applied sciences

MDPI

Article

Large Language Models and the Elliott Wave Principle: A Multi-Agent Deep Learning Approach to Big Data Analysis in Financial Markets

Michał Wawer [1], Jarosław A. Chudziak [2] and Ewa Niewiadomska-Szynkiewicz [1,*]

[1] Institute of Control and Computation Engineering, Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warsaw, Poland; michal.wawer.stud@pw.edu.pl
[2] Institute of Computer Science, Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warsaw, Poland; jaroslaw.chudziak@pw.edu.pl
* Correspondence: ewa.szynkiewicz@pw.edu.pl; Tel.: +48-22-234-74-97

Abstract: Traditional technical analysis methods face limitations in accurately predicting trends in today's complex financial markets. Meanwhile, existing AI-driven approaches, while powerful in processing large datasets, often lack interpretability due to their black-box nature. This paper presents ElliottAgents, a multi-agent system that combines the Elliott wave principle with LLMs, showcasing the application of deep reinforcement learning (DRL) and natural language processing (NLP) in financial analysis. By integrating retrieval-augmented generation (RAG) and deep reinforcement learning (DRL), the system processes vast amounts of market data to identify Elliott wave patterns and generate actionable insights. The system employs a coordinated team of specialized agents, each responsible for specific aspects of analysis, from pattern recognition to investment strategy formulation. We tested ElliottAgents on both stock and cryptocurrency markets, evaluating its effectiveness in pattern identification and trend prediction across different time scales. Our experimental results demonstrate improvements in prediction accuracy when combining classical technical analysis with AI-driven approaches, particularly when enhanced by DRL-based backtesting process. This research contributes to the advancement of financial technology by introducing a scalable, interpretable framework that enhances market analysis capabilities, offering a promising new methodology for both practitioners and researchers.

Keywords: multi-agent systems; Elliott wave principle; large language models (LLMs); investment strategies; deep reinforcement learning (DRL); financial markets

check for updates

Citation: Wawer, M.; Chudziak, J.A.; Niewiadomska-Szynkiewicz, E. Large Language Models and the Elliott Wave Principle: A Multi-Agent Deep Learning Approach to Big Data Analysis in Financial Markets. *Appl. Sci.* **2024**, *14*, 11897. https://doi.org/10.3390/app142411897

Academic Editor: Donghai Guan

Received: 18 November 2024
Revised: 14 December 2024
Accepted: 16 December 2024
Published: 19 December 2024

1. Introduction

The increasing complexity and velocity of financial markets have created unprecedented challenges in market analysis and prediction. Traditional technical analysis methods [1], while foundational to market understanding, often struggle to process and interpret the vast amounts of data generated in modern trading environments [2]. This limitation has sparked significant interest in leveraging AI and machine learning techniques to enhance market analysis capabilities.

Evolution of financial market analysis has traditionally relied on two primary approaches: fundamental analysis, which examines intrinsic value through financial metrics, and technical analysis, which focuses on price patterns and market behavior. While these approaches remain relevant across both traditional and cryptocurrency markets, their application and effectiveness vary significantly between these domains. Traditional markets benefit from established valuation metrics and regulated information disclosure [3], whereas cryptocurrency markets often rely more heavily on technical analysis due to the relative absence of standardized fundamental metrics. Technical analysts primarily use candlestick charts, as shown in Figure 1, which provide detailed visualization of price movements and market trends. Among technical analysis methods, the Elliott wave principle (EWP) [4] has emerged as a particularly effective approach for identifying market patterns based on collective investor psychology. The EWP identifies recurring wave formations in price movements, illustrated in Figure 2, which reflect cycles of market sentiment and behavior. However, the manual application of such methods faces significant challenges in today's high-speed, data-intensive trading environment.

Figure 1. Example of a modern candlestick chart; screenshot from Ref. [5].

Market analysis through wave theory was introduced by Ralph Nelson Elliott in the late 1930s. His theory postulates that financial markets progress in predictable cycles, driven by alternating phases of optimism and pessimism among market participants. As demonstrated in Figure 2, these cycles manifest as distinct wave sequences in both upward and downward trends. The figure illustrates how a complete market movement consists of a five-wave pattern in the primary trend direction, followed by reversal of the trend with new wave sequences. This structured approach to market analysis provides analysts with a framework for anticipating potential market reversals and forecasting future price movements.

Recent advancements in AI, particularly in the introduction of LLMs into multi-agent systems [6–8], offer new possibilities for enhancing these traditional approaches. This research makes two important contributions to computer analysis of financial markets. First, we introduce a multi-agent framework that successfully integrates the Elliott wave principle with modern AI technologies, demonstrating how classical technical analysis methods can be enhanced through computational intelligence. Our second contribution is the development of an innovative pattern recognition system that leverages LLMs, retrieval-augmented generation (RAG), and deep reinforcement learning (DRL) to automatically identify and classify Elliott wave formations, significantly improving the accuracy and reliability of technical analysis in high-speed trading environments.

Despite the proliferation of AI-driven market analysis tools, several critical challenges persist in the field. Existing systems often struggle to effectively combine traditional technical analysis methods with modern AI capabilities [9], resulting in either overly rigid rule-based systems or black-box AI models that lack interpretability [10]. The identification and interpretation of complex market patterns, particularly those described by the EWP, require advanced analysis that exceeds the capabilities of conventional automated systems. Furthermore, market conditions evolve rapidly, necessitating systems that can continuously learn and adapt while maintaining the reliability of their predictions. The volume and velocity of market data require efficient processing mechanisms that can extract accurate patterns while filtering out noise.

Figure 2. Example of chart with technical analysis markers—Elliott waves, marked as 1-2-3-4-5; screenshot from Ref. [5].

