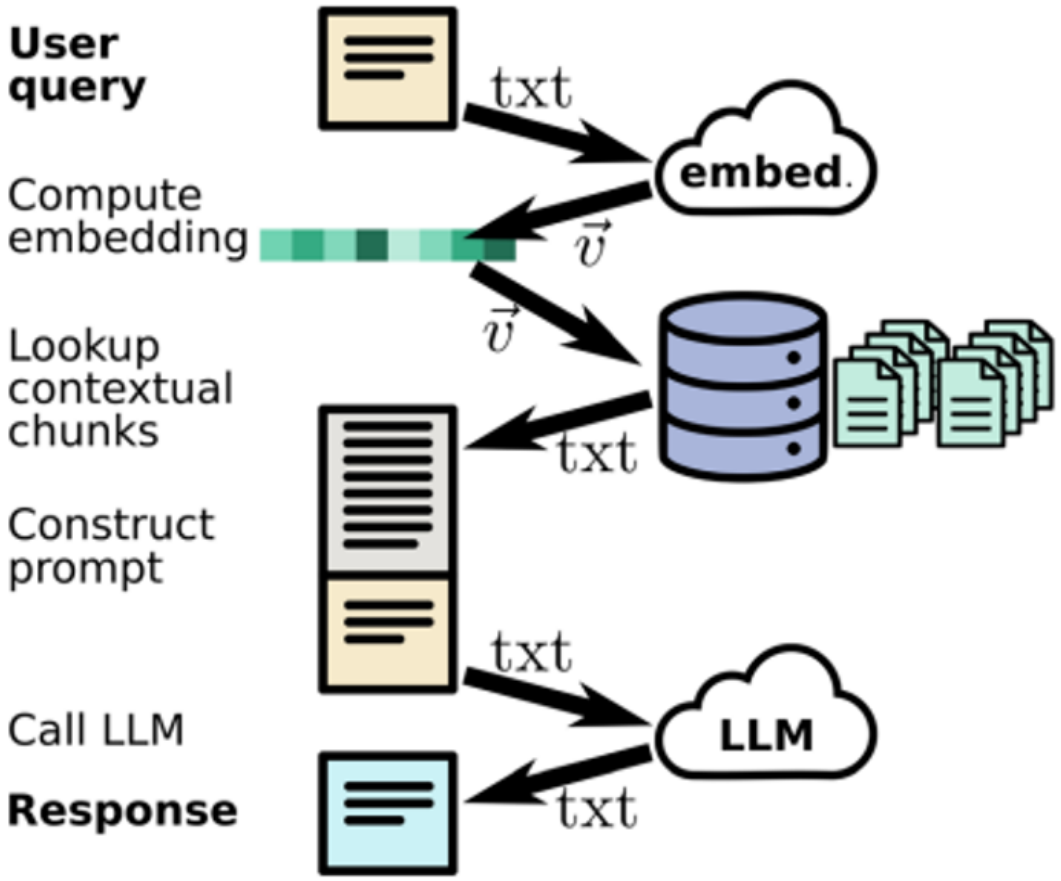


Figure 2.1: RAG Pipeline for augmenting LLM responses.

of pre-existing knowledge and information retrieved in real-time [Han et al., 2024]. For example, in the medical domain, a RAG-powered system could provide doctors with up-to-date information on the latest research findings or treatment guidelines by retrieving and synthesizing relevant documents from a medical knowledge base.

Another key application area for RAG is in customer support and helpdesk automation [Xu et al., 2024]. In this context, RAG systems can dynamically pull in relevant documentation, user manuals, or previous support cases to generate detailed, context-specific responses to customer queries, improving the efficiency and accuracy of automated customer service solutions. In addition to

these applications, LLM agents powered by RAG are increasingly being used to autonomously respond to events or generate content [Tang and Guo, 2024].

2.1.3.4 Challenges in RAG

Despite its potential, the implementation of RAG methods poses several challenges [Yu et al., 2024]. One of the primary challenges is ensuring the relevance and accuracy of the retrieved documents. The quality of the generated responses heavily depends on the quality of the retrieval process; if irrelevant or low-quality documents are retrieved, the generative model may produce inaccurate or misleading information.

Another challenge is the computational complexity involved in integrating retrieval with generation in real-time [Meduri et al., 2024]. The process of searching through large-scale vector databases and then conditioning the generation on the retrieved information requires significant computational resources, which can limit the scalability of RAG systems, especially in time-constrained applications.

2.2 The Reasoning Framework

The effective processing and interpretation of heterogeneous and complex data is critical in domains such as cloud computing and finance. To address this need, this section presents the design of a versatile reasoning framework capable of integrating various knowledge representation and retrieval mechanisms, including both traditional semantic reasoning and RAG techniques. This framework is designed to be flexible, allowing it to be applied across different use cases, such as optimizing cloud computing resources and generating structured financial summaries from multiple sources.

The Reasoning Framework is structured into several core components, each

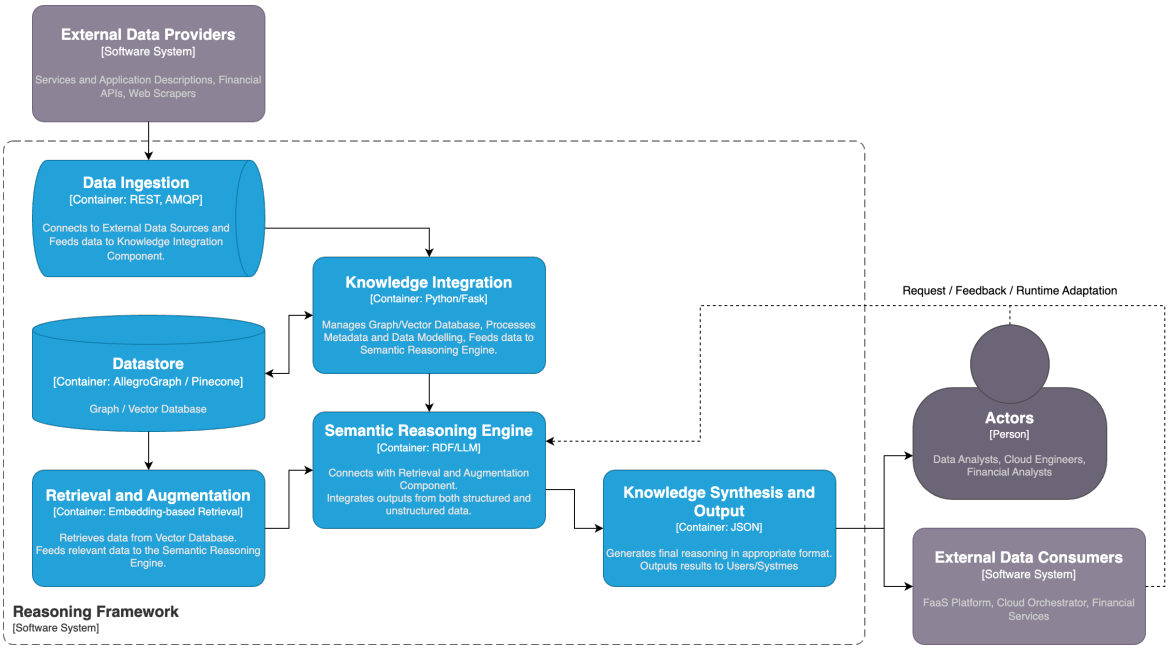


Figure 2.2: Reasoning Framework High-Level Architecture.

responsible for different aspects of knowledge processing and reasoning. The architecture is designed to be modular, allowing for the integration of various technologies depending on the specific use case. Figure 2.2 illustrates this architecture designed to support both traditional knowledge bases and RAG approaches.

Specifically, the **Knowledge Integration Layer** encompasses the Data Ingestion and Knowledge Integration components. The Data Ingestion Component is responsible for retrieving data from external sources and providers, utilizing technologies that support Representational State Transfer (REST) or Advanced Message Queuing Protocol (AMQP) protocols for real-time data acquisition. Depending on the specific use case, the Knowledge Integration component manages both structured and unstructured data. For structured data, it handles RDF triples and ontologies stored in graph databases, ensuring semantic interoperability and relationship standardization. For unstructured data, it processes text and embeddings, which are stored in vector databases. This

component is also crucial in extracting metadata via NLP techniques, organizing it within the appropriate datastore, whether graph or vector-based.

At the core of the framework is the **Semantic Reasoning Engine**, which drives the system's ability to infer insights from the integrated knowledge. This engine supports both traditional reasoning methods, leveraging ontologies and knowledge graphs to apply logical rules and derive conclusions, and dynamic reasoning approaches that use RAG techniques to process and reason over textual data. By combining these capabilities, the engine can adapt to new information and changing contexts, ensuring that the outputs it generates are both timely and relevant. The engine processes input requests or feedback from external sources, applying sophisticated reasoning to provide actionable insights across various domains.

The **Retrieval and Augmentation Module** further enhances the framework's functionality by enabling the dynamic incorporation of external data into the reasoning process. For structured data, it supports SPARQL query-based retrieval [Pérez et al., 2009], allowing for efficient data extraction from knowledge graphs. For unstructured data, the module integrates RAG methods, facilitating the retrieval of relevant documents or information snippets from vector databases. This module performs similarity searches and augments the reasoning process with pertinent information, which is then used to inform decision-making and content generation.

Finally, the **Knowledge Synthesis and Output** component is tasked with transforming the results of the reasoning processes into actionable and consumable outputs. In cloud computing contexts, this may involve generating optimized configurations or recommendations for resource allocation, which are then fed back into a cloud platform or orchestration service. In financial applications, this component generates structured summaries, reports, or analyses, combining retrieved financial data, market trends, and expert opinions to produce

outputs that are both informative and aligned with the latest developments. These outputs are then delivered in a format, such as JavaScript Object Notation (JSON), that meets the specific needs of the end users or systems interacting with the framework.

2.3 Use Cases in Cloud Computing and Finance

2.3.1 Use Case 1: Cloud Computing with Knowledge Graphs and Ontologies

2.3.1.1 Problem Context

FaaS is a novel cloud computing paradigm that allows customers to develop, run, and manage application functionalities without the complexity of managing the underlying infrastructure [Van Eyk et al., 2018]. However, deploying workflows that consist of linked functions, services, or actions in a hybrid cloud environment presents several challenges, particularly in matching application requirements with the capabilities of available resources. These challenges are compounded by the heterogeneous nature of cloud resources, including differences in computational power, latency requirements, and geographical location [Kousiouris and Kyriazis, 2021]. This section presents the application of the Reasoning Framework in addressing these challenges by leveraging knowledge graphs and ontology-based semantic reasoning.

2.3.1.2 Application of the Reasoning Framework

The Reasoning Framework proposed in this context performs semantic matching between application and resource metadata to facilitate the optimized deployment of FaaS applications in a hybrid cloud environment. The framework utilizes knowledge graphs and ontologies to model the relationships between

different resources and application components, enabling the automatic matching of application requirements with the appropriate cloud resources. In this use case the proposed framework is part of a platform being developed in the H2020 PHYSICS project[1] which enables the optimized deployment of FaaS application in hybrid cloud environments while abstracting the infrastructure layer for the customer. Figure 2.3 illustrates the positioning of the proposed service, the Reasoning Framework, with relation to other platform components.

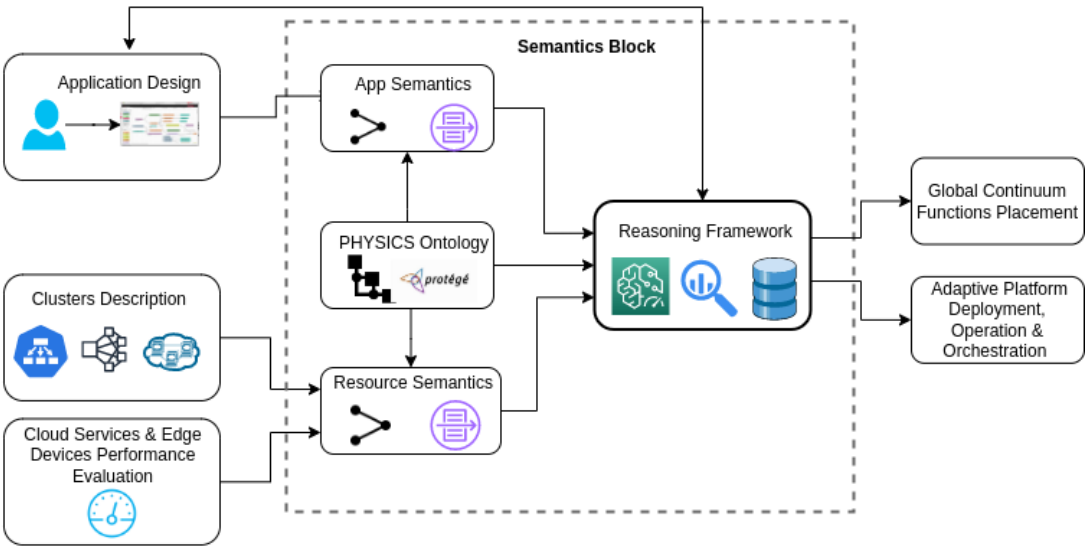


Figure 2.3: Reasoning Framework's dependencies with other components.

The framework retrieves application descriptions provided as graphs by the function editor specification (e.g., Node-RED), which includes developer-inserted annotations during the design process. These annotations are serialized in JSON-LD format and transformed into a formal application description schema based on the Web Web Ontology Language (OWL) [Bechhofer, 2004]. This schema allows the framework to interpret the application as a graph, facilitating the semantic matching process. The underlying data include all the information required for the application's deployment in a FaaS setting, such as the flows' executable docker image location, execution mode (function or

[1] https://physics-faas.eu/

service), and various annotations (e.g., locality, memory size, preferance on energy/performance etc., specific hardware needs etc.) populated by the developer, that need to be in line with the target infrastructure capabilities (see Figure 2.4).

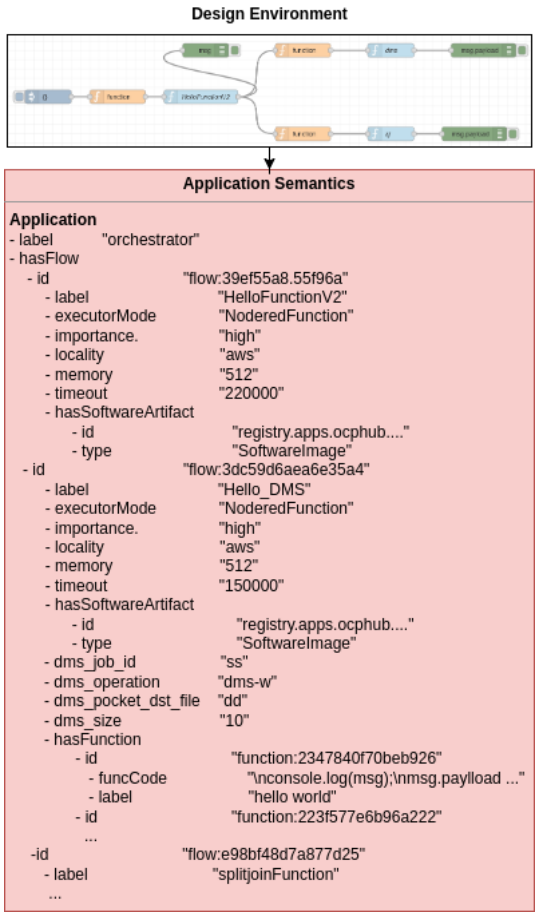


Figure 2.4: Application data as triples.

Resource descriptions from available clusters, using Kubernetes as the resource manager, are similarly ingested into the framework. These descriptions are translated into a unified format according to a pre-defined ontology [Poulakis et al., 2022], forming a resource graph. The created triples from each available cluster are ingested (also in JSON-LD format) to the Reasoning Framework forming the resource graph. The structure of sample resource data ingested in the Reasoning Framework from an Elastic Kubernetes Cluster (EKS) is il-

lustrated in Figure 2.5. This data includes information regarding the cluster locality, sizing (i.e., CPU, RAM), architecture, each cluster node's operating system, and metrics related to the performance, energy consumption, and resilience.

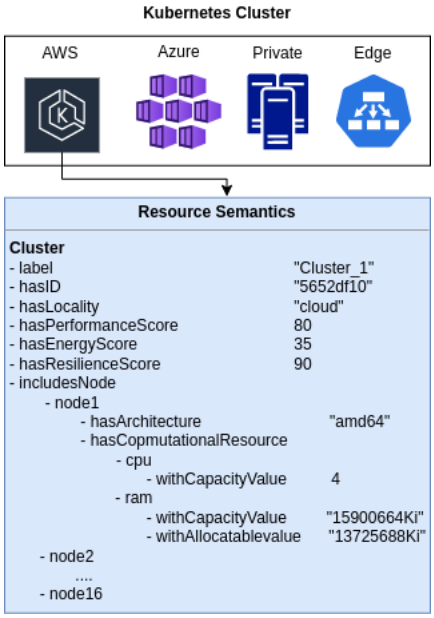


Figure 2.5: Resource data as triples.

The Reasoning Framework uses the input application and resource graphs to perform semantic matching between application requirements and available resources, effectively acting as a first-level filter for resource selection.

2.3.1.3 Workflow and Implementation

The Reasoning Framework's workflow begins with the ingestion of application and resource data into the knowledge base. The framework then applies semantic reasoning to infer relationships between the application and resource graphs. For instance, it can determine which cloud resources are best suited to deploy a given application workflow based on factors such as computational power, memory requirements, and locality constraints.

The implementation of the Reasoning Framework, illustrated in Figure 2.6 is

based on the AllegroGraph platform, a horizontally distributed, multi-model knowledge graph technology that supports sophisticated decision-making processes [Fernandes and Bernardino, 2018]. The backend service of the framework is implemented using Flask, which exposes RESTful endpoints for data ingestion, information retrieval, and inference requests. This microservices-based architecture allows the framework to be easily integrated into existing cloud platforms and to scale according to the demands of the application and resources.

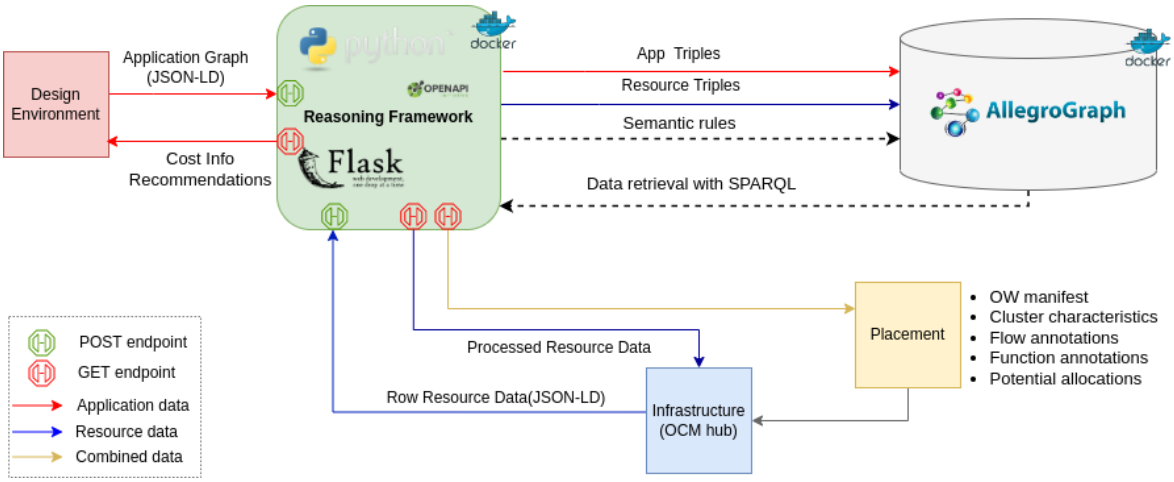


Figure 2.6: Reasoning Framework Implementation in the cloud computing use case.

2.3.1.4 Semantic Rules

After ingesting triples to the Reasoning Framework, these can be processed as graphs. With SPARQL queries to the knowledge graph, the required information, such as for example the workflows included in an application or the sizing of each cluster are made available as graph patterns [Zheng et al., 2016]. However, some of these data require multiple queries to be retrieved and this results in increased processing time for the Reasoning Framework to respond to requests. For this purpose, specific relationships between the nodes of each input graph are implemented in the form of INSERT queries. Hence, all required information per provided endpoint can be instantly retrieved with a

SELECT query. As an example, Figure 2.7 illustrates part of the available information of a target EKS cluster hosted in AWS before the INSERT query, where needed information (e.g, ram, cpu, architecture) for functions' allocations are part of each cluster node description. However, Reasoning Framework needs to aggregate such data to filter the clusters that do not meet application requirements. Such aggregations require multiple and complex queries as well transformations performed at the Fask service. On the other hand, after the INSERT operation (see Figure 2.8), performed during cluster registration in the platform, all the required data (e.g., cpu cores and ram) are available at cluster level and can be retrieved with a SELECT query.

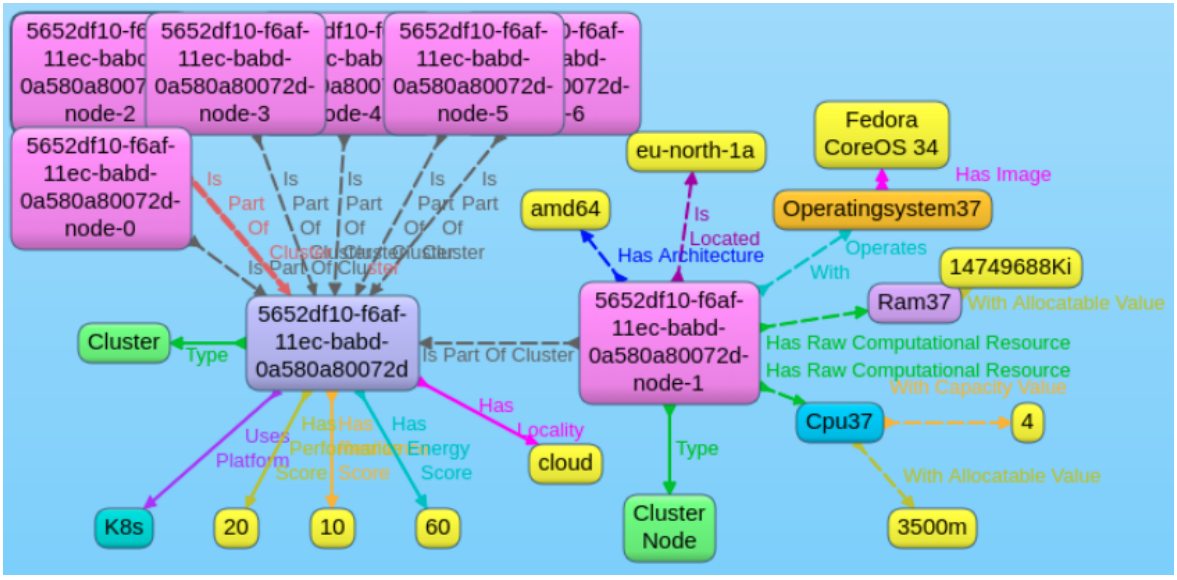


Figure 2.7: Resource Graph without INSERT queries.

Fig 2.9 illustrates an application consisting of an orchestrating flow that calls three other flows during deployment. Each of them has specific annotations, internal functions, and requirements. The target cluster should fulfill all this information for successful application deployment. Thus, application and resource matching could be cumbersome, especially for large workflows.

Although AllegroGraph's build-in reasoner automatically infers some types of

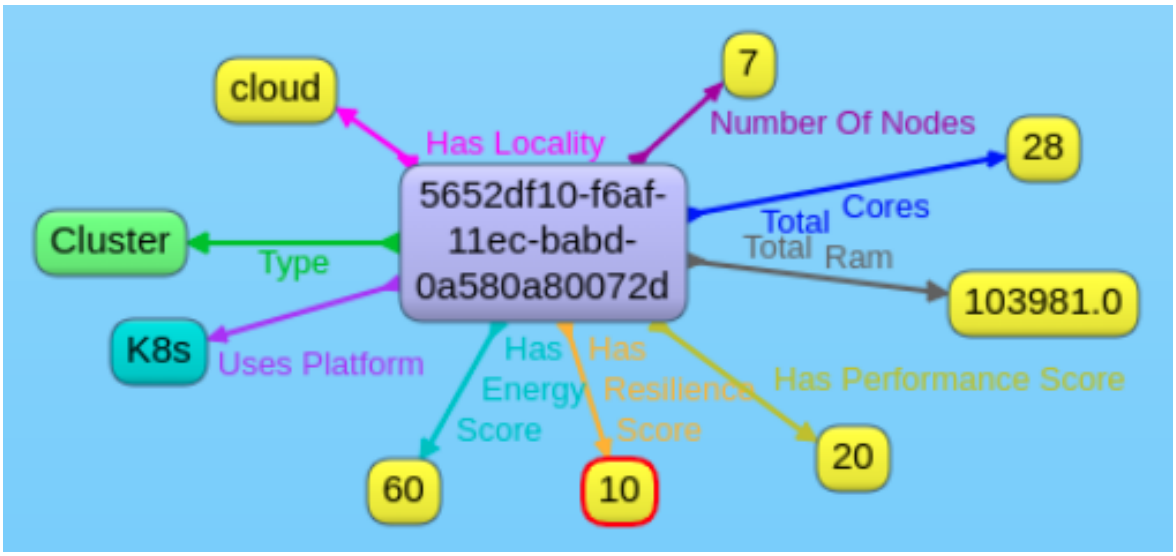


Figure 2.8: Resource Graph after updating it with INSERT queries.

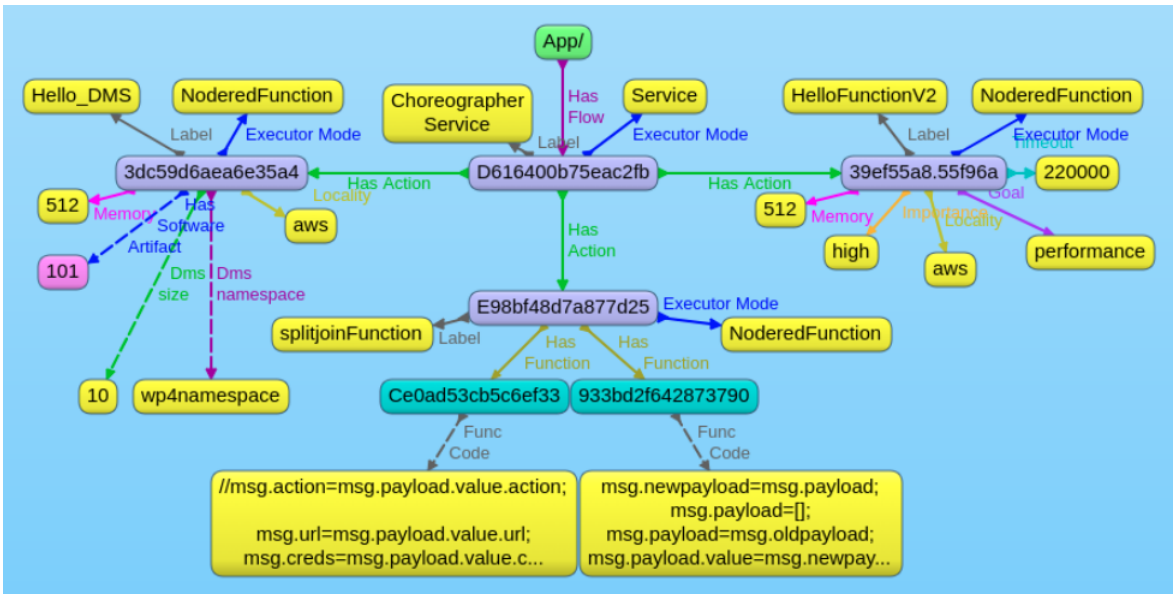


Figure 2.9: Indicative Application Graph

relationships between the nodes of the ingested graphs, they do not suffice to assign connections between application and resource graph. To this end, several rules have been implemented to automate this process and create connections in the form of edges that link each flow with candidate target clusters. These rules enable Reasoning Framework to infer triples of the form ?flow :allocatable

?cluster by comparing the requirements of the input application graph with the characteristics of the available resources. The latter allows retrieving all the information required to deploy an application with a simple SPARQL query that can be executed in a timely manner. Fig 2.10 depicts part of a deployment graph where the flows of the given application have been connected to the target cluster, as indicated by the "Allocatable" edges of the graph.

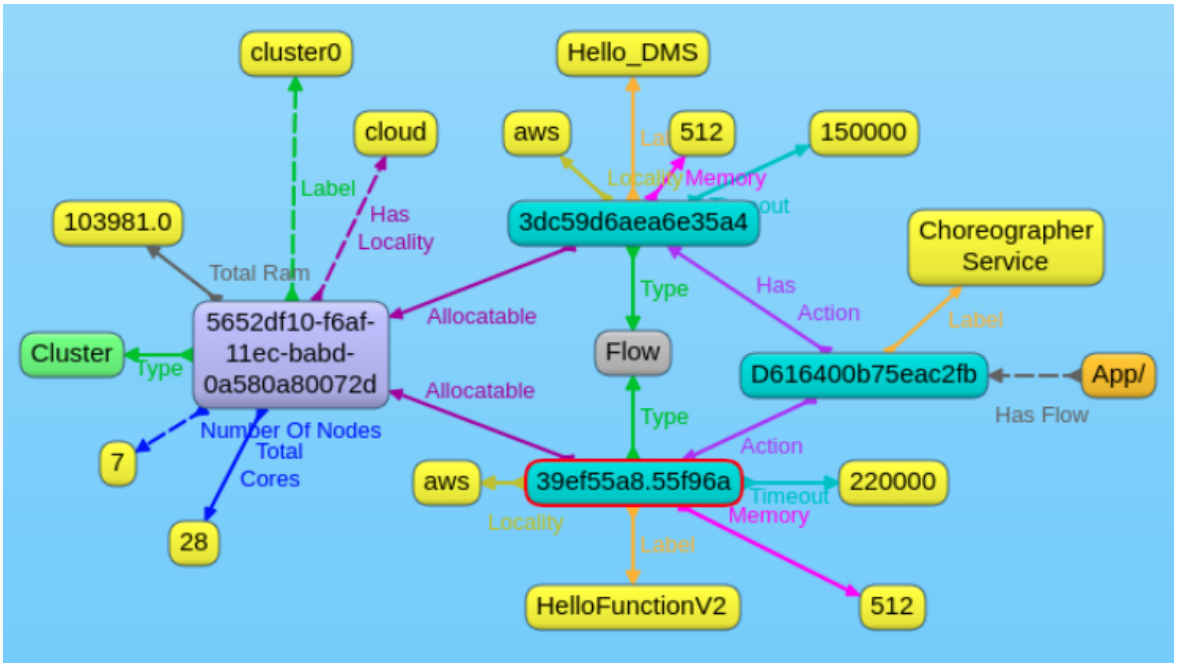


Figure 2.10: Deployment Graph.

2.3.1.5 Reasoning Framework API

While application and resource matching is Reasoning Framework's primary task, it also provides specific data stored in its KB to other components in an asynchronous manner. Platform components responsible for the placement of the input application request data such as the built image location, parameters, and several annotations of each flow/function that are then used in typical Kubernetes and FaaS platform manifest files. Table 2.3 describes the REST endpoints currently used the most from the platform.

Table 2.3: Reasoning Framework API Description

ID	Method	Endpoint[a]	Scope
EP1	POST	/cluster	Store a new cluster
EP2	GET	/cluster	Specs of stored clusters
EP3	POST	/app	Store a new app
EP4	GET	/app	Descriptions of all stored apps
EP5	GET	/app/run/a_id	Workflow allocations and manifest

[a]Only the most used endpoints are presented.

Table 2.4: Reasoning Framework's Evaluation in Cloud Computing Use Case

Endpoint	Baseline Time (ms)	Optimized Time (ms)	Improvement (%)
POST /cluster	68.99	100.45	-45.59
GET /cluster	28.32	6.58	76.76
POST /app	155.06	179.20	-15.57
GET /app	26.81	21.74	18.91
GET /app/run/id	166.42	110.71	33.48

2.3.1.6 Evaluation and Results

To evaluate the performance of the Reasoning Framework, a series of experiments were conducted to measure the response time for different Application Programming Interface (API) endpoints. These experiments compared the baseline setup, which involves complex SPARQL queries and data manipulations, with an optimized setup that leverages pre-inserted semantic rules. The results, as shown in Table 2.4, indicate that the optimized setup significantly reduces query latency, particularly for GET requests, which are critical for real-time resource allocation in cloud environments.

The experimental results demonstrate that the Reasoning Framework is effective in facilitating the deployment of FaaS applications in hybrid cloud settings, reducing the complexity and time required for resource matching and selection. By leveraging knowledge graphs and ontologies, the framework enhances the ability to manage and deploy cloud applications in a manner that is both efficient and scalable.

2.3.2 Use Case 2: Macroeconomic Analysis with RAG Approaches

2.3.2.1 Problem Context

Macroeconomic data plays a crucial role in financial analysis, serving as the backbone for understanding broad economic trends that directly impact markets, investment strategies, and portfolio management [Wang, 2022]. Key economic indicators, such as inflation rates, unemployment figures, central bank policies as well as geopolitical tensions need to be monitored continuously to make informed predictions and adjust strategies accordingly [Maitra, 2023]. However, the vast and unstructured nature of data sources—ranging from financial reports to real-time news—makes it challenging to extract relevant insights efficiently.

Traditional rule-based reasoning systems often fall short in such dynamic environments [Sun et al., 2024]. These systems rely on predefined rules and ontologies, which, while effective for structured data, lack the flexibility to adapt to rapidly changing economic conditions and the nuances found in unstructured text from various sources. Moreover, analyzing macroeconomic data typically requires significant human effort due to the varying frequencies at which this data is updated [Sijabat, 2022]. For instance, while models based on price data can be processed systematically on a daily basis, those relying on macroeconomic data necessitate thorough analysis across various aspects such as asset classes (e.g., stocks, bonds, commodities), regions, and industries, each of which experiences irregular updates and requires different levels of expertise [Fatouros et al., 2024a]. This inconsistency in data frequency leads to investment strategies that are either solely focused on macro data or exclude it entirely in favor of more advanced strategies based on price dynamics [Ciocîrlan et al., 2023]. The rigidity of rule-based systems hinders their ability to effectively identify emerging economic themes or adapt to shifts in market sentiment in a timely manner.

A Reasoning Engine based on LLMs, grounded by factual and updated data through RAG, offers a more robust solution for these scenarios. LLMs can process and understand vast amounts of text, identifying complex patterns and relationships within the data that rule-based systems might miss [Fatouros et al., 2023d]. When combined with RAG, which ensures that the reasoning is informed by the most current and relevant information, this approach enables the generation of structured, actionable insights that can be seamlessly consumed by other financial services or used to highlight specific economic themes to end users [Hivarhizov, 2024]. This dynamic and responsive system is better suited to meet the demands of modern financial analysis, where timely and accurate information is paramount.

2.3.2.2 Data Ingestion and Preprocessing

The data ingestion and preprocessing phase is a critical component of the macroeconomic analysis pipeline with the overall process illustated in Figure 2.11. It begins with a set of dedicated web scrapers, each tailored to specific organizations such as the Federal Reserve (FED), European Central Bank (ECB), JPMorgan, Goldman Sachs, and BlackRock, among others. These scrapers are designed to download publicly available reports and articles in PDF format from the websites of these organizations, based on a specified date range.

Once the PDFs are retrieved, the system extracts essential metadata, including the report's date, title, publisher, and the URL link to the report. This metadata is crucial for proper citation and ensures traceability of the sources used in subsequent analyses. The PDFs are then parsed using the PDFReader tool to extract the textual content.

Given that websites may include a mix of content, such as marketing materials or non-macroeconomic articles, a filtering step is performed using GPT-4o,

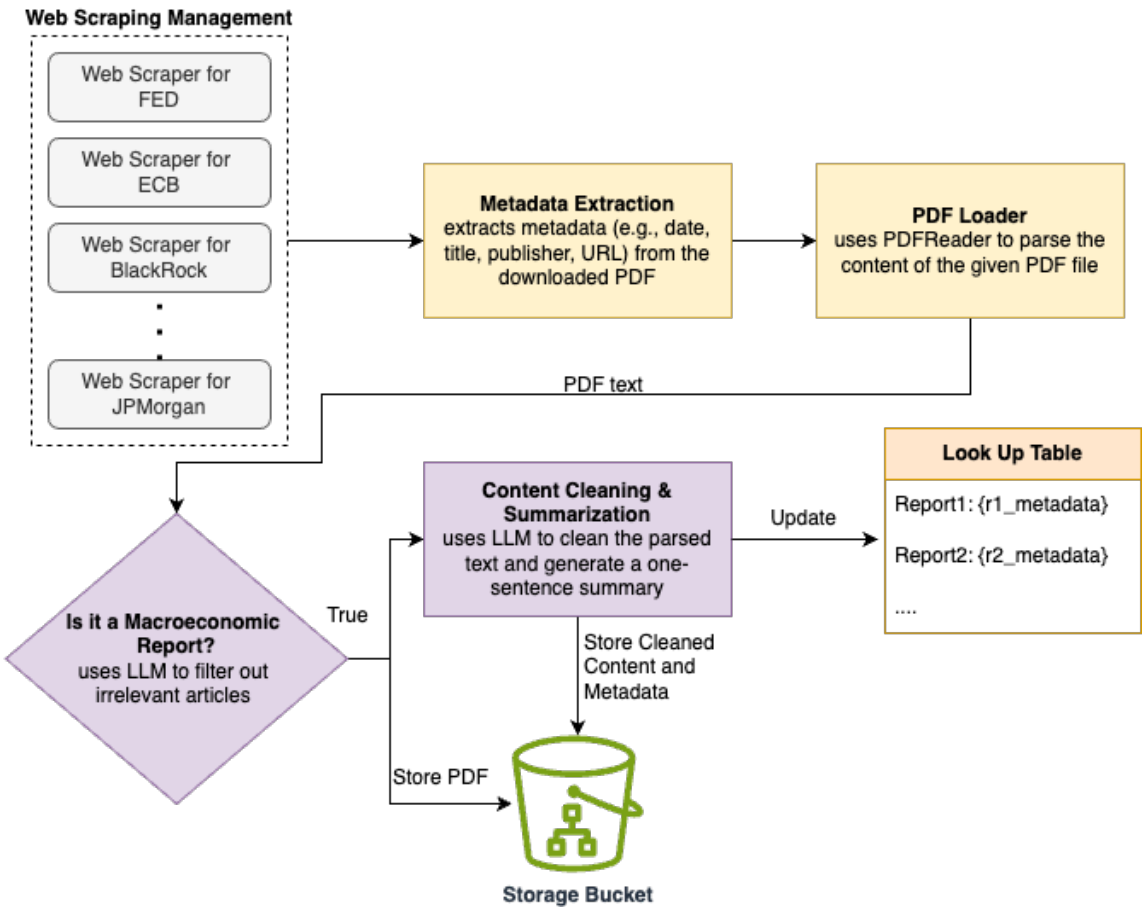


Figure 2.11: Data Ingestion and Preprocessing Workflow of the Reasoning Framework in the finance use case.

one of the most advanced LLMs currently available [OpenAI, 2024]. This step ensures that only relevant macroeconomic reports are retained for further processing. The filtered reports are then cleaned using the LLM, which removes extraneous text and refines the content. Additionally, the LLM generates a one-sentence summary for each report, which is appended to the metadata to provide a quick overview of the report's focus and facilitate better indexing.

The processed PDFs are stored in a cloud storage bucket, and a lookup table is updated with the metadata, including the location of each report in the bucket. Furthermore, for each report, a JSON file containing all the metadata and the cleaned content is generated and stored as a separate file in the bucket.

This structured storage approach, combined with comprehensive metadata indexing, ensures efficient and accurate retrieval of relevant reports during the subsequent analysis phase. By focusing on core, clean data for indexing, the system enhances both the accuracy and speed of retrieval operations.

2.3.2.3 Application of the Reasoning Framework

The application of the Reasoning Framework in the financial use case is centered around the implementation of the RAG method, designed to synthesize comprehensive macroeconomic analyses from a vast repository of financial reports and articles. This approach leverages the structured metadata and cleaned content extracted during the data ingestion and preprocessing phase (described in Section 2.3.2.2) to generate up-to-date, factually grounded insights that are crucial for financial decision-making. This approach, illustrated in Figure 2.12, integrates data ingestion, vector-based retrieval, and advanced reasoning to create accurate and actionable financial insights.

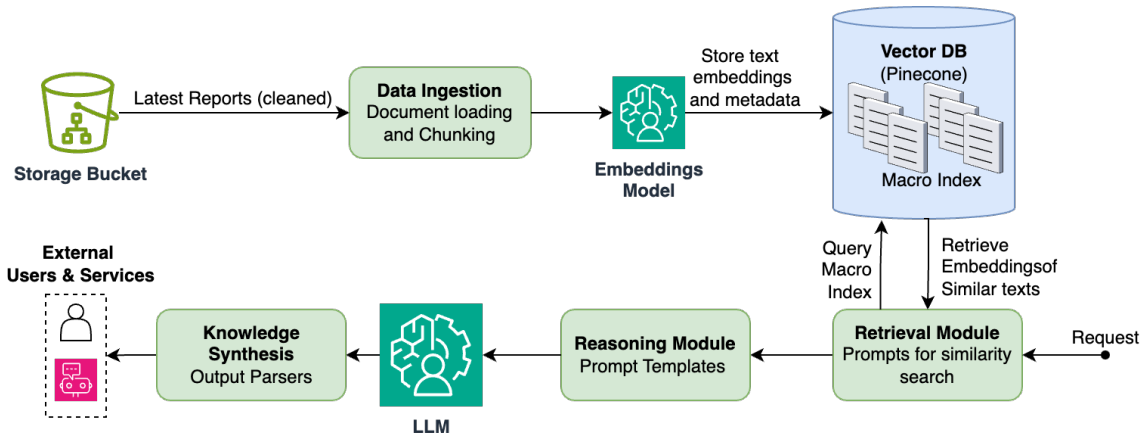


Figure 2.12: Reasoning Framework Implementation in the finance use case.

After the data ingestion and preprocessing phase, the cleaned content and metadata are retrieved using the lookup table. The processed documents are then embedded using OpenAI's embedding model and stored in a Vector Database (i.e., Pinecone). The metadata plays a crucial role in indexing these embed-

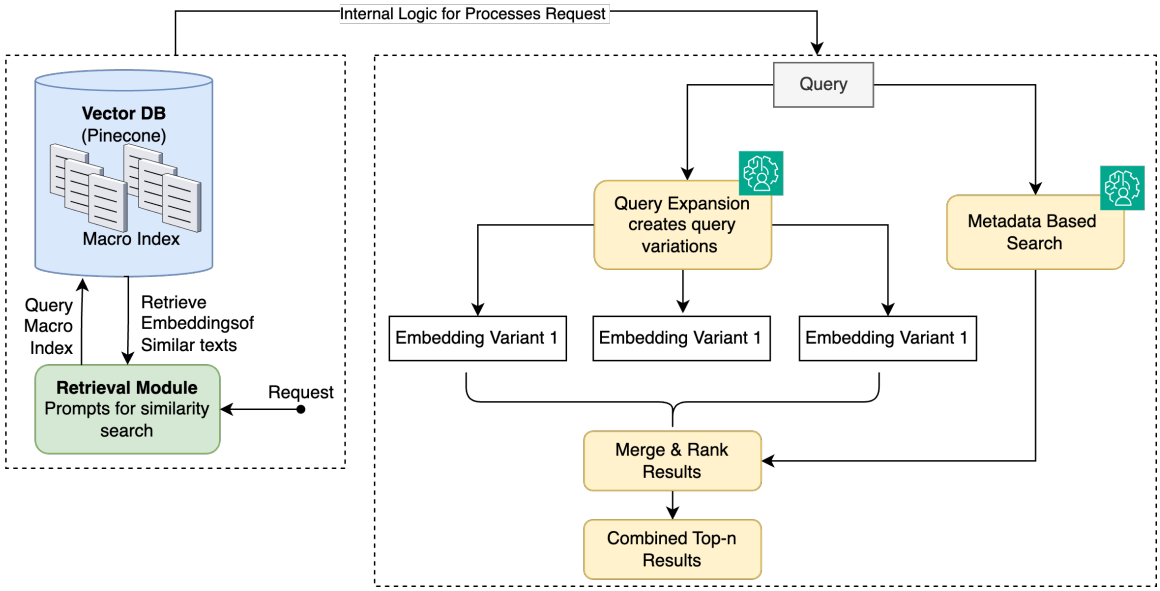


Figure 2.13: Reasoning Framework's Retrieval Module Framework as a RAG Agent.

dings, enabling fast, accurate, and time-aware retrieval based on text similarity searches.

The retrieval module illustrated in Figure 2.13 demonstrates the implementation of an advanced RAG agent, designed to support reasoning with complete and grounded information. The RAG agent upon request/query implements the following functions:

- **Query Expansion**: This step generates multiple variations of the input query using semantic enhancement, via a dedicated request to the LLM, to improve recall and identify broader relevance within the vector database.
- **Metadata-Based Search**: In parallel, metadata-driven filters refine the scope of the retrieval by leveraging date, source, or other document attributes, ensuring precision in the search results while reducing the search space. The required filters are generated by the LLM based on the input query.
- **Embedding Variants**: Different variations of the input query are to en-

sure coverage in information retrieval via similarity search.

- **Top-n results**: The original chunk texts of the to most similar embeddings are combined and then passed as context to the reasoning engine for downstream tasks.

For the Embedding Variants generation, the retrieval module uses pre-defined prompt templates. These prompts are tailored to address specific financial analysis tasks and ensure that the retrieved data is highly relevant for the reasoning process. Below is a description of the prompts used in this process:

- **Consensus Analysis Prompts**: Request comprehensive macroeconomic analyses for specific regions or markets (e.g., United States, Europe, China). Each market is addressed with a unique prompt.
- **Sentiment Analysis Prompts**: These prompts identify positive and negative sentiments regarding investment opportunities and risks.
- **Contradictory Views Prompts**: These prompts are formulated to detect and summarize any conflicting or differing viewpoints among the available macroeconomic reports.

The Reasoning Module utilizes retrieved embeddings after applying these prompts to the indexed documents in the vector database. It requests the LLM, specifically GPT-4o via OpenAI's API (although the modularity of the architecture allows for the integration of different LLMs), to generate a macroeconomic summary for each category (e.g., consensus, sentiment, contradictions) [Achiam et al., 2023]. The LLM, leveraging its generative capabilities, produces outputs that reflect current economic conditions while integrating insights from various sources, offering a comprehensive view of the financial landscape. This approach enables dynamic adaptation to new information, crucial in the rapidly evolving financial markets.

Finally, the Knowledge Synthesis and Output component processes the raw

insights generated by the LLM to ensure they are structured, accurate, and ready for practical use. This component uses tools like Pydantic to validate and enforce the structure of the output, ensuring consistency and reliability. Once validated, the output is formatted into a structured JSON format, capturing key financial insights, such as sentiment analysis, market consensus, and contradictory viewpoints. This structured report is then ready for dissemination or further analysis. It can be automatically posted to external systems, such as a financial service platform or a chatbot, ensuring that the insights are not only well-organized but also immediately actionable for decision-makers. This process ensures that the outputs are both high-quality and fit seamlessly into automated workflows, supporting real-time decision-making in dynamic environments.