This research addresses these challenges through the development of ElliottAgents, a novel multi-agent framework where specialized agents collaborate to analyze market patterns. Each agent within the system performs a specific function, from data pre-processing and pattern recognition to strategy formulation and reporting, working collectively to enhance analytical accuracy. The system integrates LLMs to enable natural language processing capabilities while implementing RAG to ensure consistent access to domain knowledge [11,12]. The platform incorporates a continuous learning architecture [13,14], where DRL enables agents to improve their pattern recognition capabilities through iterative backtesting of historical data. This adaptive learning mechanism allows the system to systematically enhance its predictive accuracy by validating identified patterns against historical market behaviors, effectively combining the structured approach of EWP with the capabilities of modern AI.

The practical applications of this research are demonstrated through testing on real market data, resulting in generation of actionable trading signals based on recognized patterns. The system's scalable architecture enables real-time market analysis, while its implementation of dynamic context management allows for improved adaptation to market conditions. The implementation of continuous learning mechanisms enhances pattern recognition accuracy, while maintaining interpretability crucial for market practitioners. Through extensive testing on historical market data, this research validates the effectiveness of combining traditional Elliott wave analysis with modern AI techniques through a multi-agent architecture. The results demonstrate that ElliottAgents provides analyses that can be utilized by both professional analysts and retail investors, contributing to both theoretical understanding and practical applications in financial forecasting.

The rest of the paper describes technologies, implementation, and experiments and is structured as follows. Section 2 describes the foundational concepts and technologies underlying our work, including the EWP, LLMs, and DRL. Section 3 presents a survey of prior research in multi-agent systems for financial market analysis. Our multi-agent framework, ElliottAgents, is overviewed in Section 4, detailing its architecture and key components. The experimental setup and performance results of the ElliottAgents system for both stock and cryptocurrency markets are presented and discussed in Sections 5 and 6. Finally, Section 7 concludes the paper and highlights future research directions.

2. Foundational Concepts and Technologies

ElliottAgents core technologies encompass foundational technical analysis and advanced computational methods to enhance financial market analysis. Key elements include the EWP, LLMs, and DRL, each integrated within a LangGraph multi-agent framework. This combination supports pattern recognition, adaptive continous learning, and efficient data processing, collectively enabling the system to generate accurate insights and trend forecasts in complex financial environments.

2.1. Elliott Wave Principle

The Elliott wave principle (EWP), developed by Ralph Nelson Elliott in the 1930s, marked a significant advancement in technical analysis. Through extensive market observation, Elliott identified repetitive patterns in stock market price movements driven by collective investor psychology [1,4]. These patterns, called waves, form the foundation of the EWP and provide a foundation for analyzing market behavior across multiple timescales.

According to the EWP, market movements can be characterized by two primary types of wave patterns. As shown in Figure 3, impulsive waves, consisting of five distinct subwaves (labeled 1–5), move in the direction of the main trend. Within these impulsive sequences, waves 1, 3, and 5 propel the market in the trend direction, while waves 2 and 4 represent necessary countertrend moves. Complementing these are corrective waves, comprising three subwaves (labeled A-B-C), which move against the predominant trend, providing market retracements.

Each wave in Elliott theory can be broken into smaller versions of the same patterns, exhibiting a fractal structure [1], where each wave contains smaller versions of the same patterns and simultaneously forms part of larger wave structures. This hierarchical organization enables traders to analyze market movements across multiple timeframes, from intraday fluctuations to multi-year trends, providing the possibility of understanding market behavior at different scales.

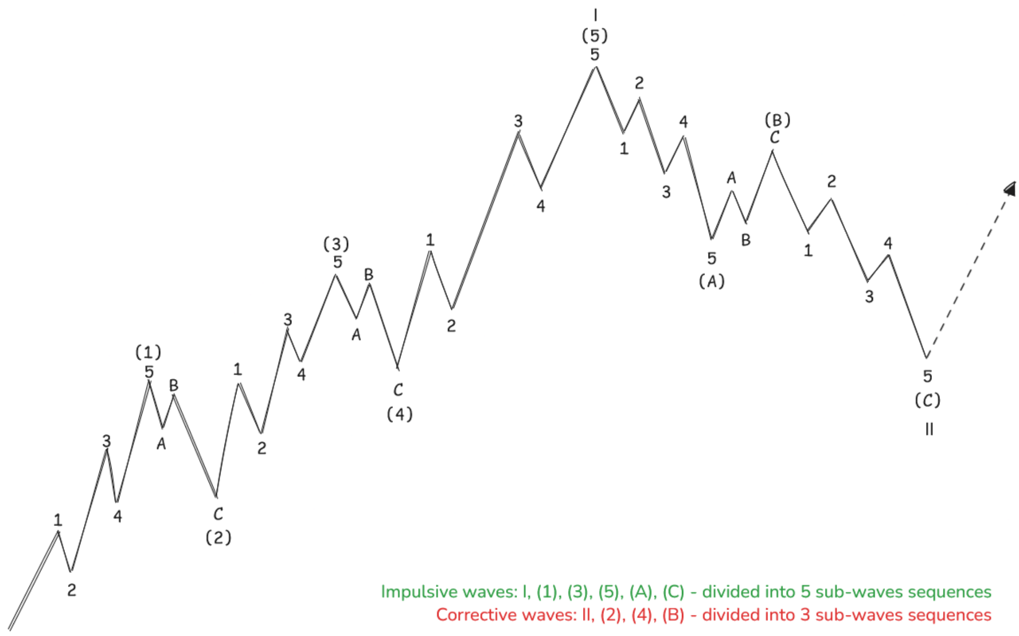


Figure 3. The fractal character of Elliott wave pattern, adapted from Refs. [4,15].

The mathematical foundation of the EWP is integrated with Fibonacci relationships [16]. The Fibonacci sequence, where each number is the sum of the previous two, generates important ratios that appear not only in financial markets but also in nature, art and architecture. The most important of these is the golden ratio (known as φ, approximately 1.618) and its reciprocal 0.618. These ratios help quantify relationships between different waves, with wave 2 often retracing 50% or 61.8% of wave 1, wave 3 frequently extending to 1.618$\times$ the length of wave 1, and wave 4 typically retracing 38.2% of wave 3 [1,4]. Figure 4 present these ratios, applied on the stock market chart, and as we can see, the lines often coincide with local tops and bottoms, predicting where the price of the asset may stop.

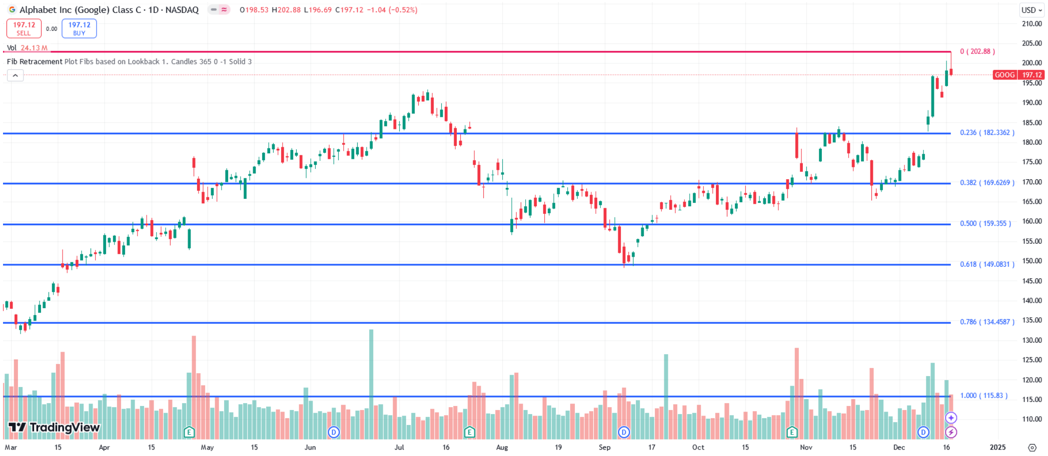


Figure 4. Horizontal lines show Fibonnaci retracement levels of 23%, 38%, 50%, 62%, and 78%, measured from the top of the uptrend; screenshot from Ref. [5].

While the EWP offers a framework for market analysis, its implementation faces several significant challenges [2]. The main challenge lies in wave pattern identification—different analysts often interpret the same market data differently, leading to varying predictions. This subjectivity is complicated by the theory's rules, which lack precise criteria for systematic testing.

Several other limitations affect the theory's effectiveness. The complexity of wave patterns and their variations can lead to overfitting, where analysts might adjust wave counts to match known market outcomes rather than make reliable predictions [17]. The theory's focus on market psychology may overlook important fundamental factors like economic indicators and geopolitical events that influence price movements. Wave patterns become particularly difficult to identify in sideways markets [4], reducing the theory's effectiveness in these conditions.

These challenges are especially relevant in today's high-speed trading environment, where algorithmic trading and rapid information flow affect market behavior. Traders using wave patterns and Fibonacci relationships must also be aware of cognitive biases, particularly confirmation bias, which can influence pattern interpretation [3]. The lack of widespread acceptance among professional traders and economists further highlights the challenges in applying the theory consistently.

2.2. Large Language Models (LLMs)

The introduction of LLMs into multi-agent systems [18,19] has enabled greater adaptability and complexity in data processing and decision-making tasks. LLMs bring advanced NLP capabilities, allowing agents to understand, generate, and interact with human language in a way that enhances the interpretability and responsiveness of the system.

2.2.1. Capabilities and Applications in the Financial Domain

LLMs represent a significant advancement in the field of AI and NLP. These models are designed to understand, generate, and interact with human language in a way that is increasingly indistinguishable from human performance. Examples of LLMs include OpenAI's GPT-4 and GPT-4o [20], Google's Gemini [21], and Claude and the LLaMA series [18]; their releases have been shown in Figure 5. They have been trained on vast amounts of text data and leverage advanced transformer architectures to perform a wide range of language-related tasks, from translation and summarization to question answering and creative writing [18].

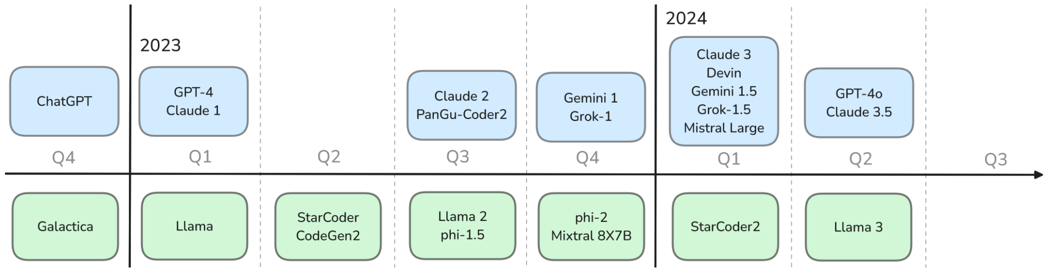


Figure 5. Timeline of foundation models released since 2023. Blue color (upper part) indicates closed-source models and green color indicates open-source models.

Applications of LLMs could be used in various domains [22]. Some of the most common applications include text generation, translation, question answering, summarization, and sentiment analysis. LLMs can generate human-like text based on a given prompt, which is used in creative writing, automated content creation, and chatbots. For example, GPT-4 [20] can write essays, poems, and even code snippets based on a few lines of input. LLMs can translate text from one language to another with high accuracy. Models like Gemini and GPT-4 have been fine-tuned for translation tasks, enabling them to understand and generate text in multiple languages. LLMs can answer questions based on a given context. This is used in applications like virtual assistants, customer support, and educational tools. Summarization models can extract the key points from a document and present them in a coherent manner. LLMs can analyze the sentiment of a given text, determining whether it is positive, negative, or neutral. This is used in social media monitoring, market research, and customer feedback analysis. Sentiment analysis models can help businesses understand customer opinions and make informed decisions.

Despite their impressive capabilities, LLMs face several challenges and limitations [23]. One of the main challenges is the computational cost of training and running these models. LLMs require significant computational resources, including powerful GPUs and large amounts of memory. This makes them expensive to train and deploy. Another challenge is the issue of bias. LLMs are trained on large datasets that may contain biased or unrepresentative data. As a result, the models can learn and propagate these biases, leading to biased or unfair outputs. LLMs also struggle with understanding context and common sense reasoning [23]. While they can generate coherent text, they may not always produce factually accurate or contextually appropriate responses. This is because LLMs rely on patterns in the training data rather than genuine understanding. For instance, an LLM might generate a plausible-sounding but incorrect answer to a factual question.