This implementation of the RAG method within the Reasoning Framework represents a significant advancement over traditional rule-based systems, offering enhanced flexibility, accuracy, and responsiveness in financial analysis. By grounding the LLM in recent, high-quality data, the framework ensures that the outputs are not only contextually relevant but also actionable, providing valuable insights that can inform investment strategies and other financial decisions.

2.3.2.4 Example Outputs

The Reasoning Framework was applied to generate concise macroeconomic analyses for three consecutive dates: June 29, 2024, July 31, 2024, and August 10, 2024. These outputs were designed to be directly consumed by the MarketSenseAI platform (see Chapter 5), offering the required macroeconomic context for stock picking.

June 29, 2024: The consensus indicated that inflation remains a central concern across major economies. The UK saw inflation fall to 2%, prompting

expectations of a rate cut by the Bank of England, while the ECB was expected to cut rates sooner than the Federal Reserve due to slower growth in the Euro area. Key sectors such as technology and AI were identified as driving short-term returns, with a focus on AI winners like Nvidia. Small- and mid-cap stocks were highlighted as undervalued, offering potential entry points for investors. Geopolitical and economic uncertainties, such as election outcomes, were recognized as significant challenges, recommending diversified portfolios with ample liquidity to navigate potential market dislocations.

July 31, 2024: The consensus analysis revealed ongoing market volatility driven by sentiment shifts rather than fundamental changes. Central banks were at a critical juncture, with potential rate cuts by the Fed and rate hikes by the BOJ leading to currency fluctuations. The U.S. economic outlook presented mixed signals, with optimism about AI-driven growth contrasted by a sluggish housing market. In Europe, economic stagnation was noted, with some regions showing better performance due to political stability and reforms. The investment strategy suggested a diversified and adaptable approach, with small-cap stocks benefiting from easing inflation and expected Fed rate cuts.

August 10, 2024: By this date, the U.S. labor market showed signs of cooling, with rising unemployment possibly leading the Fed to cut interest rates. Inflation was declining in the U.S., positively affecting risky assets but posing a potential market risk if driven by reduced demand. Central banks began cutting rates after significant hikes, with the ECB facing pressure due to high services inflation and weak economic confidence. Geopolitical factors, particularly in China and the Middle East, contributed to market uncertainty. The recommended investment strategy focused on fixed income and selective equity opportunities, favoring high-yield bonds and large-cap equities due to their growth prospects.

These outputs highlight the framework's ability to synthesize complex and

timely macroeconomic information into concise, actionable insights. By grounding the analysis in updated data, the Reasoning Framework ensures that the generated reports are relevant and directly applicable to current market conditions.

Additionally, a different configuration of the Reasoning Framework, with modified instructions in the Reasoning and Output Synthesis Modules but utilizing the same RAG process and data, produces an extended macroeconomic report. This version, available to MarketSenseAI users at https://www.marketsense-ai.com/macro-insights, includes in-depth discussions on sectoral trends, detailed policy analysis, and comprehensive investment outlooks across various regions and asset classes.

2.3.2.5 Evaluation

The evaluation of the Reasoning Framework was conducted by comparing its outputs against those generated by state-of-the-art LLMs with access to real-time data, specifically Llama 3.1 via Perplexity Pro [AI, 2024, Dubey et al., 2024] and Google Gemini 1.5 [Reid et al., 2024]. The evaluation focused on three key areas: the alignment of generated summaries with expert opinions, the quantitative assessment using automated metrics such as BERTScore and ROUGE, and qualitative feedback from a financial analyst reviewing the summaries.

2.3.2.5.1 Automated Metrics: To quantitatively assess the quality of the generated summaries, we employed two widely used metrics in NLP: BERTScore and ROUGE. In selecting BERTScore and ROUGE as the primary evaluation metrics for assessing the quality of generated macroeconomic summaries, this work aimed to capture both semantic similarity and textual overlap, which are crucial for evaluating the effectiveness of the Reasoning Framework. BERTScore, which leverages pre-trained transformer models, excels at measuring the seman-

tic alignment between the generated text and reference outputs by evaluating token similarity in a contextual manner [Zhang et al., 2019]. This makes it particularly suitable for financial text, where nuanced language and domain-specific terminology are prevalent. ROUGE, on the other hand, is a traditional metric focused on n-gram overlap, providing insights into how well the generated summaries capture key phrases and structures found in the reference texts [Lin, 2004].

These metrics were chosen because they complement each other. BERTScore focuses on the depth of understanding and semantic fidelity, while ROUGE assesses more superficial text alignment. Together, they offer a comprehensive evaluation of the generated text's quality. The comparisons yielded the results presented in Table 2.5. BERTScore results indicate that the Reasoning Framework produces summaries that are highly similar to those generated by other state-of-the-art LLMs, demonstrating its effectiveness in capturing relevant macroeconomic insights. The ROUGE scores support the BERTScore findings, confirming that the Reasoning Framework generates content that is on par with the outputs from advanced models in terms of coverage and relevance.

Table 2.5: Reasoning Framework Evaluation in Financial Use Case

Metric	RF vs Perplexity	RF vs Gemini	Perplexity vs Gemini
BERTScore (F1)	0.8374	0.8466	0.8614
ROUGE-1 (F1)	0.4659	0.4577	0.5761

2.3.2.5.2 Human Evaluation: To complement the quantitative metrics, a financial analyst was consulted to review and compare the outputs from the Reasoning Framework with those from Perplexity Pro and Google Gemini 1.5. The analyst highlighted several key strengths of the Reasoning Framework's output, particularly its forward-looking insights, sector-specific analysis, and practical investment strategies.

Key Insights from the Analyst Review:

- The Reasoning Framework (MarketSenseAI) provided forward-looking projections, especially regarding monetary policy, which are crucial for making informed investment decisions.
- The sector-specific analysis, particularly in technology and AI, offered actionable insights that were more detailed and targeted than the other models.
- The investment strategies recommended were deemed more practical and directly applicable, making it the most useful tool for investors and analysts.

The analyst concluded that, while the outputs from Perplexity Pro and Google Gemini 1.5 were robust, the Reasoning Framework's ability to combine macroeconomic insights with practical investment guidance made it superior for real-world financial applications.

2.3.2.5.3 Real-World Application: The RAG-based version of the Reasoning Framework has already integrated into the MarketSenseAI platform, providing actionable macroeconomic insights that are consumed by users for stock picking and investment strategy formulation. The real-world application of these insights has been positively received, with early feedback indicating increased user engagement and satisfaction.

2.4 Discussion

This chapter has introduced the Reasoning Framework, a versatile architecture designed to facilitate semantic matching and reasoning across diverse data types. By supporting the implementation of both traditional knowledge systems, such as ontologies and knowledge bases, with advanced retrieval-

augmented generation (RAG) methods, the framework effectively addresses the complexities of heterogeneous data processing in cloud computing and financial analysis.

One of the key contributions of this work is the seamless integration of traditional semantic reasoning with modern vector-based retrieval techniques. This combination allows the framework to leverage the strengths of both approaches, providing a robust solution for real-time decision-making in dynamic environments. The cloud computing use case demonstrated how the framework can optimize resource allocation for FaaS applications by employing knowledge graphs and ontologies to model and match application requirements with available resources. In the financial use case, the framework's application of RAG methods showed its capability to generate actionable macroeconomic insights, which are crucial for informed investment strategies.

Despite these advancements, several challenges remain. The integration of structured and unstructured data within a unified reasoning framework poses significant computational and scalability issues, particularly when dealing with large-scale, real-time data streams. Additionally, the reliance on current LLMs for semantic reasoning introduces potential limitations related to model bias.

As a direction for future work, there is considerable potential in further integrating ontologies and graph databases with LLMs and RAG approaches. For example, LLMs could be employed to automate data modeling tasks, facilitating the creation and maintenance of complex ontologies. Additionally, leveraging LLMs to enhance retrieval capabilities using SPARQL or other relevant query languages could bridge the gap between structured knowledge systems and unstructured data processing. This integration could lead to more dynamic and adaptable systems capable of more sophisticated reasoning and retrieval, thereby improving the overall performance and applicability of the Reasoning Framework in various domains.

Chapter 3

Online Learning for Time Series Prediction in Dynamic Environments

Chapter Structure

This Chapter is constructed as follows:

- **Section 3.1 - Background on Online Learning for Time Series Prediction**, reviews the necessity of online learning in dynamic environments, such as cloud computing and finance, and examines the current state of the art in this field.
- **Section 3.2 - Overview and Design of the Online Learning System**, presents an overview of the proposed AI-based system, detailing its architecture, components, and how it enables real-time updates to model parameters for time-series data processing.
- **Section 3.3 - Use Cases in Cloud Computing and Finance, and Evaluation**, evaluates the online learning system through two distinct use cases: predicting function latency in serverless cloud computing and financial risk prediction. This section also presents the outcomes of the evaluation, analyzing the system's efficiency and accuracy.

- **Section 3.4 - Discussion**, provides a summary of the chapter's contributions, highlights the challenges encountered during the implementation, and outlines future work directions, including the integration of the Reasoning Framework with online learning systems.

In dynamic environments such as cloud computing and finance, where data patterns evolve rapidly, the ability to continually adapt predictive models in real-time is crucial for maintaining accuracy and relevance. The research presented in this chapter addresses this need by enabling models to update their parameters dynamically as new data becomes available. This chapter introduces a novel AI-based system that applies online learning to time-series data, enhancing the capability to predict outcomes such as function latency in serverless cloud computing and financial risks. The findings from the evaluation of this system demonstrate its effectiveness in improving prediction accuracy and efficiency compared to traditional static models, offering significant advancements in real-time data processing and decision-making. The proposed approach not only supports the dynamic adaptation of models but also integrates seamlessly with existing frameworks, such as the Reasoning Framework discussed in Chapter 2, further solidifying its relevance and applicability in complex, real-world scenarios.

3.1 Background on Online Learning for Time Series Prediction

3.1.1 Introduction to Time Series Prediction

3.1.1.1 Definition and Characteristics of Time Series Data

Time series data refers to a sequence of data points collected or recorded at successive points in time, often at regular intervals. Unlike cross-sectional data,

which captures a single point in time, time series data is inherently temporal, meaning that its observations are dependent on the time at which they are measured. This characteristic introduces specific patterns, such as trends, seasonality, and autocorrelation, which must be considered during analysis.

For example, in financial markets, time series data may represent daily stock prices, interest rates, or trading volumes. In cloud computing, time series data might include system latency, resource usage, or error rates over time. The primary challenge in working with time series data lies in modeling these temporal dependencies accurately to make reliable predictions. Additionally, time series data often exhibits non-stationarity, where the statistical properties of the series change over time, further complicating predictive modeling.

3.1.1.2 Traditional Methods for Time Series Prediction

Traditional methods have been extensively used across various domains to forecast future data points based on historical data. Some widely used approaches include:

- **Autoregressive Integrated Moving Average (ARIMA)**: ARIMA models are widely used for forecasting univariate time series data. They combine three components: autoregression (AR), differencing to make the data stationary (I), and moving averages (MA) to model the residuals [Said and Dickey, 1985]. ARIMA models are particularly effective for series with linear relationships and can be extended to seasonal data using SARIMA (Seasonal ARIMA).

- **Exponential Smoothing (ETS)**: Exponential smoothing techniques, such as Simple Exponential Smoothing (SES) and Holt-Winters, are used for forecasting time series data by assigning exponentially decreasing weights to past observations [Kalekar et al., 2004]. These methods are especially useful for capturing trends and seasonality in data [Gardner Jr, 1985].

- **State-Space Models**: These models represent the time series as a system evolving over time, characterized by hidden states. The Kalman Filter is a well-known example, estimating the state of a dynamic system from noisy observations [Aoki, 2013, Welch et al., 1995].

- **Linear Regression Models**: When the relationship between the dependent variable and time is approximately linear, linear regression can be applied to predict future values. Although simple, they can be extended with polynomial terms or interactions to capture more complex patterns in the data.

3.1.1.3 AI Methods for Time Series Prediction

While traditional methods have proven effective in many contexts, they often struggle with non-linearities, high-dimensional data, and drifts, where the underlying data distribution changes over time. This has led to the adoption of more advanced approaches, particularly ML and DL techniques, which offer greater flexibility and accuracy for a wide range of time series prediction tasks. Key ML/DL methods include:

- **Random Forests**: Random Forests are ensemble learning techniques that combine multiple decision trees to improve predictive accuracy. These methods are particularly effective when dealing with large datasets and complex patterns like in predictive maintenance and Internet of Things (IoT) data [Papacharalampous et al., 2023, Fatouros et al., 2023c].

- **Support Vector Machines (SVM)**: SVMs are a type of supervised learning model that can be used for both classification and regression tasks, including time series prediction [Makridis et al., 2020]. By mapping input data into high-dimensional space, SVMs are able to capture complex relationships that are not easily modeled by linear techniques.

- **Recursive Neural Networks (RNNs) and Long Short-Term Memory (LSTM)**: RNNs are a class of neural networks designed to handle sequential data, making them well-suited for time series prediction [Connor et al., 1994]. LSTMs, a specialized form of RNNs, are designed to capture long-term dependencies in data by using memory cells that store information over extended periods. This makes them highly effective for time series with long-range temporal dependencies [Gallicchio et al., 2018].

- **Probabilistic Deep Neural Network (DNN)**: Probabilistic DNNs like DeepAR and DeepState are specialized models designed for probabilistic time series forecasting [Salinas et al., 2020]. These models output a full predictive distribution rather than point estimates, allowing for uncertainty quantification. These models are particularly useful in scenarios where understanding the uncertainty in predictions is crucial.

3.1.2 Challenges in Time Series Prediction

3.1.2.1 Drift and Non-Stationarity

In time series prediction, one of the major challenges is dealing with concept drift and non-stationarity.

Drift occurs when the statistical properties of the target variable change over time, making it difficult for traditional models to maintain accuracy [You et al., 2021]. Concept drift can be gradual, abrupt, or recurring. Formally, if a predictive model $f : X \rightarrow Y$ is defined, drift occurs when the joint distribution $P(X,Y)$ changes over time:

$$P_{t_1}(X,Y) \neq P_{t_2}(X,Y) \quad \text{for } t_1 \neq t_2$$

where X represents the input features, and Y represents the target variable.

Non-Stationarity is when the statistical properties of a time series, such as mean, variance, and autocorrelation, are not constant over time. Many traditional time series models, including ARIMA and ETS, assume stationarity. A time series Y_t is stationary if:

$$\mathbb{E}[Y_t] = \mu \quad \text{(constant mean)}$$

$$\text{Var}(Y_t) = \sigma^2 \quad \text{(constant variance)}$$

$$\text{Cov}(Y_t, Y_{t-k}) = \gamma_k \quad \text{(autocovariance depending only on lag } k\text{)}$$

Non-stationarity can be detected using tests such as the Augmented Dickey-Fuller (ADF) test [Chen, 2022b], where the null hypothesis H_0 is that the series is non-stationary:

$$\Delta Y_t = \alpha + \beta t + \gamma Y_{t-1} + \delta \sum_{i=1}^{p} \Delta Y_{t-i} + \varepsilon_t$$

where ΔY_t is the difference operator, and ε_t is the white noise.

Addressing Drift and Non-Stationarity: Tackling these issues is crucial for robust time series models. For instance, ARIMA models use differencing to remove non-stationarity:

$$Y'_t = Y_t - Y_{t-1}$$

This transformation stabilizes the mean by removing trends and seasonality.

Monotonic transformations, like the logarithmic transformation,

$$Y'_t = \log(Y_t)$$

or the square root transformation,

$$Y'_t = \sqrt{Y_t}$$

can stabilize variance, particularly in data with exponential growth or quadratic trends [Cheng et al., 2015].

Online learning methods continuously update model parameters, making them responsive to drift. For example, in a linear regression model:

$$Y_t = \beta_0 + \beta_1 X_t + \varepsilon_t$$

parameters β_0 and β_1 can be incrementally updated as new data arrives, allowing the model to adapt to evolving data distributions.

Advanced techniques like LSTM networks are capable of capturing and adapting to temporal dependencies, making them resilient to both non-stationarity and drift [Karaahmetoglu et al., 2020].

These challenges highlight the need for online learning techniques that can adapt to changing environments, ensuring models remain accurate and reliable over time.

3.1.2.2 High Dimensionality and Noise

Time series prediction often involves high-dimensional data and noise, both of which can complicate model accuracy and efficiency.

High-dimensional time series data features numerous variables evolving over time. This complexity increases the risk of overfitting, as models might capture noise instead of underlying patterns [Dong et al., 2020]. Dimensionality reduction techniques, like Principal Components Analysis (PCA) or autoencoders, are commonly used to transform high-dimensional data into a lower-dimensional space that retains essential information [Sun and Bozdogan, 2020].

Noise refers to random fluctuations that obscure the true underlying signal, making it difficult for models to learn and predict accurately. Noise can result

from measurement errors, external disturbances, or inherent variability. Techniques like smoothing (e.g., moving averages, exponential smoothing), filtering (e.g., Kalman filters), and denoising algorithms help reduce noise's impact on model performance [Rezaei et al., 2022]. Additionally, regularization techniques, such as Lasso or Ridge regression, can manage noise by penalizing overly complex models that might otherwise fit the noise rather than the signal.

Addressing high dimensionality and noise is essential for developing models that generalize well and provide accurate forecasts, particularly in real-world applications where data is often imperfect and high-dimensional.

3.1.2.3 Computational Constraints in Real-Time Environments

Real-time environments impose unique computational challenges for time series prediction, especially with high-frequency data or large-scale systems requiring immediate responses. Several key factors contribute to these constraints:

Latency Requirements: In real-time systems, predictions must be generated within strict time limits to be actionable. For example, in financial markets, even milliseconds can impact the success of a trading strategy. Consequently, models need to be optimized for low-latency inference, often requiring lightweight algorithms or approximate methods that deliver rapid predictions without significant sacrifices in accuracy.

Scalability: Real-time applications must often scale to handle high volumes of data or simultaneous requests. This demands models that not only perform well individually but can also scale horizontally across distributed systems or in parallel processing environments. Online learning algorithms, which update models incrementally as new data arrives, are particularly suited for maintaining real-time performance in scalable systems.

Model Adaptation: Real-time environments are dynamic, with changing conditions that can impact the performance of static models. Online learning and model adaptation are therefore crucial, allowing models to update and refine themselves in response to new data without requiring extensive retraining, which would be computationally prohibitive in real-time settings.

Addressing these computational constraints is essential for the successful deployment of time series prediction models in real-time environments, where the ability to provide timely and accurate predictions is paramount.

3.1.3 Overview of Online Learning

3.1.3.1 Types of Online Learning Approaches

This chapter focuses on the online learning strategies employed in the proposed system, which are particularly relevant to the dynamic environments of cloud computing and financial markets. The primary method utilized is the sliding window approach, which is central to the system's ability to adapt to real-time changes in data streams. The other methods, though less emphasized, complement the sliding window approach by addressing specific challenges such as drift and real-time updates.

- **Sliding Window Method**: This method continuously updates the model using the most recent data within a specified window that moves forward over time. It is particularly effective in environments like serverless cloud computing [Fatouros et al., 2023a], where rapid adaptation to fluctuating conditions, such as changes in function latency, is essential. This method ensures that the model remains up-to-date with the latest data trends, making it highly suitable for dynamic, real-time applications.
- **Incremental Learning**: Complementing the sliding window method, incremental learning updates the model with new data without requiring a

complete retraining. This approach is beneficial when the model needs to maintain an ongoing understanding of evolving trends, especially in scenarios where data flows continuously but with varying patterns.

- **Feature Drift Adaptation**: Managing feature drift—where the statistical properties of input features change over time—is crucial in dynamic sectors like finance [Fatouros et al., 2023b]. The sliding window method inherently supports this by recalibrating the model's focus on the most current data attributes, thereby mitigating the impact of feature drift on prediction accuracy.

- **Real-Time Updating**: The ability to update models in real-time is critical for maintaining predictive accuracy in environments where delays in model training can lead to suboptimal decisions. The sliding window approach facilitates rapid retraining cycles, ensuring that the model reflects the most recent data and remains effective in real-time applications, such as financial risk assessment and cloud resource management.

By emphasizing the sliding window method within the online learning framework, the proposed system ensures that models remain both responsive and accurate in continuously evolving data environments. This approach is particularly well-suited to the use cases discussed in this chapter, where real-time data processing and decision-making are paramount.

3.1.3.2 Fundamentals of Online Learning

Online learning is a ML paradigm where models are incrementally updated as new data becomes available, without the need for complete retraining. This is particularly advantageous in environments characterized by continuous data streams, where computational resources or time constraints make frequent retraining impractical.

Key Characteristics:

- **Incremental Updates**: Online learning algorithms process data sequentially, updating the model one observation or small batch at a time. This contrasts with batch learning, where the model is trained on the entire dataset at once, making online learning more suitable for real-time applications.
- **Adaptability**: These models are designed to adapt to changes over time, making them effective in non-stationary environments where data distributions evolve. This adaptability is crucial for maintaining model accuracy in dynamic settings.
- **Scalability**: By updating incrementally, online learning can efficiently handle large volumes of data, making it scalable for applications where data is too extensive to fit into memory or where the data is generated continuously.
- **Real-Time Prediction**: Online learning is essential for applications requiring real-time analysis and decision-making, as it minimizes the lag between data acquisition and model updating, thus providing timely and accurate predictions.

Challenges:

- **Stability-Plasticity Dilemma**: Online learning models must balance retaining previously learned information (stability) with the ability to integrate new information (plasticity). This balance is critical to prevent catastrophic forgetting, where new data overwrites valuable older knowledge.
- **Noise Sensitivity**: Incremental learning approaches can be more susceptible to noise and outliers since each new data point can significantly influence model parameters. Effective noise management strategies, such

as regularization, are essential to maintain model robustness.

- **Parameter Tuning**: The performance of online learning models is highly dependent on parameters such as learning rates and window sizes. These need careful tuning to achieve optimal performance, particularly in environments where data characteristics are subject to frequent changes.

- **Model Type Trade-off**: There is an inherent trade-off between using traditional time series models like ARIMA, which are quicker to retrain, and more complex deep learning models, which can capture non-linear relationships but require longer retraining times. This trade-off is particularly important in applications where rapid model updating is critical, such as in real-time financial forecasting or adaptive cloud resource allocation.

3.2 Overview and Design of the Online Learning System

The proposed online learning system is designed to adapt to dynamic environments such as cloud computing and financial markets by continuously updating predictive models based on real-time data streams. This system's architecture, illustrated in Figure 3.1, is composed of several key components, each responsible for specific tasks in the data processing, model training/update, and prediction workflow.

3.2.1 Architecture Components

3.2.1.1 Data Monitoring & Ingestion

The Data Monitoring & Ingestion component captures real-time data streams from external sources such as cloud computing environments (e.g., function latencies, resource usage metrics) and financial systems (e.g., asset prices, trading volumes). This component can be implemented using REST APIs or messaging

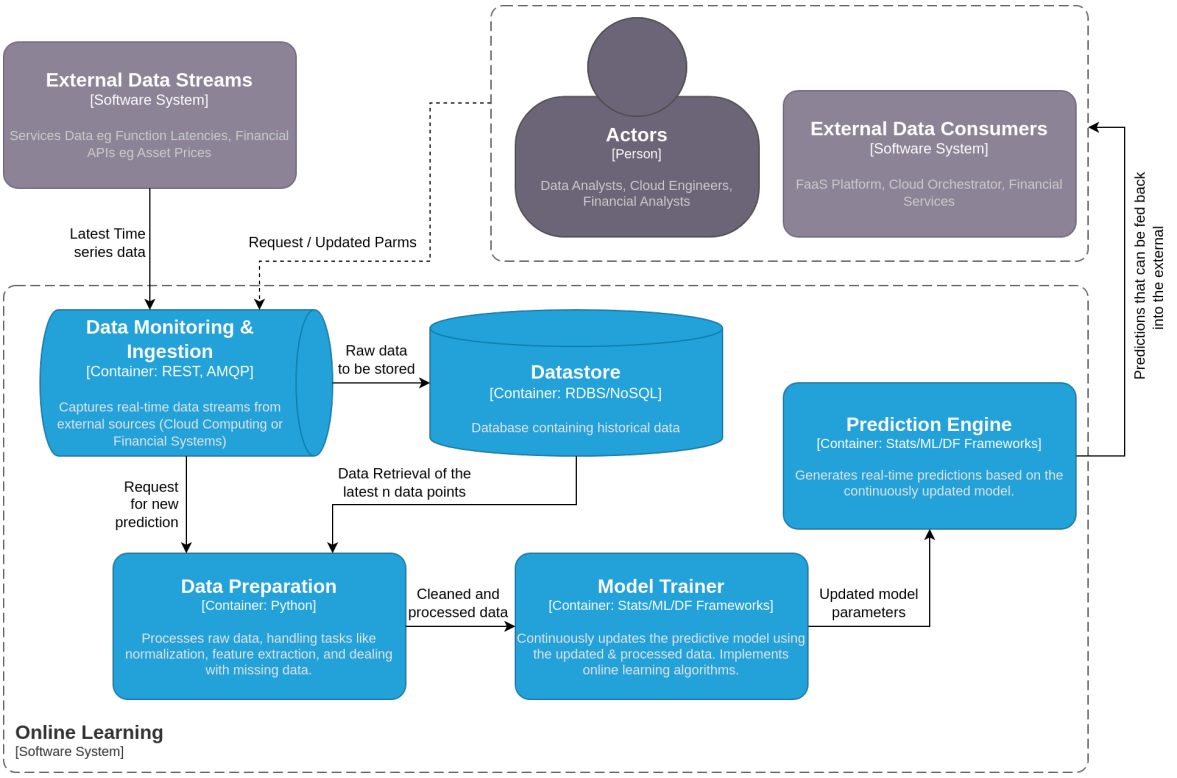


Figure 3.1: Architecture of the Online Learning System.

protocols to ensure reliable and efficient data ingestion. The raw data ingested is then sent to the Datastore and triggers the Data Preparation component when new data is available.

3.2.1.2 Datastore

The Datastore serves as a central repository for both historical and newly ingested data, supporting model training and validation processes. It can be implemented using relational (RDBMS) or NoSQL databases, depending on the system's scalability and performance requirements. The stored data is accessed by the Data Preparation component for processing and by the Model Trainer for training or updating the predictive models.

3.2.1.3 Data Preparation

The Data Preparation component is responsible for processing the raw data collected from various sources. This includes tasks such as data cleaning, normalization, feature extraction, and handling of missing values. These preprocessing steps ensure that the data is in the optimal format for model training and prediction. Once the data is prepared, it is passed to the Model Trainer for further processing.

3.2.1.4 Model Trainer

The Model Trainer component continuously updates the predictive model using the processed data from the Data Preparation component. Depending on the specific requirements and limitations of the use case, the model could be a traditional time series model or a machine learning/deep learning (ML/DL) model. For example, in cases where the available historical data covers only a short period, or when efficiency is prioritized over accuracy, a traditional model may be more appropriate. The parameters of the model should be predefined and validated through an offline process before deployment. This component is crucial for maintaining the model's accuracy and relevance in dynamic environments.

3.2.1.5 Prediction Engine

The Prediction Engine generates real-time predictions using the continuously updated model provided by the Model Trainer. This component plays a vital role in delivering actionable insights to external data consumers, such as cloud orchestrators or financial services, which rely on accurate and timely predictions for decision-making. Essentially, the Prediction Engine performs the inference task based on the most current model to provide predictions that can be used immediately by external systems.

3.2.2 System Workflow and Component Interactions

The system workflow begins with the Data Monitoring & Ingestion component capturing real-time data streams from external sources. This data is then stored in the Datastore and triggers the Data Preparation component, where it undergoes preprocessing. The processed data is used by the Model Trainer to update the predictive model, which is then utilized by the Prediction Engine to generate real-time predictions. These predictions are consumed by external systems that depend on timely and accurate data-driven insights.

3.3 Use Cases in Cloud Computing and Finance, and Evaluation

3.3.1 Use Case 1: Runtime Adaptation for Serverless Functions in Hybrid Clouds

3.3.1.1 Problem Context

Serverless computing, or FaaS, has rapidly gained popularity due to its advantages like automatic scalability, reduced operational overhead, and micro-billing. In this model, developers focus on application logic while cloud providers manage resources, dynamically scaling functions based on demand. However, the rise of serverless computing, especially in hybrid cloud environments where functions are deployed across private and public clouds (e.g., AWS, Azure), introduces challenges in optimizing performance metrics such as response latency [Shahrad et al., 2019].

In hybrid cloud setups, ensuring optimal latency is complex. Traditional static routing methods struggle to adapt to the dynamic nature of serverless functions, which can experience variability due to cold starts, resource contention, and

cloud-specific factors [Kousiouris and Pnevmatikakis, 2023]. Industries like manufacturing, eHealth, and smart agriculture, which use proprietary servers for security and cost reasons, often supplement with public cloud resources for resilience [Aryotejo et al., 2018]. This hybrid model demands sophisticated routing mechanisms that can dynamically balance performance and cost, something traditional approaches fail to provide [De Palma et al., 2020].

This proposed system addresses these challenges by introducing a dynamic routing service for hybrid cloud environments. This system leverages real-time metrics—such as function invocations, memory usage, and latency data—to intelligently route serverless functions across cloud clusters, optimizing for both performance and cost efficiency. Unlike static methods, this system adapts in real time, ensuring better resource utilization and improved response times.

The solution has been evaluated using Apache OpenWhisk clusters on AWS and Azure, demonstrating significant improvements in performance and cost management under various workloads, thereby validating the effectiveness of this dynamic routing approach in hybrid cloud environments [Fatouros et al., 2022].

3.3.1.2 Related Work

The challenge of understanding and predicting performance in serverless computing, has attracted significant research attention due to the dynamic and complex nature of FaaS deployments and their performance metrics.

3.3.1.2.1 Performance Estimation in FaaS: Recent studies have proposed various analytical methods for estimating the performance of FaaS platforms by utilizing different evaluation metrics. SAAF [Cordingly et al., 2020] provides a comprehensive set of performance profiling and resource utilization metrics for concurrent FaaS workloads, enabling accurate predictions of FaaS function

runtimes. Another approach, presented by [Mahmoudi and Khazaei, 2022], introduces a data-driven model that forecasts the average response time of functions on the KNative FaaS platform, though its evaluation is limited to a single server environment. Addressing the performance impact of concurrency, [Zafeiropoulos et al., 2022] propose autoscaling mechanisms for serverless applications using reinforcement learning techniques, tested in both real and simulated settings on the Kubeless serverless platform. The issue of handling concurrency, particularly in clusters with limited capacity, is crucial as it significantly influences both the quality of service and the operational costs of FaaS platforms [Lipitakis et al., 2023]. Moreover, the balancing of requests to optimize between wait and execution times offers multiple pathways to enhance cluster performance [Kousiouris and Pnevmatikakis, 2023].

In the realm of function orchestration, MLFaaS [Paraskevoulakou and Kyriazis, 2023] introduces a method for deploying machine learning pipelines as workflows of linked functions, determining the optimal number of functions based on individual response times. Additionally, [Lin and Khazaei, 2020] propose a framework for modeling the end-to-end latency of function workflows, using AWS Lambda metrics and graph analysis to predict and optimize both performance and cost. SLAM [Safaryan et al., 2022] focuses on estimating function execution times to recommend the best memory configurations for serverless applications, aligning with specified service level objectives and user-defined goals.

3.3.1.2.2 Implications of Bursting Invocations and Time Series Forecasting: The impact of bursting function invocations on serverless platforms has been explored by [Shahrad et al., 2019], who highlight its effect on wait times, particularly when multiple functions are invoked simultaneously on a worker node. [Qiu et al., 2021] conducted production measurements on IBM Cloud and a private cloud, examining how workload consolidation affects wait, initial-

ization, and execution latencies. Their findings indicate that function duration is correlated not only with the selected memory but also with the number of concurrent containers. Inspired by these insights, [Jegannathan et al., 2022] propose time series forecasting techniques to anticipate such performance fluctuations. Faa$T [Romero et al., 2021], for instance, leverages temporal patterns in invocation frequency to reduce cold starts, particularly for infrequently invoked applications on Azure Functions.

3.3.1.3 System Design

The primary objective of the proposed adaptive routing service is to dynamically route serverless function requests within hybrid environments, optimizing for user-defined goals such as latency or cost. This section outlines the system's design, detailing the interactions among its key components.

3.3.1.3.1 Overall Architecture: The system is built around three core components, as shown in Figure 3.2:

1. **Monitor**: Continuously tracks performance metrics across registered clusters, including average initialization, wait, and execution times, and triggers alarms when thresholds are exceeded.

2. **Forecaster**: Utilizes recent metrics to predict function latency across clusters. Based on user objectives, it calculates the optimal distribution of requests to minimize latency or cost [Amazon Web Services, 2022]).

3. **Router**: Distributes incoming function requests according to the routing strategy generated by the Forecaster.

These components work together to adaptively manage routing in response to the dynamic nature of the environment as depicted in Figure 3.2.

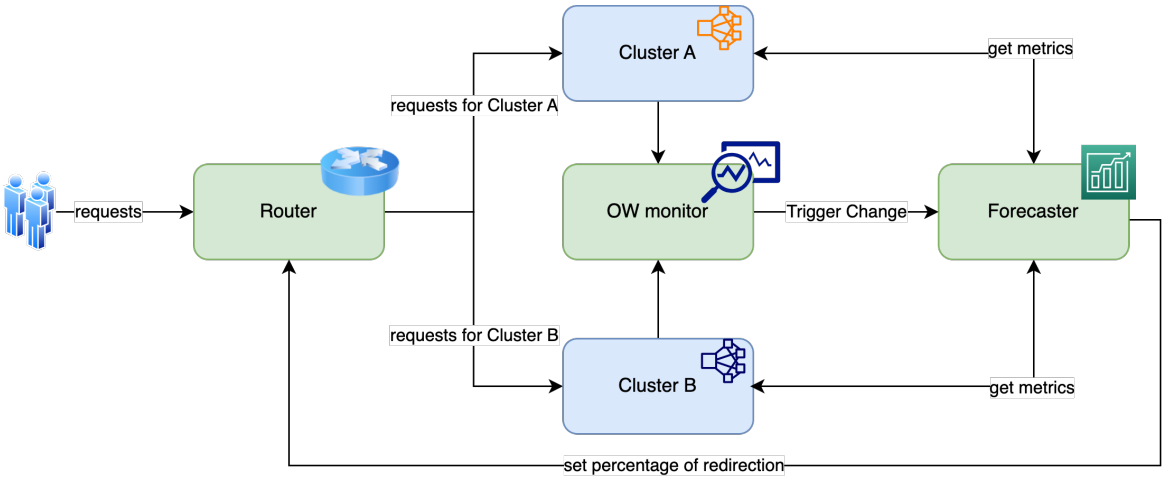


Figure 3.2: Conceptual Architecture of the Adaptive Routing Service.

3.3.1.3.2 The Monitor tracks key performance metrics across clusters, such as the number of cold starts, success rates, and averages of wait, initialization, and execution times over a configurable time window. Users can set both the time interval for calculating averages (e.g., the last 10 minutes) and the polling frequency. The Monitor can be configured to trigger alarms or invoke the Forecaster when metrics exceed predefined thresholds [Kousiouris et al., 2023].

3.3.1.3.3 The Forecaster implemented as a REST API using Flask and Gunicorn [Chesneau, 2017], retrieves performance data from OpenWhisk metrics APIs across clusters. It processes this data, focusing on a recent 15-minute window to align with typical FaaS container lifespans. Forecaster's internal architecture and components are illustrated in Figure 3.3.

Data is preprocessed to retain relevant fields such as activation ID, start and end times, wait time, duration, and memory usage. This data is then aggregated by minute, allowing for a detailed analysis. Unlike the Monitor, the Forecaster performs its own data preprocessing, enabling more sophisticated analysis.

The Forecaster employs an exponential smoothing model [Hyndman and Athana-

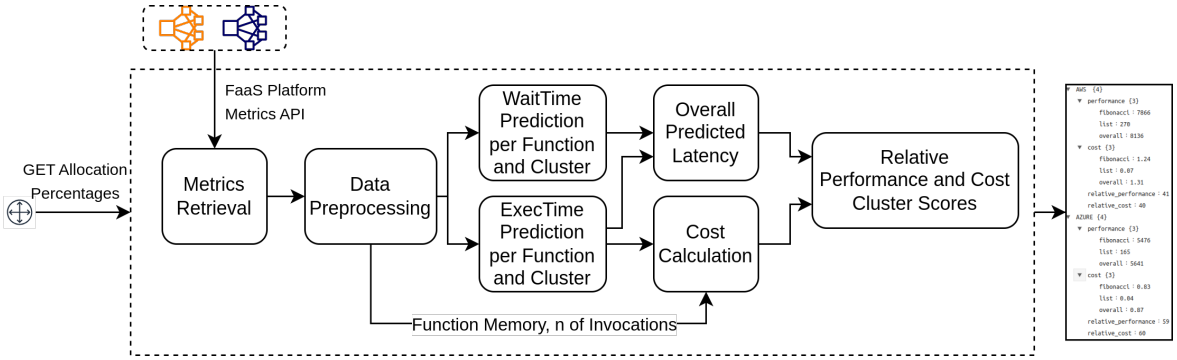


Figure 3.3: Forecaster's internal Architecture and Components.

sopoulos, 2014] to predict execution and wait times for each function. These predictions are used to calculate performance scores (total predicted latency) and cost estimates (based on invocation frequency and memory usage).

Relative scores for each cluster are computed using a reciprocal transformation of the original metrics, normalizing them for comparison across clusters using Equation 3.1. These scores guide the Router in making informed routing decisions, balancing performance and cost considerations.

$$\text{relative\_scores}[c][m] = \frac{\text{inverted\_metric}[c][m]}{\sum_{c' \in C} \text{inverted\_metric}[c'][m]} \tag{3.1}$$

The Forecaster transmits these relative scores as a JSON object to the Router, which uses them to dynamically adjust routing.

3.3.1.3.4 The Router dynamically routes incoming requests based on the Forecaster's insights. It includes a REST endpoint for applying new routing percentages in real-time. If the primary server becomes unavailable, the Router defaults to forwarding all requests to the secondary cluster.

3.3.1.4 Evaluation Setup

3.3.1.4.1 Server and Cluster Configuration: The adaptive routing service is composed of the Router and the Monitor, which are implemented as pattern subflows on a NodeRED server, with the Forecaster hosted within a Docker container. These components communicate via HTTP nodes. The system runs on a server located in Athens, Greece, equipped with 4 vCPUs and 8GB of RAM.

The function invocations are handled by two distinct OpenWhisk clusters, hosted on AWS (primary) and Azure (secondary, in a hybrid cloud context). The configurations of these clusters are compared in Table 3.1.

Table 3.1: Cluster Configuration Comparison

	AWS	**AZURE**
Container Memory	8GB	2GB
Location	Sweden	Netherlands
Orchestrator	Kubernetes	Kubernetes

3.3.1.4.2 Function Implementation: Two Python-based functions with varying computational demands were used for evaluation: a Fibonacci number calculator and a function that generates a large list. The Fibonacci function calculates the n-th Fibonacci number, while the List function creates a list with a specified length. For testing, the Fibonacci function was set with n=40, and the List function generated a list of 10 million elements.

3.3.1.4.3 Experiments: The experimental setups were designed to evaluate both performance and cost-efficiency of the adaptive routing service.

Performance-based Routing

1. Deploy the Fibonacci function with an initial routing strategy favoring AWS (100-0) and a 10-minute monitoring window, polled every 30 seconds.

2. Modify the monitoring window to 5 minutes, maintaining the same initial routing strategy as in the first experiment.
3. Conduct a baseline experiment with the Fibonacci function using a static 50-50 routing strategy.
4. Deploy both the Fibonacci and List functions with a 100-0 routing strategy and a 10-minute monitoring window.
5. Establish a baseline with both functions using a fixed 50-50 routing strategy.

Cost-based Routing

1. Evaluate cost implications using both functions with a 100-0 initial routing strategy and a 10-minute monitoring window. Costs were calculated using rates from AWS ($0.0000166667 per GB-second and $0.2 per 1M requests) and Azure ($0.000016 per GB-second and $0.2 per 1M requests). Results were compared against the performance-based routing baseline (setup five).

Alarms are triggered when the overall wait time in a cluster exceeds 3 seconds, prompting the Forecaster to reassess the routing strategy.

3.3.1.4.4 Request Rate: To maintain consistency across all experimental scenarios, a steady request rate of 10 invocations per minute per function was used, allowing for a controlled evaluation of the system's performance.

3.3.1.5 Results

3.3.1.5.1 Average Latencies Analysis: The experimental outcomes, depicted in Fig 3.4, provide insight into the performance metrics derived from various routing strategies during the tests with Fibonacci and combined (Fibonacci and List) functions. It is evident that scenarios involving both functions generally

yield lower average latencies compared to those relying solely on the Fibonacci function. This is largely due to the lower execution latency of the List function, which positively affects the overall average latency in the combined setup.

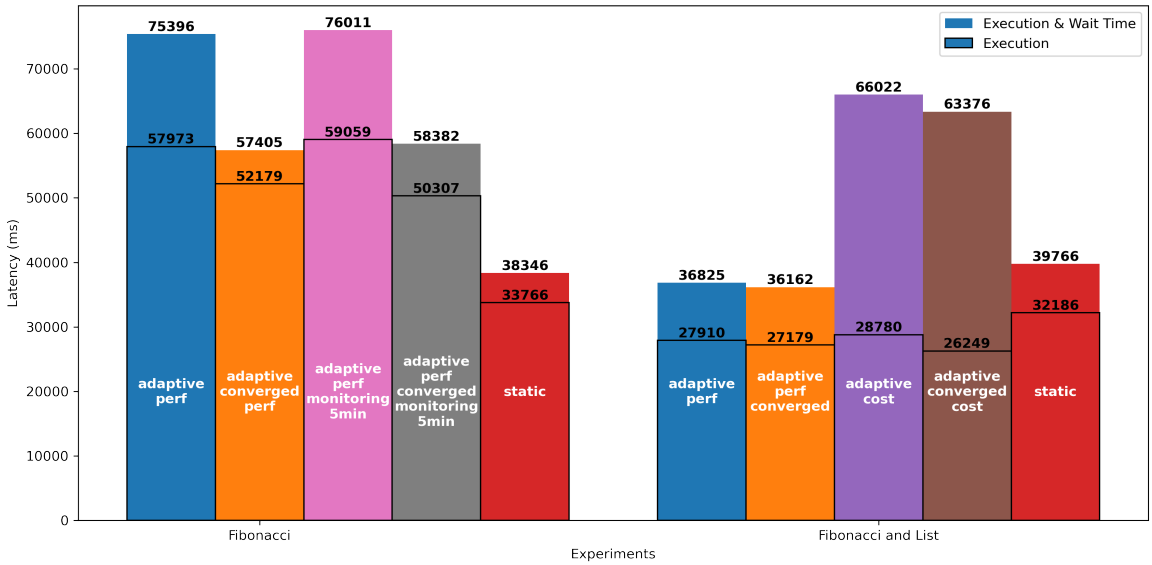


Figure 3.4: Mean Response and Execution Latency between different experiments.

For the Fibonacci function deployment, the performance-based adaptive strategy, starting with a 100-0 routing, initially encountered high latencies, with average response and execution times of 75396 ms and 57973 ms, respectively. However, a significant reduction in both latency and execution time was observed upon convergence. A reduced monitoring window of 5 minutes resulted in similar latencies but facilitated a 30% faster convergence compared to the 10-minute window interval, as shown by comparing Figure 3.5 and 3.6. The static 50-50 routing strategy exhibited lower initial latencies but lacked the adaptive optimization benefits (Figure 3.7).

In the combined deployment of the Fibonacci and List functions, the performance-based adaptive strategy also showed higher initial latencies (36825 ms response latency and 27910 ms execution time). However, it demonstrated substantial improvements upon convergence, achieving 36162 ms and 27179 ms in re-

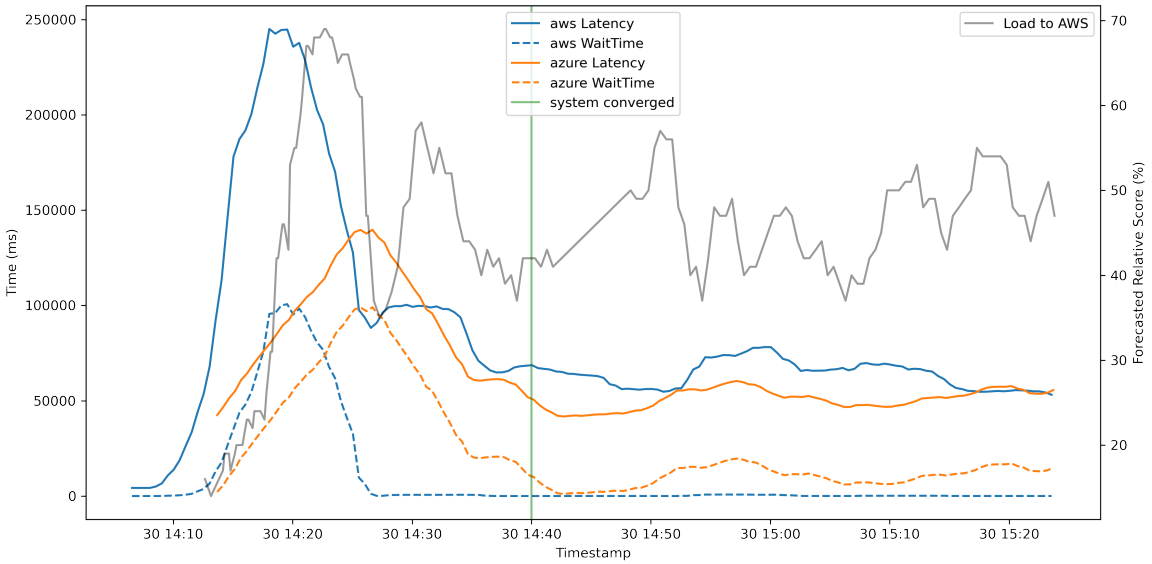


Figure 3.5: Time Series of wait and response latency with Fibonacci function, performance-based adaptation, and 10-minute monitoring pooling.