Time series prediction plays a fundamental role in financial forecasting. Classical approaches, such as ARIMA [24] and exponential smoothing, analyze data by breaking them down into trend, seasonal, and residual components to uncover patterns and forecast future values. However, these models are limited by their assumption of linear relationships, making them less effective at capturing the intricate non-linear dependencies characteristic of financial markets. LLMs leverage deep learning and extensive datasets to understand and predict sequential data [25,26]. They excel at capturing trends and seasonality, especially in datasets with clear patterns. Studies show LLMs like GPT-4 perform well on datasets with strong periodicity [23]. Techniques such as natural language paraphrasing and incorporating external knowledge via prompts can further enhance their performance [27]. However, LLMs face challenges with multi-period datasets, struggling to recognize distinct periods [28], similar to other methods. They are computationally demanding but often perform comparably to simpler models. More research is needed to fully prove their effectiveness, but distributing tasks among specialized agents in a multi-agent system may improve LLM performance on complex financial datasets. While further investigation is required, combining LLMs with multi-agent architectures holds promise for enhancing time series forecasting accuracy in finance.

2.2.2. Architecture

The core principle behind LLMs is the use of neural networks (NNs), specifically a type of network known as the transformer [19]. Transformers have revolutionized the way models process sequential data. Unlike traditional recurrent neural networks (RNNs) and long short-term memory networks (LSTMs), transformers can handle long-range dependencies more effectively, making them ideal for language tasks. Transformers utilize a mechanism called self-attention, which allows the model to weigh the importance of different words in a sentence when making predictions. This is crucial for understanding context, as the meaning of a word often depends on the surrounding words.

The architecture of a transformer consists of an encoder and a decoder, as shown in Figure 6. The encoder processes the input text, and the decoder generates the output text [19]. In practice, many LLMs use only the decoder part (as in GPT models [20]) for tasks like text generation, while others use both encoder and decoder (as in Gemini) for tasks that require understanding and generating text [29]. The encoder's role is to process the input sequence and convert it into a series of continuous representations. It does this through multiple layers of self-attention and feedforward neural networks. Each layer processes the input with a self-attention mechanism that allows the model to focus on different parts of the input sequence. The decoder also consists of multiple layers of self-attention and feedforward neural networks [20]. It takes the encoded representation from the encoder and generates the output sequence, one token at a time. During training, the decoder uses teacher forcing, where the actual previous token is used as input to predict the next token. During inference, it uses its own previous predictions to generate the next token. The self-attention mechanism is the heart of the transformer architecture. It allows the model to consider all positions of the input sequence when generating a particular token in the output sequence. This is achieved by computing a weighted sum of the input representations, where the weights are determined by the similarity between the query and key vectors of the input tokens [18].

The transformer architecture, consisting of an encoder and decoder, revolutionized sequential data processing in LLMs. Self-attention allows the model to weigh the importance of different input elements and handle long-range dependencies effectively. This architecture enables LLMs to process and generate natural language with unprecedented proficiency, opening up new possibilities in AI applications.

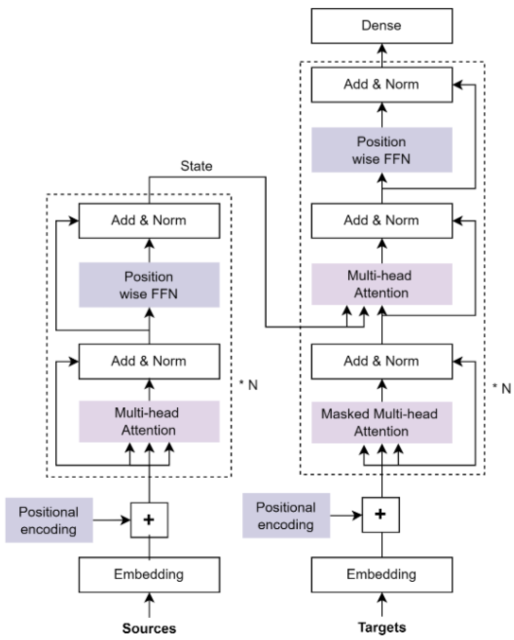


Figure 6. Transformer architecture, reprinted from Ref. [30].

2.2.3. Retrieval-Augmented Generation (RAG)

Retrieval-augmented generation represents advancement in improving the precision and reliability of generative AI systems through systematic knowledge integration. This architecture diverges from traditional generative paradigms that rely exclusively on parametric knowledge stored in model weights, instead implementing a sophisticated bi-directional retrieval mechanism that interfaces with external knowledge repositories [12]. The system operates through a multi-stage process, initially converting incoming queries into dense vector representations (embeddings), then performing efficient similarity matching within a vectorized knowledge base, and finally synthesizing the retrieved contextual information with the model's generative capabilities to produce comprehensive responses, as illustrated in Figure 7. This methodology extends beyond mere information retrieval by enabling dynamic knowledge incorporation, which substantially improves response accuracy and significantly mitigates the occurrence of fabricated information or "hallucinations"—a common challenge in traditional language models [11].

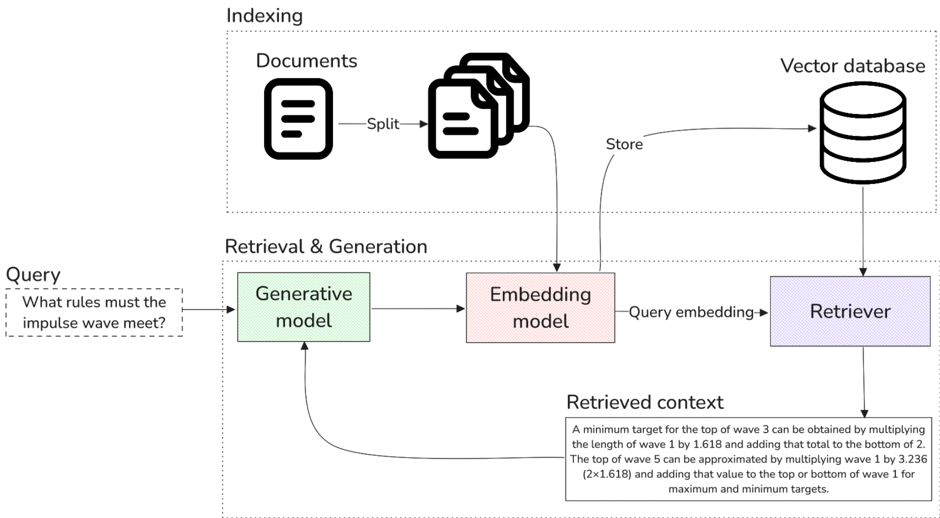


Figure 7. LangGraph RAG algorithm, adapted from Ref. [31].

One significant advantage of RAG is its ability to update knowledge by simply modifying the external knowledge base without requiring model retraining. This feature is particularly valuable in dynamic domains where information needs to be frequently updated. Additionally, RAG models have demonstrated superior performance in maintaining factual accuracy and reducing hallucinations compared to traditional language models. The effectiveness of RAG is further enhanced through the use of dense vector representations for document retrieval, which allows for more nuanced semantic matching compared to traditional keyword-based retrieval methods. The system can process and incorporate multiple retrieved documents, weighing their relevance and combining information from various sources to generate more comprehensive and accurate responses. This multi-document approach is particularly beneficial when dealing with complex queries that require synthesizing information from multiple sources. The integration of retrieval mechanisms allows RAG to maintain up-to-date knowledge bases and provide more reliable, factual responses compared to static parametric-only models

2.3. Multi-Agent Systems

Incorporating multi-agent systems facilitates efficient task distribution and collaboration, allowing specialized agents to handle distinct aspects of complex analytical processes. By enabling agents to work concurrently and share insights, this architecture enhances scalability, adaptability, and overall performance in high-demand environments like financial market analysis.

2.3.1. Agent Engineering

The design of effective agent systems requires consideration of collaboration mechanisms, scaling approaches, and task decomposition strategies [32]. In agent collaboration, two primary architectures have emerged: hierarchical and sequential processing. In hierarchical collaboration, agents communicate freely and process inputs in a network-like structure, while sequential processing involves agents following a structured sequence to build upon each other's outputs [33]. Our system adopts the hierarchical approach as illustrated in Figure 8. This allows for asynchronous execution of tasks and significantly accelerates the prediction process. This choice is motivated by the observation that hierarchical structures enable more flexible communication patterns and can better handle complex, interdependent tasks.

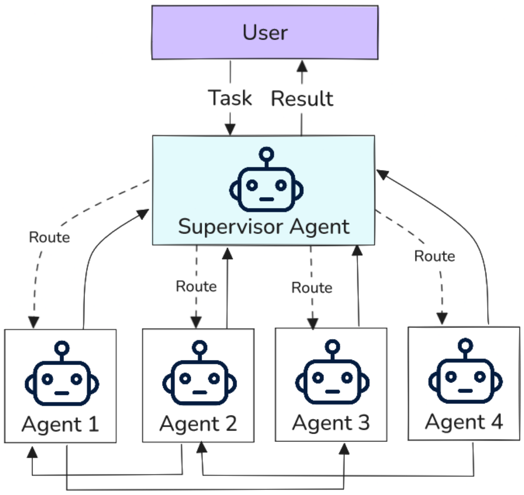


Figure 8. Diagram of supervisor agent in multi-agent hierarchical architecture, adapted from Ref. [34].

The first component of effective multi-agent system is dynamic scaling, which adjusts the number of active agents based on task complexity and available resources [33]. This approach ensures optimal resource utilization and maintains system performance under varying conditions. The system can autonomously increase or decrease the number of agents, ensuring that computational resources are allocated efficiently while maintaining the quality of outputs [35]. Dynamic scaling has proven particularly valuable in handling tasks of varying complexity, where the required computational resources may fluctuate significantly.

The decomposition of large tasks into fewer smaller ones serves as a fundamental strategy in agent engineering, enabling the breakdown of complex tasks into smaller, manageable sub-tasks [36]. In hierarchical task decomposition, tasks are organized into a structured hierarchy where each level can be further decomposed until reaching a granularity suitable for individual agents [7]. This method ensures that specialized agents can handle specific parts of a task, optimizing overall efficiency. Task decomposition not only improves the system's ability to handle complex problems but also enhances the clarity of agent responsibilities and improves overall system maintainability.

Memory management represents another component in agent engineering [32,37]. Effective agents must maintain both short-term memory for immediate context and task-related information and long-term memory for storing substantial volumes of knowledge and past experiences. The integration of these memory systems enables agents to learn from previous interactions and improve their decision-making processes over time. This is particularly important in financial analysis tasks, where historical context and pattern recognition are used for future analyses.

The integration of RAG and DRL has further enhanced agent capabilities. RAG improves the accuracy and reliability of agent outputs by enabling access to external knowledge sources [12], while DRL enables agents to learn and adapt from their experiences [38]. These technologies, combined with dynamic context management, allow agents to maintain relevance and accuracy throughout their operational life cycle.

The successful engineering of AI agent systems requires careful orchestration of multiple components and considerations. From collaboration architectures and dynamic scaling to memory management and tool integration, each aspect plays a crucial role in creating efficient systems. As the field continues to evolve, the integration of advanced technologies like RAG and DR will become increasingly important.

2.3.2. Integration with LLMs

LLMs serve as the cognitive foundation of modern AI agents, functioning effectively as their "brain" and enabling advanced reasoning, planning, and decision-making capabilities [6,8]. This integration transforms traditional agents into more versatile and intelligent systems capable of handling complex tasks through natural language understanding and generation.