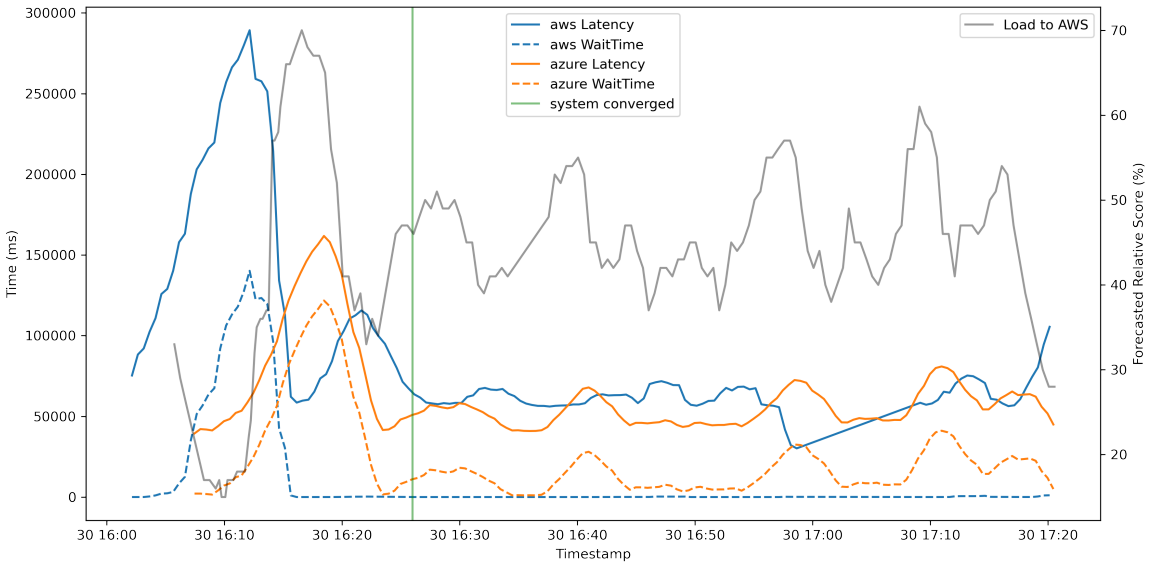


Figure 3.6: Time Series of wait and response latency with Fibonacci function, performance-based adaptation, and 5-minute monitoring pooling.

sponse latency and execution time, respectively—9% lower latency compared to the static routing (Figure 3.8 and 3.9). The cost-based adaptive approach initially showed latencies of 66022 ms and execution times of 28780 ms, which

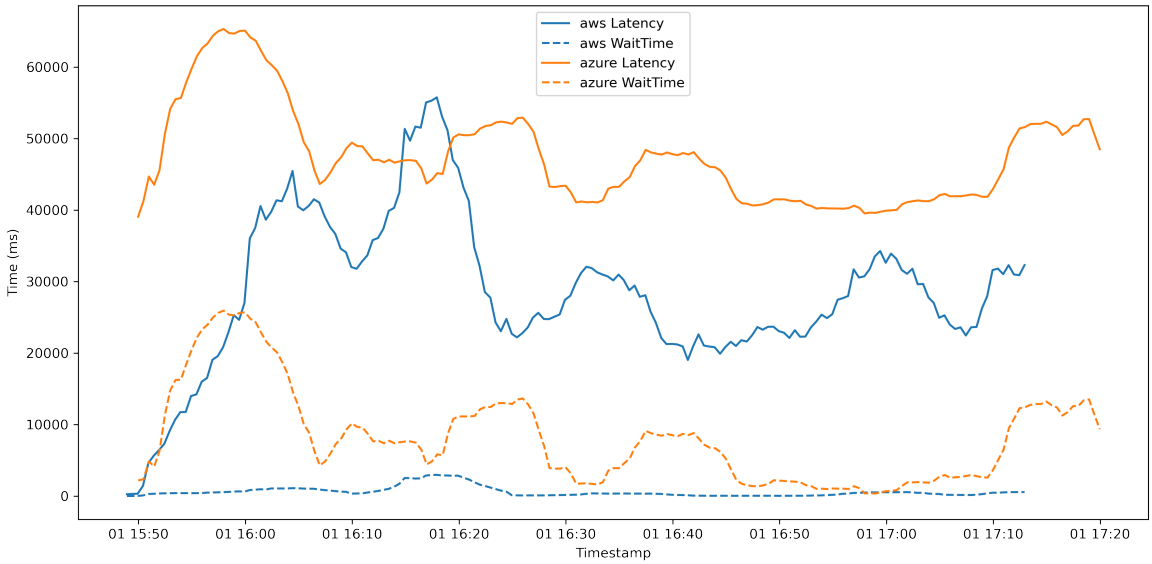


Figure 3.7: Time Series of wait and response latency with Fibonacci function and 50-50 routing policy.

improved to 63376 ms and 26249 ms post-convergence, highlighting the balance between performance and cost-efficiency (Figure 3.10).

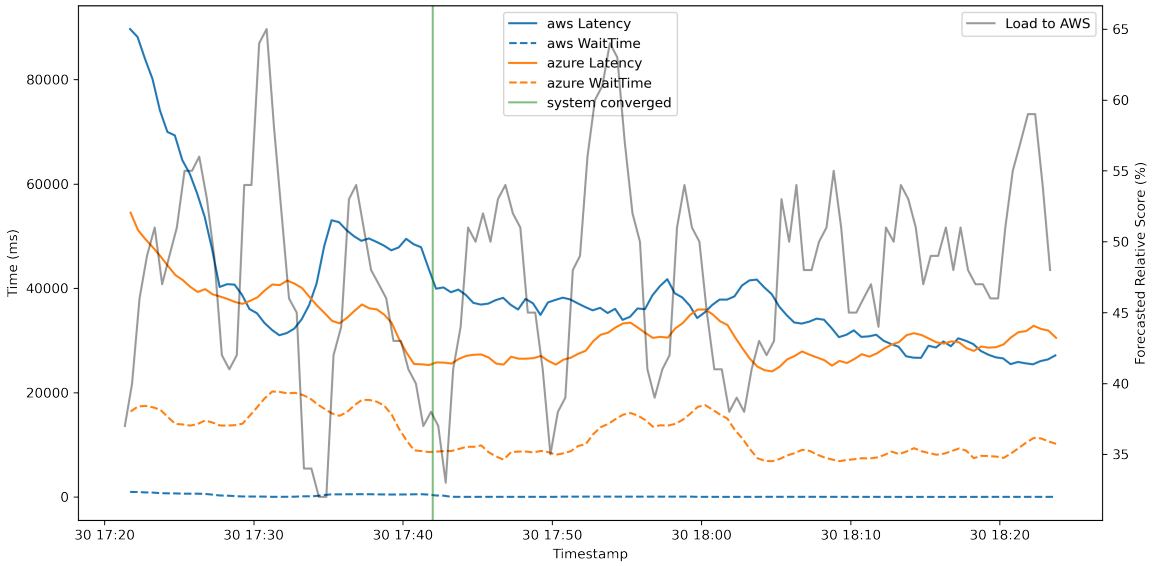


Figure 3.8: Time Series of wait and response latency with Fibonacci & List functions, performance-based adaptation, and 10-minute monitoring pooling.

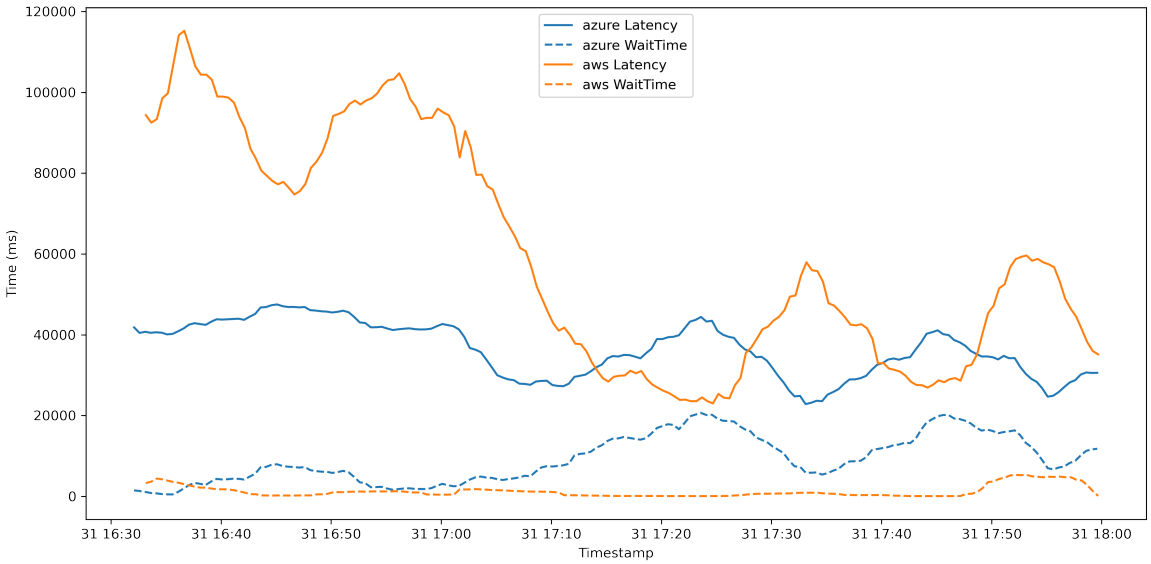


Figure 3.9: Time Series of wait and response latency with Fibonacci & List functions and 50-50 policy.

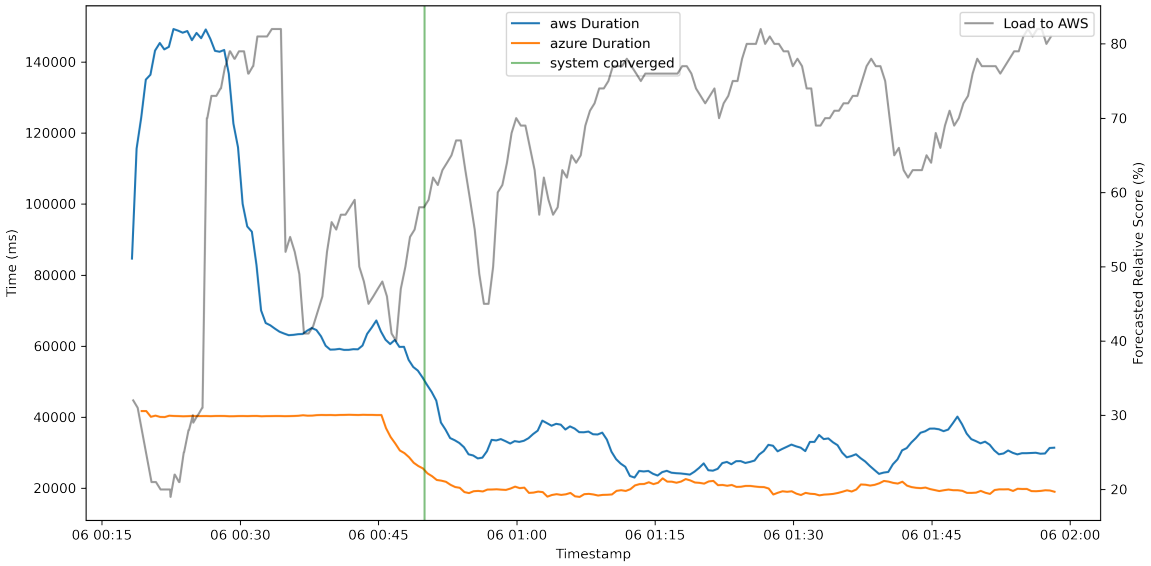


Figure 3.10: Time Series of execution latency with Fibonacci & List functions, cost-based adaptation, and 10-minute monitoring pooling.

3.3.1.5.2 Variance Analysis: Standard Deviation (STD) provides a measure of the variability in performance metrics, which is crucial for applications that require consistency. As noted in [Kousiouris and Pnevmatikakis, 2023], sta-

bility in execution times is particularly important given the FaaS cost model's dependence on execution time. The results, illustrated in Fig 3.11, reveal the standard deviation of response latency across different setups.

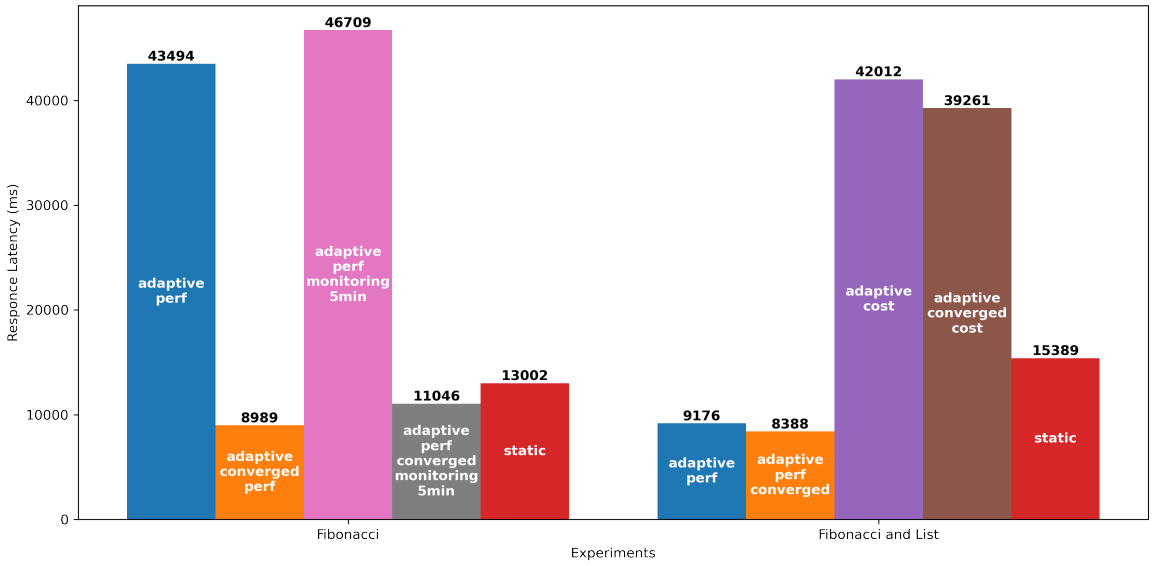


Figure 3.11: Standard Deviation of Response Latency between different experiments.

For the Fibonacci function deployment:

- The adaptive strategy initially showed high variability with a response latency STD of 43494 ms, which reduced significantly to 8989 ms after convergence.
- The 5-minute monitoring window adaptive approach followed a similar pattern, tapering to an STD of 11046 ms upon convergence.
- The static strategy exhibited approximately 30% higher latency variability compared to the converged version of the first setup.

For the combined Fibonacci and List functions deployment:

- The adaptive strategy had an STD of 9176 ms (response latency), which reduced slightly to 8388 ms upon convergence.

- The cost-optimized adaptive strategy exhibited a higher STD overall, as it prioritized execution time over latency. However, it achieved the lowest STD in execution time, reducing to 7868 ms upon convergence.
- The static strategy showed around 45% higher variability in latency compared to the converged performance-optimized adaptive setup.

3.3.1.5.3 Cost Analysis: Table 3.2 presents the cost analysis for each experimental setup, indicating both initial and converged costs. Costs are calculated based on the mean cost per minute, derived from the average execution time and the number of requests processed per minute in each cluster, assuming a function memory of 512Mb.

Table 3.2: Cost per Experiment

Experimental Setup	Cost[a]	Cost-Converged[a]
Fibonacci & List, Performance Opt	2.12	2.05
Fibonacci & List, Cost Opt	2.81	1.86
Fibonacci & List, 50-50	2.25	2.25

[a]Mean cost per minute in dollars, based on the average execution time and number of requests in each cluster, for 512Mb function memory.

The results indicate that the adaptive system, particularly in the cost-optimized setup, achieves significant cost reductions, lowering the mean cost per minute by up to 17% post-convergence. This underscores the adaptive system's ability to enhance performance while maintaining cost efficiency. Moreover, the adaptive system was successfully integrated and validated within the PHYSICS Cloud Platform, where it played a crucial role in dynamically updating resource data within the Reasoning Framework presented in the previous Chapter 2. This integration effectively retriggered the optimization process for function placement, ensuring that resource allocation decisions remained optimal and responsive to real-time data, further improving the overall efficiency and effectiveness of the cloud platform.

3.3.2 Use Case 2: Portfolio Risk Assessment leveraging Probabilistic DNNs

3.3.2.1 Problem Context

Risk assessment in the financial sector has increasingly relied on Value at Risk (VaR) models. However, traditional VaR models have shown limitations during periods of economic turmoil, such as the 2008 financial crisis and the 2020 COVID-19 pandemic, where they failed to accurately measure financial risk [De Waal et al., 2013]. Following the introduction of the Basel II Accord, VaR became crucial for determining capital requirements and risk assessment, driving institutions to seek more accurate methods to monitor and assess risk exposure [Elsinger et al., 2006, Zhao, 2020].

Despite its widespread use, VaR has faced significant criticism for its simplistic assumptions, particularly during crises when its predictions can be unreliable [Abad et al., 2014]. Critics have likened VaR to "an airbag that works all the time, except when you have a car accident" [Einhorn and Brown, 2008]. VaR represents the maximum expected loss of a portfolio over a given time horizon at a predefined confidence level, and several variations of VaR have emerged, including non-parametric, parametric, and semi-parametric models.

Non-parametric methods like Historical Simulation (HS) use past portfolio returns to estimate VaR but may fail to capture unprecedented market fluctuations [Chang et al., 2003]. Parametric models, such as the Variance-Covariance method and various GARCH variants, require assumptions about the distribution of returns, which often do not hold true for financial time series [Abad et al., 2014]. Semi-parametric models, such as those based on Monte Carlo simulations, offer a middle ground by incorporating elements from both approaches and are particularly effective for complex portfolios [Abad et al., 2014].

However, these traditional methods face challenges, especially during market downturns when portfolios may experience losses exceeding the predicted VaR. This issue is exacerbated by dependencies between VaR predictions, particularly at high confidence levels like 99%, and the presence of fat tails in financial data distributions [Angelidis and Degiannakis, 2018, Yamai et al., 2002]. Furthermore, most existing studies focus on single-asset portfolios, which may not adequately represent the complexities of diversified portfolios.

The 2008 financial crisis, resulting in a global loss of $3.4 trillion [Dattels and Miyajima, 2009], and the ongoing challenges posed by the COVID-19 pandemic underscore the need for innovative VaR prediction methodologies. In response, this research introduces a data-driven framework for predicting portfolios' VaR, addressing the dynamic nature of financial data with several key innovations:

- **Continuous Learning**: The framework continuously updates model parameters to reflect the latest market data, reducing the likelihood of clustered VaR violations.

- **Probabilistic Forecasting**: By utilizing auto-regressive recurrent neural networks, the model captures rare market events with minimal training time.

- **Portfolio-Level Analysis**: The framework extends beyond single-asset analysis to evaluate entire portfolios in near real-time, eliminating the need for frequent retraining.

While the framework is primarily evaluated on FX portfolios, its applicability extends to other financial instruments and time horizons. The evaluation demonstrates its effectiveness using various loss functions and coverage tests, suggesting that the approach can be optimized for different types of time-series data.

3.3.2.2 Related Work

The concept of Value at Risk (VaR) as a financial risk assessment tool was introduced by J.P. Morgan and Reuters in 1996 under the "RiskMetrics model" [Longerstaey and Spencer, 1996]. Despite its widespread adoption, this parametric method is criticized for relying on assumptions like the normality and independence of financial returns, which are often unrealistic.

Given these limitations, the financial sector has explored alternative approaches to VaR estimation. Research comparing various VaR methods highlights that unconditional models often lead to clustered VaR violations, although some models remain acceptable depending on the window size used for historical data [Kuester et al., 2006]. Conditional VaR models, while potentially more volatile, have also been scrutinized for their practical implications in capital allocation. Much of the focus has been on the Generalized AutoRegressive Conditional Heteroskedasticity (GARCH) family of models, with some studies suggesting that asymmetric GARCH models might offer marginally better performance, although the differences are not statistically significant [Abad et al., 2014]. Another approach that has shown promise is based on Extreme Value Theory (EVT), which is particularly effective at forecasting extreme losses during periods of financial stress. EVT-based methodologies like the Peak Over Threshold (POT) and block maxima models have been demonstrated to produce more accurate predictions of extreme losses compared to traditional methods at high confidence levels [Novak, 2011, Mcneil, 1998, Bekiros and Georgoutsos, 2005].

The advent of ML and DL has provided financial firms with advanced tools for improving VaR estimation. Recent studies have applied various deep learning architectures, such as RNNs and LSTM networks, to time-series forecasting, proving effective in capturing non-linear temporal patterns in financial data [Lim and Zohren, 2020, Sen et al., 2019]. For instance, RNN-based

models have been used in portfolio management [Neuneier, 1996, Xiong et al., 2018], while LSTM networks have shown strong potential in predicting stock market movements [Weng et al., 2017]. Additionally, Generative Adversarial Networks (GANs) have been utilized to generate synthetic financial data that closely resembles real market data, further aiding in risk modeling [Goodfellow et al., 2014, Pfenninger et al., 2021]. A combination of wavelet analysis and LSTM networks has also been employed to capture the complex characteristics of financial time series, such as non-linearity and non-stationarity [Yan and Ouyang, 2018].

Despite the advances in ML/DL, challenges remain in VaR estimation. Two major issues identified in the literature are the excess losses associated with VaR violations and the tendency of most VaR 99% models to produce more violations than the expected confidence level. The first issue highlights that even though a 99% risk measure may seem comprehensive, the 1% of events it does not capture can result in significant losses, as argued by [Hendricks, 1996]. The second issue pertains to the limited back-testing of these models regarding the coverage and independence of VaR estimations, particularly in studies focused on univariate financial indices without considering asset correlations within portfolios.

The main contributions of this research address these challenges. First, it proposes a novel probabilistic approach for VaR prediction using the DeepAR model, which provides a full predictive distribution, enabling decision-makers to optimize their risk management strategies. This approach has been shown to outperform the most prevalent VaR estimation methods, particularly at the 99% confidence level. Second, the research introduces evaluation and back-testing procedures that are applied not only to univariate assets but also to diversified portfolios, utilizing a range of evaluation metrics and statistical tests to demonstrate the effectiveness of the proposed approach.

3.3.2.3 System Design

This research proposes an innovative framework for portfolio Value at Risk (VaR) estimation using probabilistic deep neural networks [Mohebali et al., 2020]. The model calculates the VaR for each asset individually (VaR_i) and then derives the portfolio's overall VaR (VaR_p) by accounting for asset correlations and weights. This method has been evaluated using several randomly generated portfolios to ensure robustness across different scenarios.

To assess the proposed model, it was compared against five established VaR methods: GARCH, RiskMetrics (RM), Historical Simulation (HS), Bidirectional Generative Adversarial Network (BiGAN), and Monte Carlo (MC). The evaluation used metrics such as the number of VaR violations and various loss functions.

The approach uses log-returns for both VaR and Profit and Loss (PnL), with VaR being defined mathematically as the percentile of the return distribution that corresponds to a given confidence level [Christoffersen et al., 2001].

The core of the proposed framework is the DeepAR model, a probabilistic forecasting technique based on auto-regressive recurrent neural networks, designed for accurate multivariate time-series modeling [Salinas et al., 2020]. DeepAR models the conditional distribution of time-series data to produce accurate probabilistic forecasts and can be trained on multiple time-series simultaneously, enabling cross-learning among them.

As depicted in Figure 3.12 and further analyzed in terms of sequence flows by Algorithm 1, the DeepVaR framework performs estimations on asset-level (i.e. for a single VaR) at each time step t. Historical market prices x_i from t_0 to $t-1$ of multiple instruments i are ingested into the framework simultaneously. During the data preprocessing step the input data are initially resampled to match the frequency of the selected VaR time horizon and then are transformed

to log-returns ($r_{i,t}$). The latter is used to train the DeepAR model and to estimate the distribution of each time series (i.e. asset-level) for time step t. With the distribution of the assets' returns available, $VaR_{i,t}$ can be obtained from Eq. 3.2. In the last step, portfolio-level predictions, i.e. portfolio $VaR_{p,t}$ is estimated based on the returns variance-covariance matrix, $VaR_{i,t}$ and the input weight on each asset. It is also noted that the process of calculating VaR_p from VaR_i requires only matrices multiplications (see Eq. 3.3). Thus, no training is required and therefore the overall process is quite time-efficient. Thus, DeepVaR could be also used for what-if analysis comparing portfolios' risk against the different weights on input assets/instruments.

$$VaR^a = q_{1-a} \tag{3.2}$$

$$VaR_p^\alpha = \sqrt{VRV^T} \tag{3.3}$$

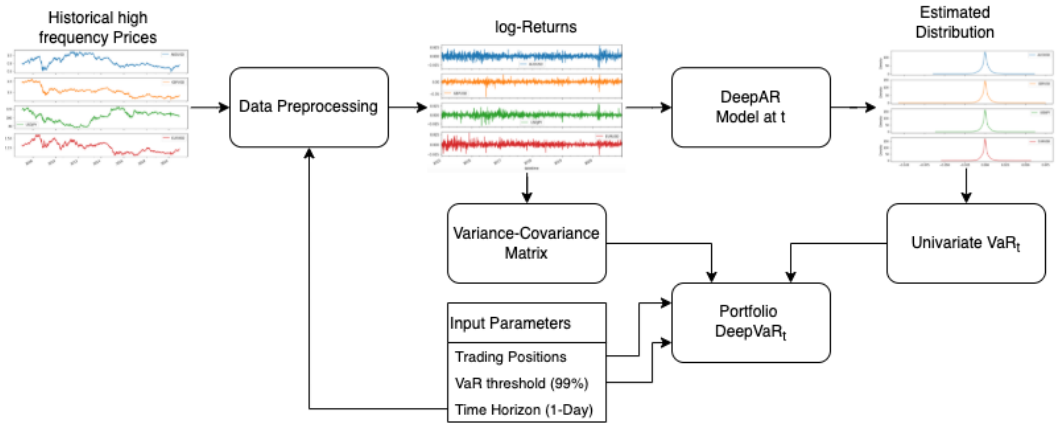


Figure 3.12: Conceptual architecture of DeepVaR framework.

DeepVaR, the implementation of DeepAR in this framework, adapts to the dynamic nature of financial data by incorporating a continuous learning approach. Unlike traditional ML pipelines that require extensive initial training, DeepVaR is retrained on the latest market data, mitigating model bias, serial correlation

Algorithm 1 VaR prediction using the DeepVaR framework

Input:
    X: Historical Prices $\in R^{T\times N}$
    w: Portfolio weights $\in R^{1\times N}$
    $freq$: Time horizon of VaR
    α: VaR confidence probability
Output:
    VaR_p^α: Portfolio VaR estimation
Data Preprocessing
$train\_ds \Leftarrow$ returns[-900 :, :] *(training dataset)*
$R \Leftarrow \text{COV}(train\_ds[-125:,:])$ *(variance covariance matrix)*
Model training and forecasting
$model \Leftarrow \text{DEEPARESTIMATOR}(predictionLength = 1,$
$contextLength = 15,$
$freq = freq,$
$numLayers = 2,$
$dropoutRrate = 0.1,$
$cellType = "lstm",$
$numCells = 50,$
$trainer = Trainer(epochs = 5, lr = 0.0001, numBatchesPerEpoch = 50))$
$estimator \Leftarrow \text{TRAIN}(model, train\_ds)$
$pred \Leftarrow \text{PREDICT}(estimator, num\_samples = 1000)$
Calculate portfolio VaR_p^α
$lower\_q \Leftarrow \text{QUANTILE}(pred, q = 1-\alpha, axis = 0)$
$upper\_q \Leftarrow \text{QUANTILE}(pred, q = \alpha, axis = 0)$
$V \Leftarrow \text{ARRAY}([1,4])$ *(initialize empty array)*
for i in $\text{RANGE}(N)$ **do**
    if $w[i] < 0$ **then**
        $V[i] \Leftarrow w[i] \times lower\_q[i]$
    else
        $V[i] \Leftarrow w[i] \times upper\_q[i]$
    end if
end for
$VaR_p^\alpha \Leftarrow -\text{SQRT}(V \times R \times V^T)$

between VaR estimations, and clustered VaR violations [Mehrabi et al., 2021]. The model is optimized for rapid retraining, making it suitable for real-time or intra-day VaR estimation.

3.3.2.4 Evaluation Setup

The proposed DeepVaR framework was evaluated against several baseline VaR estimation techniques. These include the GARCH model, known for capturing time-varying volatility in financial time-series [So and Philip, 2006]; the RiskMetrics model, a variant of GARCH [Longerstaey and Spencer, 1996]; the Historical Simulation (HS) method, which calculates VaR based on past observations; the Monte Carlo (MC) method, which estimates future returns distributions by generating random samples; and the BiGAN, which models the joint probability distribution of returns using adversarial networks [Donahue et al., 2016].

3.3.2.4.1 Dataset Description: The evaluation was conducted using daily close prices of four highly liquid foreign exchange (FX) instruments (AUDUSD, GBPUSD, USDJPY, EURUSD) from 2007 to 2020. The data was transformed into log-returns, and the VaR predictions were measured on this scale. The performance of the models was assessed using a rolling window prediction format over a test period from 2018 to 2020 as shown in Figure 3.13.

To further validate the results, 1000 randomly generated portfolios reflecting different trader behaviors were analyzed. These portfolios included both long and short positions, with asset allocation expressed as varying proportions of the four FX instruments.

3.3.2.4.2 Evaluation Metrics: The evaluation of the proposed approach employed several metrics to assess its performance. These included the number of VaR violations, violation rate, quadratic loss, smooth loss, tick loss, and firm loss, each highlighting different aspects of the models' predictive capabilities. Additionally, the validity of the VaR forecasts was tested using the Christoffersen conditional coverage test and the Dynamic Quantile (DQ) test [Engle and Manganelli, 2004], which check for the independence and correct coverage of the

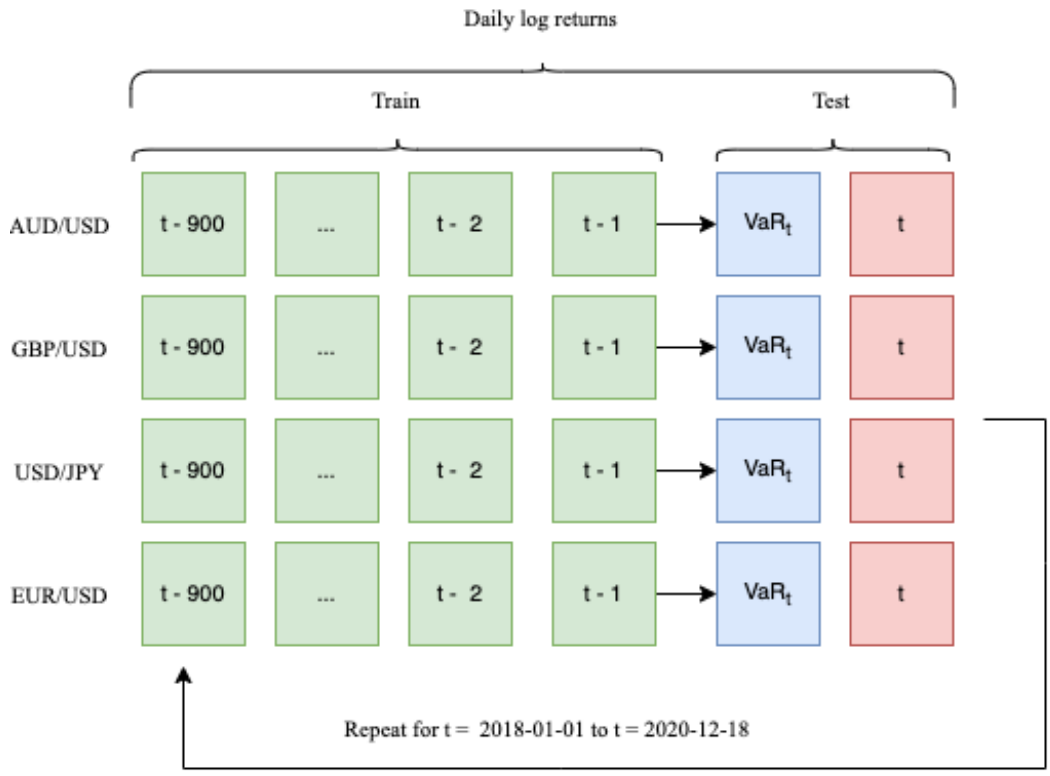


Figure 3.13: Training dataset for rolling window VaR estimation.

VaR predictions.

3.3.2.5 Results

3.3.2.5.1 Framework's Latency: Table 3.3 summarizes the required mean time in seconds per model to obtain the quantiles (e.g., q_1, q_{99}) needed for the VaR estimation of both single and four assets portfolios. The fourth column of the table indicates the relative difference in calculation time between the two different input sizes. According to these findings, it is obvious that despite the fact that deep learning-based models require significantly more time to estimate VaR than the other models, that time ($\approx 12.5s$) is very low, enabling VaR estimation even for intra-day trading applications. Moreover, the input size has a minimal effect on deep learning models' training time which leverage matrices operations to parallelize computations. In contrast, estimation time in

econometric models such as GARCH is linearly dependent on the number of the input time series.

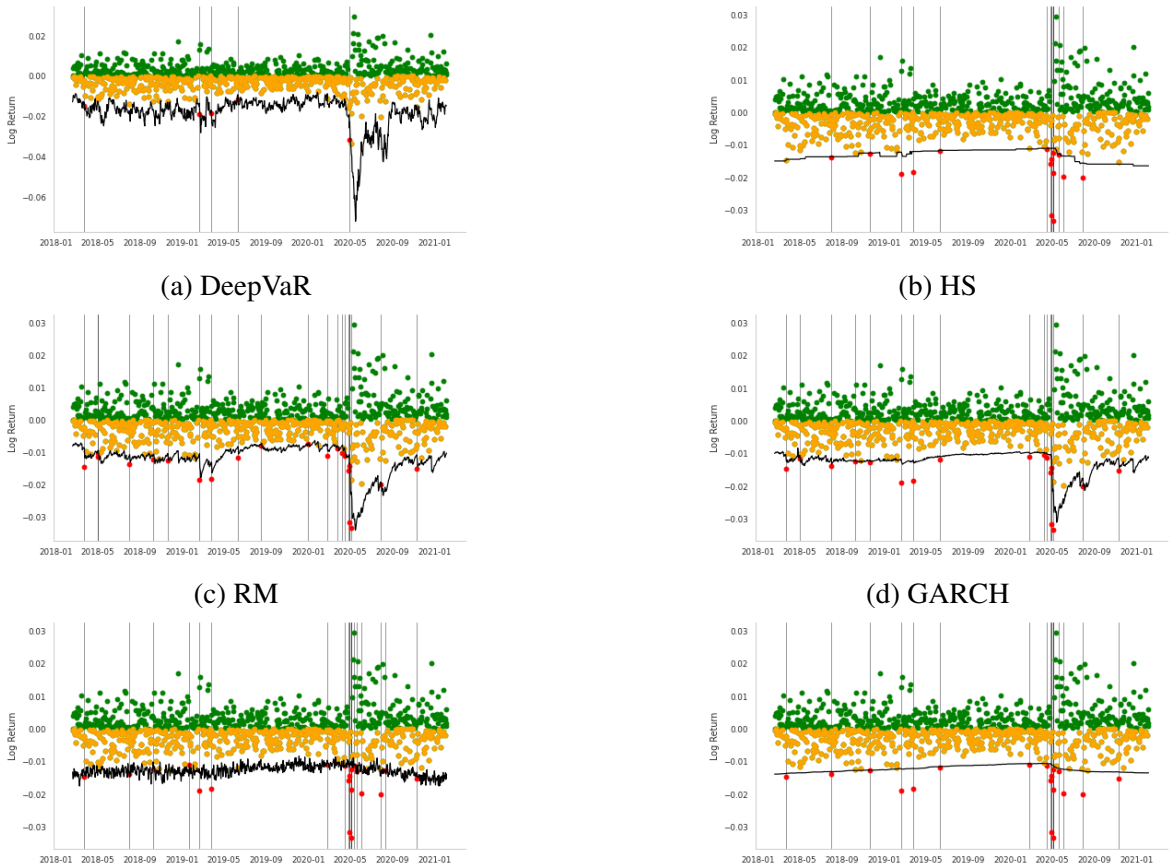
Table 3.3: Mean running time to estimate VaR quantiles.

Model	1 asset(sec)	4 assets(sec)	Rel. Difference (%)
DeepAR	12.457834	12.533903	0.61
HS	0.000201	0.000398	98.01
RM	0.003418	0.013461	293.83
GARCH	0.009435	0.038732	310.51
BiGAN	20.467264	20.947305	2.35
MC	0.000567	0.002108	271.78

3.3.2.5.2 Univariate VaR Performance: This section presents VaR estimations for each FX asset separately, covering the period from 2018-01-01 to 2020-12-18, as outlined in Section 3.3.2.4.1. The performance of various models is summarized in Tables 3.4-3.11 and Figures 3.14-3.17. Each figure compares the VaR estimates of each model (black line) against the actual portfolio returns (green and yellow dots for positive and negative returns, respectively), with red dots indicating VaR violations. The accompanying tables present evaluation metrics and statistical tests for each model.

The AUDUSD currency pair, known for its strong liquidity, is the first time series analyzed. Table 3.4 shows that the DeepVaR model outperforms the other models in all examined loss metrics. Table 3.5 indicates that, apart from DeepVaR, all models fail the Christoffersen and DQ tests for 99% VaR estimation, highlighting their inability to maintain correct unconditional coverage. Figure 3.14 further illustrates that DeepVaR adapts to increased AUDUSD volatility by providing stricter VaR estimates, while traditional models experience clustered VaR violations.

For the GBPUSD currency pair, associated with two major global economies, the DeepVaR model also shows superior performance, particularly in terms of the number of violations and various loss metrics (Table 3.6). Figure 3.15



(a) DeepVaR (b) HS

(c) RM (d) GARCH

(e) BiGAN (f) MC

Figure 3.14: AUDUSD: $VaR^{99\%}$ performance per model.

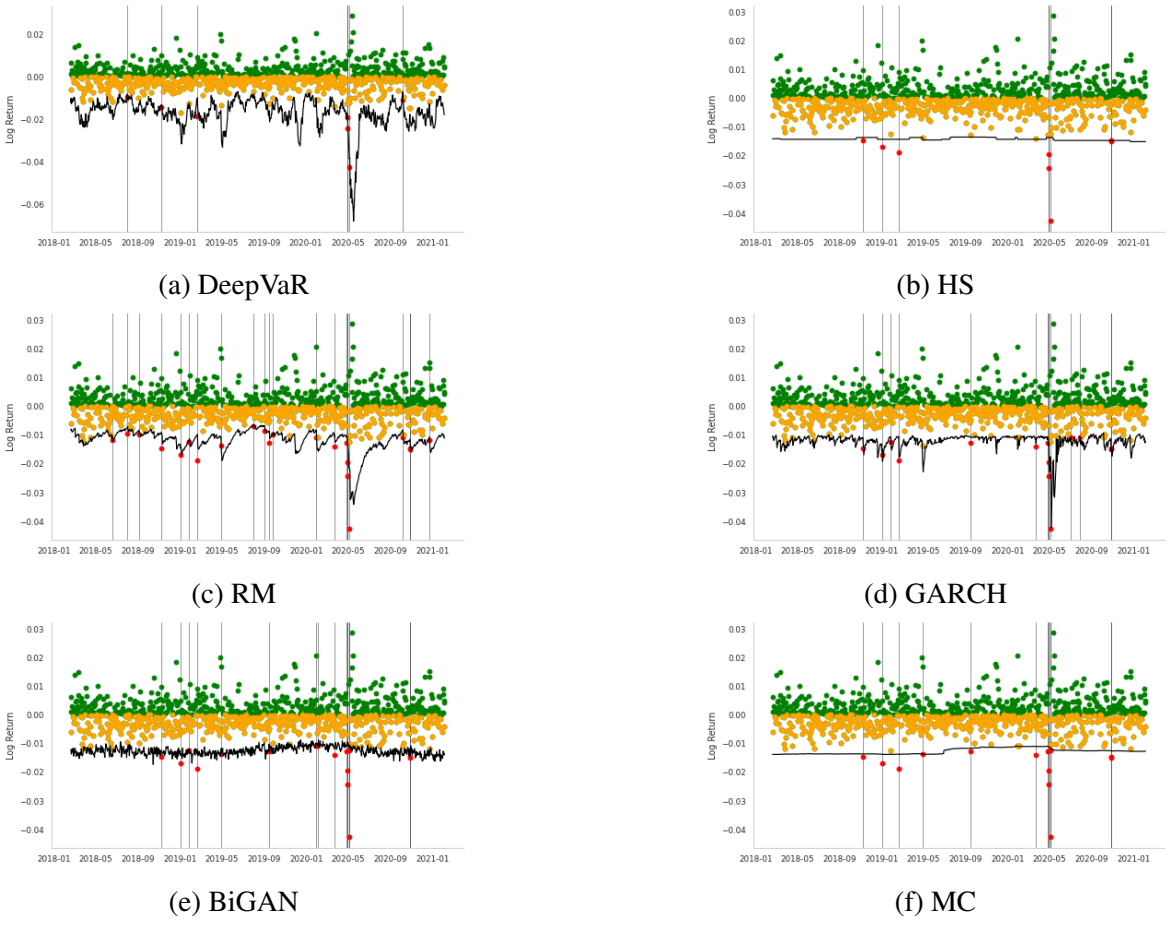
Table 3.4: Performance of $VaR^{99\%}$ models in AUDUSD series.

Model	$E[v]$	v	r_v	l_{QL}	l_Q	l_T	l_F
DeepVaR	9.28	**5**	**0.00539**	**0.00539**	**-0.00632**	**0.00020**	**0.02334**
HS	9.28	15	0.01616	0.01617	-0.00519	0.00022	0.02923
RM	9.28	21	0.02263	0.02263	-0.00472	0.00021	0.03457
GARCH	9.28	17	0.01832	0.01832	-0.00488	**0.00020**	0.03059
BiGAN	9.28	20	0.02155	0.02155	-0.00502	0.00023	0.03403
MC	9.28	18	0.01940	0.01940	-0.00489	0.00022	0.03149

indicates that both DeepVaR and HS fail to capture negative returns at similar dates, while other models exhibit more frequent VaR violations. Despite passing the Christoffersen test, all models fail the DQ test, suggesting serial dependency in the VaR violations (Table 3.7).

Table 3.5: Coverage and independence tests of $VaR^{99\%}$ models in AUDUSD series. The p-values are in brackets.

Model	LR_{uc}	LR_{ind}	LR_{cc}	DQ
DeepVaR	2.386 [0.122]	0.054 [0.816]	2.441 [0.295]	2.158 [0.905]
HS	3.014 [0.083]	10.682** [0.001]	13.696** [0.001]	124.125** [0.0]
RM	11.036** [0.001]	2.931 [0.087]	13.966** [0.001]	31.907** [0.0]
GARCH	5.224* [0.022]	1.013 [0.314]	6.237* [0.044]	17.616** [0.007]
BiGAN	9.424** [0.002]	7.211** [0.007]	16.635** [0.0]	106.141** [0.0]
MC	6.512* [0.011]	8.445** [0.004]	14.957** [0.001]	108.176** [0.0]



(a) DeepVaR (b) HS

(c) RM (d) GARCH

(e) BiGAN (f) MC

Figure 3.15: GBPUSD: $VaR^{99\%}$ performance per model.

The USDJPY currency pair, second only to EURUSD in trading volume, is also evaluated. DeepVaR consistently shows superior performance across most metrics, except for tick loss, where GARCH performs slightly better (Table 3.8).

Table 3.6: Performance of $VaR^{99\%}$ models in GBPUSD series.

Model	$E[v]$	v	r_v	l_{QL}	l_Q	l_T	l_F
DeepVaR	9.28	**7**	**0.00754**	**0.00754**	**-0.00617**	**0.00019**	0.02454
HS	9.28	8	0.00862	0.00862	-0.00557	0.00020	**0.02280**
RM	9.28	22	0.02371	0.02371	-0.00467	0.00021	0.03530
GARCH	9.28	14	0.01509	0.01509	-0.00485	**0.00019**	0.02710
BiGAN	9.28	16	0.01724	0.01724	-0.00502	0.00022	0.02969
MC	9.28	13	0.01401	0.01401	-0.00508	0.00021	0.02663

Table 3.7: Coverage and independence tests of $VaR^{99\%}$ models in GBPUSD series. The p-values are in brackets.

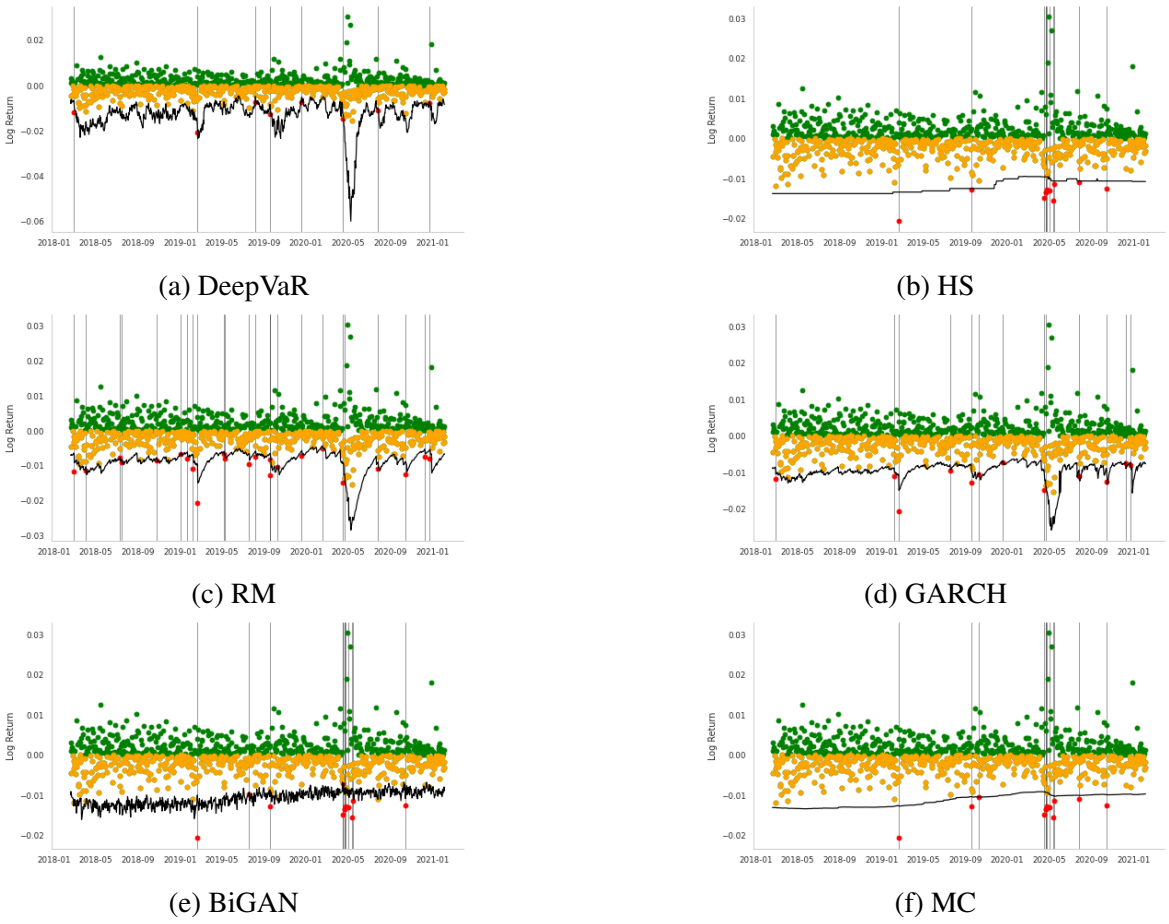
Model	LR_{uc}	LR_{ind}	LR_{cc}	DQ
DeepVaR	0.613 [0.434]	4.258* [0.039]	4.871 [0.088]	27.815** [0.0]
HS	0.184 [0.668]	3.713 [0.054]	3.897 [0.142]	31.297** [0.0]
RM	12.745** [0.0]	0.366 [0.545]	13.11** [0.001]	36.725** [0.0]
GARCH	2.108 [0.147]	1.623 [0.203]	3.731 [0.155]	35.0** [0.0]
BiGAN	4.055* [0.044]	4.868* [0.027]	8.923* [0.012]	98.703** [0.0]
MC	1.347 [0.246]	6.491* [0.011]	7.838* [0.02]	77.377** [0.0]

The coverage and independence tests in Table 3.9 indicate promising results for both DeepVaR and GARCH, with DeepVaR achieving fewer VaR violations than the nominal threshold. As shown in Figure 3.16, DeepVaR provides stricter VaR estimates during periods of high volatility, such as May 2020, by leveraging the dependencies between the examined time-series.