The architecture of an LLM-based agent comprises several key components that work in concert to create an intelligent system. At its core, the LLM functions as the central processing unit [33,37], interpreting inputs, generating responses, and coordinating various agent functions through natural language. This integration allows agents to leverage the extensive knowledge and reasoning capabilities embedded within the LLM while maintaining structured interaction with their environment.

As presented on Figure 9, agent cognitive architecture is built around four fundamental components:

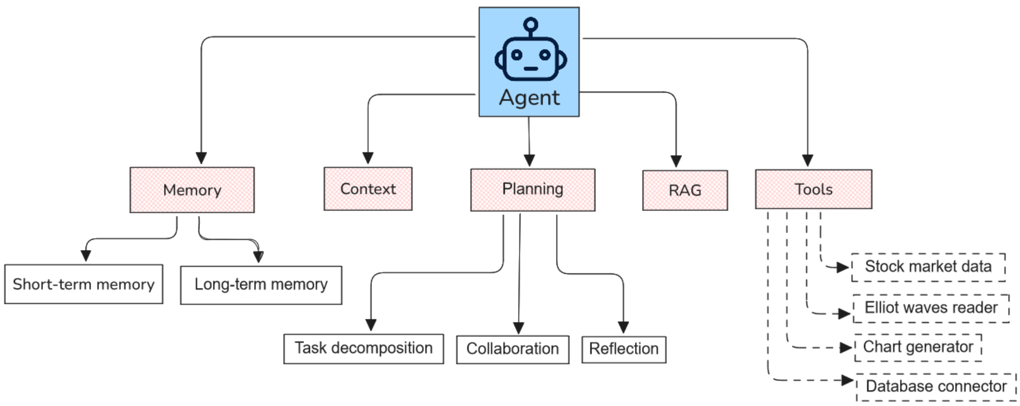


Figure 9. Overview of a LLM autonomous agent, adapted from Ref. [37].

- Planning: The planning component harnesses the LLM's reasoning capabilities to devise action sequences based on objectives and environmental constraints [32]. This module employs various techniques such as Chain-of-Thought (CoT) and Tree-of-Thought (ToT) prompting, enabling the agent to break down complex tasks into manageable steps. The LLM's natural language processing abilities allow it to understand goals, generate plans, and adapt strategies based on changing circumstances.
- Memory: The memory architecture consists of two primary components: short-term memory for immediate context and task-related information and long-term memory for storing substantial volumes of knowledge and past experiences [37]. The LLM interacts with both memory types to maintain context awareness and improve decision-making over time. This dual-memory system enables the agent to learn from past interactions while remaining responsive to current situations.
- Tools: LLM-based agents are equipped with various specialized tools that extend their capabilities beyond pure language processing [8]. The tools component presents specific instruments that were created for our system to adjust it for analysis of financial markets:

  - Stock market data: provides access to real-time and historical market information, essential for informed decision-making.
  - Elliott wave reader: tool for technical analysis, helping the agent identify and interpret Elliott wave patterns in price movements.
  - Chart generator: allows the agent to visualize market data, especially Elliott waves, creating graphical representations of detected waves. This tool is directly connected to Elliott waves reader tool.
  - Database connector: enables the agent to access and manage structured data of backtesting results.
- Context: The context management system enables the agent to maintain and update its understanding of the current situation. The LLM processes incoming information and integrates it with existing knowledge [39], allowing the agent to adapt its responses and decisions based on evolving circumstances. This component ensures that the agent's analyses and recommendations remain relevant and well-grounded in current conditions.

The integration between these components is facilitated by the LLM's natural language capabilities. For example, when an agent needs to analyze market trends, the LLM can interpret complex market data provided by the tools, apply relevant analytical frameworks stored in memory, and generate coherent analysis and recommendations [7,37]. The LLM's ability to process and generate natural language makes it an ideal interface between the agent's various components and external stakeholders.

This architecture enables LLM-based agents to perform sophisticated tasks that were previously challenging for traditional agents. The combination of powerful language understanding, structured tool usage, and effective memory management creates a system that can adapt to new situations, learn from experience, and provide nuanced responses to complex queries.

2.3.3. ReAct Agents

The ReAct (Reasoning and Acting) [40] agent is designed to enable LLMs to engage in dynamic decision-making and problem-solving through iterative cycles of reasoning and action. The ReAct framework, which integrates memory, reasoning, and task execution, allows an agent to navigate complex environments by balancing its cognitive and operational processes. This architecture, depicted in Figure 10, exemplifies a modular system where LLMs interact with tools, memory, and the external environment to achieve task outcomes.

The reasoning and acting process begins when a user assigns it a task. The LLM then initiates reasoning processes based on both current observations and stored memories, allowing it to draw on past interactions or knowledge relevant to the task [37]. ReAct agent abilities include advanced problem-solving, logical reasoning, and the handling of complex tasks, which tend to manifest in models with larger parameter counts. In the ReAct agent, these emergent abilities enable it to navigate complex situations, solve novel problems, and dynamically adjust its actions as needed.

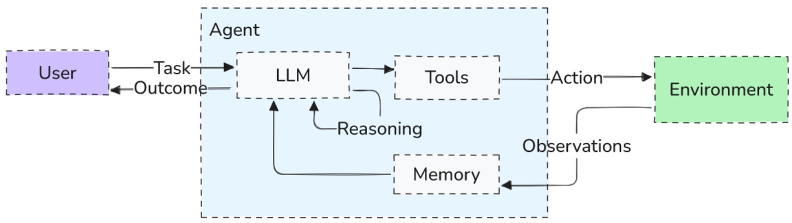


Figure 10. ReAct agent components, adapted from Ref. [41].

The ReAct system utilizes a multi-agent architecture where each agent [37] is specialized in a specific task, such as data retrieval, pattern recognition, or decision-making. These agents collaborate to solve complex problems by sharing information and building upon each other's outputs [8]. The system's flexibility allows it to handle a wide range of tasks, from question-answering and fact verification to interactive decision-making in simulated environments.

One of the key strengths of ReAct is its ability to overcome limitations faced by traditional reasoning or acting approaches used in isolation. For instance, in question-answering tasks, ReAct can interact with external APIs to verify information and avoid hallucinations that might occur with pure reasoning approaches.

ReAct has demonstrated impressive performance across various benchmarks [40], often outperforming baselines with just a few in-context examples [33]. Its success in tasks ranging from multi-hop question answering to navigating text-based games highlights the potential of this approach for developing more capable AI systems.

2.4. Deep Reinforcement Learning (DRL)

Deep reinforcement learning represents a fusion of deep learning and reinforcement learning (RL) principles within machine learning. As illustrated in Figure 11, DRL follows a cyclical interaction pattern where an agent engages with its environment through a continuous feedback loop [42]. In this process, the agent observes the current state, takes an action at based on that state, and receives both a reward and transitions to a new state from the environment. The agent's goal is to learn an optimal policy that maximizes cumulative rewards through these repeated interactions. Unlike traditional machine learning approaches, DRL enables agents to learn complex decision-making strategies through direct environmental interaction, making it particularly well suited for sequential decision tasks.

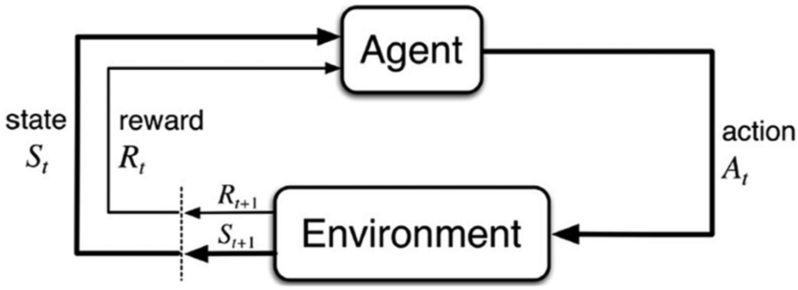


Figure 11. The agent–environment interaction in the Markov Decision Process (MPD) for ranking information with reinforcement learning, reprinted from Ref. [43].

Deep learning, utilizing neural networks with several layers [42], enhances RL by effectively managing complex state and action spaces. Prominent DRL approaches include Deep Q-Networks (DQN), which leverage neural networks to approximate Q-values (the expected rewards associated with specific actions), and Policy Gradient methods, which focus on directly optimizing the policy [38]. DRL employs techniques such as experience replay, where previously encountered experiences are stored and reused to improve training efficiency, and target networks, which ensure training stability by maintaining consistent target values.

At the core of DRL is the concept of reinforcement learning, a type of machine learning where an agent learns to make decisions by performing actions in an environment to maximize some notion of cumulative reward. The agent observes the state of the environment, takes actions, and receives feedback in the form of rewards or penalties. Over time, the agent learns to take actions that maximize the total reward, which is formally defined as the expected sum of discounted future rewards:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \quad (1)$$

where R_t is the reward at time step t and γ is the discount factor that determines the present value of future rewards.

One of the seminal works in DRL is the development of the Deep Q-Network (DQN) [42]. The DQN algorithm combines Q-learning, a popular RL algorithm, with deep neural networks.

In Q-learning, the agent learns a Q-function, which estimates the expected cumulative reward for taking a given action in a given state and following the optimal policy thereafter. The Q-function is updated using the Bellman equation:

$$Q(s_t, a_t) = R_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a') \tag{2}$$

where $Q(s_t, a_t)$ is the action-value function, representing the expected return (reward) of taking action a_t in state s_t. R_{t+1} is the reward received after taking action a_t at time step t. γ is the discount factor, which determines the present value of future rewards. $\max_{a'} Q(s_{t+1}, a')$ represents the maximum expected future reward, starting from the next state s_{t+1} and considering all possible actions a'.

Traditional Q-learning stores this Q-function in a table, which becomes impractical for large state spaces. Instead, DQN uses a neural network to approximate the Q-function, where the Q-values for all possible actions are output for a given state.

The training of a DQN involves two main components: the neural network and an experience replay buffer [44]. The neural network takes the state as input and outputs Q-values for all possible actions. The experience replay buffer stores the agent's experiences, which are tuples of the form (state s_t, action a_t, reward r_{t+1}, next state s_{t+1}). During training, random batches of experiences are sampled from the replay buffer to update the neural network. This process helps to break the correlations between consecutive experiences and stabilizes the training.

3. Related Work

Multi-agent systems for financial market analysis have evolved significantly over time. Early frameworks relied heavily on predefined rules and simple algorithms for stock market trading [45]. Some researchers applied fuzzy logic in conjunction with multi-agent systems for stock technical analysis [9]. While innovative, these early systems demonstrated limited adaptability to rapidly changing market conditions.

The rigidity of traditional multi-agent systems in the face of dynamic financial markets has been a significant limitation. Research has shown that these systems often struggled with unexpected market changes or new price patterns not accounted for in their original design [10]. Recent advancements in LLMs have opened new possibilities in financial analysis. LLMs offer enhanced capabilities in natural language understanding and reasoning, enabling more sophisticated market information analysis.

There is a notable scarcity of systems that effectively combine traditional technical analysis methods with modern AI techniques [46,47]. While various methods of hierarchical optimization and their parallel implementation for portfolio selection have been proposed [48], and some studies have applied Elliott wave theory to forecast market price trends [2], these approaches have not leveraged the advanced capabilities of AI, which evolves over time.

LLMs, on the other hand, leverage deep learning techniques and extensive datasets to understand and predict time series data [19,20,29]. These models leverage their capability to understand and generate sequential data, which is crucial for accurate forecasting of time series characterized by trends and seasonal patterns [24]. There are also other transformer-type models that are successfully used in stock market prediction, like Hidformer models [49]. However, challenges remain, particularly with multi-period datasets, where LLMs struggle to recognize distinct periods [23], and similar problems apply to all other methods [26,28]. Despite their computational demands, LLMs often perform on par with simpler models, suggesting that they hold potential, but more research is needed to prove their effectiveness. The use of agents may be a factor that will greatly improve the results of time series prediction by distributing tasks among agents [6], enabling a more robust analysis of complex big sets of data.