Table 3.8: Performance of $VaR^{99\%}$ models in USDJPY series.

Model	$E[v]$	v	r_v	l_{QL}	l_Q	l_T	l_F
DeepVaR	9.28	**8**	**0.00862**	**0.00862**	**-0.00493**	0.00015	**0.02146**
HS	9.28	11	0.01185	0.01185	-0.00489	0.00016	0.02387
RM	9.28	24	0.02586	0.02586	-0.00366	0.00015	0.03456
GARCH	9.28	13	0.01401	0.01401	-0.00396	**0.00013**	0.02351
BiGAN	9.28	12	0.01293	0.01293	-0.00439	0.00016	0.02354
MC	9.28	12	0.01293	0.01293	-0.00460	0.00016	0.02411

The final currency pair examined is EURUSD, the most widely traded forex pair. The results, summarized in Table 3.10, show that DeepVaR has the



(a) DeepVaR (b) HS

(c) RM (d) GARCH

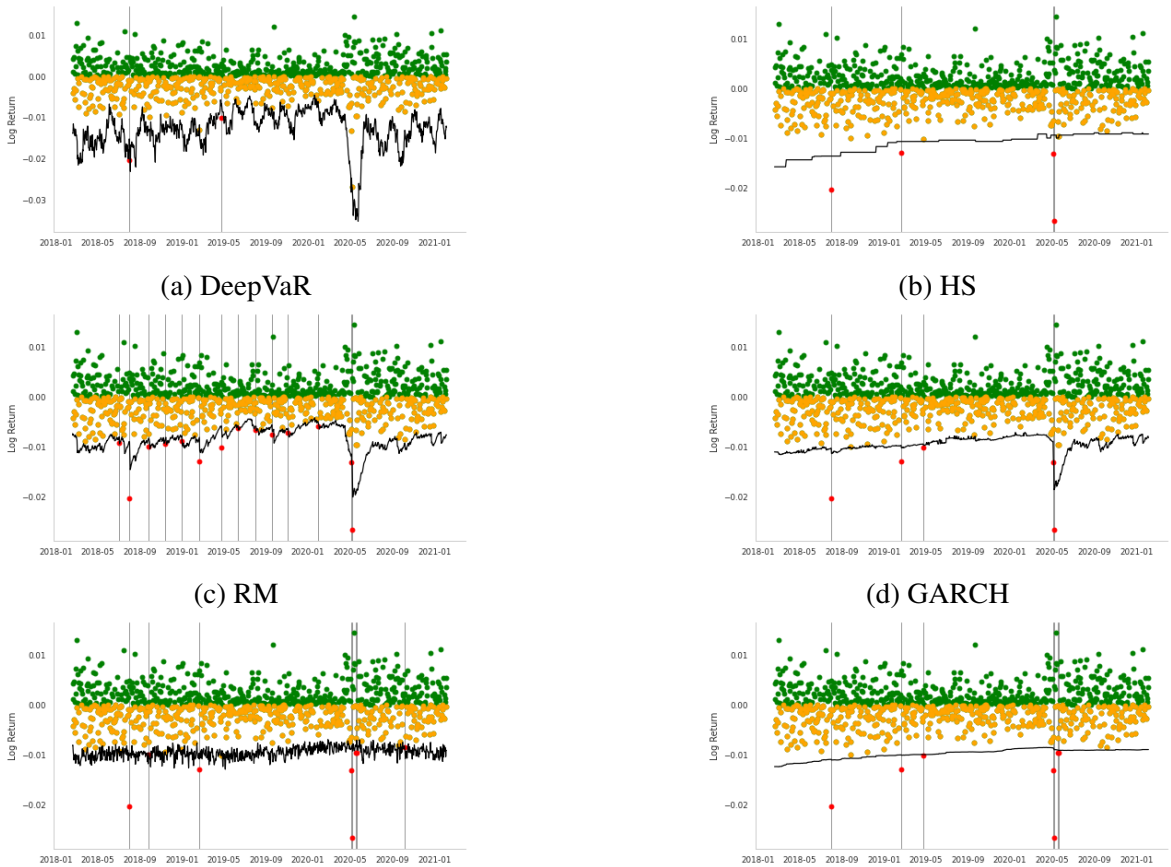
(e) BiGAN (f) MC

Figure 3.16: USDJPY: $VaR^{99\%}$ performance per model.

Table 3.9: Coverage and independence tests of $VaR^{99\%}$ models in USDJPY series. The p-values are in brackets.

Model	LR_{uc}	LR_{ind}	LR_{cc}	DQ
DeepVaR	0.184 [0.668]	0.139 [0.709]	0.324 [0.851]	2.857 [0.827]
HS	0.308 [0.579]	2.477 [0.116]	2.785 [0.248]	49.276** [0.0]
RM	16.439** [0.0]	0.207 [0.649]	16.646** [0.0]	36.755** [0.0]
GARCH	1.347 [0.246]	0.37 [0.543]	1.717 [0.424]	8.448 [0.207]
BiGAN	0.743 [0.389]	7.143** [0.008]	7.886* [0.019]	81.484** [0.0]
MC	0.743 [0.389]	2.159 [0.142]	2.902 [0.234]	45.667** [0.0]

lowest quadratic and smooth loss, as well as the fewest VaR violations. The GARCH model performed better in terms of tick and firm loss. Table 3.11 shows that all models passed the Christoffersen coverage and independence tests, but



(a) DeepVaR (b) HS

(c) RM (d) GARCH

(e) BiGAN (f) MC

Figure 3.17: EURUSD: $VaR^{99\%}$ performance per model.

only DeepVaR and RM succeeded in the DQ test. Figure 3.17 demonstrates that DeepVaR is the only model that avoided any VaR violations during the EURUSD volatility shift in May 2020.

Table 3.10: Performance of $VaR^{99\%}$ models in EURUSD series.

Model	$E[v]$	v	r_v	l_{QL}	l_Q	l_T	l_F
DeepVaR	9.28	**2**	**0.00216**	**0.00216**	**-0.00508**	0.00014	0.01501
HS	9.28	4	0.00431	0.00431	-0.00459	0.00014	0.01539
RM	9.28	14	0.01509	0.01509	-0.00362	**0.00013**	0.02354
GARCH	9.28	5	0.00539	0.00539	-0.00404	**0.00013**	**0.01494**
BiGAN	9.28	8	0.00862	0.00862	-0.00404	0.00014	0.01814
MC	9.28	7	0.00754	0.00754	-0.00416	0.00014	0.01738

The results for each FX time series can be attributed to the inherent charac-

Table 3.11: Coverage and independence tests of $VaR^{99\%}$ models in EURUSD series. The p-values are in brackets.

Model	LR_{uc}	LR_{ind}	LR_{cc}	DQ
DeepVaR	8.463** [0.004]	0.009 [0.926]	8.472* [0.014]	5.786 [0.448]
HS	3.846 [0.05]	0.035 [0.852]	3.881 [0.144]	29.349** [0.0]
RM	2.108 [0.147]	0.429 [0.512]	2.537 [0.281]	9.598 [0.143]
GARCH	2.386 [0.122]	0.054 [0.816]	2.441 [0.295]	22.326** [0.001]
BiGAN	0.184 [0.668]	0.139 [0.709]	0.324 [0.851]	26.285** [0.0]
MC	0.613 [0.434]	0.107 [0.744]	0.72 [0.698]	28.095** [0.0]

teristics of the examined models, with the key factor being how each model adapts to changes in return volatility. Models that closely fit the true returns are more prone to VaR breaches, while those that generalize better tend to produce higher firm losses (Section 3.3.2.4.2).

For instance, the HS model relies on the last 1000 historical returns to estimate VaR, leading to slow adaptation to sudden and permanent changes in volatility, as reflected in Figures 3.14-3.17. Similarly, the MC model, which also uses historical data for input parameters, shows slow response to volatility shifts.

The BiGAN model, as illustrated in Figures 3.14-3.17, shows frequent oscillations in its VaR estimates, but it fails to capture sudden negative returns, suggesting limited effectiveness in predicting VaR during rare financial events.

The GARCH and RM models, both GARCH(1,1) types, effectively capture time-varying volatility. However, the RM model uses a shorter history (74 days) compared to the 900 days used by GARCH, leading to RM producing the most VaR violations across all time series due to its closer fit to actual PnL.

DeepVaR mitigates this trade-off by training over 900 days and predicting return distribution parameters based on the last 15 values of the input series. This allows DeepVaR to "memorize" past information while accurately capturing time-series volatility using recent data for parameter estimation.

3.3.2.5.3 Multivariate VaR Performance: The multivariate performance evaluation of VaR models was conducted using 1000 randomly generated portfolios, each including both long and short positions on various FX assets. The goal was to assess the models in a realistic portfolio context where the total absolute value of positions equals one, simulating a scenario that minimizes commission fees—a common strategy in portfolio management. Notably, no commission fees or extra charges were considered in this evaluation, simplifying the analysis as these could be modeled as a constant across all VaR models without significantly affecting the outcomes.

To compute the portfolio VaR, the correlations between the FX instruments were accounted for, using the correlation matrix R, which was calculated based on the last 125 daily returns. The portfolio VaR for a given day was then estimated using the weighted VaR estimates of each instrument, taking into account both long and short positions. This process is detailed in Algorithm 2.

The average performance across the random portfolios is summarized in Table 3.12. DeepVaR consistently achieved the lowest loss across all metrics, except for tick loss, where it showed a slight disadvantage. Table 3.13 further highlights the robustness of DeepVaR, with a significant percentage of portfolios passing the coverage and independence tests, outperforming the other models.

Table 3.12: Average performance of $VaR^{99\%}$ models over the FX portfolios.

Model	$E[v]$	v	r_v	l_{QL}	l_Q	l_T	l_F
DeepVaR	9.28	**2.90310**	**0.00319**	**0.00313**	**-0.00427**	0.00011	**0.01351**
HS	9.28	7.29	0.00784	0.00784	-0.00361	0.00011	0.01618
RM	9.28	12.14	0.01308	0.01308	-0.00305	**0.00010**	0.02002
GARCH	9.28	8.61	0.00928	0.00928	-0.00321	**0.00010**	0.01660
BiGAN	9.28	11.51	0.01240	0.01240	-0.00320	0.00011	0.01966
MC	9.28	10.31	0.01111	0.01111	-0.00327	0.00011	0.01854

The evaluation results are also presented as box plots in Figure 3.18, providing

Algorithm 2 Portfolio VaR rolling window estimation

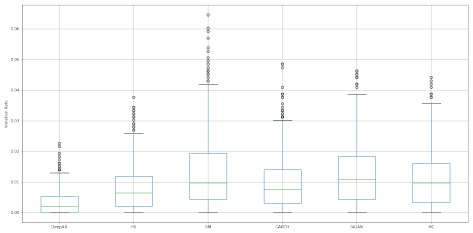
Generate random portfolio weights $w \in \mathbb{R}^{1\times 4}$
Set confidence probability a (i.e. $\alpha = 0.99$) of VaR estimation
Set window t_w = 125, for the calculation of returns correlation matrix $R \in \mathbb{R}^{4\times 4}$
Split FX returns to train $R_{train} \in \mathbb{R}^{125\times 4}$ and test set $R_{test} \in \mathbb{R}^{928\times 4}$
for test day t = 1 to T **do**
    $R \Leftarrow \text{CORR}(R_{train})$
    for m in $models$ **do**
        Initialize zero vector $V \in \mathbb{R}^{1\times 4}$
        for i in w **do**
            if $w_i < 0$ **then**
                $V_i \Leftarrow w_i VaR^{1-\alpha}_{i,m,t}$
            else
                $V_i \Leftarrow w_i VaR^{\alpha}_{i,m,t}$
            end if
        end for
        $VaR^{\alpha}_{m,t} \Leftarrow -\sqrt{VRV^T}$
    end for
    Append $R_{test}[t]$ to R_{train}
    $PnL_t \Leftarrow \sum_{i=1}^{4} w_i R_{train,i}[-1]$
end for

Table 3.13: Percentage of portfolios passed the coverage and independence tests of $VaR^{99\%}$ per model in significant level 95%

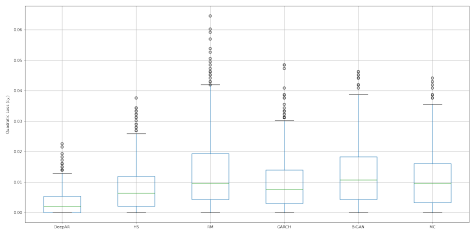
Model	LR_{uc}	LR_{ind}	LR_{cc}	DQ
DeepVaR	72.8	**95.5**	**80.6**	**84.6**
HS	**76.9**	72.0	59.5	36.7
RM	65.2	95.3	68.3	55.9
GARCH	76.5	92.9	77.5	63.0
BiGAN	64.3	71.5	53.5	26.1
MC	70.4	68	54.8	26.7

a visual comparison of the models' performance across the portfolios. DeepVaR stands out as the only model with a violation rate below the 1-α (i.e., 0.01) confidence probability in most portfolios, as shown in Figure 3.18a. The other models showed higher variability, with some portfolios significantly exceeding the nominal threshold.

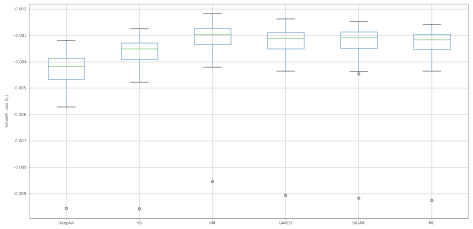
DeepVaR's advantage is also evident in the quadratic loss (Figure 3.18b) and



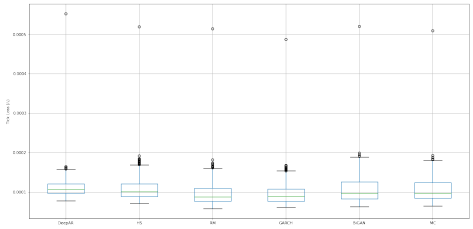
(a) Violation Rate



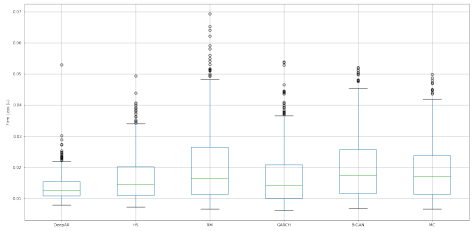
(b) Quadratic Loss



(c) Smooth Loss



(d) Tick Loss



(e) Firm Loss

Figure 3.18: Box-plots of the $VaR^{99\%}$ performance per model over 1000 random portfolios.

smooth loss (Figure 3.18c) metrics, where it outperformed all other models. Although DeepVaR underperformed slightly in tick loss (Figure 3.18d), its performance was more stable across portfolios compared to the other models, which exhibited higher variability.

Finally, the firm loss metric (Figure 3.18e) highlights DeepVaR's efficiency, showing that it resulted in lower opportunity costs for firms compared to the other models.

In summary, the multivariate evaluation demonstrated that DeepVaR consistently outperformed the other models in most metrics across a broad set of random portfolios, making it a robust choice for portfolio VaR estimation.

3.4 Discussion

This chapter presented a novel approach to online learning for time series prediction in dynamic environments, such as cloud computing and financial markets. The primary contributions include the development of an adaptable system architecture that continuously updates predictive models based on real-time data streams, ensuring high accuracy and relevance. The system's effectiveness was demonstrated through two distinct use cases: optimizing serverless function latency in hybrid cloud environments and assessing financial risk using probabilistic deep neural networks. Both use cases showed significant improvements in predictive performance, validating the system's utility in real-world scenarios.

However, several challenges were encountered during implementation. Managing drift and non-stationarity in time series data proved particularly difficult, requiring sophisticated data preprocessing and model adaptation strategies. Additionally, the computational constraints of real-time environments posed challenges in balancing model complexity with the need for low-latency predictions. Despite these challenges, the system demonstrated robustness and adaptability, particularly when integrated with advanced online learning techniques.

Chapter 4
Prompt Engineering for Financial Sentiment Analysis

Chapter Structure

This Chapter is constructed as follows:

- **Section 4.1 - Introduction to Financial Sentiment Analysis**, introduces the concept of sentiment analysis within the financial domain, discussing its importance and the challenges specific to financial texts.
- **Section 4.2 - Background on Financial Sentiment Analysis**, reviews previous research in sentiment analysis for finance, focusing on the role of AI tools and the evolution of techniques leading up to the use of LLMs.
- **Section 4.3 - Methodology**, details the approach used to evaluate ChatGPT for financial sentiment analysis, including the dataset, experimental setup, and the metrics for performance evaluation.
- **Section 4.4 - Experimental Results**, presents and analyzes the results of the experiments, comparing ChatGPT's performance with FinBERT, and discussing the impact of different prompt designs.

- **Section 4.5 - Discussion**, summarizes the key findings, highlights the strengths and limitations of ChatGPT in this context, and suggests implications for real-world applications and future research.

Building on the foundation laid in the previous chapters, where the dissertation thesis explored knowledge systems, reasoning engines and the dynamics of online learning systems for time series prediction, this chapter transitions to the domain of text analytics. In the financial sector, where unstructured data such as news articles, analyst reports, and social media posts abound, extracting actionable insights through text analytics is crucial. The power of LLMs is explored to meet these challenges, leveraging their advanced capabilities in NLP. This chapter explores how emerging models like GPT outperform previous state-of-the-art language models, offering unprecedented accuracy and efficiency in tasks like sentiment analysis [Fatouros et al., 2023d]. Text analytics not only offers valuable standalone insights but also plays a pivotal role in enhancing predictive systems. For instance, when integrated with time series models, such as those presented in Chapter 3, text-derived features can provide additional context, improving the accuracy and robustness of time series models. This chapter delves into the application of LLMs in financial sentiment analysis, demonstrating how these models transform raw text into actionable intelligence, thereby enriching knowledge and predictive systems in dynamic environments like financial markets.

4.1 Introduction to Financial Sentiment Analysis

4.1.1 Motivation

The financial services industry has always been at the forefront of adopting new technologies, constantly evolving to keep pace with the rapidly shifting global landscape. From the early days of electronic trading platforms to the

more recent emergence of financial technologies (Fintech), the sector has undergone a significant transformation, especially with the incorporation of AI and ML [Arner et al., 2015]. The growth of the Fintech market underscores the profound impact of these innovations, with applications ranging from credit scoring and fraud detection to robo-advisory services and AI-powered chatbots for customer service [Fatouros et al., 2023b, Kotios et al., 2022].

Among the various technological advancements, sentiment analysis has become an essential tool for understanding market dynamics and predicting future trends. By analyzing market sentiment, financial institutions can gain valuable insights into the collective mood of the market, thereby enabling more informed and strategic decision-making.

Historically, sentiment analysis in finance has relied on manually curated lexicons and simple ML algorithms [Schumaker and Chen, 2009]. However, with the rapid advancements in NLP, more sophisticated techniques have emerged. DL models like BERT (Bidirectional Encoder Representations from Transformers) [Devlin et al., 2018] and FinBERT [Liu et al., 2021], a version of BERT tailored specifically for financial text, have significantly enhanced the accuracy and reliability of sentiment analysis.

Despite these advancements, the financial domain presents unique challenges for sentiment analysis. Financial texts, particularly news headlines, often contain domain-specific terminology and nuanced sentiments that complicate the sentiment classification process [Loughran and McDonald, 2011]. For instance, a typical characteristic of FX news is the intertwining of sentiments about multiple currencies within a single statement. Consider the headline, "CAD is one of the better places to sell the US dollar on a pullback." Conventional sentiment analysis tools might incorrectly identify this as a positive sentiment overall, missing the nuanced implications for specific currency pairs like USD/CAD. Such tools often fail to infer the specific financial instruments related to the

sentiment, which is crucial for accurate financial analysis.

Moreover, traditional sentiment analysis models often lack the flexibility to adapt their output based on specific use-case contexts, limiting their broader applicability [Poria et al., 2016]. For example, the sentiment expressed in a discussion about a new government regulatory policy may vary depending on whether the context is an investor forum focused on market impacts or a consumer forum concerned with service fees. Standard sentiment analysis models might struggle to capture these contextual differences, underscoring the need for more context-aware models that can offer a more comprehensive and accurate sentiment analysis in financial applications [Poria et al., 2017]. Addressing these challenges presents significant opportunities for advancing financial sentiment analysis.

Among the AI tools reshaping the financial landscape, Generative Pre-trained Transformers (GPT) have shown considerable promise. GPT, a state-of-the-art language model developed by OpenAI, has been trained on vast amounts of data, enabling it to understand and replicate complex patterns in human language with remarkable accuracy [Radford et al., 2018]. Since the introduction of the first GPT model in 2018, OpenAI has made significant improvements in the architecture, culminating in the latest version, GPT-4 [OpenAI, 2023a]. The widespread impact of GPT became particularly evident with the release of ChatGPT, a specialized version fine-tuned for conversational tasks, towards the end of 2022.

ChatGPT holds significant potential for enhancing existing financial applications, including risk analysis through sentiment analysis. By leveraging ChatGPT's advanced natural language understanding capabilities, financial institutions can significantly improve the accuracy and depth of sentiment analysis. This model can effectively process and interpret vast amounts of unstructured data, such as news articles and financial reports, providing a comprehensive

view of market sentiment. Improved sentiment analysis can, in turn, inform critical areas such as investment strategies, risk management, and portfolio optimization, leading to more informed decision-making and potentially higher returns [Tetlock, 2007, Chen et al., 2014]. Additionally, ChatGPT's conversational capabilities can make complex risk analysis findings more accessible to both expert and novice users, enhancing the practical utility of financial insights [Yue et al., 2023].

The advanced language understanding capabilities of ChatGPT open up a wealth of opportunities for innovation in the financial domain, particularly in sentiment analysis. Despite its potential, this area remains largely underexplored, presenting a significant opportunity for research and development. ChatGPT's ability to interpret complex language patterns, which are often prevalent in unstructured financial data, offers a promising solution for enhancing the depth and accuracy of sentiment analysis in financial contexts.

4.1.2 Value Proposition

The field of financial sentiment analysis has seen the application of various Language Model (LM), but the potential of LLMs, in the context of the FX market remains relatively unexplored, as discussed in Section 4.2. FX news is particularly challenging due to its unique nature, often conveying sentiments about two different assets within a single statement, adding an extra layer of complexity. Recognizing this intricacy, this chapter breaks new ground by investigating ChatGPT 3.5's ability to discern these subtle sentiment cues. The value propositions of the research presented in this chapter are outlined as follows:

- A zero-shot prompting strategy is employed to assess ChatGPT's proficiency in interpreting FX-related financial text, emphasizing its ability to

achieve accurate sentiment analysis without the need for domain-specific fine-tuning. This approach highlights the advanced capabilities of generative AI in specialized financial sentiment analysis tasks and its potential for real-world applications.

- A carefully curated and annotated dataset of FX-related news headlines has been developed and made publicly available. This dataset facilitates a comprehensive evaluation of various ChatGPT prompts, showcasing the model's extensive training on diverse datasets and its adaptability across different sectors, including finance. Sharing this dataset contributes a valuable resource to the research community, encouraging further innovation in financial sentiment analysis and financial services.

- The evaluation includes traditional metrics such as precision, recall, f1-score, and Mean Absolute Error (MAE) for sentiment classification. Additionally, it extends to analyzing the correlation between predicted sentiment and market returns. This comprehensive evaluation covers sentiment class prediction and sentiment score estimation for single and multiple news headlines, as well as complete news articles, demonstrating ChatGPT's versatility and proficiency in handling complex financial texts.

- Benchmarking ChatGPT's performance against the state-of-the-art model FinBERT and a naive approach provides a well-rounded perspective on its capabilities in the financial domain. The exploration of multiple prompts underscores the importance of prompt engineering, particularly in zero-shot contexts, to optimize performance and enhance the effectiveness of sentiment analysis.

- As LLMs become increasingly integrated into real-world applications, this chapter offers valuable insights for researchers and stakeholders on how to effectively harness the model's potential. The publicly available dataset and annotations further solidify the contribution, serving as a useful re-

source for future research endeavors in financial sentiment analysis.

4.2 Background on Financial Sentiment Analysis

4.2.1 Sentiment Analysis in Finance

Sentiment analysis, also known as opinion mining, involves the application of computational methods to extract subjective information, such as opinions and sentiments, from text data [Bing, 2012]. This technique has been utilized across various domains, including customer reviews, social media content, and news articles. In the financial sector, sentiment analysis is particularly valuable for assessing market sentiment, a critical factor that can influence the price movements of financial assets [Bollen et al., 2011, Siering et al., 2018].

The concept that sentiment or public mood can impact financial markets has been recognized since the time of John Maynard Keynes [Keynes, 1937], who described market behavior as being influenced by "animal spirits," or waves of optimism and pessimism. Numerous empirical studies have supported this idea, showing that shifts in investor sentiment can significantly affect asset prices and trading volumes [Baker and Wurgler, 2007, Tetlock, 2007].

Initially, sentiment analysis in finance relied on manually crafted financial lexicons, where words were pre-classified as positive, negative, or neutral [Loughran and McDonald, 2011]. However, this approach often oversimplified the context in which words were used. For example, while the word "crisis" typically carries a negative connotation, it could be used in a positive context, such as in the phrase "the company successfully averted a crisis."

The advent of ML techniques marked a significant advancement in sentiment analysis. ML models, trained on labeled financial news articles, were able to learn from the context in which words appeared, leading to more accurate senti-

ment classifications [Schumaker and Chen, 2009]. Despite these improvements, traditional ML methods, such as Naive Bayes, support vector machines, and decision trees, often struggled to capture long-term dependencies in the text and were challenged by the high dimensionality of textual data [Chen et al., 2018].

The evolution of DL techniques, particularly in the field of NLP, has further enhanced sentiment analysis. This progress is exemplified by the introduction of advanced language models like ELMo [Peters et al., 2018], ULMFit [Howard and Ruder, 2018], and transformer-based architectures such as BERT, which have demonstrated superior performance in sentiment classification tasks. These models excel at understanding context, capturing long-term dependencies, and reducing data dimensionality through high-quality embeddings [Refaeli and Hajek, 2021]. Domain-specific adaptations like FinBERT have further refined these capabilities, particularly for financial texts, thereby improving the accuracy and reliability of financial sentiment analysis. For instance, [Leippold, 2023] highlighted FinBERT's robustness against adversarial text manipulations compared to keyword-based methods in financial sentiment analysis. [Farimani et al., 2022] demonstrated the use of FinBERT models for real-time market predictions, emphasizing the importance of sentiment-aware time series derived from financial news related to FX and cryptocurrency markets. Their approach, which integrated sentiment scores with technical indicators, showed significant improvements over existing baselines. These advancements underscore the transformative potential of LMs in enhancing the understanding and prediction of financial sentiments.

However, financial sentiment analysis still faces several challenges. These include the need to comprehend domain-specific terminology, disentangle subtle sentiments related to multiple financial instruments within the same context, and adjust sentiment output based on specific use cases or prior topics. For

example, a statement might be interpreted as negative for the overall economy but positive for a particular stock. Current models like FinBERT often reflect the perspective of the experts who annotated the training data [Malo et al., 2014], limiting their flexibility in adjusting sentiment output based on finer nuances. This context-dependent framing of sentiment remains an open challenge, highlighting the need for future research that enhances the adaptability and performance of sentiment analysis in finance. In this regard, the emergence of LLMs like GPT variants presents promising potential. These models, with their inherent contextual understanding and adaptability, offer valuable solutions to these challenges and open new avenues for exploration in financial sentiment analysis [Radford et al., 2019].

4.2.2 ChatGPT and Related AI Tools

The success of GPT has spurred the development of various LLMs that have demonstrated impressive performance across a range of NLP tasks. Each of these models brings distinct advantages and capabilities to the field of NLP, including financial applications.

Among these, BloombergGPT is a notable example, excelling in several financial NLP tasks [Wu et al., 2023a]. Developed by Bloomberg's AI team, this model has been trained on an extensive corpus of financial texts. However, despite its success in various financial tasks, BloombergGPT currently lacks an open API and is primarily used internally within Bloomberg as of May 2023.

Another significant player in the realm of LLMs is Google's Bard, a direct competitor to ChatGPT. Powered by Google's LAMDA (Language Model for Dialogue Applications), Bard combines features of both BERT and GPT architectures [Thoppilan et al., 2022]. While Bard has great potential for creating engaging and contextually aware dialogues, it also shares the limitation of

BloombergGPT in that it does not yet offer a widely accessible API.

BLOOM, an open-source alternative to GPT-3, has also garnered attention among LLMs [Scao et al., 2022]. Although BLOOM is open-source, it presents its own challenges, requiring specialized knowledge and substantial computational resources for effective use. Additionally, there isn't a version of BLOOM specifically tuned for conversational tasks, a feature that makes models like ChatGPT particularly valuable.

Since the introduction of ChatGPT, numerous LLMs tailored for specific tasks, such as code completion [Dakhel et al., 2023], content creation, and marketing, have emerged. These models, although more specialized, add new dimensions of utility and specialization, further broadening the potential applications and impacts of LLMs. Despite the proliferation of LLMs, ChatGPT continues to hold a leading position in the field [JasperAI, 2023]. Its accessibility through an open API, extensive training data, and versatility across a wide range of tasks highlight its substantial potential and practical utility.

Despite the advancements in applying ChatGPT across various fields, such as healthcare and education [Sallam, 2023], its specific application to financial sentiment analysis remains relatively unexplored. Notably, there is a lack of studies that directly assess ChatGPT's performance through its API, which is essential for broader adoption and integration into third-party applications. To date, no studies have evaluated ChatGPT's performance in financial sentiment analysis, particularly within the context of the FX market. This gap in the existing research presents an opportunity to explore and contribute to this emergent field. By focusing on ChatGPT's application and performance in sentiment analysis, and by directly interacting with its API, new insights and findings can be introduced to the field, underscoring the real-world applicability and practicality of integrating ChatGPT into financial services.

4.3 Methodology

This section outlines the methodology used to evaluate ChatGPT's performance in financial sentiment analysis, covering data collection, preprocessing, and the deployment of ChatGPT via its API. It also details the experimental setup, including prompt design and evaluation criteria, with a focus on the practical implications of the findings in the financial sector.

4.3.1 Dataset Creation and Annotation

The dataset was compiled by collecting news headlines relevant to key FX pairs: AUDUSD, EURCHF, EURUSD, GBPUSD, and USDJPY. The data was sourced from FX Live[1] and FXstreet[2] over 86 days, between January and May 2023 (Figure 4.1). The dataset comprises 2,291 unique news headlines, each associated with a FX pair, timestamp, source, author, URL, and article text. Data was gathered using a custom web scraping service deployed on a virtual machine, retrieving the latest news approximately every 15 minutes (Figure 4.2).

Each headline was manually annotated for sentiment, based on its potential short-term impact on the corresponding FX pair, acknowledging the sensitivity of currency markets to economic news [Evans and Lyons, 2005]. Sentiments were categorized as 'positive', 'negative', or 'neutral', corresponding to bullish, bearish, and hold sentiments, respectively. Table 4.1 provides examples of annotated headlines, while Figure 4.3 and 4.4 depict the overall and per-FX-pair sentiment distributions.

The dataset's token distribution was also analyzed, with common tokens in positive and negative sentiment headlines illustrated in Figure 4.5 and 4.6.

[1] https://www.forexlive.com/
[2] https://www.fxstreet.com/

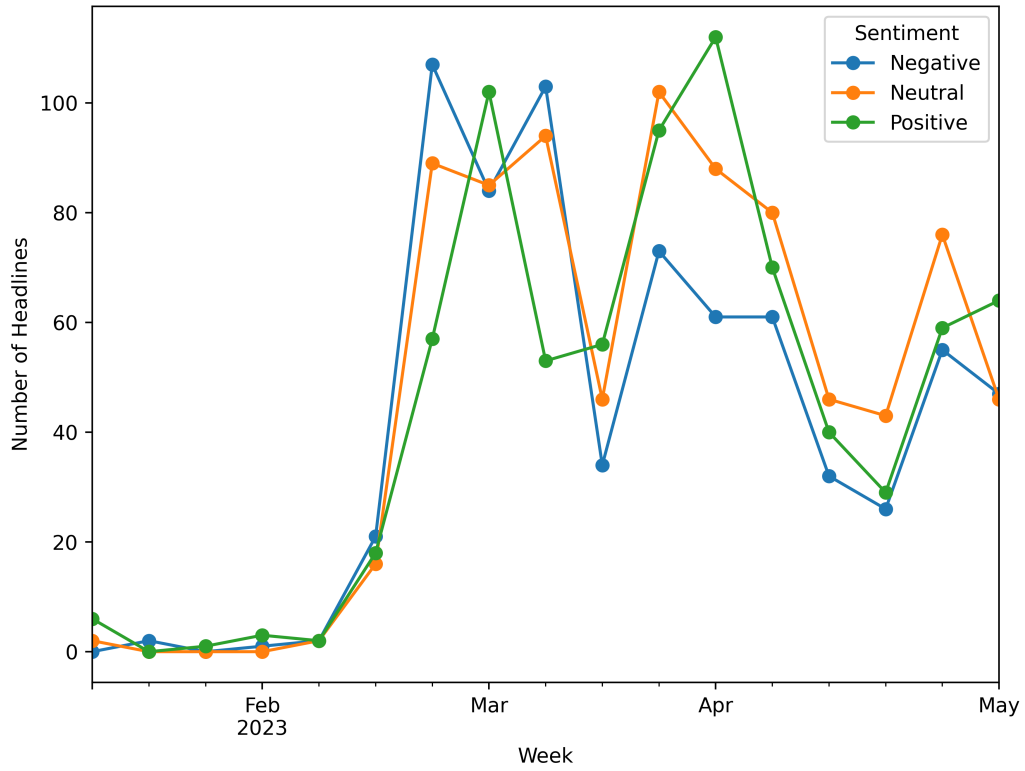


Figure 4.1: Time Series Plot of Sentiment Labels per Week

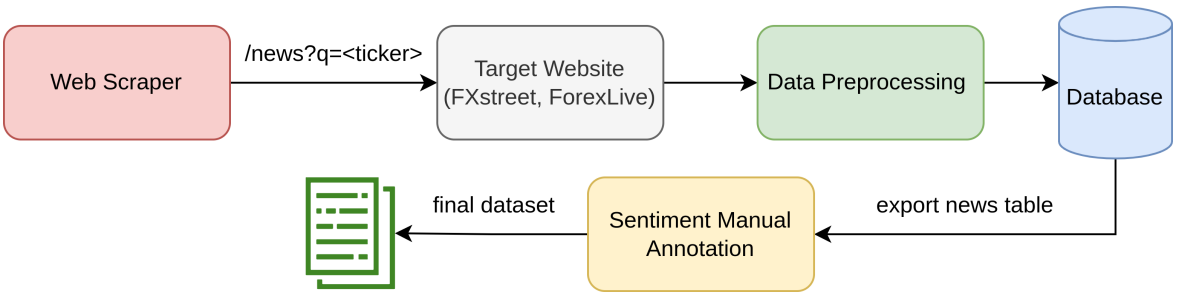


Figure 4.2: Dataset Creation Process

Table 4.2 offers detailed statistics on token distribution, the number of articles, and the daily average of articles per FX pair.

The dataset reveals variability in article frequency, ranging from 55 articles for EURCHF to 758 for EURUSD, and in token count, with an average of 263

tokens per article and 12 tokens per headline. The length of texts influences the response latency, cost, and performance of language models like ChatGPT, with longer texts requiring more computational resources and time, potentially increasing latency and cost.

Table 4.1: Examples of Annotated Headlines

FX Pair	Headline	Sentiment	Explanation
GBPUSD	Diminishing bets for a move to 12400	Neutral	Lack of strong sentiment in either direction
GBPUSD	No reasons to dislike Cable in the very near term as long as the Dollar momentum remains soft	Positive	Positive sentiment towards GBPUSD (Cable) in the near term
GBPUSD	When are the UK jobs and how could they affect GBPUSD	Neutral	Poses a question and does not express a clear sentiment
USDJPY	BoJ's Ueda: Appropriate to continue monetary easing to achieve 2% inflation target with wage growth	Positive	Monetary easing from Bank of Japan (BoJ) could lead to a weaker JPY in the short term due to increased money supply
USDJPY	Dollar rebounds despite US data. Yen gains amid lower yields	Neutral	Since both the USD and JPY are gaining, the effects on the USDJPY FX pair might offset each other
USDJPY	USDJPY to reach 124 by Q4 as the likelihood of a BoJ policy shift should accelerate Yen gains	Negative	USDJPY is expected to reach a lower value, with the USD losing value against the JPY.
AUDUSD	RBA Governor Lowe's Testimony High inflation is damaging and corrosive	Positive	Reserve Bank of Australia (RBA) expresses concerns about inflation. Typically, central banks combat high inflation with higher interest rates, which could strengthen AUD.

To support further research, the dataset has been made publicly available on the Zenodo repository[3] [Fatouros and Kouroumali, 2023]. This resource is expected to be valuable for exploring machine learning techniques in financial sentiment analysis, contributing to a deeper understanding of the interaction between financial news sentiment, AI models, and market behavior.

[3]https://zenodo.org/record/7976208

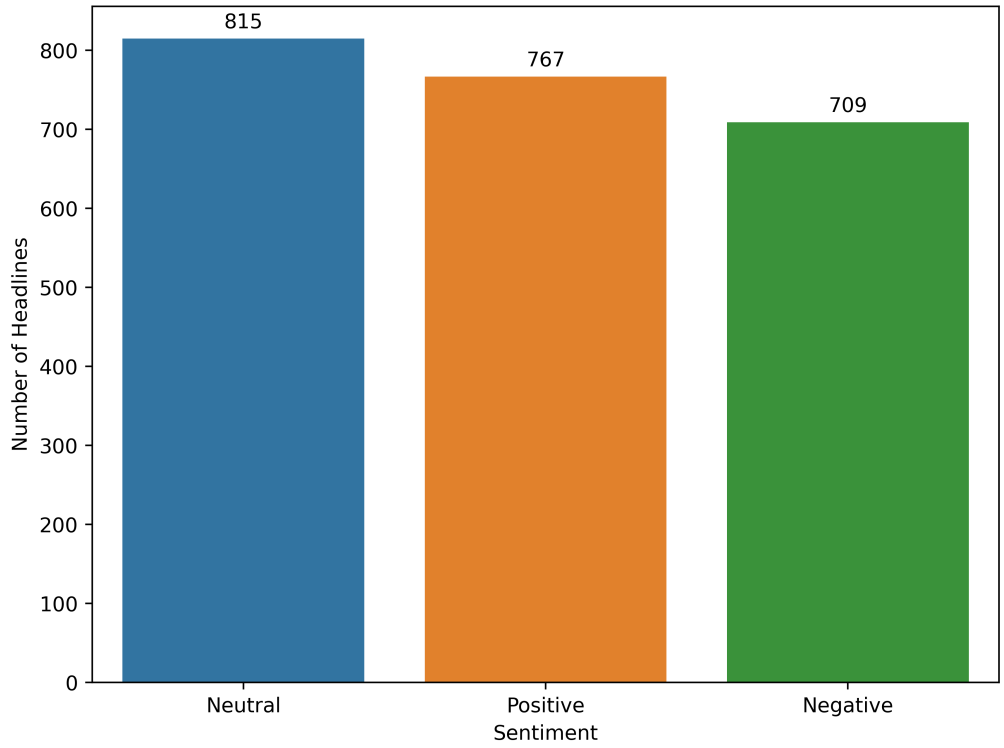


Figure 4.3: Sentiment Distribution

4.3.2 Establishing Baseline: Sentiment Classification using FinBERT

To assess the effectiveness of ChatGPT in financial sentiment analysis, its performance is compared against FinBERT, a widely recognized model known for its accuracy in financial text sentiment classification. FinBERT serves as a solid benchmark, allowing for a meaningful evaluation of ChatGPT's capabilities in this specialized domain.

For the implementation, the Transformers library from Hugging Face was employed to utilize FinBERT [Wolf et al., 2020]. This Python-based API offers a user-friendly interface for integrating pre-trained transformer models into applications. The process involved loading the FinBERT model and its corresponding tokenizer from the Hugging Face model repository, tokenizing the

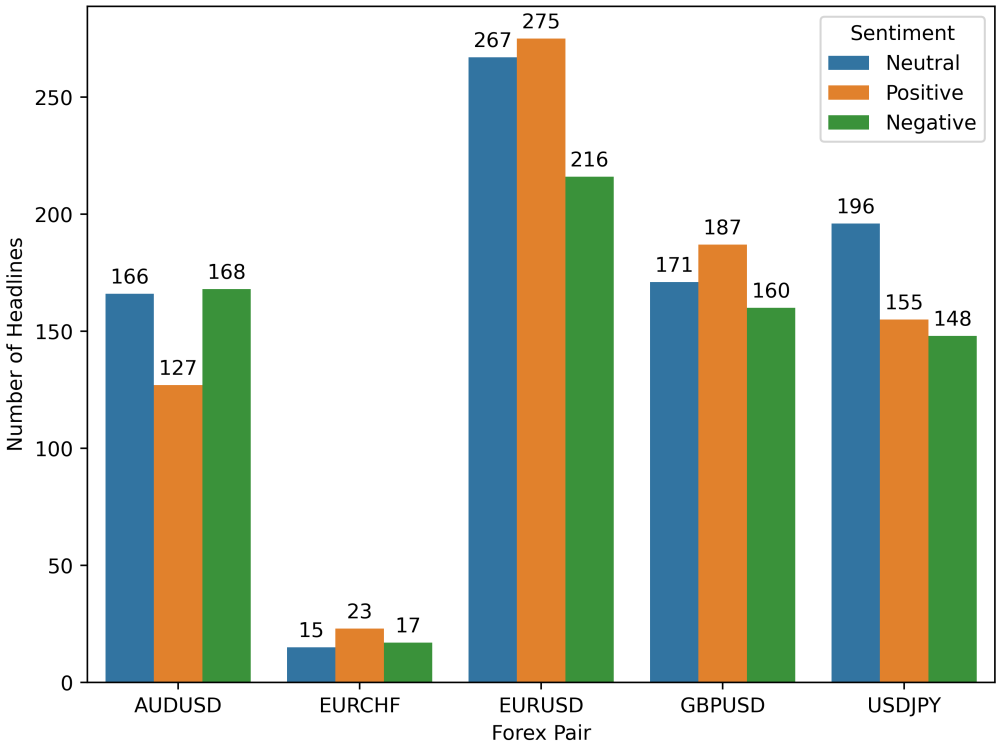


Figure 4.4: Sentiment Distribution per FX Pair

news headlines, and passing them through the model. FinBERT then generates probabilities for each sentiment category—positive, negative, and neutral—reflecting its confidence in classifying the input text. These probabilities are used to determine the predicted sentiment class and compute a sentiment score (denoted as FinBERT-N) for each headline. The sentiment score is calculated by subtracting the probability of the negative class from the probability of the positive class.

4.3.3 Sentiment Classification with ChatGPT

The primary objective of this chapter is to investigate the potential of ChatGPT in the domain of financial sentiment analysis. Given its extensive training on a

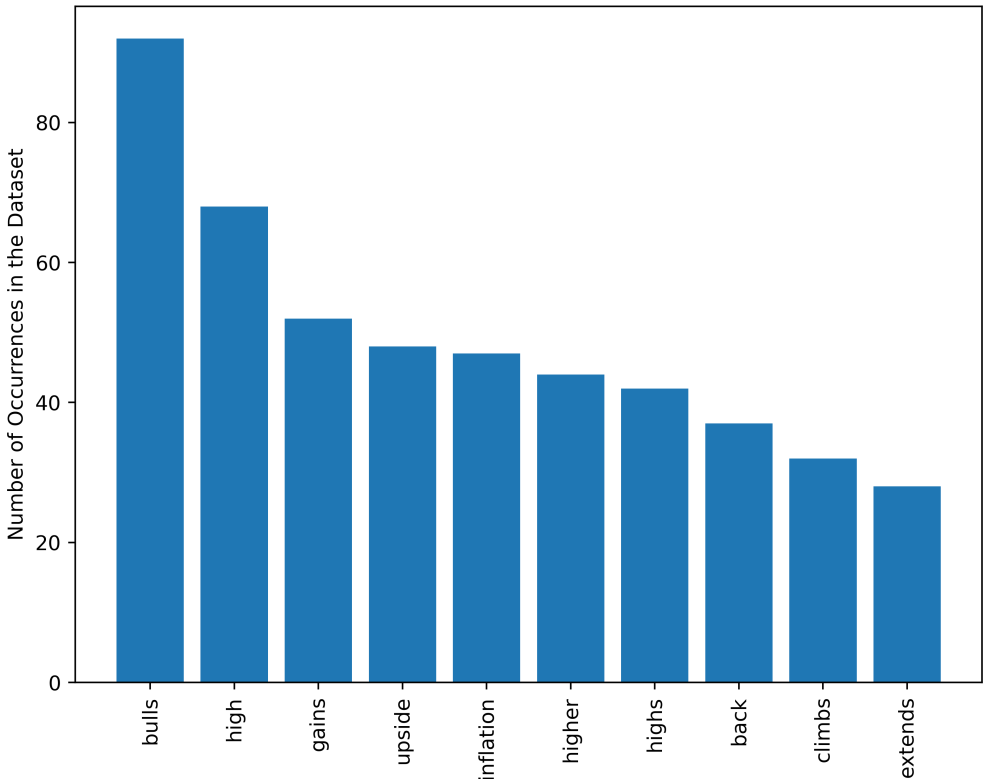


Figure 4.5: Most Common Tokens on Positive Headlines

diverse range of texts, ChatGPT is anticipated to exhibit an advanced contextual understanding, which is crucial for interpreting sentiment in financial news. The rationale behind utilizing ChatGPT lies in its capability to grasp not only the literal meanings of words but also the underlying implications, idioms, and sentiments that may be specific to a particular industry or topic. For example, in the FX market, a news item might convey a positive sentiment for one currency (e.g., JPY) but a negative sentiment for the corresponding FX pair (e.g., USDJPY). This nuanced contextual understanding, which is vital for most sentiment analysis tasks, is where ChatGPT could potentially surpass traditional language models like BERT.

To assess ChatGPT's capabilities in financial sentiment analysis, a series of ex-

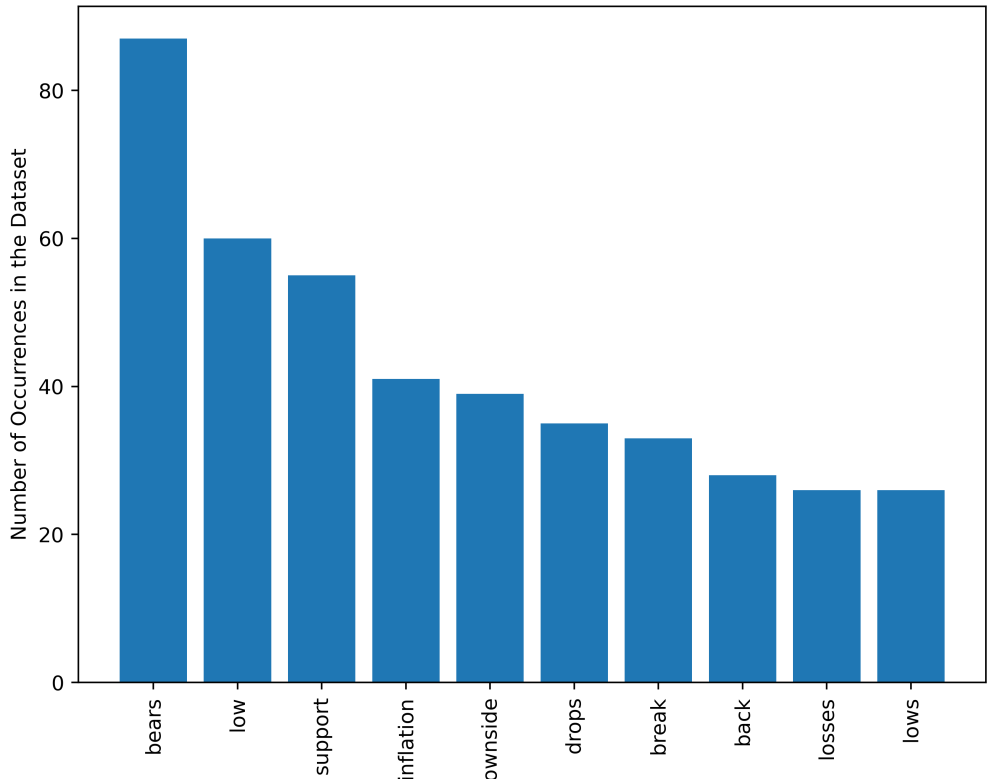


Figure 4.6: Most Common Tokens on Negative Headlines

periments were designed and conducted using a zero-shot prompting approach. Zero-shot prompting leverages ChatGPT's pre-existing training and comprehension of language patterns and context to generate desired responses without any additional task-specific fine-tuning or further training. This approach allows for a direct evaluation of ChatGPT's inherent capabilities in sentiment analysis, highlighting its potential for immediate integration and application in various scenarios.

Table 4.3 outlines the prompts used for sentiment classification. These prompts range from simulating the perspective of a financial analyst evaluating news (GPT-P1 and P2) to that of a sentiment analysis AI model, such as FinBERT, assessing a headline (GPT-P3), a FX trader reacting to market updates (GPT-

Table 4.2: FX Dataset Statistics

FX Pair	No of Articles	Daily Articles	Headline Tokens	Article Tokens
AUDUSD	461	5.36	11.51	288.42
		(3.97)	(2.85)	(149.4)
EURCHF	55	0.64	12.58	162.62
		(0.92)	(3.06)	(98.9)
EURUSD	758	8.81	11.56	255.87
		(7.08)	(2.9)	(180.53)
GBPUSD	518	6.02	11.95	263.1
		(4.67)	(2.55)	(148.29)
USDJPY	499	5.8	11.68	259.38
		(4.29)	(2.88)	(142.82)
Total	2291	26.64	11.69	262.58
		(21.71)	(2.82)	(158.99)

Note: Values in parenthesis represent standard deviations.

P4), and prompts designed to evaluate the sentiment of all available daily news per FX pair (GPT-P5) and all available daily news (GPT-P6). Additionally, variations of these prompts were implemented to require numerical sentiment output in the form of a sentiment score (GPT P1N-P6N) ranging from $[-1,1]$ instead of a sentiment class label.

Another variant, applied to both FinBERT and GPT-P4 (FinBERT-A, FinBERT-AN, GPT-P4A, and GPT-P4AN), was established to analyze model performance in sentiment analysis when both the article text and news headline are provided. This aims to determine whether the inclusion of the article's text—serving as additional context—enhances the model's accuracy.

Prompts GPT-P6 and GPT-P6N were specifically designed to ask ChatGPT to process a set of headlines and return a collective sentiment in a structured format, such as JSON. Evaluating this feature is crucial to understanding ChatGPT's adaptability and its potential for seamless integration into existing platforms. This capability is particularly important for practical scenarios where the generated outputs are directly utilized by other systems or services.

Table 4.3: Experimental ChatGPT Prompts for Sentiment Classification

Prompt Abbr.	Prompt
GPT-P1	Act as a financial expert holding {ticker}. How do you feel about the headline {headline}? Answer in one token: positive, negative, or neutral.
GPT-P2	Act as a financial expert. Classify the sentiment for {ticker} based only on the headline {headline}. Answer in one token: positive, negative, or neutral.
GPT-P3	Act as a sentiment analysis model trained on financial news headlines. Classify the sentiment of the headline {headline}. Answer in one token: positive, negative, or neutral.
GPT-P4	Act as an expert at FX trading holding {ticker}. Based only on the headline {headline}, will you buy, sell or hold {ticker} in the short term? Answer in one token: positive for buy, negative for sell, or neutral for hold position.
GPT-P5	Act as an expert at FX trading holding {ticker}. Based only on the following list of headlines {ticker_daily_headlines}, will you buy, sell or hold {ticker} in the short term? Answer in one token: positive for buy, negative for sell, or neutral for hold position
GPT-P6	Act as a sentiment analysis service of a financial platform. Based only on the following list of headlines {all_daily_headlines}, provide a summary of the daily sentiment for the FX pairs: {tickers}. Provide only the sentiment per FX pair in JSON format eg {'USDJPY': 'positive', 'EURUSD': 'neutral'}. The sentiment can be positive for buy, negative for sell, or neutral for hold position.

4.3.4 Experimental Setup

The sentiment analysis experiments were conducted using the GPT-3.5-turbo model available through OpenAI's Python library [Brown et al., 2020]. Each prompt was submitted to the model as a single user instruction, with several parameters fine-tuned to optimize the model's output depending on the prompt. For instance, the *max_tokens* parameter in GPT P1-P5 was set to 1 to limit the model's response to a single token. For prompts GPT P1N-P4N, P5N, P6-P6N, the *max_tokens* parameter was set to 10, 20, and 200 tokens respectively, based on the expected length of the response. Subsequent processing was applied to the response text to extract the necessary information for each prompt. For example, a simple function with a regular expression was developed to extract

the JSON (JavaScript Object Notation) from P6's response. The *temperature* parameter was set to 0.2 to encourage the model to generate more deterministic outputs. It is important to note that these prompts were refined through extensive prompt engineering to ensure that GPT's responses included the required information (i.e., sentiment class or score).

For each prompt, the time taken to receive a response from the API was monitored to track task latency and evaluate GPT's efficiency. Additionally, the number of tokens used in the completion and the number of tokens in the prompt were logged for cost management and resource usage tracking.

The experiments were iteratively executed over the FX dataset described in Section 4.3.1. Error handling mechanisms were incorporated to address potential API exceptions or response errors, ensuring smooth execution of the iterative process. While running these sentiment analysis experiments on large datasets incurs costs in line with OpenAI's pricing structure, the insights gained from the analysis provide invaluable intelligence that significantly enhances decision-making in finance.

4.3.5 Evaluation Metrics

The evaluation strategy for sentiment classification in this chapter employed a dual approach, involving both a traditional evaluation based on comparison with the true sentiment class and a market-related model evaluation. The traditional method focused on models that predicted a sentiment class for each headline (i.e., FinBERT and GPT's P1-P4), where the predicted sentiment class was compared against the actual sentiment class annotated in the dataset. In contrast, the market-related evaluation was applied to all models under consideration, where the predicted sentiment classes were assigned corresponding integer codes. For FinBERT, the sentiment score was calculated based on the

predicted class probabilities, as explained in Section 4.3.2.

4.3.5.1 Evaluation Based on True Class

To assess the performance of the sentiment classification models, an evaluation framework was established using several key performance metrics, including accuracy, precision, recall, and F1-score. These metrics provide insights into how effectively the models predict sentiment classes as defined in our annotated dataset.

A particularly important metric used in this evaluation is the Sentiment Mean Absolute Error (S-MAE), represented by Equation 4.1, where N denotes the total number of instances or data points, y_i is the true sentiment class for the i-th instance, and $\hat{y}_i$ is the predicted sentiment class for that instance. S-MAE measures the average absolute difference between the integer values of the true sentiment classes and the predicted ones, with -1, 0, and 1 representing negative, neutral, and positive sentiments, respectively.

$$S-MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{4.1}$$

The use of S-MAE is particularly justified by its ability to penalize models that significantly misclassify sentiment. This is crucial in financial sentiment analysis, where the interpretation of sentiment can be complex and nuanced. For instance, a headline expressing a positive sentiment for a particular currency (e.g., USD) could imply a negative sentiment for its paired currency (e.g., EURUSD). Similarly, negative sentiment towards an economy following a central bank announcement might actually indicate positive market sentiment for the associated currency. These subtleties highlight the importance of penalizing both incorrect polarity predictions and misjudgments in sentiment intensity.

Additionally, the evaluation includes a comparative analysis of different models.

Specifically, the performance of the FinBERT model is compared with that of ChatGPT under various prompts (P1-P4). This comparison aims to reveal the influence of prompt selection on ChatGPT's performance, offering a deeper understanding of how different models perform in sentiment classification tasks.

4.3.5.2 Model Evaluation in Relation to the Market

Beyond traditional classification metrics, the models were also assessed in terms of their relevance to actual market returns. This evaluation approach is particularly important given the primary application of these models—to inform trading decisions in financial markets.

For this analysis, sentiment scores were aggregated on a daily basis for each FX pair. Specifically, the sentiment scores from all headlines related to a particular FX pair within a single day were summed to produce a daily sentiment score (refer to Equation 4.2). This aggregated score was then compared with the actual daily return of the corresponding FX pair using Pearson correlation. The correlation was calculated to determine whether the sentiment scores predicted by the models had any relationship with real market movements.

$$S_d = \sum_{i=1}^{N_d} s_i \tag{4.2}$$

Where:

- S_d is the total sentiment score for day d.
- N_d is the number of headlines related to the FX pair on day d.
- s_i is the sentiment score of the i^{th} headline on day d.

In addition to assessing correlation with market returns, the models were also evaluated on their ability to correctly predict the direction of market movement, which is a crucial factor in financial decision-making [Blaskowitz and Herwartz,

2011]. This metric, known as Directional Accuracy (DA), measures the percentage of days where the predicted sentiment direction (positive/negative) matches the actual direction of market returns. For instance, if the model accurately predicted the market movement direction on 65 out of 100 days, the model's DA would be 65%.

For this market-related evaluation, the sentiment score for prompts GPT-P1 through P6 was treated as an integer value corresponding to the predicted sentiment class. This approach allowed for the sentiment score to be quantified and aggregated in a manner that reflects its intended use—as an indicator of the overall daily sentiment towards a specific FX pair.

To provide a comprehensive benchmark, both a naive baseline for DA and the DA based on the true sentiment were calculated. The naive baseline simply predicts the most common sentiment direction for each FX pair in the annotated dataset. This baseline offers a fundamental comparison point, especially given the imbalances often found in real-world data. Meanwhile, the DA of the true sentiment provides a direct measure of how well sentiment aligns with market movements in the dataset.

This evaluation encompassed all models (FinBERT, GPT P1-P6, and P1N-P6N) and, by integrating DA with Pearson correlation, offered a more complete perspective on the models' performance in a financial setting. The findings highlight the significance of advanced sentiment analysis methods, particularly when contrasted with the naive baseline, in capturing the complexities of financial market behavior.

4.4 Experimental Results

This section presents the outcomes from the in-depth analysis and experiments conducted to evaluate ChatGPT's effectiveness in financial sentiment analysis.

A detailed discussion is provided, comparing the performance of different ChatGPT prompts against the established FinBERT baseline across various metrics. The code used to reproduce these results is available on GitHub[4].

4.4.1 Sentiment Classification Performance

Table 4.4 illustrates the sentiment classification performance across various models, including FinBERT and different ChatGPT prompts (GPT-P1 to P4) applied to our dataset. The evaluation criteria include metrics such as Accuracy, Precision, Recall, F1-Score, and S-MAE.

The results reveal that GPT models consistently outperform FinBERT in all metrics. FinBERT, a specialized language model for financial text, shows an accuracy of 0.561, a precision of 0.560, and a recall of 0.562. Its F1-Score stands at 0.556, and it produces an S-MAE of 0.540. While these figures are respectable, they are significantly outperformed by the GPT models. Notably, GPT-P3, which emulates FinBERT as a sentiment analysis model trained on financial news headlines, performs well but falls short of other GPT models that incorporate FX pair information into their prompts.

GPT-P1, designed to act as a financial expert analyzing sentiment in headlines while considering the related FX pair, also performs well in sentiment classification. However, it slightly lags behind GPT-P2 by about 8%, indicating that while the emotion-based approach of P1 is effective, the sentiment-focused approach of P2 is more precise in this context.

GPT-P2 and P4, which function as financial experts factoring in the related FX pair during sentiment analysis, demonstrate significant improvements in sentiment classification. GPT-P2 leads with the highest accuracy, recall, and F1-Score, all around 0.790. GPT-P4, modeled as a FX trader making decisions

[4]https://github.com/giorgosfatouros/Financial-Sentiment-Analysis-with-ChatGPT

based on headline sentiment, exhibits similar performance metrics to GPT-P2, despite the different prompt framing. This alignment with the dataset annotation suggests that role-specific prompts, such as those used in GPT-P4 and GPT-P2, are particularly effective.

Table 4.4: Performance Results in Sentiment Classification

Model	**Accuracy**	**Precision**	**Recall**	**F1**	**S-MAE**
FinBERT	0.561	0.560	0.562	0.556	0.540
GPT-P1	0.730	0.760	0.730	0.725	0.300
GPT-P2	**0.790**	0.797	**0.790**	**0.790**	0.227
GPT-P3	0.735	0.780	0.735	0.737	0.282
GPT-P4	0.784	**0.804**	0.784	0.789	**0.221**

Table 4.5 provides a more detailed breakdown of the models' performance, evaluating their precision, recall, and F1-score for each sentiment class: Positive, Negative, and Neutral. For the Positive sentiment class, GPT-P1 achieved the highest precision, while GPT-P4 led in recall and F1-score. In the Negative sentiment class, GPT-P4 demonstrated the highest precision, with GPT-P2 excelling in recall and F1-score. For the Neutral sentiment class, GPT-P2 outperformed in precision, while GPT-P3 had the highest recall. GPT-P2 and GPT-P4 both achieved the top F1-score for Neutral sentiment. This detailed analysis highlights the strengths and weaknesses of each model across different sentiment classes, which is crucial for applications where accurate classification of specific sentiment classes is more critical than overall performance.

Furthermore, as shown in Table 4.6, the performance of GPT models varies depending on the specific FX pair. For the AUDUSD pair, GPT-P2 shows superior performance in accuracy, recall, F1-score, and S-MAE, while GPT-P4 leads in precision. For the USDJPY pair, GPT-P2 excels in all performance metrics. The evaluation of the EURCHF pair reveals a split in performance dominance, with GPT-P2 leading in accuracy, recall, and F1-score, while GPT-P4 excels in precision and S-MAE. Notably, GPT-P4 outperforms in all metrics for both the EURUSD and GBPUSD pairs, highlighting its effectiveness in sentiment

Table 4.5: Performance Metrics for Individual Sentiment Classes Across Models

	FinBert	P1	P2	P3	P4
Positive					
Precision	0.59	**0.91**	0.87	0.89	0.88
Recall	0.61	0.54	0.71	0.59	**0.79**
F1-score	0.60	0.68	0.79	0.71	**0.82**
Negative					
Precision	0.56	0.74	0.81	0.87	**0.88**
Recall	0.68	**0.90**	0.88	0.72	0.76
F1-score	0.61	0.81	**0.85**	0.79	0.81
Neutral					
Precision	0.53	0.64	**0.71**	0.60	0.66
Recall	0.41	0.77	0.78	**0.88**	0.83
F1-score	0.46	0.69	**0.74**	0.72	**0.74**

analysis related to these FX pairs.

Table 4.6: Best Performing Model in Sentiment Classification per FX pair

FX Pair	Accuracy	Precision	Recall	F1	S-MAE
AUDUSD	P2	P4	P2	P2	P2
EURCHF	P2	P4	P2	P2	P4
EURUSD	P4	P4	P4	P4	P4
GBPUSD	P4	P4	P4	P4	P4
USDJPY	P2	P2	P2	P2	P2

It is noteworthy that 75.6% of the headlines in the dataset explicitly mention the related FX pair at the beginning. This suggests that the performance difference between prompts such as GPT-P1, P2, and P4, which incorporate FX pair information, could be even more pronounced with more complex headlines. To further investigate, the models were tested on a subset of the data containing headlines that do not explicitly mention the associated FX pair. This assessment provides insights into how well each model handles sentiment analysis in a more challenging context where the relevant topic (i.e., FX pair) is not directly stated. The results from this analysis, presented in Table 4.7, show that GPT-P4 continues to outperform, achieving the highest scores across all metrics. This finding highlights GPT-P4's ability to interpret implicit context

within headlines.

In contrast, FinBERT's performance slightly decreases when the FX pair is not directly mentioned, with accuracy, precision, recall, F1-score, and S-MAE at 0.543, 0.543, 0.543, 0.538, and 0.539, respectively. GPT-P1, P2, and P3 also show a dip in performance compared to the complete dataset results. However, even with reduced performance, GPT-P1, P2, and P3 still outperform FinBERT, underscoring the superiority of GPT models in handling complex scenarios. Despite the decrease in performance, GPT-P2 still achieves high scores with an accuracy of 0.711, precision of 0.728, recall of 0.711, F1-score of 0.714, and S-MAE of 0.326, demonstrating its robustness in sentiment classification.

This analysis underscores the strength of GPT models, particularly GPT-P4, in sentiment analysis tasks. These models' ability to maintain high performance even in more challenging scenarios makes them promising tools for real-world applications. Importantly, these results also emphasize the significance of prompt design in guiding ChatGPT. By carefully crafting prompts tailored to the task, the model's effectiveness in sentiment analysis can be significantly enhanced.

Table 4.7: Performance Results in Sentiment Classification (filtered data)

Model	Accuracy	Precision	Recall	F1	S-MAE
FinBERT	0.543	0.543	0.543	0.538	0.539
GPT-P1	0.663	0.715	0.663	0.672	0.410
GPT-P2	0.711	0.728	0.711	0.714	0.326
GPT-P3	0.665	0.657	0.665	0.657	0.384
GPT-P4	**0.765**	**0.772**	**0.765**	**0.763**	**0.249**

4.4.2 Sentiment Score Relation to the Financial Market

The second evaluation method assesses the alignment between the sentiment scores predicted by each model and actual market price movements. This comparison aims to evaluate the potential utility of these models in a trading

context by analyzing how closely their sentiment scores correspond to the daily returns of the respective FX pairs. The process for calculating the overall daily sentiment score is outlined in Section 4.3.5.2. This comparison is conducted using Pearson correlation and DA, as discussed in Section 3.3.2.3.

4.4.2.1 Correlation with Market Returns

The Pearson correlation coefficient is a crucial metric for evaluating the effectiveness of the models in predicting sentiment that aligns with market returns. This coefficient ranges from -1 to +1, where a higher positive value indicates a stronger relationship between sentiment scores and market returns. A higher positive correlation suggests that as the sentiment improves, market returns also tend to increase. Therefore, the model with the highest correlation between its sentiment scores and market returns can be considered the most accurate in reflecting the market's reaction to sentiment changes.

The correlation matrix shown in Figure 4.7 reveals the relationships among various models, the annotated sentiment, and the daily market returns calculated using pairwise correlation on our dataset aggregated on a daily basis as discussed in Section 4.3.5.2. It is expected that GPT-P4 demonstrates the highest correlation with the true sentiment, as P4 closely mirrors the approach used for annotating sentiment in the dataset. Similarly, FinBERT shows the strongest correlation with GPT-P3, which mimics a sentiment analysis model trained on financial data. However, a particularly interesting observation is that GPT-P6N exhibits a higher correlation with market returns than the true sentiment itself. This highlights GPT's ability to process all daily news collectively, enhancing its capacity to more accurately gauge market sentiment. Models that generate sentiment scores ranging from -1 to 1, rather than performing sentiment classification, tend to align more closely with market movements. This alignment is plausible because these models are better equipped to capture subtle variations

in news sentiment, whether slightly positive or negative.

Interestingly, while the GPT models that consider entire article texts (specifically, GPT-P4A and GPT-P4AN) did not outperform their counterparts focusing solely on headlines, the opposite was true for FinBERT, which showed improved performance when given a broader context.

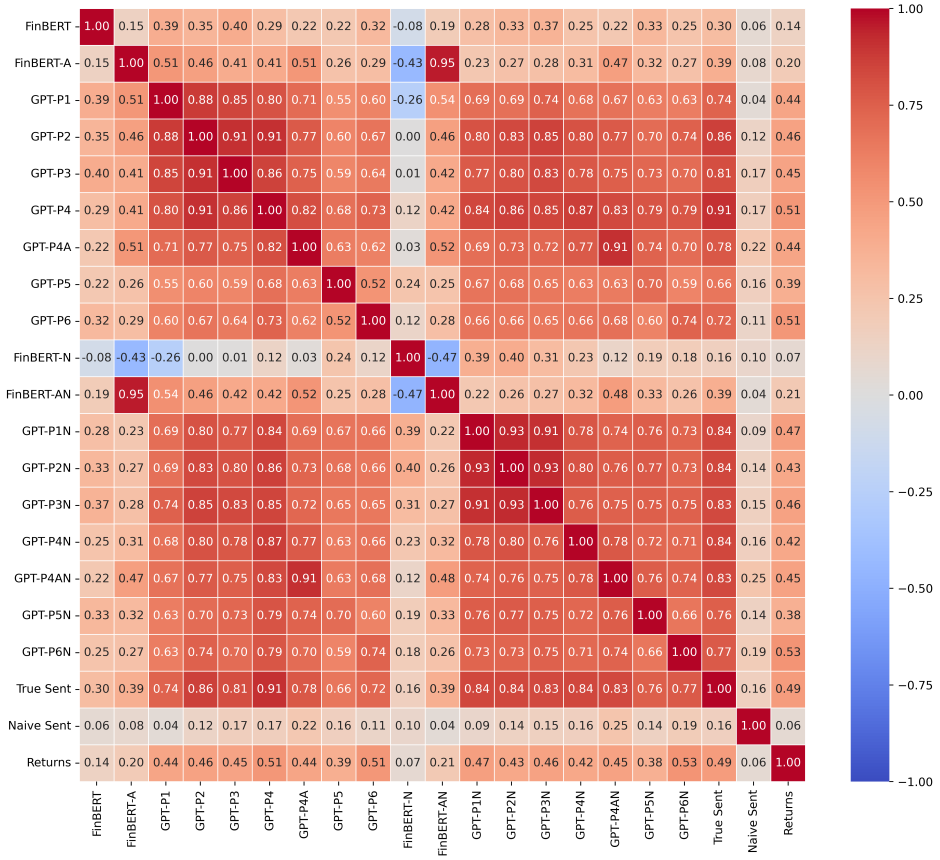


Figure 4.7: Correlation Matrix of Predicted Sentiment and Market Returns

It's also important to note that GPT-P5 exhibited the lowest correlation with both daily returns and true sentiment, suggesting that this prompt may not be as effective for sentiment prediction in relation to market price forecasting as the others.

Moreover, the high correlation between the categorical and numerical versions of the models (GPT-P1 and P1N through GPT-P6 and P6N) indicates that these different versions largely agree on the sentiment of the text, demonstrating the consistency of GPT's outputs.

The correlation between true sentiment and daily returns is approximately 0.49, indicating a moderate linear relationship. This suggests that while sentiment can explain some of the variations in market prices, other factors and market dynamics are also likely influencing these movements.

Table 4.8 provides a more detailed breakdown of the correlation coefficients between predicted sentiments by various models and returns for individual FX pairs. Although different models show varying degrees of correlation strength with different currency pairs, the pair-specific correlations for each model generally reflect the trends highlighted in Figure 4.7. A notable and consistent observation across all models is the low or negative correlation with EURCHF pair returns, which is likely due to the limited volume and scarcity of news related to EURCHF.

4.4.2.2 Directional Accuracy of Sentiment Scores

Tables 4.9 and 4.10 provide an overview of the DA achieved by various sentiment scoring models, alongside comparisons with a naive baseline and the DA based on true sentiment. DA measures the percentage of instances where the predicted sentiment direction aligns with the actual direction of market movement.

For reference, the naive approach, which simply predicts the most frequent sentiment direction for each FX pair, achieves a DA of 52.1%. This figure serves as a baseline for evaluating the models. In contrast, the DA based on true sentiment is significantly higher, standing at 69.6%.

Table 4.8: Correlation of Predicted Sentiment and FX pair returns

Model	**AUD/ USD**	**EUR/ CHF**	**EUR/ USD**	**GBP/ USD**	**USD/ JPY**
FinBERT	0.004	-0.214	0.192	0.206	0.209
FinBERT-N	-0.206	-0.028	0.155	0.172	0.113
FinBERT-A	0.282	-0.134	0.169	0.051	0.448
FinBERT-AN	0.313	-0.220	0.122	0.085	0.505
GPT-P1	0.401	-0.025	0.461	0.398	0.659
GPT-P1N	0.353	**0.235**	0.547	0.461	0.627
GPT-P2	0.413	-0.060	0.446	0.441	0.659
GPT-P2N	0.317	-0.061	0.476	0.438	0.606
GPT-P3	0.405	-0.118	0.440	0.460	0.619
GPT-P3N	0.390	0.035	0.464	0.451	0.650
GPT-P4	0.508	0.061	0.519	0.526	0.641
GPT-P4N	0.466	0.058	0.388	0.414	0.542
GPT-P4A	0.499	0.004	0.473	0.335	0.646
GPT-P4AN	0.532	0.0	0.466	0.342	0.581
GPT-P5	0.132	0.131	0.428	0.335	**0.669**
GPT-P5N	0.335	0.132	0.416	0.404	0.448
GPT-P6	0.580	0.072	0.461	0.527	0.628
GPT-P6N	**0.589**	-0.098	**0.585**	**0.557**	0.620
True Sent	0.517	0.173	0.425	0.496	0.650

Upon reviewing the tables, GPT-P1N emerges as the model with the highest DA, reaching 67.2%, indicating its strong capability in accurately predicting the direction of market movement based on sentiment scores. GPT-P3N and GPT-P6N follow closely, with DA values around 66%. Among the categorical models, GPT-P5 leads with the highest DA, closely trailed by GPT-P6 and GPT-P4. Overall, the numerical models generally demonstrate superior DA compared to their categorical counterparts, highlighting their enhanced ability to capture directional shifts in market sentiment. Notably, the GPT models exhibit a performance level that is nearly comparable to the human-annotated sentiment, underscoring their effectiveness in this task.

Table 4.11 presents the DA of numerical sentiment analysis models across different FX pairs. The results reveal that models GPT-P1N through GPT-P3N demonstrate consistent performance across all FX pairs, suggesting that,

Table 4.9: DA of Non-Numerical and Naive Models

Model	DA
FinBERT	0.595
FinBERT-A	0.498
GPT-P1	0.583
GPT-P2	0.607
GPT-P3	0.607
GPT-P4	0.640
GPT-P4A	0.587
GPT-P5	**0.652**
GPT-P6	0.640
Naive Sent.	0.521

Table 4.10: DA of Numerical and True Sent. Models

Model	DA
FinBERT-N	0.599
FinBERT-AN	0.526
GPT-P1N	**0.672**
GPT-P2N	0.644
GPT-P3N	0.660
GPT-P4N	0.611
GPT-P4AN	0.648
GPT-P5N	0.623
GPT-P6N	0.652
True Sent.	0.696

despite the different instructions in their prompts, these models capture similar sentiment aspects from the headlines.

GPT-P4AN, which integrates the entire article content, stands out as the top performer for the AUDUSD pair. However, for the other FX pairs, its performance is comparable to that of the GPT models that rely solely on headline analysis.

For the EURUSD and GBPUSD pairs, GPT-P6N significantly surpasses other models in performance. This superior result may be attributed to GPT-P6N's capability to process an extensive list of daily headlines, thereby effectively capturing the collective sentiment, which could be more reflective of the market

movements for these specific pairs.

However, for the EURCHF pair, the DA generally tends to be lower. This could be due to the relatively limited amount of data available for this pair, highlighting the crucial importance of data availability in predicting market direction accurately. Although FinBERT, when including the entire article content, achieves the highest DA for EURCHF, the small sample size makes it difficult to draw definitive conclusions about the significance of incorporating broader context.

Overall, these findings underscore the complex relationship between news sentiment and market movements, emphasizing the need for thorough testing across diverse datasets to validate models. They also highlight the importance of understanding the unique dynamics and trends of each market to tailor sentiment analysis approaches accordingly.

Table 4.11: Directional Accuracy Results for each Numerical Model per Ticker

Model	AUDUSD	EURCHF	EURUSD	GBPUSD	USDJPY
FinBERT	0.537	0.469	0.538	0.547	0.482
FinBERTA	0.481	**0.562**	0.481	0.491	0.625
GPT-P1	0.648	0.500	0.673	0.679	**0.696**
GPT-P2	0.648	0.438	0.577	0.679	0.679
GPT-P3	0.630	0.500	0.635	0.755	0.679
GPT-P4	0.537	0.188	0.500	0.660	0.536
GPT-P4A	**0.667**	0.312	0.615	0.660	0.679
GPT-P5	0.593	0.469	0.615	0.660	0.589
GPT-P6	0.593	0.344	**0.731**	**0.792**	0.607

Note: The results refer to the numerical (N) versions of the models providing a continuous sentiment value in [-1,1].

4.4.2.3 Performance and Cost Analysis

In the deployment of machine learning models, especially within production environments, both performance and cost are critical considerations. These factors play a significant role in determining a model's scalability, efficiency,

and overall practicality. Therefore, the evaluation of GPT prompts includes an analysis of processing time and token consumption, which directly impact operational costs and performance. Table 4.12 summarizes the average processing time and number of tokens processed by each prompt.

Table 4.12: Average Time and Tokens Processed per ChatGPT Prompt

Prompt	Pr. time	Pr. tokens	Hdln. time	Hdln. tokens
P1	0.94	63.0	0.94	63.17
P2	0.81	67.0	0.81	67.15
P3	0.97	66.0	0.97	65.91
P4	0.82	84.0	0.82	84.39
P4A	0.66	437.0	0.82	437.09
P5	0.83	193.0	0.11	24.74
P6	5.99	592.0	0.22	22.22
P1N	2.29	90.0	2.29	90.34
P2N	1.42	85.0	1.42	84.79
P3N	1.23	80.0	1.23	79.98
P4N	1.06	91.0	1.06	90.81
P4AN	0.71	494.0	1.06	494.09
P5N	1.85	201.0	0.24	25.68
P6N	6.10	600.0	0.23	22.54

Among the sentiment classification prompts, P1, P2, and P3 exhibit similar average times and token usage, generating concise responses in less than a second with approximately 65 tokens. The top-performing prompt, P4, generates more tokens, which could potentially lead to higher costs. In contrast, P5 and P6, while producing significantly more tokens, benefit from processing multiple headlines at once. This capability, along with their higher accuracy, can result in cost savings when handling large volumes of data. Prompts that generate numerical outputs generally require more time and tokens due to the increased complexity involved in generating accurate responses. Furthermore, experimental results indicate that the additional cost incurred by analyzing entire news articles—which can involve processing more than five times the number of tokens per prompt—does not necessarily correspond to a significant improvement in performance.

It is important to note that processing times can be influenced by various factors and may not always be consistent. The load on the ChatGPT model, which varies depending on the time of day, global usage patterns, and the location of the end-user, can significantly affect processing times. Therefore, while the average times presented are useful for relative comparisons, they may not always reflect real-time performance accurately, and thus should be interpreted with caution.

Considering that the cost of using OpenAI's API for the ChatGPT-3.5 model is 0.002 USD per 1,000 tokens, the financial implications of integrating ChatGPT into existing services appear to be relatively low, even when processing large volumes of data daily. For instance, Bloomberg News publishes approximately 5,000 articles per day [Bloomberg, 2023].

To illustrate, using the P6N model—which generates the highest number of tokens among the models tested—the daily cost can be estimated as follows: 600 tokens (average generated by P6N per article) multiplied by 5,000 articles, then multiplied by 0.002 USD per 1,000 tokens. This calculation results in a total daily cost of approximately 6 USD.

Therefore, even when handling large data volumes, such as those produced by Bloomberg, the cost of using advanced language models like ChatGPT for sentiment analysis remains quite affordable. This cost-effectiveness, combined with the strong performance demonstrated by these models, makes them a viable option for real-world financial applications.

For applications requiring real-time insights and enhanced accuracy, a mixed approach may be necessary. This approach could involve generating quick and concise responses for individual headlines using prompts P1-P4 (or P1N-P4N), while also processing larger batches of headlines with prompts P5, P6 (or P5N, P6N) when time allows.

4.5 Discussion

4.5.1 ChatGPT's Performance in Financial Sentiment Analysis

The findings from this research indicate that ChatGPT consistently surpasses the performance of the FinBERT model in financial sentiment analysis, regardless of the prompts employed. Notably, the prompts GPT-P4, P6, and P6N demonstrated significant effectiveness in sentiment analysis, offering valuable insights that could enhance market prediction capabilities. The strong performance of ChatGPT, coupled with the flexibility afforded by prompt engineering, suggests a high potential for further refinement and application optimization. For instance, by carefully framing the task, as seen with prompts GPT-P4 and P6, it is possible to boost ChatGPT's accuracy even further. However, it is essential to acknowledge that the effectiveness of different prompts varies, underscoring the need for continued optimization and extensive testing on larger datasets.

An intriguing observation was that GPT-P6N exhibited a stronger correlation with market returns than the actual sentiment itself, suggesting that the model might have a superior grasp of overall market sentiment when processing a full day's worth of news simultaneously. This capability may stem from ChatGPT's ability to detect subtle sentiment variations across multiple news articles, which could be more challenging for individual human annotators who review texts in isolation.

Furthermore, models that generate sentiment scores on a scale from -1 to 1 (N versions) demonstrated a closer alignment with market trends, indicating their proficiency in capturing the finer nuances of news sentiment that might be more reflective of market behavior. On the other hand, despite GPT-P5's approach of processing all daily news for each FX pair, it showed a weaker correlation with both market returns and true sentiment, highlighting the critical

importance of effective prompt engineering and thorough prompt evaluation before implementation.

The strong correlation observed across pairs such as GPT-P1 and P1N to GPT-P6 and P6N also highlights the consistency and dependability of GPT's outputs, positioning these models as promising tools for practical applications.

4.5.2 Potential Applications in Financial Services

The results of this research point to several practical applications within the financial services industry. By carefully selecting prompts, models like ChatGPT can be utilized for financial sentiment analysis, offering actionable insights that may assist in forecasting market trends. It is essential to recognize, however, that the optimal prompt may differ based on the specific use case, the financial instrument in question, and the acceptable margin of error in predictions.

It is also important to emphasize that the observed correlation between the model's sentiment scores and market movements does not inherently indicate an ability to predict future price changes. Financial markets are intricate systems influenced by numerous factors, including macroeconomic indicators, geopolitical events, and technical considerations. Understanding this complexity, a similar methodology to that employed in this study could be expanded by incorporating a broader set of inputs. Instead of relying solely on domain-specific news headlines, the model could be fed additional relevant data, such as economic indicators and other market information, potentially leading to even more comprehensive insights. Consequently, while sentiment analysis through language models like ChatGPT can provide valuable viewpoints, these models should be integrated into a more comprehensive approach to financial market analysis to maximize their effectiveness.

4.5.3 Limitations and Future Work

While this study presents promising results, it also uncovers certain limitations of ChatGPT and highlights areas for future research.

First, the limited duration of the dataset used in this study is a notable constraint. The analysis is based on a specific time frame, and while the outcomes are encouraging, the models' performance may differ when applied to data from other periods or under different market conditions. Financial markets are influenced by various factors over time, and the short duration of the dataset may not capture all these dynamics. This limitation suggests the need for further studies that cover longer periods to validate and generalize these findings.

Additionally, even though the models showed significant correlation with market sentiment, they did not fully align with market movements, indicating that sentiment explains only a portion of the variations in market prices. This observation highlights the complexity of financial markets and suggests that future research should aim to refine these models to better capture market behavior.

In the context of generative AI, potential issues like model collapse, particularly when relying heavily on synthetic training data, should be considered [Brock et al., 2018]. Such phenomena can cause models to produce repetitive or uniform outputs across different inputs [Goodfellow et al., 2014]. In this study, while prompts guided ChatGPT's responses, it's important to note that ChatGPT's foundational knowledge is built on extensive and diverse real-world text data, which reduces the reliance on synthetic data alone [Brown et al., 2020]. This approach helps mitigate the risk of overfitting to synthetic patterns [Zhang et al., 2021].

As the development of LLMs rapidly progresses, future research should focus on comparing and evaluating the unique capabilities of both existing and

emerging models. For instance, newer models like GPT-4 [OpenAI, 2023a] have the potential to further improve the performance demonstrated by ChatGPT in this study. Additionally, with the advent of open-source LLMs like META's Llama, it would be valuable to assess these models, their effectiveness in financial sentiment analysis, and their potential integration into commercial systems. Continuous exploration of advancements in LLMs is essential to fully harness their capabilities in the complex and dynamic field of financial services.

While the initial findings emphasize the strengths of models like ChatGPT in financial sentiment analysis, it is important to broaden the scope of research. Specifically, exploring how sentiment scores derived from ChatGPT influence market dynamics in subsequent days could provide insights into the temporal relationship between sentiment and market movements.

Regarding processing time, more extensive testing under varied conditions is required to establish more accurate and consistent response time measurements. For practical applications that utilize LLM services, such as the ChatGPT API, it would be prudent to implement mechanisms to manage potential variations in response time due to factors like system load fluctuations.

Overall, the findings of this study underscore the significant potential of ChatGPT in financial sentiment analysis. By addressing the identified areas for future research, more sophisticated and accurate sentiment analysis models for financial markets with real business value can be developed.

Chapter 5

AI on Diverse Data for Holistic Financial Analysis

Chapter Structure

This Chapter is constructed as follows:

- **Section 5.1 - Introduction**, introduces MarketSenseAI, an advanced AI-driven framework designed for holistic financial analysis, outlining its key components, objectives and the motivation behind its development in response to the challenges faced in contemporary financial markets.
- **Section 5.2 - Background on AI in Financial Analysis**, reviews the evolution of AI tools in financial analysis, highlighting the transition from sentiment analysis, as discussed in the previous chapter, to a more comprehensive market analysis approach.
- **Section 5.3 - Methodology**, details the architectural design of MarketSenseAI, including the integration of news summarization, fundamental analysis, and macroeconomic assessment, along with the data sources and experimental setup.
- **Section 5.4 - Experimental Results**, presents the outcomes of the em-

pirical tests conducted with MarketSenseAI, comparing its performance with established benchmarks and demonstrating its practical application in stock selection and market analysis.

- **Section 5.5 - Discussion**, interprets the results, evaluates the strengths and limitations of MarketSenseAI, and discusses its potential implications for financial markets.

Building on the insights gained from Chapters 2 and 4, where the potential of LLMs in knowledge mining and financial sentiment analysis was explored, this chapter transitions to a more holistic approach to financial market analysis. MarketSenseAI, the framework introduced here, leverages the strengths of sentiment analysis and integrates them with other critical aspects of financial analysis, including fundamental and technical analysis. The research in financial sentiment analysis with ChatGPT has laid the groundwork for this broader exploration, setting the stage for the advanced capabilities of LLMs in comprehensive financial market analysis. This chapter's focus shifts from sentiment analysis to the development of a sophisticated tool that provides actionable insights across various dimensions of the financial markets, revolutionizing financial engineering through the exclusive use of NLP.

5.1 Introduction

5.1.1 Background and Motivation

Capital markets are essential for the efficient allocation of capital within an economy, and the process of price discovery is crucial for maintaining the health and stability of the financial system [Kidwell et al., 2016]. This process is influenced by a complex array of factors, including company-specific data, sectoral trends, macroeconomic indicators, momentum effects, and political as

well as geopolitical events [Lewellen, 2002]. Market participants engage collectively in this intricate process, contributing to the overall efficiency of financial markets [Malkiel, 2003].

The selection of stocks operates as a mechanism of price discovery, wherein investors identify stocks that they believe are "mispriced" relative to the broader market, offering potential returns. This approach underpins the concept of value investing [Greenwald et al., 2020]. However, the notion of "mispricing" extends beyond simple valuation discrepancies; it can also encompass market perceptions of a stock's fair price, which may differ from its fundamental value. This can include expectations of future growth, often emphasized in "growth investing," where current fundamentals are sometimes overlooked. Additionally, stock selection is complicated by factors such as the rise of passive investing, capital flows, derivative-related activities, and macroeconomic conditions, all of which contribute to a financial system that is inherently probabilistic and often chaotic [Bouchaud et al., 2003].

Retail investors, in particular, face challenges in analyzing individual stocks due to their limited capacity to process information, susceptibility to behavioral biases, and lack of robust risk management skills. As a result, they may miss promising investment opportunities or expose themselves to unnecessary risks. Exchange Traded Funds (ETFs) present a practical solution for these investors, allowing them to engage with the broader market more effectively, a strategy often referred to as investing in "beta."

Small to medium-sized asset and wealth management firms also encounter difficulties in conducting detailed stock analysis due to resource constraints and a limited scope of selection. For these firms, ETFs offer an appealing option, providing a more diversified and manageable investment approach.

In contrast, larger professional firms are typically equipped with advanced technology, infrastructure, and skilled personnel, enabling them to conduct

superior analysis and risk management of their portfolios. These firms often employ teams of stock analysts, economists, and traders whose collective knowledge and expertise are directed toward identifying investment opportunities. However, despite these advantages, outperforming the market remains challenging. Large organizations face issues such as silos, poor communication, and conflicting incentives, which can hinder their ability to achieve consistent success [Weiss-Cohen et al., 2019].

Over the past 15 years, particularly following the 2008 financial crisis, there have been significant shifts in the structure and operation of capital markets, with lasting effects on price discovery. Key developments include:

1. **Central Bank Policies**: The 2008 crisis fostered a belief among market participants that central banks would intervene to stabilize markets using all available tools. However, excessive reliance on central bank interventions risks distorting market mechanisms and incentives, leading to the underpricing of risk, moral hazard, and potential systemic externalities [BIS, 2022].

2. **The Rise of Passive Investing**: ETFs facilitate "blind" participation in market-weighted indices, treating all included stocks equally regardless of their fundamental value. This can cause stocks to deviate significantly from their fair value, particularly those widely held by passive investors [Goyal and He, 2015].

3. **Impact of Retail Investors**: The increasing influence of retail investors, who have access to gamified, leveraged, and derivative-enabled trading platforms, has also disrupted price discovery. For example, Zero-Day Expiry options (0DTE), which expire within a single day, accounted for approximately 43% of the total S&P 500 options volume in 2022, compared to just 6% in 2017 [Brogaard et al., 2023]. Similarly, the phenomenon of "meme stocks," such as GameStop, where retail herd behavior drove

prices to extreme levels, exemplifies the impact of retail investors [Anand and Pathak, 2022].

These factors collectively disrupt the functioning of price discovery, reducing the incentives for investors to accurately assess risk and value assets. In this complex market environment, there is a pressing need for more sophisticated tools that can enhance the analytical capabilities of human decision-makers. Such tools are essential for navigating the increasingly data-rich and complex financial landscape, enabling investors to achieve precision and insight necessary for informed decision-making.

5.1.2 Potential of LLMs in Stock Selection and Financial Analysis

Pretrained foundational models have demonstrated exceptional versatility and effectiveness across various domains [Usha Ruby et al., 2024]. The emergence of LLMs like ChatGPT offers promising advancements in financial analysis and stock selection [Mao et al., 2024]. These advanced AI systems, trained on vast and diverse datasets, have shown the ability to replicate intricate aspects of human cognition, and in many cases, surpass them [OpenAI, 2023a].

LLMs can quickly analyze vast amounts of financial data, discerning detailed insights from earnings reports to macroeconomic studies, and processing unstructured data such as news articles and expert opinions more efficiently than human analysts [Guo et al., 2023]. This deep content analysis enables LLMs to identify patterns often overlooked in traditional analyses [Alshami et al., 2023].

Moreover, LLMs play a critical role in reducing biases in stock selection. Unlike human analysts, LLMs are not influenced by emotional or overconfidence biases, offering a more objective perspective in financial analysis [Tjuatja et al., 2023]. While some biases from training data may persist, LLMs significantly

diminish the impact of human biases, such as overconfidence or confirmation bias, on investment decisions [Abramski et al., 2023,Atreides and Kelley, 2023]. Furthermore, LLMs can process and analyze extensive financial data, surpassing the limitations of individual or team analysts.

Although these AI systems may outperform humans in specific tasks, their primary value lies in augmenting human capabilities. They serve as powerful tools that enhance decision-making, improve the quality of analysis, and increase overall productivity [Noy and Zhang, 2023]. For example, complex tasks like consolidating financial statements from multiple subsidiaries of a large corporation can be streamlined by an LLM-based system [Kim et al., 2023]. Such systems can highlight discrepancies, flag outliers, and provide executive summaries, tasks that would be time-consuming and error-prone if done manually.

Recent developments in the financial sector validate these observations. Major players such as JPMorgan and Bloomberg have launched AI-driven initiatives, including an AI-enabled advisory platform [CNBC, 2023] and a finance-centric LLM [Wu et al., 2023b]. Additionally, Morgan Stanley has utilized OpenAI's models to create a chatbot that assists financial advisors by leveraging the bank's extensive research data [OpenAI, 2023b]. Similarly, Broadridge, through its subsidiary LTX, introduced BondGPT, a chatbot powered by GPT-4 designed to assist institutional investors in bond trading [LTXtrading, 2023]. Although many major financial institutions like Goldman Sachs and BlackRock have dedicated AI departments working on specialized projects, the specifics of these innovations often remain proprietary and are not fully disclosed to the public.

5.1.3 Contributions

This chapter makes several key contributions to the integration of AI and financial analysis through the introduction of MarketSenseAI[1], an innovative service for stock analysis rooted in the power of LLMs. The primary contributions are as follows:

1. **A Novel LLM-Driven Investment Service**: MarketSenseAI represents a significant advancement by integrating various text data sources to provide holistic stock investment insights, surpassing traditional models that rely solely on quantitative data and models.

2. **Explainable Investment Signals**: MarketSenseAI generates actionable and interpretable investment insights, ensuring transparency and increasing user acceptance of AI-generated signals.

3. **Versatile Use Cases**: The design of MarketSenseAI allows for individual usage of service components, catering to diverse investor needs and offering flexibility and adaptability to different investment strategies.

4. **Integration of Macroeconomic Analysis**: Unlike traditional models, MarketSenseAI incorporates macroeconomic conditions directly into the model. This integration allows the system to account for factors such as interest rates and economic growth, which are typically considered separately in traditional approaches. The holistic view provided by MarketSenseAI enables more informed and accurate predictions, reflecting real-world complexities more effectively.

5. **Empirical Evaluation**: Demonstrating the reliability and statistical significance of its recommendations, MarketSenseAI's empirical testing on the S&P 100 stocks over a 15-month period showcases its superior performance, delivering excess alpha of 10% to 30% and achieving a cumulative

[1] MarketSenseAI is available at www.marketsense-ai.com.

return of up to 72%, while maintaining a risk profile comparable to the broader market.

6. **Superior Performance**: MarketSenseAI establishes itself as a pioneering tool in the integration of generative AI into financial analytics by outperforming high-performing indices and traditional strategies.

7. **An Independent Financial Advisor**: MarketSenseAI democratizes access to premium investment insights for retail investors, asset managers, and other stakeholders.

In summary, this paper pioneers the integration of multi-source data analysis with the cognitive capabilities of LLMs to redefine stock selection and portfolio management. Unlike traditional financial engineering methods such as pricing models, time series analysis, or sentiment indicators, MarketSenseAI utilizes NLP exclusively to analyze diverse data types (i.e., news, financial reports, and stock prices) and rank the generated stock signals. This comprehensive and interpretable methodology represents a significant advancement in financial analytics, demonstrating the potential of LLMs to disrupt the financial industry by providing actionable investment insights based entirely on text analytics.

At the core of MarketSenseAI, the LLM generates concise summaries from vast amounts of numerical and textual data, extracting crucial insights about a company's developments and stock potential. The architecture is designed to process complex and large datasets effectively by dividing tasks into manageable components, each focusing on specific aspects of the data. It then analyzes these summaries, considering the investment horizon, to make stock investment suggestions. Based on the explanations of these signals, the LLM ranks the stocks, allowing for the selection of the most robust signals. This multi-process approach, harnessing the summarization and analytical power of AI, offers a sophisticated tool for investors navigating the complexities of the stock market.

Furthermore, MarketSenseAI's modular architecture allows for diverse applications in the financial domain. Each component of this architecture provides specific insights, such as news, fundamentals, and macroeconomic summaries, which can be used independently. It can facilitate the construction of AI-based portfolios using the generated signals and their explanations, offering a revolutionary approach to asset management. This framework can be tailored to make personalized investment decisions, taking into account user preferences on risk, investment horizon, and goals. This adaptability demonstrates the framework's potential in various areas of finance, extending far beyond stock selection.

Overall, MarketSenseAI bridges domain knowledge with the latest AI advancements, providing a novel and applicable system in investment finance. This tool holds significant promise for asset managers and retail investors seeking advanced financial advice, especially those with limited resources and access to premium financial services.

5.2 Background on AI in Financial Analysis

The application of AI techniques in the financial sector has surged significantly over the past decade, gradually eclipsing traditional statistical or algorithmic approaches [OECD, 2021]. Various facets of the financial domain, including risk assessment [Fatouros et al., 2023b], banking services [Kotios et al., 2022], and trading operations [Bloomberg, 2019], have increasingly integrated AI technologies. Concurrently, the rise of foundational models, particularly the successive versions of GPT and their associated chat interfaces, has driven transformative changes across diverse sectors. The financial industry, in particular, has begun incorporating insights derived from these models into its business operations [Chui et al., 2023].

Despite the relatively recent public availability of these models — such as the ChatGPT API, which only became accessible in March 2023 — numerous research efforts have emerged, exploring how these Generative AI frameworks can enhance investment strategies.

Research by [Zaremba and Demir, 2023] highlights the potential of ChatGPT in finance, especially for tasks requiring NLP capabilities, such as sentiment analysis of financial news and summarization of earnings reports. These tasks show a significant correlation with stock market behavior [Tetlock et al., 2008]. Additionally, [Lopez-Lira and Tang, 2023] demonstrate ChatGPT's accuracy in financial news sentiment analysis, showing a positive correlation between ChatGPT-generated sentiment scores and subsequent stock returns. Under various investment strategies, ChatGPT was found to outperform traditional sentiment analysis methods, with news sentiment contributing to notable returns.

In comparative analyses, [Li et al., 2023] argue that ChatGPT and GPT-4 outperform domain-specific models like FinBERT [Araci, 2019] and BloombergGPT [Wu et al., 2023b] in tasks such as named entity recognition and news classification. While FinBERT excels in financial sentiment analysis compared to ChatGPT, the study lacked prompt engineering and used a dataset inherently favoring FinBERT. On the other hand, [Fatouros et al., 2023d] provide evidence that ChatGPT surpasses FinBERT in financial sentiment analysis, both in classification performance and correlation with actual returns, even when using zero-shot prompting. Zero-shot prompting allows ChatGPT to perform tasks without specific prior training, indicating its effectiveness in sentiment analysis based on broad-based training, despite not being explicitly trained on financial data.

Furthermore, [Kim et al., 2023] emphasize the utility of GPT-3.5 in summarizing corporate disclosures, suggesting that sentiment derived from these

summaries more accurately predicts stock market reactions than the original documents. This finding underscores the value of GPT models for investors seeking concise, targeted information. [Kirtac and Germano, 2024] extended this research, showing GPT-3.5's effectiveness in sentiment analysis for predicting next-day stock returns, outperforming FinBERT and lexicon-based models. Additionally, [Yu et al., 2023] and [Chen et al., 2023] explored sentiment indicators derived from LLMs, demonstrating how these indicators enhance forecasting models for stock movements, further indicating the superior performance of LLMs in predicting stock directions.

In summary, current research on LLMs in financial applications supports and reinforces the methodologies underlying each component of the proposed system. However, while these studies highlight the utility of LLMs in various financial tasks, few focus specifically on stock selection. Most existing research is limited to specific types of text data, such as news articles or tweets, with narrow use cases like sentiment analysis or text summarization, and often presents limited evaluation with less structured methodologies. In contrast, MarketSenseAI presents a framework that leverages multi-modal financial data, including news, financial reports, stock prices, and macroeconomic insights, to provide actionable and interpretable investment recommendations for analyzed stocks, while outperforming high-performing ETFs.

Unlike traditional methods that heavily rely on quantitative analysis, where sentiment indicators are used as features in predictive models, MarketSenseAI emphasizes language understanding and reasoning to generate investment insights after processing both numerical and textual data. MarketSenseAI's innovative approach also incorporates macroeconomic analysis through expert opinion, overcoming the limitations of traditional quantitative models that struggle to effectively use infrequent macroeconomic data or integrate expert perspectives. This approach enables the provision of detailed, AI-generated explanations for

each recommendation, enhancing the interpretability and trustworthiness of investment decisions. Additionally, the evaluation considers transaction costs and the number of trades, underscoring MarketSenseAI's applicability in real-world settings.

5.3 Methodology

The architectural framework of MarketSenseAI, as illustrated in Figure 5.1, integrates four primary components responsible for processing data inputs, alongside a fifth component dedicated to formulating the final investment recommendation (i.e., buy, hold, or sell) for a specific stock. This final component synthesizes all the gathered information and provides a clear explanation for the decision. Each component leverages OpenAI's API and employs the GPT-4 model [OpenAI, 2023a], utilizing zero-shot prompting and in-context learning to perform distinct tasks [Dong et al., 2022].

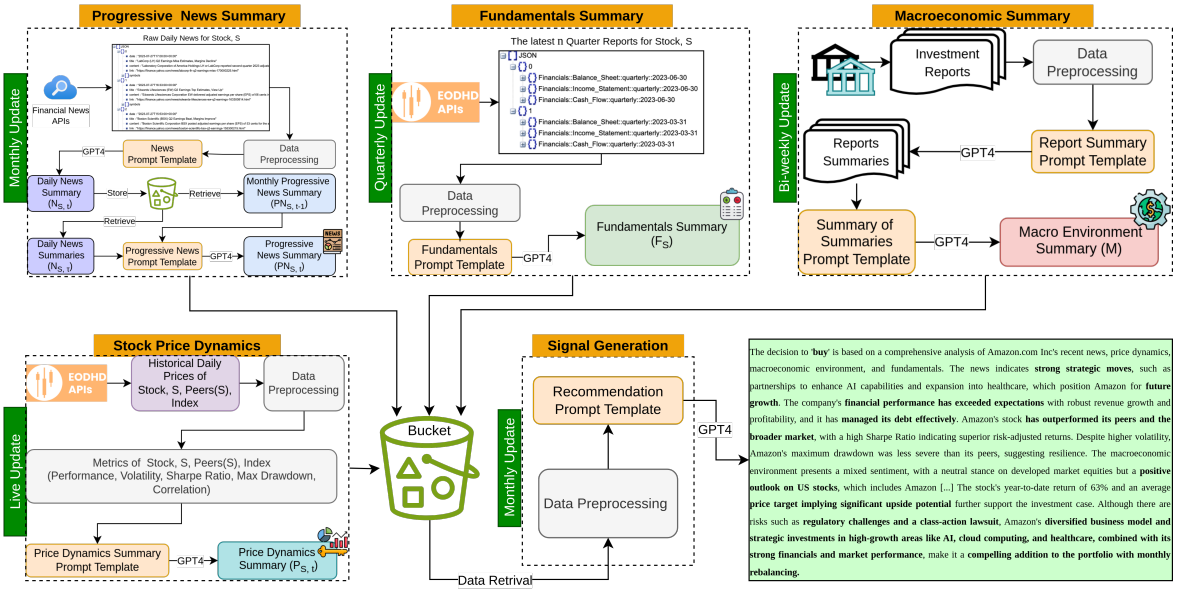


Figure 5.1: Conceptual architecture of MarketSenseAI, highlighting the core components, data flow, and outcome for a selected stock (e.g., Amazon).

The framework is structured to mimic the decision-making process of a profes-

sional investment team. This process encompasses monitoring recent developments related to the company or its sector (via a news summarizer), analyzing the company's latest financial statements (through a fundamentals summarizer), and performing a macroeconomic analysis of the current environment, while considering price action dynamics (through macro and price dynamics summaries). The overarching architecture includes the following components, with their data flow depicted in Figure 5.2:

1. **Progressive News Summarizer**: Gathers and condenses daily news items into ongoing summaries.

2. **Fundamentals Summarizer**: Preprocesses and examines quarterly financial data to create a summary of key financial metrics.

3. **Price Dynamics Summarizer**: Compares the target stock's performance with that of similar stocks and the broader market context.

4. **Macroeconomic Environment Summary (MarketDigest)**: Synthesizes investment reports and research articles from financial institutions into a concise macroeconomic summary.

5. **Signal Generation**: Integrates outputs from the above components to generate investment recommendations, accompanied by detailed justifications.

The subsequent sections provide a detailed description of each key component in this architecture.

5.3.1 Progressive News Summarizer

The impact of company-specific news—ranging from announcements, reports, analyst opinions, to research findings—on market sentiment and subsequent stock prices is significant [Malik, 2011]. Depending on the nature of the news,

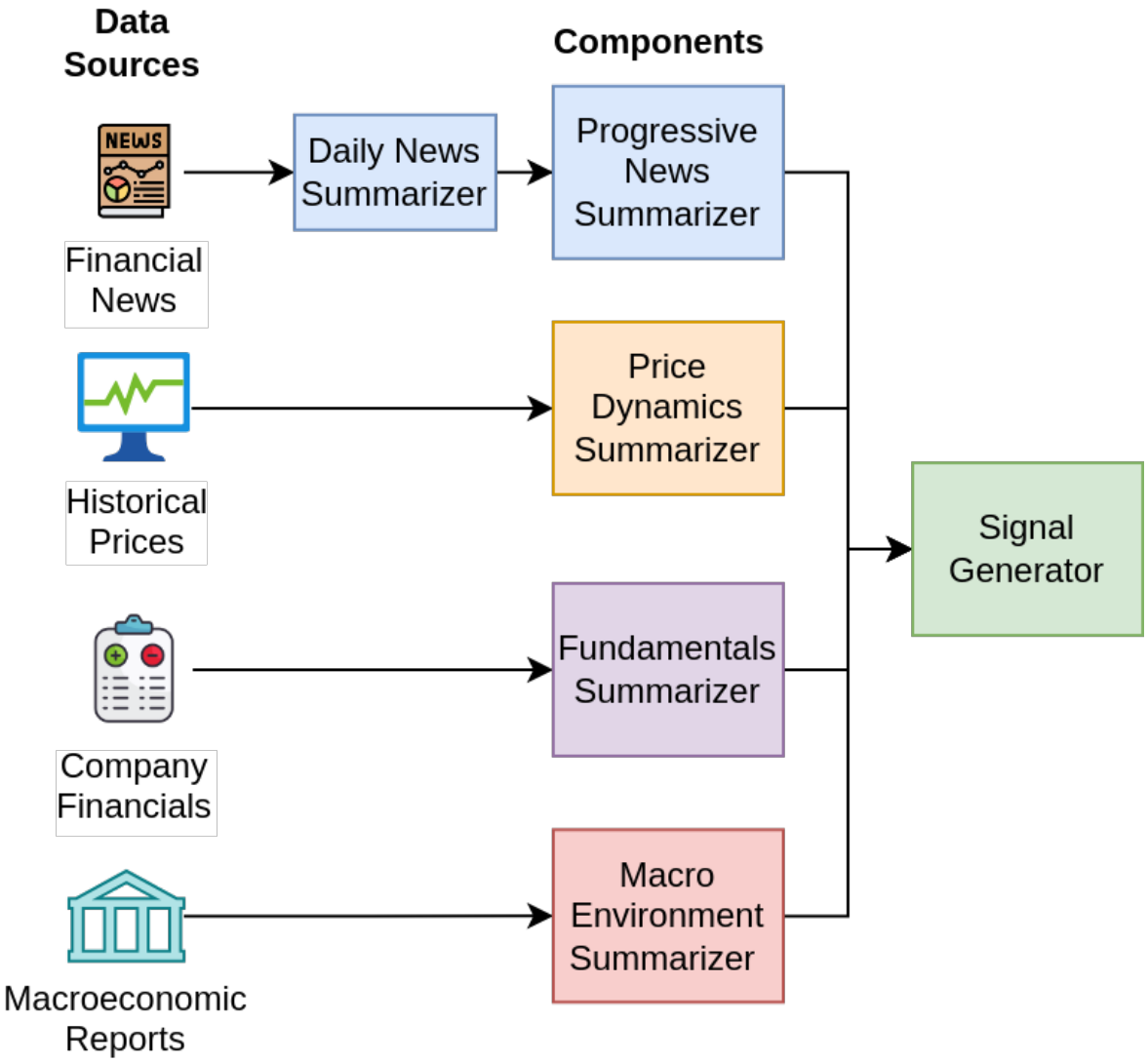


Figure 5.2: Data flow within MarketSenseAI.

its influence can be short-term, long-term, or minimal [Alqahtani et al., 2020]. Therefore, the sourcing and interpretation of news require careful management in the context of stock analysis.

The Progressive News Summarizer ($PN_{S,t}$) is tasked with acquiring news, condensing it, and creating a progressive synopsis of the most impactful news related to a particular stock. As depicted in Figure 5.3, daily news items relevant to a specific stock are retrieved from available APIs. In this study, the

EODHD[2] Stock Market and Financial News API was utilized for sourcing the news. The processes for generating daily and progressive news summaries are modeled by Equations 5.1 and 5.2, respectively.

$$N_{S,\tau} = \bigoplus_{i=t-\tau}^{t} \text{Summarize}(N_{S,i}) \tag{5.1}$$

$$PN_{S,t} = \text{Summarize}(PN_{S,t=t-1}, N_{S,\tau}) \tag{5.2}$$

Where $PN_{S,t}$ represents the Progressive News Summary for stock S at time t, and $N_{S,\tau}$ is the aggregated news summaries for stock S over the most recent τ days. Additionally, $N_{S,i}$ refers to the news articles available for stock S on day i. The parameter τ defines the number of days included in the aggregation, representing the time window for the progressive news summary.

The daily news for a company undergoes preprocessing to exclude unrelated text, such as clickbait articles, ensuring that the content is in an appropriate format for inclusion in the prompt. GPT-4, accessed via OpenAI's API, is then systematically prompted to distill the daily news and generate a concise summary ($N_{S,i}$) for each day. These summaries are stored in a centralized repository.

While this method produces a summary for a specific date, it is crucial to integrate the ongoing narrative of news-related content for the company, particularly older news that remains significant. For instance, in scenarios such as a merger or legal dispute, an announcement about the company's better-than-expected results might hold less weight in the overall decision-making process. The Progressive News Summarizer addresses this by merging the latest news summaries ($N_{S,\tau}$) with the preceding progressive summary ($PN_{S,t-1}$).

[2] https://eodhd.com/financial-apis/stock-market-financial-news-api/

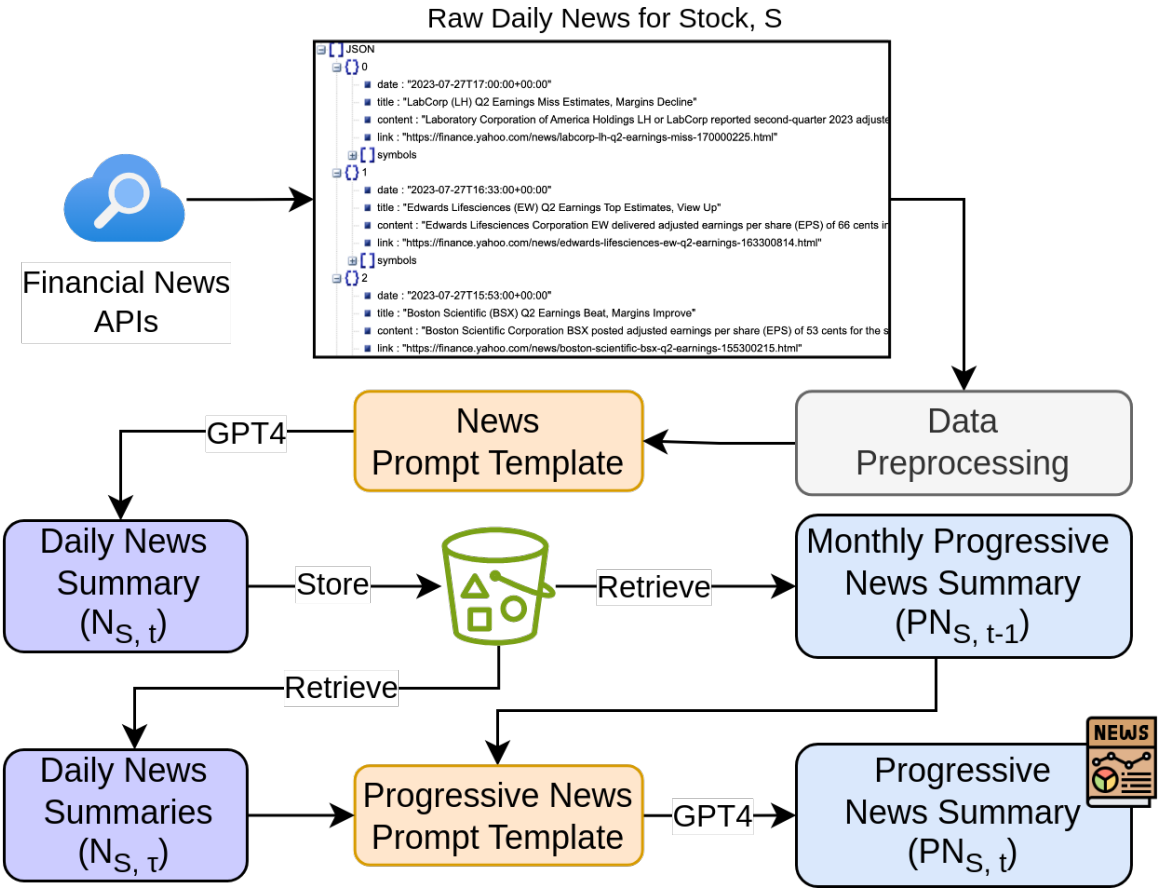


Figure 5.3: Progressive News Summarizer

Specifically, the prompt is structured to instruct GPT-4 to emulate the role of a financial analyst tasked with synthesizing an updated summary for a specific stock by integrating various types of information. The prompt includes:

- **Current Summary**: The most recent summary of the company and its stock as of a specific month and year.
- **Daily News Summary**: News articles divided into factual news and analysts' opinions concerning the company over a given month.
- **Instructions**: Directions to integrate the most relevant information, distinguishing between factual news and analysts' opinions.

This process ensures that the summary consistently reflects the latest, most

relevant, and significant developments, offering a comprehensive and up-to-date snapshot of the company's standing in the news.

Although this study's Progressive News Summarizer is tailored for monthly intervals, its design allows for adaptability to different frequencies, aligning with various investment strategies. Future research could explore the effects of adjusting the summarizer's frequency on the effectiveness of different investment approaches.

Table 5.1 illustrates how the Progressive News Summarizer effectively captures the evolving narrative around Apple Inc. (AAPL.US) across two different months in 2023. The summaries encompass a wide array of topics, from the company's ongoing financial performance to its strategic initiatives and market challenges.

In October, the focus was on Apple's significant role in the tech industry, marked by the launch of the iPhone 15 series and updates to its Apple Watch and AirPods. This period also highlighted challenges in smartphone sales, the impact of geopolitical issues, and fluctuations in stock performance. Unique to this month were reports on Apple's interest in acquiring Formula 1 broadcasting rights and the sale of company stock by CEO Tim Cook, underscoring the company's diverse strategic interests and executive decisions.

By November, while many earlier themes continued, new elements emerged. The summary highlighted Apple's sales slowdown, competitive pressures in the smartphone market, and its strategic shift in partnerships, including the termination of its credit card agreement with Goldman Sachs. Notably, the company's sustainability efforts and the launch of the M3 chip were highlighted, alongside its expansion in streaming content.

This table showcases the Progressive News Summarizer's ability to integrate the latest corporate developments, market dynamics, and strategic maneuvers. It

serves as a valuable tool for providing a comprehensive and up-to-date analysis of investment opportunities and industry trends. An example of a summary generated by this component is provided in Table 5.2.

Table 5.1: Apple Inc. Progressive News Summary (October vs November 2023)

Topic	October	November
Financial Performance	Yes	Yes
Guidance for Q4 2023	Yes	Yes
iPhone 15 Launch	Yes	Yes
New Apple Watch and AirPods	Yes	No
Smartphone Sales Challenges	Yes	Yes
Geopolitical Issues Impact	Yes	Yes
Stock Performance	Yes	Yes
Partnership with DuckDuckGo	Yes	No
CEO Tim Cook's Stock Sale	Yes	No
Interest in Formula 1 Broadcasting	Yes	No
Product Lineup including M3 Chip	No	Yes
Sales Slowdown and Competition	No	Yes
Appeal Win in UK	No	Yes
Partnership with Goldman Sachs	No	Yes
Regulatory Issues in Payment Apps	No	Yes

Table 5.2: Apple Inc. Progressive News Summary (November 2023)

Category	News Summary
Market Position	Dominant in the tech sector with record services revenue and robust product lineup, including the iPhone 15 series and new Mac products.
Sales Performance	Experiencing a slowdown, potentially dropping 5% in iPhone sales due to challenges in China and Japan.
Stock Analysis	Recent dip viewed as a buying opportunity by analysts, seasonal performance aligns with product launch cycle.
Legal and Strategic Moves	Won UK appeal on mobile browser and cloud gaming services, potentially ending credit card partnership with Goldman Sachs.
Innovation and Sustainability	Launched M3 silicon chip based on 3-nanometer technology, expanding streaming content on Apple TV+.
Regulatory Challenges	Faces oversight of digital wallets and payment apps, navigating geopolitical and economic risks.
Investment Consideration	Despite challenges, presents a promising opportunity with strategic expansion, innovative products, and strong services division.

5.3.2 Fundamentals Summarizer

Fundamental data is a cornerstone in predictive financial analytics, offering quantifiable metrics that reveal a company's current health and potential future trajectory. As shown in Figure 5.4, we source this quarterly information using EODHD's Fundamental Data API. To ensure consistency and clarity when comparing financial data, we preprocess the data before incorporating it into the prompt. This preprocessing involves a numerical abbreviation technique that converts large numbers into a more compact format, using prefixes like "million," "billion," or "thousand." For instance, numbers in the billions are formatted as 'X billion,' where X is the original number divided by one billion, rounded to two decimal places. This standardization is crucial for allowing GPT-4 to accurately compare and interpret complex financial figures.

Additionally, financial data from different quarters is arranged side by side in a table format. The resulting prompt, inputted into the GPT-4 model, focuses on aspects such as profitability, revenue trajectory, debt metrics, and cash flow dynamics by comparing the most recent quarterly financial statements. By emphasizing recent data, the LLM can detect shifts in financial performance, potentially correlating these with ongoing news developments. Specifically, MarketSenseAI models a company's financial condition using the following formula:

$$F_s = \text{Summarize}\left(\text{Standardize}\left(\bigcup_{i=1}^{n} \text{FD}_{s,q_i}\right)\right) \tag{5.3}$$

Where FD_{s,q_i} represents the financial data for a given stock s in quarter q_i, with $i = 1,\ldots,n$ indicating that the data spans the last n quarters. The Standardize() function is applied to this data, incorporating standardization techniques such as numerical abbreviation to ensure uniformity across different data sets.

The prompt structure guides GPT-4 to take on the role of a financial analyst focused on recent trends. The AI is tasked with evaluating the financial health of a specified company's stock by analyzing its latest quarters. The prompt includes:

- **Financial Tables**: Key financial data from the Balance Sheet, Income Statement, and Cash Flow Statement for the latest quarters.
- **Analysis Focus**: Emphasis on recent trends and developments in profitability, revenue growth, debt levels, and cash flow generation.
- **Instructions**: Conduct a bullet-point analysis.

Although LLMs traditionally encounter challenges in interpreting complex numerical data, our preprocessing approach, combined with GPT-4's capabilities, ensures accurate comparison and analysis. The fundamentals summarizer is designed to present an unbiased, factual overview of a company's financial status, avoiding any direct investment recommendations. Table 5.3 illustrates how the Fundamentals Summarizer distills key insights from financial statements, including income, balance, and cash flow statements. Like the Progressive News Summarizer, this component can be used independently to provide a concise financial overview of the company under analysis.

5.3.3 Stock Price Dynamics Summarizer

The Stock Price Dynamics Summarizer is a critical component of MarketSenseAI, designed to analyze and contextualize the price movements and financial metrics of stocks. As illustrated in Figure 5.5, this component not only evaluates the target stock but also compares its performance with the five most similar stocks, selected based on company description and sector, along with the broader market context represented by the S&P 500 index. The Stock Price Dynamics Summary is mathematically represented by Equation 5.4.

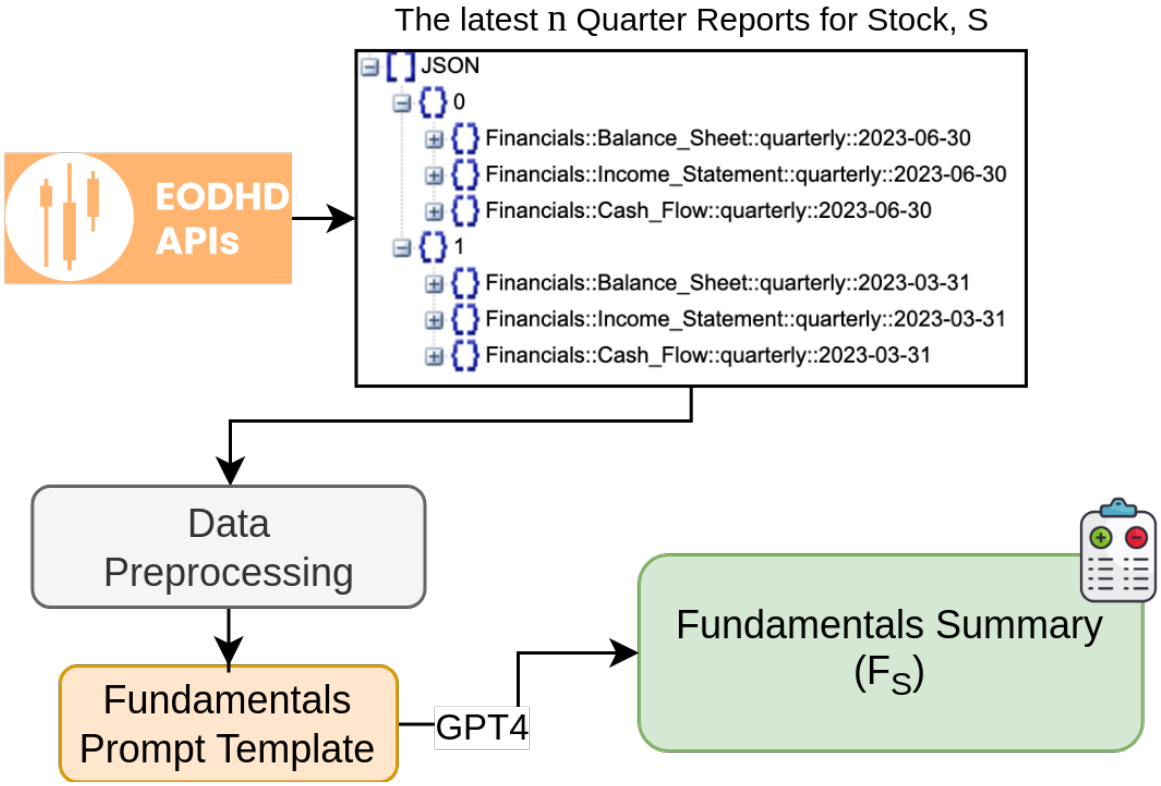


Figure 5.4: Stock's Fundamentals Summary

$$P_{S,t} = \text{Summarize}\left(\text{M}_{S,t}, \bigcup_{j=1}^{n} \text{M}_{P_j,t}\right) \tag{5.4}$$

In this equation, $P_{S,t}$ represents the Stock Price Dynamics Summary for stock S at time t, $\text{M}_{S,t}$ denotes the performance metrics of stock S at the same time point, and $\text{M}_{P_j,t}$ refers to the performance metrics for each peer stock P_j at time t. The structure of the prompt includes:

- **Performance Metrics**: This involves analyzing the stock's performance using metrics such as Cumulative Returns, Volatility, Sharpe Ratio, Maximum Drawdown, and a Correlation Matrix.
- **Comparative Analysis**: The stock's performance is compared with that of similar stocks and the S&P 500 index.

Table 5.3: Apple Inc. Fundamentals Summary (2023-Q3)

Category	Fundamentals Summary
Profitability	Apple's net income increased from $19.88 billion in Q2 2023 to $22.96 billion in Q3 2023, indicating strong profitability. The company's gross profit also rose from $36.41 billion to $40.43 billion over the same period.
Revenue Growth	Total revenue grew from $81.80 billion in Q2 2023 to $89.50 billion in Q3 2023, reflecting positive revenue growth.
Debt Levels	The company's total liabilities increased from $274.76 billion in Q2 2023 to $290.44 billion in Q3 2023. While long-term debt slightly decreased from $98.07 billion to $95.28 billion, short-term debt increased from $11.21 billion to $15.81 billion. The net debt also saw a slight increase from $80.87 billion to $81.12 billion, signaling a rising debt level that could pose concerns if not managed properly.
Cash Flow	Net cash from operating activities decreased from $26.38 billion in Q2 2023 to $21.60 billion in Q3 2023. However, the end period cash flow rose from $29.90 billion to $30.74 billion, indicating positive cash flow generation.
Assets and Equity	Total assets grew from $335.04 billion in Q2 2023 to $352.58 billion in Q3 2023. Similarly, total stockholder equity increased from $60.27 billion to $62.15 billion, reflecting growth in the company's assets and equity.
Conclusion	Apple Inc. demonstrates strong profitability and revenue growth. However, the increasing debt level requires careful monitoring. The company maintains positive cash flow generation, and its assets and equity are expanding.

- **Instructions**: Summarize the findings in a concise and factual report.

The methodology for identifying similar stocks is described in Algorithm 3, which employs the MPNet language model to generate embeddings and compute similarity scores [Song et al., 2020]. Specifically, stock descriptions are encoded into high-dimensional vectors by MPNet, capturing each company's unique characteristics and activities. This allows for the computation of pairwise similarity scores among companies listed in the S&P 500, ensuring that the selected stocks for comparison are genuinely similar to the target stock. This process is crucial for conducting a comprehensive comparative analysis that integrates individual stock performance with broader market trends.

The summarizer retrieves market data for the target stock, its similar stocks, and the S&P 500 index. It analyzes key financial indicators, including cumulative returns and Sharpe ratios over 3, 6, and 12 months, and also evaluates

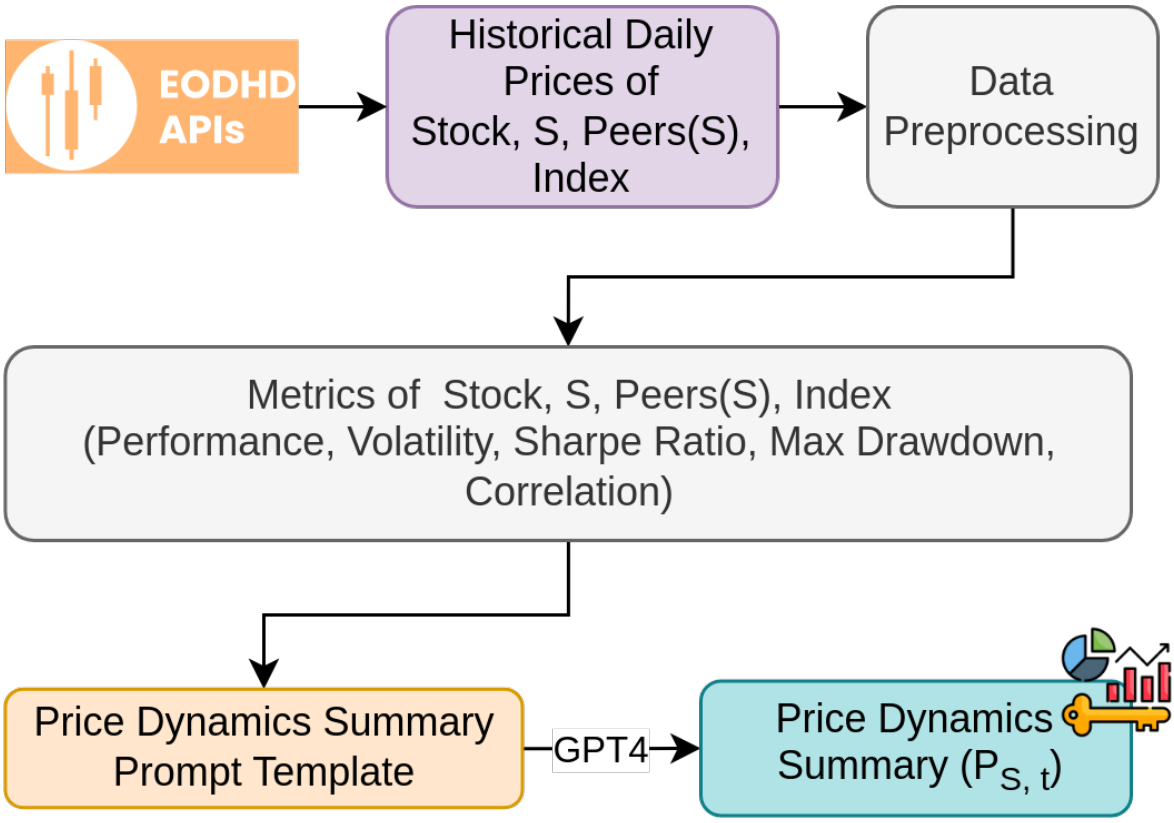


Figure 5.5: Stock Price Dynamics Summary

Algorithm 3 Stock Universe Identification

```
procedure StockUniverse(targetStock, descriptions, n)
    Generate embeddings for each stock description using a pre-trained language model
    Compute similarity scores between the target stock's embedding and others
    Rank the stocks based on their similarity scores to the target stock
    Select the top n stocks with the highest similarity scores
    return List of top n similar stocks based on embeddings
end procedure
```

volatility and maximum drawdown. These metrics are particularly important as they provide insights into the risk-adjusted returns and resilience of the stocks during market downturns [Korn et al., 2022]. This analysis offers a comprehensive understanding of the stock's performance relative to its peers and the broader market.

An example of the summarizer's output is shown in Table 5.4, highlighting its

ability to distill complex data into accessible insights. This approach provides a multi-dimensional view of the stock's market dynamics and momentum in relation to similar companies and overall market trends.

Table 5.4: Apple Inc. Stock Price Dynamics Summary (November 2023)

Metric	Price Dynamics Summary
Cumulative Return	Apple Inc demonstrated a 29.0% return, outperforming the S&P 500 index's 13.7% but underperforming tech peers like Adobe Systems and Amazon.com Inc.
Sharpe Ratio	Apple's Sharpe Ratio of 1.34 indicates a favorable risk-adjusted return compared to the market index Sharpe Ratio of 0.99, suggesting better compensation for the risk taken.
Volatility	Apple's volatility at 21.7% is lower than that of Alphabet Inc, Adobe, and Amazon, indicating less erratic stock price movements.
Maximum Drawdown	Apple experienced a maximum drawdown of -16.0%, which is less severe than the drawdowns of Adobe, Amazon, and Best Buy Co. Inc.
Correlation	Apple shows a high correlation with the S&P 500 (0.76) and moderate correlation with other tech stocks like Microsoft Corporation and Amazon.
Conclusion	Apple has shown resilience and strong risk-adjusted performance relative to the broader market and some tech peers, with lower volatility and a relatively modest maximum drawdown.

5.3.4 Macroeconomic Environment Summary

Conducting an in-depth macroeconomic analysis is crucial for making informed investment decisions and effective capital allocation. Such analysis provides vital insights into the overall economic health and performance, which significantly influence the profitability and value of individual companies as well as the broader stock market. By considering major forces that shape the investment landscape, such as global events like the Covid-19 pandemic or the war in Ukraine, investors can make more informed decisions.

To support this, MarketSenseAI includes a component called MarketDigest, illustrated in Figure 5.6 and presented in the second use case of Chapter 2. This component synthesizes investment reports and research articles on a biweekly

basis, providing concise summaries of complex economic data and trends. MarketDigest sources its information from a variety of publicly accessible reports from leading banks and investment institutions, including Goldman Sachs, Morgan Stanley, UBS, and BlackRock. The mathematical representation for MarketSenseAI's Macroeconomic Environment Summary component, MarketDigest, is formulated as follows:

$$M_t = \text{Summarize}\left(\bigcup_{j=1}^{N} \text{Summarize}\left(\text{Report}_{j,t}\right)\right) \tag{5.5}$$

Here, M_t represents the MarketDigest output at time t, and $\text{Report}_{j,t}$ denotes the investment report or article j at time t. The variable N quantifies the number of reports or articles analyzed at time t.

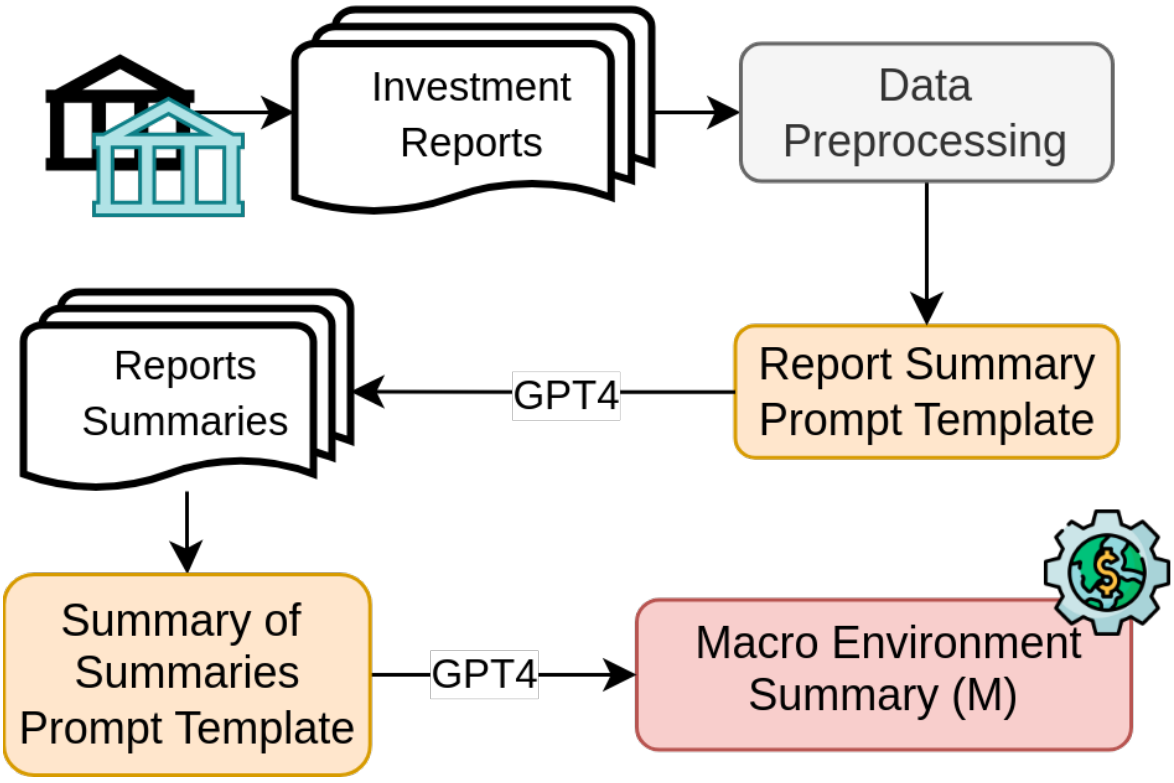


Figure 5.6: Macroeconomic Environment Summary (MarketDigest)

Initially, these reports and articles are transformed into text form. MarketDi-

gest then leverages GPT-4 to clean and filter the individual reports, extract their metadata, store them in a vector database and subsequently, using RAG transform the individual reports into a comprehensive overview. This method enables MarketDigest to integrate diverse perspectives and analyses into a cohesive narrative, offering a consensus view on the macroeconomic climate, central bank policies, preferred sectors or countries, and geopolitical trends. The output is concise yet thorough, accounting for potential contradictions or differing viewpoints among market analysts and experts. The prompt structure for MarketDigest includes:

- **Initial Summary Focus**: Summarization of individual reports, emphasizing critical macroeconomic elements, including central bank policies, geopolitical insights, and market outlooks.
- **Synthesis and Sentiment Analysis**: An in-depth analysis of all reports to extract consensus and divergent views, with a focus on sentiment analysis categorized by asset class or investment dimension.
- **Instructions**: Demand for a detailed and factual report, with emphasis on prevailing market sentiments and analytical categorization by asset class.

Table 5.5 provides an example of a MarketDigest summary. Notably, KM Cube Asset Management[3] initiated MarketDigest in March 2023, serving as a critical analytical tool, providing succinct market overviews to its clients. Distributed biweekly, it enhances the understanding of market dynamics for both private and institutional clients, thereby informing their investment decision-making processes [Metaxas, 2023]. Furthermore, MarketDigest plays a central role in the company's monthly investment committee meetings, substantially contributing to the deliberation and formulation of strategies for investment portfolios and the management of discretionary products.

[3] https://www.km3am.com/

Table 5.5: Macroeconomic Environment Summary (November 2023)

Category	Macro Summary
Inflation	US core PCE inflation eased to 3.5% in October, indicating a disinflation trend.
Interest Rates	Market-implied pricing suggests potential rate cuts in March 2024 for both the US and Europe.
Japan's Monetary Policy	Bank of Japan expected to weaken or abandon yield curve control due to domestic inflation.
ECB Policy	European Central Bank has begun balance sheet unwind.
Bond Market Outlook	Positive outlook on short- to medium-term developed market sovereign bonds.
Equity Market Stance	Neutral stance on developed market equities, with US stocks as the largest allocation.
Global Growth	Global economy expected to experience below-trend growth in 2024.
Investment Strategy	Portfolios should maintain neutral exposure to risk and equities, overweight allocation to quality fixed income.
US Dollar	US dollar's position as the leading global reserve currency shows signs of vulnerability.
Employment Risks	Risks to employment are on the downside, with leading indicators of employment deteriorating significantly.
Market Rally	Global financial markets experiencing significant rally, boosted by cooling inflation and falling Treasury yields.
Contradictions	Positive outlook on bonds but neutral on equities; US dollar vulnerability but remains a key currency.
Positive Sentiment	Short- to medium-term bonds, inflation-linked bonds, private market income, quality fixed income, US stocks.
Negative Sentiment	Credit, US Treasury, private markets, small-cap equities, Chinese equities.
Neutral Sentiment	Developed market equities, investment-grade credit, real estate, private equity funds, emerging markets outside China.

5.3.5 Signal Generation

The signal generation component, representing the final stage in the MarketSenseAI pipeline (Figure 5.1), synthesizes the textual outputs from the news, fundamentals, price dynamics, and macroeconomic analysis components to produce an investment recommendation for a specific stock, accompanied by a detailed rationale.

The underlying premise is that an investment decision for a stock should be

informed by key developments related to the company, reflected in the news, the company's financial health, the stock's performance relative to competitors and the market, and the broader macroeconomic environment. This decision-making model can be mathematically represented as:

$$I_s = f(N_s, F_s, P_s, M) \tag{5.6}$$

Here, N_s, F_s, P_s, and M denote stock-specific (s) textual representations of current news, fundamentals, price dynamics, and macroeconomic conditions, respectively, derived from the components discussed in Sections 5.3.1-5.3.4.

We argue that the state-of-the-art LLM, GPT-4, is well-equipped to evaluate and reason across these diverse data categories, as evidenced by its demonstrated proficiency in complex financial reasoning tasks [Callanan et al., 2023]. Within this framework, GPT-4 is prompted to assume the role of an expert financial analyst. This approach utilizes the Chain of Thought methodology [Wei et al., 2022], guiding the model through a logical, multi-step reasoning process that mirrors the analytical approach of a seasoned financial expert. By employing this technique, MarketSenseAI can effectively analyze and synthesize the news, company fundamentals, stock performance data, and macroeconomic factors that influence the stock, thereby providing structured and reasoned insights into stock selection. This method is particularly advantageous in complex domains like finance, where navigating multifaceted data and reasoning like an expert is crucial. Additionally, in-context learning is applied to dynamically adjust the analysis based on current financial conditions and evolving market data [Dong et al., 2022]. This dual strategy enables MarketSenseAI to deliver deep insights that adapt to changing market conditions and investor preferences, marking a significant advancement in AI-driven financial analysis. The prompt structure includes:

- **News Analysis**: The model first evaluates the latest news summaries related to the target company, assessing their potential impact on the stock's performance.
- **Price Dynamics Analysis**: Next, it compares the stock's price dynamics with those of related stocks and the broader market, offering a relative performance perspective.
- **Macroeconomic Environment Analysis**: The model then examines the broader macroeconomic environment and its implications for the target company, considering global economic trends and events.
- **Fundamentals Analysis**: The final step involves a review of the company's fundamental financial data, analyzing its current financial health and future prospects.

The output from the model is presented in a concise format, including a decision ("buy", "sell", or "hold") along with a clear, step-by-step explanation of the reasoning behind this choice. In this context, "buy" and "sell" refer to portfolio positioning (long and short positions, respectively), while "hold" indicates that the stock should not be included in the portfolio composition.

Table 5.6 illustrates MarketSenseAI's functionality by providing an explainable investment recommendation for Apple Inc. at the end of November 2023, suggesting a "hold" position in the rebalanced portfolio for the following month. The table reveals mixed signals, balancing positive developments such as strong financials and successful product launches against challenges like sales slowdowns, competitive pressures, and regulatory scrutiny. The overall decision to "hold" reflects a cautious approach, considering the various contrasting factors at play.

The analysis of text similarity within MarketSenseAI, as depicted in Figure 5.7 and detailed in Table 5.7, illustrates how the model's reasoning aligns with

Table 5.6: Apple Inc. Generated Signal and Explanation (November 2023)

Decision	Explanation
HOLD	The analysis [...] presents a mixed picture. Starting with the news analysis, while Apple remains a dominant player in the tech sector with solid financials and a successful product launch in South Korea, it faces challenges such as a sales slowdown, competitive pressures, and regulatory scrutiny. The Chinese government ban and sluggish sales in China and Japan are concerning, but analysts see the recent stock dip as a buying opportunity, suggesting underlying strength. The company's strategic moves, such as potentially ending its credit card partnership with Goldman Sachs, indicate a focus on its services business, which is a positive sign. However, the legal challenges and geopolitical risks cannot be ignored. In terms of price dynamics, Apple has outperformed the S&P 500 [...] strong risk-adjusted performance. However, the maximum drawdown and the high correlation with the market suggest that Apple is not immune to market downturns. The macro environment analysis reveals a disinflation trend in the US and potential rate cuts in 2024, which could be favorable for equities. However, the global economy is expected to grow below trend, and there is a neutral stance on developed market equities, suggesting caution. The positive outlook on quality fixed income and US stocks is encouraging, but the vulnerability of the US dollar and downside risks to employment add uncertainty. Fundamentals analysis shows strong profitability and revenue growth for Apple, with increased net income and gross profit. The company's debt levels have risen, which warrants monitoring, but the positive cash flow generation and growth in assets and equity are reassuring. Considering the mixed signals [...] a "hold" decision is prudent[...]

Green/Red text highlights positive/negative developments for the company or stock.

the dynamics of the input data, which is crucial for generating monthly investment signals. The high similarity scores in the "News" and "Price Dynamics" summaries (mean scores of 0.923 and 0.907, respectively) reflect the model's emphasis on these rapidly changing factors, recognizing their immediate impact on stock prices. This emphasis is particularly relevant for short-term, monthly predictions, where current developments and price trends can significantly influence market behavior.

Conversely, outputs from the "Fundamentals" and "Macro" components, with mean similarity scores of 0.849 and 0.803, respectively, exhibit a lesser direct influence on the model's monthly decisions. The fundamentals, updated quarterly, provide a stable but less frequently changing view of a company's financial health, while macroeconomic data, being broader and more generalized,

have a more moderate impact on short-term investment decisions.

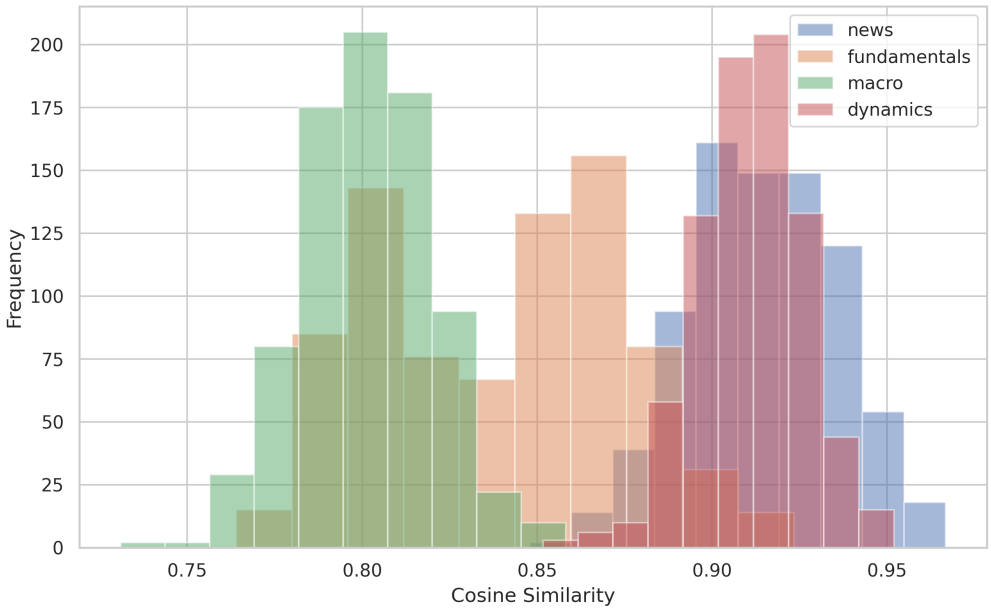


Figure 5.7: MarketSenseAI Components' Text Similarity with Signal

Table 5.7: Statistics of Text Similarity between Signals and Components

Component	mean	std	min	max
News	0.92	0.02	0.87	0.97
Price Dynamics	0.91	0.01	0.85	0.94
Fundamentals	0.85	0.03	0.77	0.93
Macro	0.80	0.02	0.75	0.87

This analysis highlights MarketSenseAI's capability to process and integrate various data types, tailoring its decision-making process to the nature of the input data. This approach is essential for delivering accurate and timely investment recommendations that align with the investor's time horizon.

5.3.6 Experimental Setup

This section outlines the data sources and methodologies employed in MarketSenseAI's analysis, along with the experimental setup used to evaluate and interpret the generated investment signals and explanations.

5.3.6.1 Data

The evaluation of MarketSenseAI's performance utilizes stocks from the S&P 100 index, which includes the 100 largest and most established companies in U.S. equity markets. These stocks, due to their prominence and the substantial amount of analysis they receive, create a challenging environment for achieving superior stock-selection performance, particularly when accounting for transaction costs [Fontinelle, 2022].

The assessment period extends from December 1, 2022, to March 31, 2024. During this period, MarketSenseAI drew on various datasets for its in-context learning processes:

1. **News**: A total of 163,483 articles published between December 1, 2022, and February 29, 2024, averaging 4.57 articles per day per stock, with a standard deviation of 5.49. The average number of tokens per article was 867, with a standard deviation of 1196. This dataset resulted in 35,229 daily, company-specific news summaries and 1,500 monthly progressive summaries.

2. **Fundamentals**: Financial data were sourced from 612 quarterly reports of S&P 100 stocks, starting from the second quarter of 2022. This data set produced 608 unique fundamentals summaries, averaging about 6 per stock.

3. **Descriptions**: Concise descriptions of each stock and its sector were used by Algorithm 3 to identify similar stocks.

4. **Prices**: Historical daily stock prices (adjusted close) from January 1, 2022, to February 29, 2024, were analyzed to compute stock price dynamics. This data was utilized by the Stock Price Dynamics component to produce one summary per month for each stock, totaling 1,500 summaries.

5. **Macro**: 187 investment reports (each 20-30 pages long) from major financial institutions, published between April 2023 and February 2023, were analyzed by MarketDigest. For predictions made between January and March 2023, macroeconomic summaries were unavailable for the signal generation component, resulting in 11 macroeconomic summaries being used for signal generation.

To evaluate the system, the "Signal Generation" component was provided with the latest available summaries (news, fundamentals, price dynamics, macro) at the end of each month. While macroeconomic summaries were identical across all stocks, fundamental summaries were updated only when a new quarterly report for a stock became available. News and price summaries, being more dynamic, provided updated stock-specific insights each month. The investment signals generated by MarketSenseAI were then assessed based on the actual stock performance in the month following the signal. In total, 1,500 signals were generated (15 months x 100 stocks), with a distribution of 338 “buy”, 1,150 “hold”, and 12 “sell” signals.

5.3.6.2 Data Sources, Preprocessing, and Parameter Selection

The empirical testing employed data from S&P 100 stocks, encompassing market trends, news articles, quarterly financial statements, and one year of historical daily stock prices sourced from EODHD APIs, alongside publicly available macroeconomic data from investment reports by institutions like Goldman Sachs, Morgan Stanley, UBS, and BlackRock, as well as statements from the Federal Reserve.

News data underwent filtering to exclude irrelevant articles and was summarized daily using GPT-4, followed by aggregation into monthly summaries. Financial data was standardized using numerical abbreviations and organized quarterly to facilitate comparative analysis. Stock price data was adjusted for splits and dividends, with key metrics such as cumulative returns, volatility, Sharpe ratio, and maximum drawdown calculated. Macroeconomic data was converted into text and summarized using GPT-4.

Feature engineering was incorporated into the prompts used for GPT-4, ensuring the extraction of significant events and analyst opinions from news data, profitability and debt metrics from financial data, performance metrics from stock price data, and consensus and market sentiment from macroeconomic data.

Key parameters were determined through empirical testing and domain expertise. The number of similar stocks in the Price Dynamics Component was set to five peers, balancing granularity and relevance. The number of quarters in the Fundamentals Summarizer was set to five to ensure a comprehensive financial health analysis. Signal update frequency was set to one month, balancing responsiveness and stability. These steps ensure robust data inputs for MarketSenseAI, enabling the generation of accurate and actionable investment signals.

5.3.6.3 Evaluation

The evaluation of MarketSenseAI's generated signals was designed to rigorously assess the model's performance through multiple, robust methodologies. We focused on comparing MarketSenseAI's signals against bootstrapped signals to evaluate their statistical significance and against actual stock price movements under various investment strategies to measure practical performance. Additionally, we utilized GPT-4 to rank "buy" signals based on their explanations,

providing an indirect assessment of the quality and context of the generated signals. This multi-faceted evaluation approach ensures a comprehensive understanding of MarketSenseAI's effectiveness and its potential advantages over traditional and naive investment strategies.

5.3.6.3.1 Bootstrapping: Bootstrapping is a statistical method that involves resampling data with replacement to estimate the variability of specific statistics, including standard errors, confidence intervals, and various accuracy metrics [Efron and Tibshirani, 1986]. This method is particularly useful when dealing with complex or unknown data distributions.

In this study, bootstrapping was employed to evaluate MarketSenseAI's performance and its statistical significance against randomized investment signals. For this purpose, a matrix of signals for MarketSenseAI ($[n_{months} \times n_{stocks}]$) was used, from which samples were drawn randomly, representing "sell" (-1), "hold" (0), and "buy" (1) positions. It is crucial to note that the distribution of these randomly generated signals might not mirror the distribution within the MarketSenseAI dataset.

This approach ensures that the findings are not due to random chance and provides a robust assessment of the model's performance. After an iterative examination, we settled on creating 10,000 random portfolios for bootstrapping, observing that additional samples did not significantly alter the evaluation outcomes.

In this context, multiple randomized signals were generated for the stocks under consideration over the designated time frame. MarketSenseAI's performance was then compared against these randomized signals using two primary metrics. Firstly, the portfolio's cumulative returns were calculated by adhering to "buy", "sell", or both signals, applying equal weight to each and implementing monthly rebalancing (as depicted in Figure 5.8 and defined in Equation

5.7). Secondly, the effectiveness of the signals was evaluated using a hit ratio, with the following month's actual returns serving as the reference benchmark (as detailed in Equation 5.8).

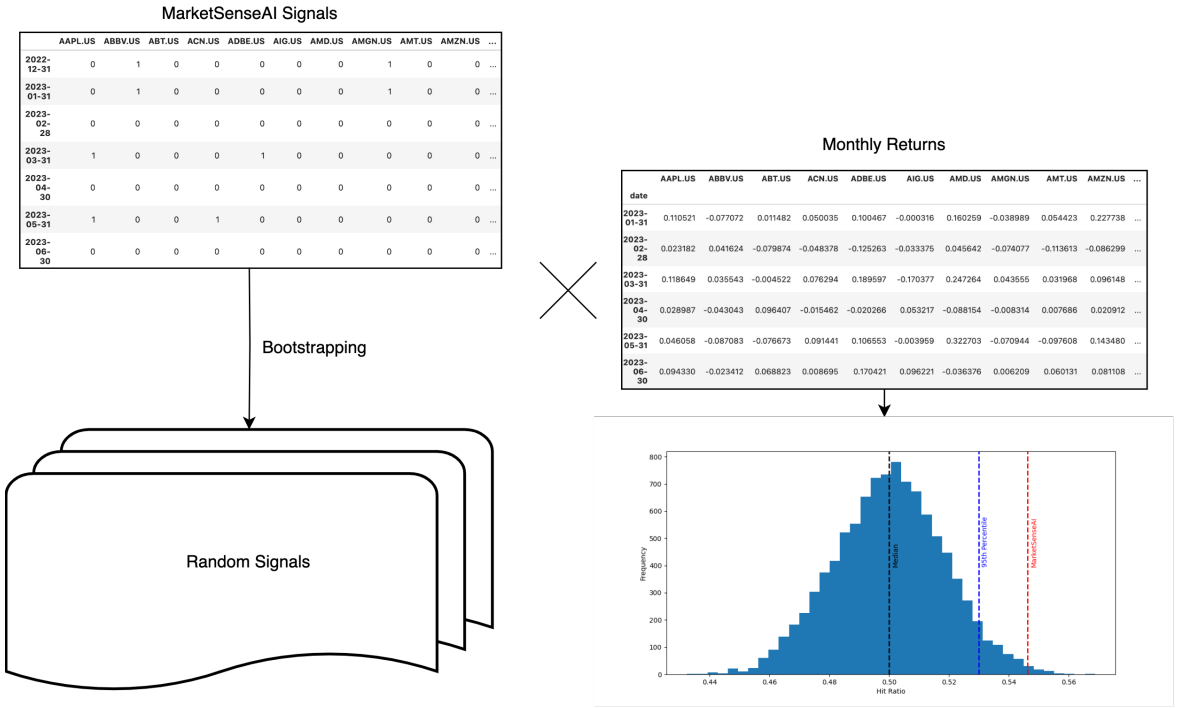


Figure 5.8: Bootstrapping-based Evaluation

The portfolio performance (cumulative return) is given by:

$$\text{Performance} = \prod_{i=1}^{n} \left(1 + \frac{\sum_{j=1}^{N} P_L(i,j)}{\text{signals per month at } i} \right) \tag{5.7}$$

The hit ratio is calculated as:

$$HR_L = \frac{\sum_{(i,j) \in V_L} \mathrm{I}(P_L(i,j) > 0)}{\text{length}(V_L)} \tag{5.8}$$

where,

$P_L(i,j)$: Performance of asset j at time i, defined as $P_L(i,j) = m(i,j) \times r(i,j)$.

L : Indicator representing the evaluated signals, L_{long} for long, L_{short} for short, and L_{both} for both signals.

$m(i,j)$: Model predictions (signals) for asset j at time i.

$r(i,j)$: Actual returns for asset j at time i.

V_L : Set of returns based on model predictions and L.

$\mathrm{I}(x)$: Indicator function, returning 1 if x is true and 0 otherwise.

This methodology evaluates the real impact of MarketSenseAI by determining whether its recommendations offer tangible benefits compared to random trading signals.

5.3.6.3.2 Market Performance: This part of the assessment focuses on providing a practical evaluation of MarketSenseAI by comparing the performance of portfolios constructed based on its signals with actual market prices. Various investment strategies were employed to demonstrate the robustness and versatility of MarketSenseAI's recommendations. The design of the MarketSenseAI-based portfolios, the baseline portfolios, and the evaluation metrics are outlined in Table 5.8 and Table 5.9, respectively. The MarketSenseAI portfolios were created by following the service's signals, which were generated on the last day of each month after market closure, and held for one month.

By evaluating MarketSenseAI's signals against actual market prices, we aim to showcase the real-world applicability and effectiveness of the generated recommendations. Utilizing different investment strategies highlights how MarketSenseAI can be adapted to various investment approaches and objectives, emphasizing its flexibility. The inclusion of traditional market indices and

naive strategies as benchmarks provides a clear context for evaluating the performance improvements offered by MarketSenseAI. The use of comprehensive evaluation metrics, such as Sharpe Ratio and Maximum Drawdown, ensures that the assessment considers both returns and risk, offering a balanced view of performance.

Table 5.8: Evaluated Investments Strategies on S&P 100 Stocks

Abbreviation	Description
MS	Equally weighted portfolio rebalanced monthly based on both "buy" and "sell" signals of MarketSenseAI.
MS-L	Equally weighted portfolio rebalanced monthly based on the "buy" signals of MarketSenseAI.
MS-L-Cap	Capitalization-weighted portfolio rebalanced monthly based on the "buy" signals of MarketSenseAI.
MS-Top10-SR	Equally weighted portfolio rebalanced monthly based on the 10 stocks with the best Sharpe Ratio among those with a "buy" signal.
S&P100-Eq	Equally weighted portfolio of all the stocks of the S&P 100 index.
S&P100	Capitalization-weighted S&P 100 index (OEF ETF).
Naive	Equally weighted portfolio rebalanced monthly for all S&P 100 stocks with prices above their corresponding 200-day moving average, fully allocated.
Naive-Top10	Equally weighted portfolio rebalanced monthly based on the 10 stocks with the best Sharpe Ratio and prices above their corresponding 200-day moving average, fully allocated.
MS-TopN-GPT	Equally weighted portfolio rebalanced monthly based on the N stocks with the best score produced by GPT-4 after processing all the stocks with a "buy" signal.
MS-High-GPT	Equally weighted portfolio rebalanced monthly based on stocks with a score greater than 7/10 produced by GPT-4 after processing all the stocks with a "buy" signal.
MS-Low-GPT	Equally weighted portfolio rebalanced monthly based on stocks with a score lower than or equal to 7/10 produced by GPT-4 after processing all the stocks with a "buy" signal.
MS-TopN-Cap-GPT	Capitalization-weighted portfolio rebalanced monthly based on the N stocks with the best score produced by GPT-4 after processing all the stocks with a "buy" signal.

5.3.6.3.3 Ex-post Evaluation with Ranking: To further assess the quality and relevance of the explanations provided for the "buy" signals, an additional layer of analysis was incorporated using a ranking mechanism, as illustrated in

Table 5.9: Portfolio Evaluation Metrics

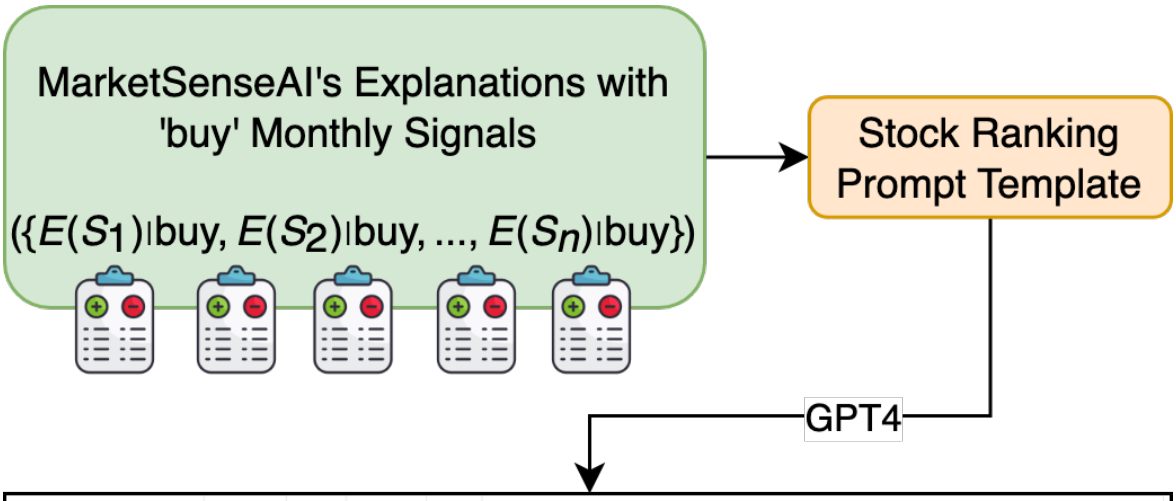
Metric	Description
Total Return	The portfolio's cumulative returns (%) over a specific period (Equation 5.7)
Sharpe Ratio	A measure of risk-adjusted return, calculated as the average return earned in excess of the risk-free rate per unit of volatility.
Sortino Ratio	Similar to the Sharpe Ratio, but focuses on downside risk by measuring returns relative to negative asset volatility.
Volatility	A statistical measure of the dispersion of returns for a given security or market index, typically measured using standard deviation.
Win Rate	The percentage of trades that are profitable out of the total number executed.
Maximum Drawdown (Ddn)	The maximum observed percentage loss from a peak to a trough of a portfolio, before a new peak is reached.

Figure 5.9. In this process, GPT-4 was presented with all the explanations that resulted in "buy" signals for each stock during a particular month. This set comprised all explanations $E(S_i)$ associated with a "buy" recommendation for each stock $S_{i|S_1=buy}$. The prompt instructed GPT-4 to rank these explanations on a scale from 0 to 10, where 10 represented a strong buy.

The rationale behind this approach is that if MarketSenseAI's outputs are truly insightful, detailed, and actionable, GPT-4 should be capable of ranking them effectively [Shu et al., 2023, Liu et al., 2023]. Stocks with higher rankings are expected to perform better in the portfolios. This method also introduces an alternative mechanism for ranking, filtering, and weighting in portfolio management. This evaluation approach was applied in the last four investment strategies listed in Table 5.8.

5.3.6.4 Setup

MarketSenseAI was implemented using Python 3.11, with the LangChain framework [Chase, 2022] employed for constructing prompts and utilizing OpenAI's API to access the GPT-4 model. Each component of MarketSenseAI, described in Section 3.3.2.3, operates independently as a standalone script. The out-



Company	Ticker	Signal	Notes	Score	Summary
NVIDIA Corporation	NVDA.US	Buy	New	9.0	Strong leadership in GPU and AI, robust financials, and strategic adaptability. High growth potential despite volatility.
Microsoft Corporation	MSFT.US	Buy	Reiterate	8.5	Strong financial performance, strategic AI and cloud investments, and diversified revenue streams. Bullish market confidence.
Mastercard Inc	MA.US	Buy	Reiterate	8.5	Expansion into new markets and innovative technologies. Solid financials and favorable risk-return profile.
Amazon.com Inc	AMZN.US	Buy	Reiterate	8.0	Strategic AI and healthcare expansion, robust financial performance, and strong market outperformance.
Visa Inc. Class A	V.US	Buy	Reiterate	8.0	Innovation in digital payments, strong earnings growth, and shareholder value actions. Resilient investment with growth potential.
Meta Platforms Inc.	META.US	Buy	Reiterate	7.5	Recovery with strategic cost-cutting and focus on AI/metaverse. Solid financial growth and market resilience.

Figure 5.9: Signal Ranking by GPT-4

puts generated by these components are systematically stored in a datastore, ensuring organized and efficient data management. By leveraging OpenAI's API for GPT-4, MarketSenseAI benefits from the consistent processing times provided by OpenAI's infrastructure, while OpenAI handles the underlying computational and hardware management.

For the backtesting of portfolios during the evaluation process, the VectorBT PRO library[4] was selected. This library is renowned for its versatility and efficiency in financial analysis and backtesting investment strategies, making it an ideal choice for rigorously assessing the performance of portfolios generated by MarketSenseAI.

[4] https://vectorbt.pro/

To determine the most similar stocks for the Stock Price Dynamics Summarizer, the `mpnet-base-v2` model from Hugging Face's Transformers framework was utilized, alongside the `cosine_similarity` function from the Scikit-Learn Python library. Data preprocessing was conducted using Pandas and NumPy libraries, ensuring that the price data was clean and properly prepared for analysis.

The experiments were conducted on a desktop computer equipped with an AMD Ryzen 5 5600x 6-Core CPU, 32GiB of RAM, and an NVIDIA GeForce RTX 3070 GPU. For data storage, a dedicated cloud-based S3 bucket hosted on Amazon Web Services (AWS) was employed.

5.4 Experimental Results

This section presents the empirical findings obtained from the multifaceted evaluations outlined in Section 3.3.2.4.

5.4.1 Bootstrapping Evaluation Results

Table 5.10 displays the outcomes of the bootstrapping evaluation, which is instrumental in contrasting the efficacy of MarketSenseAI with various bootstrapped portfolios. This evaluation also includes an assessment of MarketSenseAI's performance with detrended returns, providing a refined analysis of its signal generation capability. The detrending of returns is mathematically expressed as:

$$r'(i,j) = r(i,j) - \overline{r(i,\cdot)} \tag{5.9}$$

In this formula, $r'(i,j)$ represents the detrended return for asset j at time i, $r(i,j)$ is the actual return, and $\overline{r(i,\cdot)}$ is the average return at time i across all

assets. This detrending process is crucial as it helps to isolate the performance of individual stocks from the broader market trends, thereby offering a clearer perspective on MarketSenseAI's signal precision.

The table evaluates both cumulative returns (R) and hit ratios (HR), along with their respective quantiles (Q_R and Q_{HR}), delivering an extensive view of the system's effectiveness in comparison to randomized strategies.

Table 5.10: MarketSenseAI vs Bootstrapped Portfolios

Signals	R	Q_R	HR	Q_{HR}
Detrend-Buy	7.67	97.85	51.86	99.50
Buy	35.48	98.70	60.34	99.35
Detrend-Sell	22.62	100	72.73	100
Sell	0.10	100	63.64	100
Detrend-Both	8.41	99.55	52.61	94.20
Both	33.87	100	60.46	100

The results from the bootstrapping evaluation reveal that MarketSenseAI's signals significantly outperform random chance, as evidenced by the high quantiles achieved in both cumulative returns (R) and hit ratios (HR) across diverse signal categories. This superior performance holds true even when assessing detrended returns, indicating MarketSenseAI's proficiency in identifying profitable investment opportunities independent of broader market trends.

A particularly noteworthy observation is the high hit ratio quantile for the "Buy" signals following detrending. Given the upward market trend during the evaluation period, this suggests that MarketSenseAI's recommendations have a higher probability of success compared to randomly generated signals. This finding underscores the model's ability to effectively pinpoint potential market-outperforming opportunities.

In summary, the bootstrapping evaluation robustly demonstrates MarketSenseAI's capacity to produce trading signals that significantly exceed what would be expected by mere chance.

5.4.2 Market Performance Evaluation Results

It is crucial to highlight that the period during which the experiments were conducted was marked by varied performances among different stocks. While technology giants and companies centered around AI enjoyed a strong year, others delivered more modest returns [Thompson, 2023]. To address this disparity, our analysis places a special focus on equal-weighted indices, which helps to balance these differences and more accurately reflect MarketSenseAI's potential. Additionally, all results presented here incorporate transaction costs, ensuring a realistic assessment of the practical applicability and effectiveness of strategies derived from MarketSenseAI.

5.4.2.1 Vanilla Strategies

The evaluation of MarketSenseAI's vanilla strategies, as detailed in Table 5.11 and illustrated in Figure 5.10, demonstrates the effectiveness of LLM-driven investment strategies. The strategy that follows the full set of MarketSenseAI's signals (MS), equally weighted, achieves a total return of 35.48% (32.94% after accounting for transaction costs), with a Sharpe ratio of 2.49 and a Sortino ratio of 3.87. The long-only strategy (MS-L), which considers only the "buy" signals generated by MarketSenseAI, produces similar results, reflecting the relatively low number of "sell" signals issued.

Table 5.11: MarketSenseAI Performance of Vanilla Strategies

Strategy	Total Return[1]	Sharpe	Sortino	Vol	Win Rate	Max Ddn
MS	35.48 (32.94)	2.49	3.87	15.76	65.68	8.47
MS-L	35.79 (34.82)	2.41	3.75	16.44	65.02	9.00
S&P100-Eq	25.22 (25.12)	1.98	3.02	14.72	76.36	10.66
Naive	17.89 (17.45)	1.47	2.14	14.71	65.09	11.00
MS-L-Cap	**66.22 (65.25)**	**2.90**	**4.95**	22.81	65.88	9.64
S&P100	43.27 (43.20)	2.86	4.61	**16.17**	N/A	**9.24**

These results significantly outperform the equally-weighted S&P 100 (S&P100-

Eq) both in total and risk-adjusted returns, with an "alpha" of approximately 10% and 28% higher Sortino ratio. The naive trend-following strategy (Naive), often used by market participants, yielded significantly lower results.

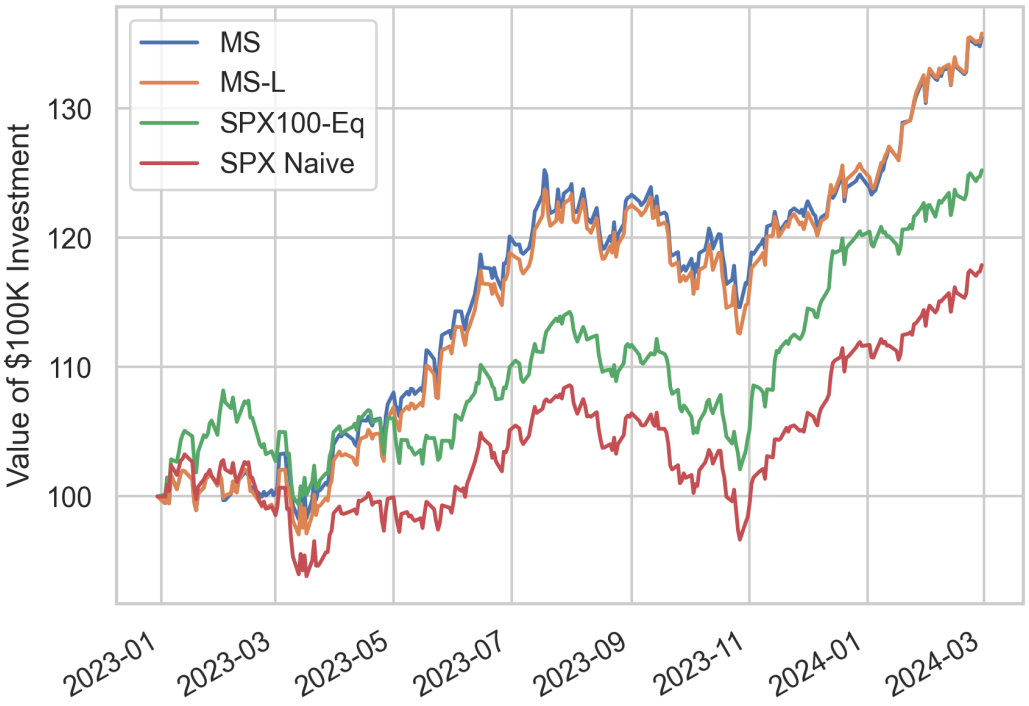


Figure 5.10: Performance of Equally-Weighted Portfolios

In terms of capitalization-weighted performance, the MS-L-Cap emerges as a top performer in terms of Sharpe and Sortino ratios, as well as total return, reaching an 60% total return. This is a significant outperformance of 43% to the S&P100 ETF as depicted in Figure 5.11.

5.4.2.2 Rank-based Strategies

Continuing the analysis of MarketSenseAI's effectiveness, Table 5.12 and Figure 5.12 present an in-depth examination of rank-based strategies derived

Figure 5.11: Performance of Capitalization-Weighted Portfolios

from MarketSenseAI's signals. This section explores the practical applications of these signals, particularly focusing on portfolios with a more manageable number of assets (approximately 10), and emphasizes how different strategic implementations can influence investment outcomes.

Table 5.12: MarketSenseAI Performance of Rank-Based Strategies

Strategy	Total Return	Sharpe	Sortino	Vol	Win Rate	Max Ddn
MS-Top10-SR	23.13 (22.12)	1.45	2.11	19.27	67.8	12.66
MS-Top5-GPT	50.96 (49.67)	2.26	3.69	24.01	68.42	11.39
MS-Top10-GPT	49.09 (48.07)	2.68	4.29	19.37	**74.1**	**7.66**
MS-High-GPT	39.47 (38.35)	2.28	3.44	19.08	71.9	9.73
MS-Low-GPT	25.66 (24.27)	1.76	2.64	**17.04**	55.1	12.22
Naive-Top10	29.01 (28.18)	1.67	2.48	20.29	69.3	9.79
MS-Top10-Cap-GPT	**72.87 (71.64)**	**2.80**	**4.89**	25.61	71.2	10.77

The MS-Top10-SR strategy, which selects "buy" stocks with the highest Sharpe ratios, achieved a total return of 23.13% (22.12% after accounting for transaction costs). However, strategies that employ GPT-4 to rank stocks demonstrate significantly better performance, both in terms of total returns and risk-adjusted metrics. This underscores the advanced analytical abilities of the LLM in identifying high-potential investments, as well as the value of the insights generated by MarketSenseAI.

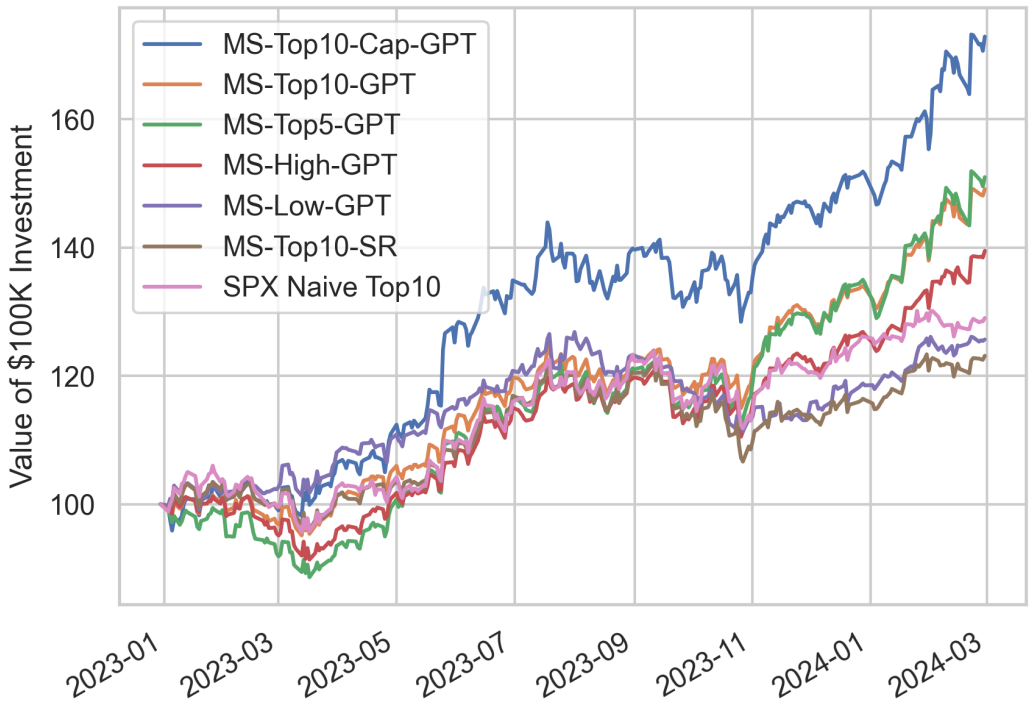


Figure 5.12: Performance of Ranked Portfolios

The MS-Top5-GPT and MS-Top10-GPT strategies, which focus on the top 5 and top 10 stocks as ranked by GPT-4, delivered exceptional total returns of 50.96% (49.67% after costs) and 49.09% (48.07% after costs), respectively. These strategies not only outperformed the MS-Top10-SR and the Naive-Top10 benchmarks but also exhibited impressive Sharpe and Sortino ratios, highlight-

ing their strong performance relative to the risk involved.

Notably, the MS-Top10-GPT strategy achieved the highest win rate at 74.1% and the lowest maximum drawdown at 7.66%, further demonstrating the robustness of GPT-4's ranking mechanism in managing market volatility and capturing growth opportunities.

The comparison between the MS-High-GPT and MS-Low-GPT strategies, which yielded returns of 39.47% (38.35% after costs) and 25.66% (24.27% after costs) respectively, reveals the significant advantage of selecting stocks based on the detailed explanations provided by MarketSenseAI. This differentiation highlights the value of the insights that support each stock's recommendation, introducing a novel GPT-based ranking method for stocks that are classified with the same signal.

Moreover, the MS-Top10-Cap-GPT strategy, which applies a capitalization-weighted approach to the top GPT-4 ranked stocks, stands out with a total return of 72.87% (71.64% after costs), marking the highest return among all the strategies evaluated.

5.4.2.3 GPT Ranking

Expanding on the insights derived from GPT-4's rankings of MarketSenseAI signals, Figure 5.13 visually represents the quality of explanations associated with MarketSenseAI's "buy" signals over the evaluation period. The figure combines a bar plot that shows the frequency of "buy" signals for stocks with at least five such signals, with a scatter plot that displays the evaluative scores assigned by GPT-4. These scores reflect the strength, depth, and relevance of the explanations accompanying each "buy" recommendation. In this scatter plot, points are positioned according to the average score assigned to each stock, offering an intuitive view of how GPT-4 assessed the quality and persuasiveness of the explanations provided for each stock.

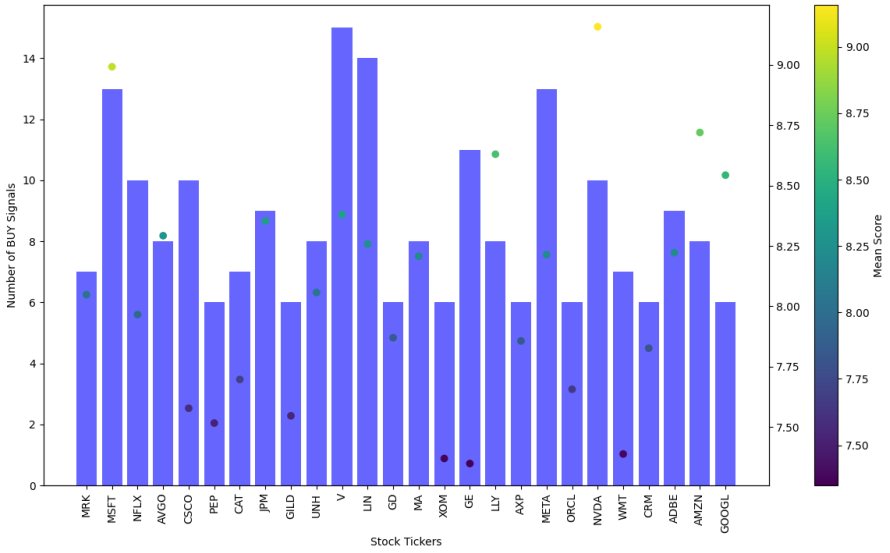


Figure 5.13: Signals Ranking by GPT-4

A notable observation from Figure 5.13 is the significantly higher scores given to technology and AI-related stocks, such as Nvidia, Microsoft, and Amazon. This trend reflects the heightened market enthusiasm for these sectors during the evaluation period. The tendency for these stocks to receive higher scores not only validates the relevance of MarketSenseAI's "buy" signals but also highlights the model's ability to capture and convey prevailing market sentiment and potential in its explanations.

5.5 Discussion

The notable performance of MarketSenseAI can largely be attributed to the advanced capabilities inherent in LLMs like GPT-4. These models are trained on an extensive and diverse corpus that includes research articles, financial documents, and historical records of significant financial events. This comprehensive training allows GPT-4 to recognize and differentiate between effective and ineffective investment strategies that have been employed on Wall Street.

For instance, the model can understand various investment methodologies such as factor investing, momentum investing, and growth investing, as well as how macroeconomic factors, like interest rate policies, influence asset prices over time.

MarketSenseAI leverages this deep, contextual understanding by providing GPT-4 with inputs similar to those used by professional investment teams, including news articles, financial statements, stock price data, and macroeconomic reports. This approach enables the AI to perform a detailed analysis of each stock, generating investment decisions that are consistently updated with the latest information. The system's ability to analyze each stock individually and adjust based on real-time updates ensures that its decisions are both rational and well-informed.

One of the key strengths of MarketSenseAI is its elimination of human biases and emotions from the investment process. Human decision-making is often influenced by psychological factors, such as overconfidence or fear, which can lead to suboptimal investment choices. By contrast, MarketSenseAI relies purely on data-driven insights, resulting in objective and potentially more reliable stock selection.

The empirical results presented in this chapter demonstrate that MarketSenseAI can generate accurate and actionable investment signals. These results underline the potential of GPT-4 not only to replicate but also to enhance the decision-making processes traditionally carried out by professional investment teams. This capability suggests a significant leap forward in the application of AI to financial markets, offering both retail and institutional investors a powerful tool to navigate the complexities of stock selection and portfolio management.

However, it is important to recognize the limitations of this work. The assumptions made regarding the factors influencing investment decisions and

the effectiveness of GPT-4 across different market conditions may not hold universally. The relatively short duration of the evaluation period also limits the generalizability of the findings, pointing to the need for further research over longer periods and in diverse market environments. Additionally, while GPT-4 was chosen for its advanced capabilities, future research should explore the potential benefits and limitations of other LLMs, as well as the effects of fine-tuning these models on the performance of MarketSenseAI.

Overall, the presented findings highlight the considerable promise of LLMs in financial analysis. MarketSenseAI represents a significant advancement in AI-driven financial decision-making, contributing to the ongoing development of tools that could enhance the stability and efficiency of the financial system.

Chapter 6

Conclusions and Future Work

Chapter Structure

This Chapter is constructed as follows:

- **Section 6.1 - Conclusions**, presents the concluding remarks of this thesis. This Section summarizes the key findings of the research and highlights its significance.
- **Section 6.2 - Future Work**, outlines potential directions for future research that could build upon the findings of the current thesis.

6.1 Conclusions

This doctoral dissertation has explored the design, optimization, and development of AI systems, with a particular focus on the challenges posed by multidimensional, high-dimensional, and complex datasets. The research conducted addresses significant gaps in the fields of cloud computing and finance, providing novel methodologies and frameworks that advance the state of the art in these domains.

One of the primary contributions of this research is the development of a

data-centric Knowledge-Based Reasoning Framework, which effectively handles and extracts insights from heterogeneous, complex, and multi-frequency datasets. This framework, evaluated in both cloud computing and financial contexts, demonstrated its ability to enhance resource allocation and optimize decision-making processes. This directly addresses RQ1, which sought to develop methods for managing diverse data sources, particularly in the complex domains of finance and cloud computing.

The research also introduced advanced techniques for real-time adaptability and performance in AI systems focusing on time series data. By integrating online learning and adaptive algorithms, the proposed methods allow AI applications to maintain high performance and swiftly adapt to new information, which is critical in dynamic environments. This work answers RQ2, focusing on enhancing the real-time adaptability and robustness of AI systems in rapidly changing conditions.

Another significant contribution lies in the optimization and comparative evaluation of LLMs against traditional transfer learning methods for text classification tasks, such as sentiment analysis. The findings indicate that LLMs, with appropriate prompt engineering, can outperform traditional methods, especially in handling complex and context-rich text data. This research addresses RQ3, investigating the conditions under which LLMs offer superior performance compared to conventional approaches.

Finally, the dissertation explored methods for enhancing the utility of LLMs by incorporating domain-specific knowledge to improve their relevance for specialized applications. This involved integrating proprietary data and developing strategies to mitigate the inherent limitations of LLMs, such as hallucinations and challenges in processing structured data. This contribution addresses RQ4, focusing on methods to enhance the applicability and reliability of LLMs in specialized domains, particularly in finance.

Overall, this research makes substantial contributions to the advancement of data-centric AI methodologies, providing innovative solutions to key challenges in handling complex datasets, enhancing real-time adaptability, optimizing language models, and extending their applicability to specific domains. These contributions collectively pave the way for more robust, scalable, and impactful AI applications across a variety of industries.

6.2 Future Work

As a direction for future work, there is considerable potential in further integrating ontologies and graph databases with LLMs and RAG approaches. In particular, LLMs could be utilized to automate complex data modeling tasks, streamlining the creation and maintenance of intricate ontologies. By leveraging LLMs to enhance retrieval capabilities using SPARQL or other relevant query languages, it would be possible to bridge the gap between structured knowledge systems and unstructured data processing. This integration could result in more dynamic and adaptable systems, capable of sophisticated reasoning and retrieval, thereby significantly improving the overall performance and applicability of the Reasoning Framework across various domains.

Moreover, while GPT-4 was selected for its advanced capabilities, future research should investigate the potential benefits and limitations of other LLMs. Exploring different models and the effects of fine-tuning these models could offer valuable insights into optimizing the performance of systems like MarketSenseAI. This could include experimenting with domain-specific fine-tuning to better align the LLMs with specialized tasks in finance, cloud computing, or other sectors.

Another promising direction lies in applying the knowledge gained from this research to build human-centered interfaces, such as chatbots, that utilize pro-

prietary, domain-specific, and real-time knowledge as well as LLM-powered agents. These interfaces could assist users in making informed decisions in a personalized manner. By integrating real-time and proprietary data, leveraging the advanced reasoning capabilities of LLMs for orchestrating multiple data pipelines and models, such interfaces could provide tailored recommendations and insights, enhancing user experience and decision-making processes in various industries.

Bibliography

[Abad et al., 2014] Abad, P., Benito, S., and López, C. (2014). A comprehensive review of value at risk methodologies. *The Spanish Review of Financial Economics*, 12(1):15–32.

[Abramski et al., 2023] Abramski, K., Citraro, S., Lombardi, L., Rossetti, G., and Stella, M. (2023). Cognitive network science reveals bias in gpt-3, gpt-3.5 turbo, and gpt-4 mirroring math anxiety in high-school students. *Big Data and Cognitive Computing*, 7(3):124.

[Achiam et al., 2023] Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. (2023). Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

[Adnan and Akbar, 2019] Adnan, K. and Akbar, R. (2019). Limitations of information extraction methods and techniques for heterogeneous unstructured big data. *International Journal of Engineering Business Management*, 11:1847979019890771.

[Agbaegbu et al., 2021] Agbaegbu, J., Arogundade, O. T., Misra, S., and Damaševičius, R. (2021). Ontologies in cloud computing—review and future directions. *Future Internet*, 13(12):302.

[AI, 2024] AI, P. (2024). About perplexity ai. https://www.perplexity.ai/hub/about. Accessed: 2024-08-10.

[Ai et al., 2023] Ai, Q., Bai, T., Cao, Z., Chang, Y., Chen, J., Chen, Z., Cheng, Z., Dong, S., Dou, Z., Feng, F., et al. (2023). Information retrieval meets large

language models: a strategic report from chinese ir community. *AI Open*, 4:80–90.

[Al-Antari, 2023] Al-Antari, M. A. (2023). Artificial intelligence for medical diagnostics—existing and future ai technology!

[Alqahtani et al., 2020] Alqahtani, A., Wither, M. J., Dong, Z., and Goodwin, K. R. (2020). Impact of news-based equity market volatility on international stock markets. *Journal of Applied Economics*, 23(1):224–234.

[Alshami et al., 2023] Alshami, A., Elsayed, M., Ali, E., Eltoukhy, A. E., and Zayed, T. (2023). Harnessing the power of chatgpt for automating systematic review process: Methodology, case study, limitations, and future directions. *Systems*, 11(7):351.

[Amazon Web Services, 2022] Amazon Web Services (2022). Amazon lambda pricing model. Accessed: 2024-08-21.

[Anand and Pathak, 2022] Anand, A. and Pathak, J. (2022). The role of reddit in the gamestop short squeeze. *Economics Letters*, 211:110249.

[Angelidis and Degiannakis, 2018] Angelidis, T. and Degiannakis, S. A. (2018). Backtesting var models: A two-stage procedure. *Available at SSRN 3259849*.

[Aoki, 2013] Aoki, M. (2013). *State space modeling of time series*. Springer Science & Business Media.

[Araci, 2019] Araci, D. (2019). Finbert: Financial sentiment analysis with pre-trained language models. Preprint at https://arxiv.org/abs/1908.10063.

[Arner et al., 2015] Arner, D. W., Barberis, J., and Buckley, R. P. (2015). The evolution of fintech: A new post-crisis paradigm. *Geo. J. Int'l L.*, 47:1271.

[Aryotejo et al., 2018] Aryotejo, G., Kristiyanto, D. Y., et al. (2018). Hybrid cloud: bridging of private and public cloud computing. In *Journal of Physics: Conference Series*, volume 1025, page 012091. IOP Publishing.

[Atreides and Kelley, 2023] Atreides, K. and Kelley, D. (2023). Cognitive biases in natural language: Automatically detecting, differentiating, and measuring bias in text.

[Bajraktari et al., 2018] Bajraktari, L., Ortiz, M., and Šimkus, M. (2018). Combining rules and ontologies into clopen knowledge bases. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

[Baker and Wurgler, 2007] Baker, M. and Wurgler, J. (2007). Investor sentiment in the stock market. *Journal of economic perspectives*, 21(2):129–151.

[Bechhofer, 2004] Bechhofer, S. (2004). Owl web ontology language reference, w3c recommendation. *http://www. w3. org/TR/owl-ref/*.

[Bekiros and Georgoutsos, 2005] Bekiros, S. D. and Georgoutsos, D. A. (2005). Estimation of value-at-risk by extreme value and conventional methods: a comparative evaluation of their predictive performance. *Journal of International Financial Markets, Institutions and Money*, 15(3):209–228.

[Bing, 2012] Bing, L. (2012). Sentiment analysis and opinion mining (synthesis lectures on human language technologies). *University of Illinois: Chicago, IL, USA*.

[BIS, 2022] BIS (2022). Market dysfunction and central bank tools. https://www.bis.org/publ/mc_insights.pdf. Accessed September 28, 2023.

[Blaskowitz and Herwartz, 2011] Blaskowitz, O. and Herwartz, H. (2011). On economic evaluation of directional forecasts. *International journal of forecasting*, 27(4):1058–1065.

[Bloomberg, 2019] Bloomberg (2019). What's an "algo wheel?" and why should you care? | bloomberg professional services. https://www.bloomberg.com/professional/blog/whats-algo-wheel-care/. Accessed September 24, 2023.

[Bloomberg, 2023] Bloomberg (2023). Bloomberg media distribution. Accessed:May 26, 2023.

[Bollen et al., 2011] Bollen, J., Mao, H., and Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of computational science*, 2(1):1–8.

[Bouchaud et al., 2003] Bouchaud, J.-P., Gefen, Y., Potters, M., and Wyart, M. (2003). Fluctuations and response in financial markets: the subtle nature ofrandom'price changes. *Quantitative finance*, 4(2):176.

[Brock et al., 2018] Brock, A., Donahue, J., and Simonyan, K. (2018). Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*.

[Brogaard et al., 2023] Brogaard, J., Han, J., and Won, P. Y. (2023). How does zero-day-to-expiry options trading affect the volatility of underlying assets? Available at SSRN: https://ssrn.com/abstract=4426358 or http://dx.doi.org/10.2139/ssrn.4426358.

[Brown et al., 2020] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

[Callanan et al., 2023] Callanan, E., Mbakwe, A., Papadimitriou, A., Pei, Y., Sibue, M., Zhu, X., Ma, Z., Liu, X., and Shah, S. (2023). Can gpt models be financial analysts? an evaluation of chatgpt and gpt-4 on mock cfa exams. Preprint at https://arxiv.org/abs/2310.08678.

[Chalupsky et al., 2002] Chalupsky, H., MacGregor, R., and INST, U. O. S. C. M. D. R. I. S. (2002). Ontologies, knowledge bases and knowledge management. Technical report, USC Information Sciences Insti-tute.

[Chang et al., 2003] Chang, Y.-P., Hung, M.-C., and Wu, Y.-F. (2003). Non-parametric estimation for risk in value-at-risk estimator. *Communications in Statistics-Simulation and Computation*, 32(4):1041–1064.

[Chase, 2022] Chase, H. (2022). Langchain. https://github.com/langchain-ai/langchain. Accessed December 29, 2023.

[Chen et al., 2018] Chen, C., Fengler, M. R., Härdle, W. K., and Liu, Y. (2018). Textual sentiment, option characteristics, and stock return predictability. Available at SSRN: https://ssrn.com/abstract=3210585.

[Chen, 2022a] Chen, D. (2022a). Constructing a data-driven model of english language teaching with a multidimensional corpus. *Mathematical Problems in Engineering*, 2022(1):2715408.

[Chen et al., 2014] Chen, H., De, P., Hu, Y. J., and Hwang, B.-H. (2014). Wisdom of crowds: The value of stock opinions transmitted through social media. *The Review of Financial Studies*, 27(5):1367–1403.

[Chen, 2022b] Chen, X. (2022b). Influence of structural break on the power of adf unit root test. *Journal of New Economics and Finance*.

[Chen et al., 2023] Chen, Z., Zheng, L. N., Lu, C., Yuan, J., and Zhu, D. (2023). Chatgpt informed graph neural network for stock movement prediction. Preprint at https://arxiv.org/abs/2306.03763.

[Cheng et al., 2015] Cheng, C., Sa-Ngasoongsong, A., Beyca, O., Le, T., Yang, H., Kong, Z., and Bukkapatnam, S. T. (2015). Time series forecasting for nonlinear and non-stationary processes: a review and comparative study. *Iie Transactions*, 47(10):1053–1071.

[Chesneau, 2017] Chesneau, B. (2017). Gunicorn documentation.

[Christoffersen et al., 2001] Christoffersen, P., Hahn, J., and Inoue, A. (2001). Testing and comparing value-at-risk measures. *Journal of empirical finance*,

8(3):325–342.

[Chui et al., 2023] Chui, M., Hazan, E., Roberts, R., Singla, A., Smaje, K., Sukharevsky, A., Yee, L., and Zemmel, R. (2023). The economic potential of generative ai: The next productivity frontier. Technical report, McKinsey & Company. https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/the-economic-potential-of-generative-ai-the-next-productivity-frontier. Accessed September 24, 2023.

[Church, 2017] Church, K. W. (2017). Word2vec. *Natural Language Engineering*, 23(1):155–162.

[Ciocîrlan et al., 2023] Ciocîrlan, C., Zwak-Cantoriu, M.-C., Stancea, A., and Plăcintă, D.-D. (2023). European macroeconomic dynamics on financial markets and economic policy: A cross country study for spillover effects. *Studia Universitatis Babes-Bolyai Oeconomica*, 68:40 – 63.

[CNBC, 2023] CNBC (2023). Jpmorgan ai investment advisor. https://www.cnbc.com/2023/05/25/jpmorgan-develops-ai-investment-advisor.html. Accessed September 24, 2023.

[Connor et al., 1994] Connor, J. T., Martin, R. D., and Atlas, L. E. (1994). Recurrent neural networks and robust time series prediction. *IEEE transactions on neural networks*, 5(2):240–254.

[Cordingly et al., 2020] Cordingly, R., Shu, W., and Lloyd, W. J. (2020). Predicting performance and cost of serverless computing functions with saaf. In *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, pages 640–649. IEEE.

[Dakhel et al., 2023] Dakhel, A. M., Majdinasab, V., Nikanjam, A., Khomh, F., Desmarais, M. C., and Jiang, Z. M. (2023). Github copilot ai pair programmer: Asset or liability? *Journal of Systems and Software*, page 111734.

[Dattels and Miyajima, 2009] Dattels, P. and Miyajima, K. (2009). Will emerging markets remain resilient to global stress? *Global Journal of Emerging Market Economies*, 1(1):5–24.

[De Palma et al., 2020] De Palma, G., Giallorenzo, S., Mauro, J., and Zavattaro, G. (2020). Allocation priority policies for serverless function-execution scheduling optimisation. In *Service-Oriented Computing: 18th International Conference, ICSOC 2020, Dubai, United Arab Emirates, December 14–17, 2020, Proceedings 18*, pages 416–430. Springer.

[De Waal et al., 2013] De Waal, B., Petersen, M. A., Hlatshwayo, L. N., and Mukuddem-Petersen, J. (2013). A note on basel iii and liquidity. *Applied Economics Letters*, 20(8):777–780.

[Devlin et al., 2018] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[Donahue et al., 2016] Donahue, J., Krähenbühl, P., and Darrell, T. (2016). Adversarial feature learning. *arXiv preprint arXiv:1605.09782*.

[Dong et al., 2022] Dong, Q., Li, L., Dai, D., Zheng, C., Wu, Z., Chang, B., Sun, X., Xu, J., and Sui, Z. (2022). A survey for in-context learning. Preprint at https://arxiv.org/abs/2301.00234.

[Dong et al., 2020] Dong, Y., Liu, Y., and Qin, S. J. (2020). Efficient dynamic latent variable analysis for high-dimensional time series data. *IEEE Transactions on Industrial Informatics*, 16:4068–4076.

[Douze et al., 2024] Douze, M., Guzhva, A., Deng, C., Johnson, J., Szilvasy, G., Mazaré, P.-E., Lomeli, M., Hosseini, L., and Jégou, H. (2024). The faiss library. *arXiv preprint arXiv:2401.08281*.

[Dubey et al., 2024] Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. (2024). The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

[Efron and Tibshirani, 1986] Efron, B. and Tibshirani, R. (1986). Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Statistical science*, 1(1):54–75.

[Einhorn and Brown, 2008] Einhorn, D. and Brown, A. (2008). Private profits and socialized risk. *Global Association of Risk Professionals*, 42:10–26.

[Elshawi et al., 2018] Elshawi, R., Sakr, S., Talia, D., and Trunfio, P. (2018). Big data systems meet machine learning challenges: towards big data science as a service. *Big data research*, 14:1–11.

[Elsinger et al., 2006] Elsinger, H., Lehar, A., and Summer, M. (2006). Risk assessment for banking systems. *Management science*, 52(9):1301–1314.

[Emmert-Streib, 2021] Emmert-Streib, F. (2021). From the digital data revolution toward a digital society: Pervasiveness of artificial intelligence. *Machine Learning and Knowledge Extraction*, 3(1):284–298.

[Engle and Manganelli, 2004] Engle, R. F. and Manganelli, S. (2004). Caviar: Conditional autoregressive value at risk by regression quantiles. *Journal of business & economic statistics*, 22(4):367–381.

[Erling, 2012] Erling, O. (2012). Virtuoso, a hybrid rdbms/graph column store. *IEEE Data Eng. Bull.*, 35(1):3–8.

[Evans and Lyons, 2005] Evans, M. D. and Lyons, R. K. (2005). Do currency markets absorb news quickly? *Journal of International Money and Finance*,

24(2):197–217.

[Farimani et al., 2022] Farimani, S. A., Jahan, M. V., Fard, A. M., and Tabbakh, S. R. K. (2022). Investigating the informativeness of technical indicators and news sentiment in financial market price prediction. *Knowledge-Based Systems*, 247:108742.

[Fatouros and Kouroumali, 2023] Fatouros, G. and Kouroumali, K. (2023). Forex news annotated dataset for sentiment analysis. [Data set].

[Fatouros et al., 2023a] Fatouros, G., Kousiouris, G., Lohier, T., Makridis, G., Polyviou, A., Soldatos, J., and Kyriazis, D. (2023a). Enhancing smart agriculture scenarios with low-code, pattern-oriented functionalities for cloud/edge collaboration. In *2023 19th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT)*, pages 285–292. IEEE.

[Fatouros et al., 2023b] Fatouros, G., Makridis, G., Kotios, D., Soldatos, J., Filippakis, M., and Kyriazis, D. (2023b). Deepvar: a framework for portfolio risk assessment leveraging probabilistic deep neural networks. *Digital finance*, 5(1):29–56.

[Fatouros et al., 2023c] Fatouros, G., Makridis, G., Mavrogiorgou, A., Soldatos, J., Filippakis, M., and Kyriazis, D. (2023c). Comprehensive architecture for data quality assessment in industrial iot. In *2023 19th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT)*, pages 512–517.

[Fatouros et al., 2024a] Fatouros, G., Metaxas, K., Soldatos, J., and Kyriazis, D. (2024a). Can large language models beat wall street? evaluating gpt-4's impact on financial decision-making with marketsenseai. *Neural Computing and Applications*.

[Fatouros et al., 2022] Fatouros, G., Poulakis, Y., Polyviou, A., Tsarsitalidis, S., Makridis, G., Soldatos, J., Kousiouris, G., Filippakis, M., and Kyriazis, D. (2022). Knowledge graphs and interoperability techniques for hybrid-cloud deployment of faas applications. In *2022 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 91–96. IEEE.

[Fatouros et al., 2023d] Fatouros, G., Soldatos, J., Kouroumali, K., Makridis, G., and Kyriazis, D. (2023d). Transforming sentiment analysis in the financial domain with chatgpt. *Machine Learning with Applications*, 14:100508.

[Fatouros et al., 2024b] Fatouros, G., Touloupos, M., and Soldatos, J. (2024b). Large language models and extended reality convergence for personalized training at the industrial metaverse. In *In Proceedings of the 2024 32nd European Conference on Information Systems (ECIS)*.

[Fernandes and Bernardino, 2018] Fernandes, D. and Bernardino, J. (2018). Graph databases comparison: Allegrograph, arangodb, infinitegraph, neo4j, and orientdb. In *Data*, pages 373–380.

[Fontinelle, 2022] Fontinelle, A. (2022). Can anybody beat the market? https://www.investopedia.com/ask/answers/12/beating-the-market.asp, Accessed January 01, 2024.

[Friedman and Broeck, 2020] Friedman, T. and Broeck, G. (2020). Symbolic querying of vector spaces: Probabilistic databases meets relational embeddings. In *Conference on Uncertainty in Artificial Intelligence*, pages 1268–1277. PMLR.

[Gallicchio et al., 2018] Gallicchio, C., Micheli, A., and Pedrelli, L. (2018). Comparison between deepesns and gated rnns on multivariate time-series prediction. *arXiv preprint arXiv:1812.11527*.

[Gao et al., 2023] Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., and Wang, H. (2023). Retrieval-augmented generation for large language

models: A survey. *arXiv preprint arXiv:2312.10997*.

[Gardner Jr, 1985] Gardner Jr, E. S. (1985). Exponential smoothing: The state of the art. *Journal of forecasting*, 4(1):1–28.

[Goodfellow et al., 2014] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.

[Goyal and He, 2015] Goyal, A. and He, Z. (2015). Passive investing and market liquidity. *The Review of Financial Studies*, 28(7):2167–2203.

[Greenwald et al., 2020] Greenwald, B. C., Kahn, J., Bellissimo, E., Cooper, M. A., and Santos, T. (2020). *Value investing: from Graham to Buffett and beyond*. John Wiley & Sons, NY, USA.

[Guan et al., 2023] Guan, X., Wang, J., Sun, Z., Zhang, Z., Duan, T., Deng, S., Liu, F., and Cui, H. (2023). New problems in active sampling for mobile robotic online learning. In *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 1155–1160. IEEE.

[Guo et al., 2023] Guo, B., Zhang, X., Wang, Z., Jiang, M., Nie, J., Ding, Y., Yue, J., and Wu, Y. (2023). How close is chatgpt to human experts? comparison corpus, evaluation, and detection. Preprint at https://arxiv.org/abs/2301.07597.

[Guo et al., 2022] Guo, R., Luan, X., Xiang, L., Yan, X., Yi, X., Luo, J., Cheng, Q., Xu, W., Luo, J., Liu, F., et al. (2022). Manu: a cloud native vector database management system. *arXiv preprint arXiv:2206.13843*.

[Hameed et al., 2016] Hameed, A., Khoshkbarforoushha, A., Ranjan, R., Jayaraman, P. P., Kolodziej, J., Balaji, P., Zeadally, S., Malluhi, Q. M., Tziritas, N., Vishnu, A., et al. (2016). A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems. *Computing*, 98:751–774.

[Han et al., 2024] Han, R., Zhang, Y., Qi, P., Xu, Y., Wang, J., Liu, L., Wang, W. Y., Min, B., and Castelli, V. (2024). Rag-qa arena: Evaluating domain robustness for long-form retrieval augmented question answering. *arXiv preprint arXiv:2407.13998*.

[Hellerstein et al., 2018] Hellerstein, J. M., Faleiro, J., Gonzalez, J. E., Schleier-Smith, J., Sreekanti, V., Tumanov, A., and Wu, C. (2018). Serverless computing: One step forward, two steps back. *arXiv preprint arXiv:1812.03651*.

[Hendricks, 1996] Hendricks, D. (1996). Evaluation of value-at-risk models using historical data. *Economic policy review*, 2(1).

[Hivarhizov, 2024] Hivarhizov, I. (2024). Development and implementation of systems based on llm in finance. *State and Regions. Series: Economics and Business*.

[Howard and Ruder, 2018] Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

[Hu and Lu, 2024] Hu, Y. and Lu, Y. (2024). Rag and rau: A survey on retrieval-augmented language model in natural language processing. *arXiv preprint arXiv:2404.19543*.

[Hyndman and Athanasopoulos, 2014] Hyndman, R. J. and Athanasopoulos, G. (2014). Forecasting: principles and practice, 2013. *URL: https://www. otexts. org/fpp [accessed 2018-02-15][WebCite Cache ID 6xFJlXCQI]*.

[JasperAI, 2023] JasperAI (2023). The ai in business trend report. Accessed:May 26, 2023.

[Jegannathan et al., 2022] Jegannathan, A. P., Saha, R., and Addya, S. K. (2022). A time series forecasting approach to minimize cold start time in

cloud-serverless platform. In *2022 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, pages 325–330. IEEE.

[Jung, 2018] Jung, Y. (2018). Nonlinear regression models for heterogeneous data with massive outliers. *Journal of Applied Statistics*, 46:1456 – 1477.

[Kalekar et al., 2004] Kalekar, P. S. et al. (2004). Time series forecasting using holt-winters exponential smoothing. *Kanwal Rekhi school of information Technology*, 4329008(13):1–13.

[Karaahmetoglu et al., 2020] Karaahmetoglu, O., Ilhan, F., Balaban, I., and Kozat, S. S. (2020). Unsupervised online anomaly detection on irregularly sampled or missing valued time-series data using lstm networks. *arXiv preprint arXiv:2005.12005*.

[Keynes, 1937] Keynes, J. M. (1937). The general theory of employment. *The quarterly journal of economics*, 51(2):209–223.

[Khoshraftar et al., 2021] Khoshraftar, S., Mahdavi, S., and An, A. (2021). Centrality-based interpretability measures for graph embeddings. In *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–10. IEEE.

[Kidwell et al., 2016] Kidwell, D. S., Blackwell, D. W., and Whidbee, D. A. (2016). *Financial institutions, markets, and money*. John Wiley & Sons, NY, USA.

[Kim et al., 2023] Kim, A. G., Muhn, M., and Nikolaev, V. V. (2023). Bloated disclosures: Can chatgpt help investors process information? Available at SSRN: https://ssrn.com/abstract=4425527 or http://dx.doi.org/10.2139/ssrn.4425527.

[Kingston, 2001] Kingston, J. (2001). High performance knowledge bases: four approaches to knowledge acquisition, representation and reasoning for

workaround planning. *Expert Systems with Applications*, 21(4):181–190.

[Kirtac and Germano, 2024] Kirtac, K. and Germano, G. (2024). Sentiment trading with large language models. Available at SSRN: https://ssrn.com/abstract=4706629.

[Korn et al., 2022] Korn, O., Möller, P. M., and Schwehm, C. (2022). Drawdown measures: Are they all the same? *The Journal of Portfolio Management*, 48(5):104–120.

[Kotios et al., 2022] Kotios, D., Makridis, G., Fatouros, G., and Kyriazis, D. (2022). Deep learning enhancing banking services: a hybrid transaction classification and cash flow prediction approach. *Journal of big Data*, 9(1):100.

[Kousiouris et al., 2023] Kousiouris, G., Ambroziak, S., Zarzycki, B., Costantino, D., Tsarsitalidis, S., Katevas, V., Mamelli, A., and Stamati, T. (2023). A pattern-based function and workflow visual environment for faas development across the continuum. In *Companion of the 2023 ACM/SPEC International Conference on Performance Engineering*, ICPE '23 Companion, page 165–172, New York, NY, USA. Association for Computing Machinery.

[Kousiouris and Kyriazis, 2021] Kousiouris, G. and Kyriazis, D. (2021). Functionalities, challenges and enablers for a generalized faas based architecture as the realizer of cloud/edge continuum interplay. In *CLOSER*, pages 199–206.

[Kousiouris and Pnevmatikakis, 2023] Kousiouris, G. and Pnevmatikakis, A. (2023). Performance experiences from running an e-health inference process as faas across diverse clusters. In *Companion of the 2023 ACM/SPEC International Conference on Performance Engineering*, pages 289–295.

[Krzywicki et al., 2016] Krzywicki, A., Wobcke, W., Bain, M., Martinez, J. C., and Compton, P. (2016). Data mining for building knowledge bases: tech-

niques, architectures and applications. *The Knowledge Engineering Review*, 31(2):97–123.

[Kuester et al., 2006] Kuester, K., Mittnik, S., and Paolella, M. S. (2006). Value-at-risk prediction: A comparison of alternative strategies. *Journal of Financial Econometrics*, 4(1):53–89.

[Kukreja et al., 2023] Kukreja, S., Kumar, T., Bharate, V., Purohit, A., Dasgupta, A., and Guha, D. (2023). Vector databases and vector embeddings-review. In *2023 International Workshop on Artificial Intelligence and Image Processing (IWAIIP)*, pages 231–236. IEEE.

[Langer and Mukherjee, 2023] Langer, A. and Mukherjee, A. (2023). Data strategy for exponential growth. In *Developing a Path to Data Dominance: Strategies for Digital Data-Centric Enterprises*, pages 65–112. Springer.

[Leippold, 2023] Leippold, M. (2023). Sentiment spin: Attacking financial sentiment with gpt-3. *Finance Research Letters*, page 103957.

[Lewellen, 2002] Lewellen, J. (2002). Momentum and autocorrelation in stock returns. *The Review of Financial Studies*, 15(2):533–564.

[Li et al., 2023] Li, X., Zhu, X., Ma, Z., Liu, X., and Shah, S. (2023). Are chatgpt and gpt-4 general-purpose solvers for financial text analytics? an examination on several typical tasks. Preprint at https://arxiv.org/abs/2305.05862.

[Lim and Zohren, 2020] Lim, B. and Zohren, S. (2020). Time series forecasting with deep learning: A survey. *arXiv preprint arXiv:2004.13408*.

[Lin and Khazaei, 2020] Lin, C. and Khazaei, H. (2020). Modeling and optimization of performance and cost of serverless applications. *IEEE Transactions on Parallel and Distributed Systems*, 32(3):615–632.

[Lin, 2004] Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

[Lipitakis et al., 2023] Lipitakis, A.-D., Kousiouris, G., Nikolaidou, M., Bardaki, C., and Anagnostopoulos, D. (2023). Empirical investigation of factors influencing function as a service performance in different cloud/edge system setups. *Simulation Modelling Practice and Theory*, 128:102808.

[Liu et al., 2023] Liu, Y., Iter, D., Xu, Y., Wang, S., Xu, R., and Zhu, C. (2023). Gpteval: Nlg evaluation using gpt-4 with better human alignment. Preprint at https://arxiv.org/abs/2303.16634.

[Liu et al., 2021] Liu, Z., Huang, D., Huang, K., Li, Z., and Zhao, J. (2021). Finbert: A pre-trained financial language representation model for financial text mining. In *Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence*, pages 4513–4519.

[LlamaIndex, 2024] LlamaIndex (2024). Sec filings reader. https://docs.llamaindex.ai/en/stable/api_reference/readers/sec_filings/. Accessed: 2024-08-10.

[Longerstaey and Spencer, 1996] Longerstaey, J. and Spencer, M. (1996). Riskmetricstm—technical document. *Morgan Guaranty Trust Company of New York: New York*, 51:54.

[Lopez-Lira and Tang, 2023] Lopez-Lira, A. and Tang, Y. (2023). Can chatgpt forecast stock price movements? return predictability and large language models. Preprint at https://arxiv.org/abs/2304.07619.

[Lopez-Veyna et al., 2022] Lopez-Veyna, J. I., Castillo-Zuñiga, I., and Ortiz-Garcia, M. (2022). A review of graph databases. In *International Conference on Software Process Improvement*, pages 180–195. Springer.

[Loughran and McDonald, 2011] Loughran, T. and McDonald, B. (2011). When is a liability not a liability? textual analysis, dictionaries, and 10-ks. *The Journal of finance*, 66(1):35–65.

[LTXtrading, 2023] LTXtrading (2023). Bondgpt: Introducing ltx's generative ai application for corporate bond trading. https://www.ltxtrading.com/bondgpt Accessed January 19, 2024.

[Mahmoudi and Khazaei, 2022] Mahmoudi, N. and Khazaei, H. (2022). Performance modeling of metric-based serverless computing platforms. *IEEE Transactions on Cloud Computing*.

[Maitra, 2023] Maitra, S. (2023). Impact of economic uncertainty, geopolitical risk, pandemic, financial & macroeconomic factors on crude oil returns–an empirical investigation. *arXiv preprint arXiv:2310.01123*.

[Makridis et al., 2020] Makridis, G., Kyriazis, D., and Plitsos, S. (2020). Predictive maintenance leveraging machine learning for time-series forecasting in the maritime industry. In *2020 IEEE 23rd international conference on intelligent transportation systems (ITSC)*, pages 1–8. IEEE.

[Malik, 2011] Malik, F. (2011). Estimating the impact of good news on stock market volatility. *Applied Financial Economics*, 21(8):545–554.

[Malkiel, 2003] Malkiel, B. G. (2003). The efficient market hypothesis and its critics. *Journal of economic perspectives*, 17(1):59–82.

[Malo et al., 2014] Malo, P., Sinha, A., Korhonen, P., Wallenius, J., and Takala, P. (2014). Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65(4):782–796.

[Mao et al., 2024] Mao, Y., Chen, B., Chen, W., Deng, Y., Zeng, J., and Du, M. (2024). A comprehensive review of vertical applications in the financial

sector based on large language models. In *Proceedings of the 3rd International Conference on Big Data Economy and Digital Management, BDEDM 2024, January 12–14, 2024, Ningbo, China*.

[Mcneil, 1998] Mcneil, A. J. (1998). Calculating quantile risk measures for financial time series using extreme value theory.

[Meduri et al., 2024] Meduri, K., Nadella, G. S., Gonaygunta, H., Maturi, M. H., and Fatima, F. (2024). Efficient rag framework for large-scale knowledge bases.

[Mehrabi et al., 2021] Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., and Galstyan, A. (2021). A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35.

[Metaxas, 2023] Metaxas, K. (2023). Marketdigest. https://www.km3am.com/2023/03/13/marketdigest-new-ai-powered-tool-for-wealth-management-insights/. Accessed September 24, 2023.

[Miller, 2013] Miller, J. J. (2013). Graph database applications and concepts with neo4j. In *Proceedings of the southern association for information systems conference, Atlanta, GA, USA*, volume 2324, pages 141–147.

[Mohebali et al., 2020] Mohebali, B., Tahmassebi, A., Meyer-Baese, A., and Gandomi, A. H. (2020). Probabilistic neural networks: a brief overview of theory, implementation, and application. *Handbook of Probabilistic Models*, pages 347–367.

[Munir and Anjum, 2018] Munir, K. and Anjum, M. S. (2018). The use of ontologies for effective knowledge modelling and information retrieval. *Applied Computing and Informatics*, 14(2):116–126.

[Najork, 2023] Najork, M. (2023). Generative information retrieval. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1–1.

[Naveen, 2023] Naveen, H. (2023). Log analysis with genai: A hands-on guide. https://medium.com/@hari10.nitw/log-analysis-with-genai-with-handson-0e05224aed96. Accessed: 2024-08-10.

[Neuneier, 1996] Neuneier, R. (1996). Optimal asset allocation using adaptive dynamic programming. *Advances in Neural Information Processing Systems*, 8:952–958.

[Nguyen et al., 2022] Nguyen, P., Kertkeidkachorn, N., Ichise, R., and Takeda, H. (2022). Mtab4d: Semantic annotation of tabular data with dbpedia. *Semantic Web*, (Preprint):1–25.

[Novak, 2011] Novak, S. Y. (2011). *Extreme value methods with applications to finance*. CRC Press.

[Noy and Zhang, 2023] Noy, S. and Zhang, W. (2023). Experimental evidence on the productivity effects of generative artificial intelligence. *Science*, 381(6654):187–192.

[OECD, 2021] OECD (2021). Artificial intelligence, machine learning and big data in finance: Opportunities, challenges, and implications for policy makers. https://www.oecd.org/finance/financial-markets/Artificial-intelligence-machine-learning-big-data-in-finance.pdf. Accessed September 24, 2023.

[OpenAI, 2023a] OpenAI (2023a). Gpt-4 technical report.

[OpenAI, 2023b] OpenAI (2023b). Morgan stanley wealth management deploys gpt-4 to organize its vast knowledge base. https://openai.com/

customer-stories/morgan-stanley. Accessed January 19, 2024.

[OpenAI, 2024] OpenAI (2024). Hello gpt-4o: Model evaluations and overview. https://openai.com/index/hello-gpt-4o/. Accessed: 2024-08-10.

[Ovadia et al., 2023] Ovadia, O., Brief, M., Mishaeli, M., and Elisha, O. (2023). Fine-tuning or retrieval? comparing knowledge injection in llms. *arXiv preprint arXiv:2312.05934*.

[Papacharalampous et al., 2023] Papacharalampous, G., Tyralis, H., Doulamis, A., and Doulamis, N. (2023). Comparison of tree-based ensemble algorithms for merging satellite and earth-observed precipitation data at the daily time scale. *Hydrology*, 10(2):50.

[Paraskevoulakou and Kyriazis, 2023] Paraskevoulakou, E. and Kyriazis, D. (2023). Ml-faas: Towards exploiting the serverless paradigm to facilitate machine learning functions as a service. *IEEE Transactions on Network and Service Management*.

[Pennington et al., 2014] Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

[Pérez et al., 2009] Pérez, J., Arenas, M., and Gutierrez, C. (2009). Semantics and complexity of sparql. *ACM Transactions on Database Systems (TODS)*, 34(3):1–45.

[Peters et al., 2018] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*,

Volume 1 (Long Papers), pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

[Petrova et al., 2017] Petrova, G., Tuzovsky, A., and Aksenova, N. V. (2017). Application of the financial industry business ontology (fibo) for development of a financial organization ontology. In *Journal of Physics: Conference Series*, volume 803, page 012116. IOP Publishing.

[Pfenninger et al., 2021] Pfenninger, M., Rikli, S., and Bigler, D. N. (2021). Wasserstein gan: Deep generation applied on financial time series. *Available at SSRN 3877960*.

[Poria et al., 2017] Poria, S., Cambria, E., Bajpai, R., and Hussain, A. (2017). A review of affective computing: From unimodal analysis to multimodal fusion. *Information fusion*, 37:98–125.

[Poria et al., 2016] Poria, S., Cambria, E., and Gelbukh, A. (2016). Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems*, 108:42–49.

[Poulakis et al., 2022] Poulakis, Y., Fatouros, G., Kousiouris, G., and Kyriazis, D. (2022). Hocc:an ontology for holistic description of cluster settings. In *19th International Conference, GECON 2022, September 13-15*.

[Qiu et al., 2021] Qiu, H., Jha, S., Banerjee, S. S., Patke, A., Wang, C., Hubertus, F., Kalbarczyk, Z. T., and Iyer, R. K. (2021). Is function-as-a-service a good fit for latency-critical services? In *Proceedings of the Seventh International Workshop on Serverless Computing (WoSC7) 2021*, pages 1–8.

[Radford et al., 2018] Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. (2018). Improving language understanding by generative pre-training.

[Radford et al., 2019] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask

learners. *OpenAI blog*, 1(8):9.

[Ramprasad and Sivakumar, 2024] Ramprasad, A. and Sivakumar, P. (2024). Context-aware summarization for pdf documents using large language models. In *2024 International Conference on Expert Clouds and Applications (ICOECA)*, pages 186–191. IEEE.

[Rausch et al., 2021] Rausch, T., Rashed, A., and Dustdar, S. (2021). Optimized container scheduling for data-intensive serverless edge computing. *Future Generation Computer Systems*, 114:259–271.

[Refaeli and Hajek, 2021] Refaeli, D. and Hajek, P. (2021). Detecting fake online reviews using fine-tuned bert. In *Proceedings of the 2021 5th International Conference on E-Business and Internet*, pages 76–80.

[Reid et al., 2024] Reid, M., Savinov, N., Teplyashin, D., Lepikhin, D., Lillicrap, T., Alayrac, J.-b., Soricut, R., Lazaridou, A., Firat, O., Schrittwieser, J., et al. (2024). Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

[Rezaei et al., 2022] Rezaei, M. R., Popovic, M. R., Lankarany, M., and Yousefi, A. (2022). Deep direct discriminative decoders for high-dimensional time-series data analysis. *arXiv preprint arXiv:2205.10947*.

[Romero et al., 2021] Romero, F., Chaudhry, G. I., Goiri, Í., Gopa, P., Batum, P., Yadwadkar, N. J., Fonseca, R., Kozyrakis, C., and Bianchini, R. (2021). Faa$t: A transparent auto-scaling cache for serverless applications. In *Proceedings of the ACM Symposium on Cloud Computing*, pages 122–137.

[Safaryan et al., 2022] Safaryan, G., Jindal, A., Chadha, M., and Gerndt, M. (2022). Slam: Slo-aware memory optimization for serverless applications. In *2022 IEEE 15th International Conference on Cloud Computing (CLOUD)*, pages 30–39. IEEE.

[Said and Dickey, 1985] Said, S. E. and Dickey, D. A. (1985). Hypothesis testing in arima (p, 1, q) models. *Journal of the American Statistical Association*, 80(390):369–374.

[Salinas et al., 2020] Salinas, D., Flunkert, V., Gasthaus, J., and Januschowski, T. (2020). Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191.

[Sallam, 2023] Sallam, M. (2023). Chatgpt utility in healthcare education, research, and practice: systematic review on the promising perspectives and valid concerns. In *Healthcare*, volume 11, page 887. MDPI.

[Scao et al., 2022] Scao, T. L., Fan, A., Akiki, C., Pavlick, E., Ilić, S., Hesslow, D., Castagné, R., Luccioni, A. S., Yvon, F., Gallé, M., et al. (2022). Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

[Schumaker and Chen, 2009] Schumaker, R. P. and Chen, H. (2009). Textual analysis of stock market prediction using breaking financial news: The azfin text system. *ACM Transactions on Information Systems (TOIS)*, 27(2):1–19.

[Sen et al., 2019] Sen, R., Yu, H.-F., and Dhillon, I. (2019). Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting. *arXiv preprint arXiv:1905.03806*.

[Shahrad et al., 2019] Shahrad, M., Balkind, J., and Wentzlaff, D. (2019). Architectural implications of function-as-a-service computing. In *Proceedings of the 52nd annual IEEE/ACM international symposium on microarchitecture*, pages 1063–1075.

[Shao et al., 2023] Shao, Z., Gong, Y., Shen, Y., Huang, M., Duan, N., and Chen, W. (2023). Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. *arXiv preprint arXiv:2305.15294*.

[Shi et al., 2023] Shi, J., Chaurasiya, V., Liu, Y., Vij, S., Wu, Y., Kanduri, S., Shah, N., Yu, P., Srivastava, N., Shi, L., et al. (2023). Embedding based retrieval in friend recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3330–3334.

[Shu et al., 2023] Shu, L., Wichers, N., Luo, L., Zhu, Y., Liu, Y., Chen, J., and Meng, L. (2023). Fusion-eval: Integrating evaluators with llms. Preprint at https://arxiv.org/abs/2311.09204.

[Siemer, 2019] Siemer, S. (2019). Exploring the apache jena framework. *George August University: Göttingen, Germany*.

[Siering et al., 2018] Siering, M., Muntermann, J., and Rajagopalan, B. (2018). Explaining and predicting online review helpfulness: The role of content and reviewer-related signals. *Decision Support Systems*, 108:1–12.

[Sijabat, 2022] Sijabat, R. (2022). Examining the impact of economic growth, poverty and unemployment on inflation in indonesia (2000-2019): Evidence from error correction model. *Jurnal Studi Pemerintahan*, pages 25–58.

[Singh et al., 2023] Singh, P. N., Talasila, S., and Banakar, S. V. (2023). Analyzing embedding models for embedding vectors in vector databases. In *2023 IEEE International Conference on ICT in Business Industry & Government (ICTBIG)*, pages 1–7. IEEE.

[Siripurapu et al., 2023] Siripurapu, S., Darimireddy, N. K., Chehri, A., Sridhar, B., and Paramkusam, A. (2023). Technological advancements and elucidation gadgets for healthcare applications: An exhaustive methodological review-part-i (ai, big data, block chain, open-source technologies, and cloud computing). *Electronics*, 12(3):750.

[So and Philip, 2006] So, M. K. and Philip, L. (2006). Empirical analysis of garch models in value at risk estimation. *Journal of International Financial*

Markets, Institutions and Money, 16(2):180–197.

[Soldatos and Kyriazis, 2022] Soldatos, J. and Kyriazis, D. (2022). *Big Data and artificial intelligence in digital finance: Increasing personalization and trust in digital finance using Big Data and AI*. Springer Nature.

[Song et al., 2020] Song, K., Tan, X., Qin, T., Lu, J., and Liu, T.-Y. (2020). Mpnet: Masked and permuted pre-training for language understanding. *Advances in Neural Information Processing Systems*, 33:16857–16867.

[Sun et al., 2024] Sun, W., Wang, J., Guo, Q., Li, Z., Wang, W., and Hai, R. (2024). Cebench: A benchmarking toolkit for the cost-effectiveness of llm pipelines. *arXiv preprint arXiv:2407.12797*.

[Sun and Bozdogan, 2020] Sun, Y. and Bozdogan, H. (2020). Segmentation of high dimensional time-series data using mixture of sparse principal component regression model with information complexity. *Entropy*, 22(10):1170.

[Tang and Guo, 2024] Tang, Y. and Guo, W. (2024). Automatic retrieval-augmented generation of 6g network specifications for use cases. *arXiv preprint arXiv:2405.03122*.

[Tetlock, 2007] Tetlock, P. C. (2007). Giving content to investor sentiment: The role of media in the stock market. *The Journal of finance*, 62(3):1139–1168.

[Tetlock et al., 2008] Tetlock, P. C., Saar-Tsechansky, M., and Macskassy, S. (2008). More than words: Quantifying language to measure firms' fundamentals. *The journal of finance*, 63(3):1437–1467.

[Thompson, 2023] Thompson, C. (2023). Magnificent 7 stocks: What you need to know. https://www.investopedia.com/magnificent-seven-stocks-8402262, Accessed January 01, 2024.

[Thoppilan et al., 2022] Thoppilan, R., De Freitas, D., Hall, J., Shazeer, N., Kulshreshtha, A., Cheng, H.-T., Jin, A., Bos, T., Baker, L., Du, Y., et al.

(2022). Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

[Tjuatja et al., 2023] Tjuatja, L., Chen, V., Wu, S. T., Talwalkar, A., and Neubig, G. (2023). Do llms exhibit human-like response biases? a case study in survey design. Preprint at https://arxiv.org/abs/2311.04076.

[Toorajipour et al., 2021] Toorajipour, R., Sohrabpour, V., Nazarpour, A., Oghazi, P., and Fischl, M. (2021). Artificial intelligence in supply chain management: A systematic literature review. *Journal of Business Research*, 122:502–517.

[Touya and Lokhat, 2020] Touya, G. and Lokhat, I. (2020). Deep learning for enrichment of vector spatial databases: Application to highway interchange. *ACM Transactions on Spatial Algorithms and Systems (TSAS)*, 6(3):1–21.

[Tran et al., 2022] Tran, K. V., Phan, H. P., Quach, K. N. D., Nguyen, N. L.-T., Jo, J., and Nguyen, T. T. (2022). A comparative study of question answering over knowledge bases. In *International Conference on Advanced Data Mining and Applications*, pages 259–274. Springer.

[Usha Ruby et al., 2024] Usha Ruby, A., George Chellin Chandran, J., Chaithanya, B., Swasthika Jain, T., and Patil, R. (2024). Wheat leaf disease classification using modified resnet50 convolutional neural network model. *Multimedia Tools and Applications*, pages 1–19.

[Van Eyk et al., 2018] Van Eyk, E., Iosup, A., Abad, C. L., Grohmann, J., and Eismann, S. (2018). A spec rg cloud group's vision on the performance challenges of faas cloud architectures. In *Companion of the 2018 acm/spec international conference on performance engineering*, pages 21–24.

[Van Eyk et al., 2017] Van Eyk, E., Iosup, A., Seif, S., and Thömmes, M. (2017). The spec cloud group's research vision on faas and serverless

architectures. In *Proceedings of the 2nd International Workshop on Serverless Computing*, pages 1–4.

[Wang et al., 2021] Wang, J., Yi, X., Guo, R., Jin, H., Xu, P., Li, S., Wang, X., Guo, X., Li, C., Xu, X., et al. (2021). Milvus: A purpose-built vector data management system. In *Proceedings of the 2021 International Conference on Management of Data*, pages 2614–2627.

[Wang, 2022] Wang, Y. (2022). The analysis on the relation between cusip data and macroeconomic financial factors and its correlation to the capital markets. In *Proceedings of the 2022 13th International Conference on E-business, Management and Economics*, pages 361–366.

[Wang et al., 2024] Wang, Z., Gao, Z., Yang, Y., Wang, G., Jiao, C., and Shen, H. T. (2024). Geometric matching for cross-modal retrieval. *IEEE Transactions on Neural Networks and Learning Systems*.

[Wei et al., 2022] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

[Weiss-Cohen et al., 2019] Weiss-Cohen, L., Ayton, P., Clacher, I., and Thoma, V. (2019). Behavioral biases in pension fund trustees' decision making. *Review of Behavioral Finance*, 11(2):128–143.

[Welch et al., 1995] Welch, G., Bishop, G., et al. (1995). An introduction to the kalman filter.

[Weng et al., 2017] Weng, B., Ahmed, M. A., and Megahed, F. M. (2017). Stock market one-day ahead movement prediction using disparate data sources. *Expert Systems with Applications*, 79:153–163.

[Wolf et al., 2020] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

[Wu et al., 2023a] Wu, S., Irsoy, O., Lu, S., Dabravolski, V., Dredze, M., Gehrmann, S., Kambadur, P., Rosenberg, D., and Mann, G. (2023a). Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*.

[Wu et al., 2023b] Wu, S., Irsoy, O., Lu, S., Dabravolski, V., Dredze, M., Gehrmann, S., Kambadur, P., Rosenberg, D., and Mann, G. (2023b). Bloomberggpt: A large language model for finance. Preprint at https://arxiv.org/abs/2303.17564.

[Xiong et al., 2018] Xiong, Z., Liu, X.-Y., Zhong, S., Yang, H., and Walid, A. (2018). Practical deep reinforcement learning approach for stock trading. *arXiv preprint arXiv:1811.07522*.

[Xiong et al., 2024] Xiong, Z., Papageorgiou, V., Lee, K., and Papailiopoulos, D. (2024). From artificial needles to real haystacks: Improving retrieval capabilities in llms by finetuning on synthetic data. *arXiv preprint arXiv:2406.19292*.

[Xu et al., 2024] Xu, Z., Cruz, M. J., Guevara, M., Wang, T., Deshpande, M., Wang, X., and Li, Z. (2024). Retrieval-augmented generation with knowledge graphs for customer service question answering. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2905–2909.

[Yadav et al., 2023] Yadav, A., Ahmad, N., Khan, I. R., Agarwal, P., and Kaur, H. (2023). Role of ai, big data in smart healthcare system. In *2023 6th*

International Conference on Information Systems and Computer Networks (ISCON), pages 1–8. IEEE.

[Yamai et al., 2002] Yamai, Y., Yoshiba, T., et al. (2002). Comparative analyses of expected shortfall and value-at-risk: their estimation error, decomposition, and optimization. *Monetary and economic studies*, 20(1):87–121.

[Yan and Ouyang, 2018] Yan, H. and Ouyang, H. (2018). Financial time series prediction based on deep learning. *Wireless Personal Communications*, 102(2):683–700.

[Ye et al., 2023] Ye, J., Chen, X., Xu, N., Zu, C., Shao, Z., Liu, S., Cui, Y., Zhou, Z., Gong, C., Shen, Y., et al. (2023). A comprehensive capability analysis of gpt-3 and gpt-3.5 series models. *arXiv preprint arXiv:2303.10420*.

[You et al., 2021] You, X., Zhang, M., Ding, D., Feng, F., and Huang, Y. (2021). Learning to learn the future: Modeling concept drifts in time series prediction. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2434–2443.

[Yu et al., 2024] Yu, H., Gan, A., Zhang, K., Tong, S., Liu, Q., and Liu, Z. (2024). Evaluation of retrieval-augmented generation: A survey. *arXiv preprint arXiv:2405.07437*.

[Yu et al., 2023] Yu, X., Chen, Z., Ling, Y., Dong, S., Liu, Z., and Lu, Y. (2023). Temporal data meets llm–explainable financial time series forecasting. Preprint at https://arxiv.org/abs/2306.11025.

[Yue et al., 2023] Yue, T., Au, D., Au, C. C., and Iu, K. Y. (2023). Democratizing financial knowledge with chatgpt by openai: Unleashing the power of technology. *Available at SSRN 4346152*.

[Zafeiropoulos et al., 2022] Zafeiropoulos, A., Fotopoulou, E., Filinis, N., and Papavassiliou, S. (2022). Reinforcement learning-assisted autoscaling mech-

anisms for serverless computing platforms. *Simulation Modelling Practice and Theory*, 116:102461.

[Zaremba and Demir, 2023] Zaremba, A. and Demir, E. (2023). Chatgpt: Unlocking the future of nlp in finance. Available at SSRN: https://ssrn.com/abstract=4323643 or http://dx.doi.org/10.2139/ssrn.4323643.

[Zhang et al., 2021] Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2021). Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115.

[Zhang et al., 2019] Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2019). Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

[Zhao, 2020] Zhao, K. (2020). Analysis of investment risk assessment model of financial institutions under economic growth. In *International Conference on Urban Intelligence and Applications*, pages 233–240. Springer.

[Zheng et al., 2016] Zheng, W., Zou, L., Peng, W., Yan, X., Song, S., and Zhao, D. (2016). Semantic sparql similarity search over rdf knowledge graphs. *Proc. VLDB Endow.*, 9(11):840–851.