This gap in the literature underscores the need for more comprehensive systems that can harness both proven technical analysis methods and AI. Integrating these approaches has the potential to significantly enhance the accuracy of predictions and the adaptability of financial market analysis systems.

4. ElliottAgents System Architecture

The architecture of ElliottAgents is designed to streamline the analysis of financial data through a multi-layered, modular system implemented in Python v3.11. At its core, the platform combines specialized agents built using the LangGraph framework within LangChain, each responsible for specific tasks, including data retrieval, pattern recognition, backtesting, and report generation. Market data are sourced through the yfinance API and processed using pandas for efficient data manipulation. The system employs a neo4j graph database for storing backtesting results and LangChain chroma for vector storage, while the OpenAI API powers the language model capabilities of each agent. This design leverages a centralized coordinator to manage agent workflows, enabling seamless data flow and integration of insights across agents, with a streamlit-based GUI providing accessible visualization capabilities. By incorporating DRL for continuous improvement and adaptable decision-making, the ElliottAgents system architecture enhances predictive accuracy and adaptability, catering to the complexities of dynamic financial markets.

4.1. Main Assumptions of the Proposed System

The objectives of the ElliottAgents platform are designed to address the challenges faced by traders and financial analysts in the cryptocurrency market. The primary aim is to develop a platform that improves the accuracy and timeliness of predictions. The following objectives have been formulated to address the research question and guide the development of the platform:

1. User-defined analysis parameters: The system allows users to specify the symbol, time frame, and interval of the analysis.
2. Integration of EWP into a multi-agent architecture: This objective focuses on the design and implementation of a multi-agent system that incorporates the EWP. Agents should accurately identify and classify Elliott wave patterns in stock market data.
3. Real-Time data integration: Creates mechanisms for efficient processing of recent market data by AI agents.
4. Multi-faceted analysis and user-friendly presentation: Involves developing a system for stock market data analysis and creating an interface for clear presentation of analytical results. The results should be presented in an easy-to-understand manner and include visualization of the results.
5. Implementation of DRL: Implements a DRL mechanism for AI agents to improve performance based on previous interactions.
6. Scalability and future development: Explores scalability of the proposed system and potential avenues for future enhancements and applications in broader financial contexts.

4.2. Multi-Agent Framework

The multi-agent framework within ElliottAgents organizes agents into specialized teams that collaboratively execute complex financial analysis tasks. Each agent is designed with a distinct role, allowing for precise task decomposition and efficient information flow across the system. Through structured coordination and dynamic task allocation, agents can adapt their focus based on real-time requirements, improving responsiveness and scalability.

The ElliottAgents system implements a hierarchical multi-agent architecture where specialized agents are organized into analysis and reporting teams to optimize data flow and task execution. Figure 12 illustrates the system's agent hierarchy and information flow pathways.

The system's components are distributed, with specific agents having access to dedicated analytical tools, including chart generators for visual analysis, database connectors for historical data management, and a wave reader algorithm and APIs for real-time market data acquisition. This distributed toolset architecture enables specialized processing capabilities while maintaining system cohesion through centralized coordination.

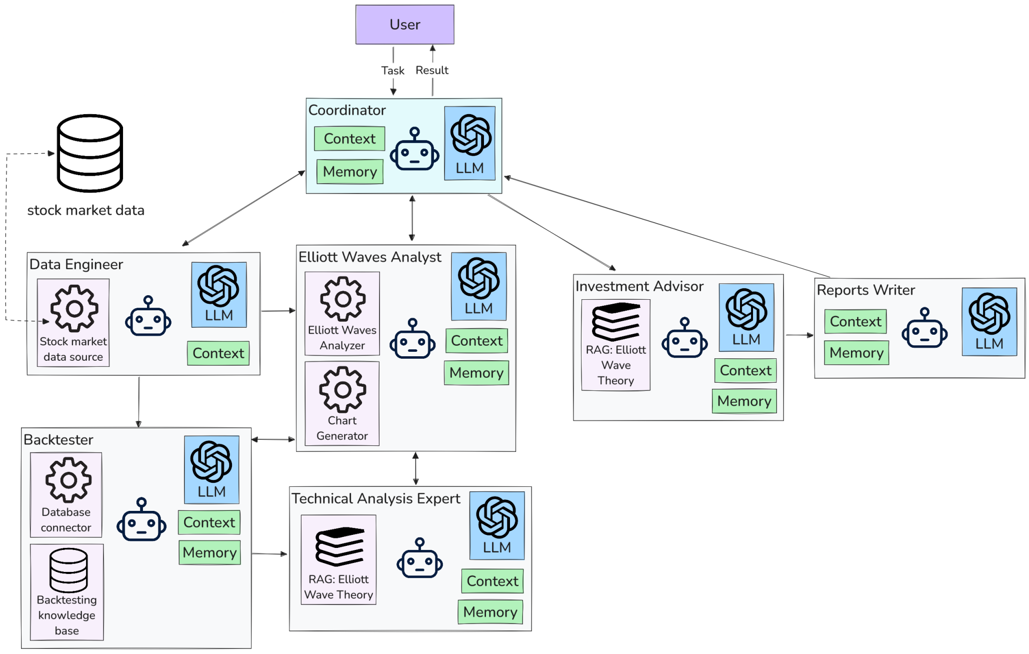


Figure 12. Diagram showing the components used by each agent and the flow of data between them.

The system's implementation incorporates RAG mechanisms and external knowledge bases, which enhance the analytical capabilities of specific agents. The Technical Analysis Expert agent utilizes RAG to access comprehensive Elliott wave theory documentation, enabling pattern recognition and analysis. This implementation is demonstrated in the agent's context definition, shown in Figure 13.

```
technical_analysis_expert_prompt_template = '''
    You are a technical analysis expert specializing in Elliott Wave Principle.
    Use the context provided by the RAG tool to analyze and identify Elliott Wave patterns in the given stock price data.
    Your goal is to provide a comprehensive analysis of the current wave structure and potential future price movements.

    You have access to the following tool: {tools}

    When analyzing the stock price data:
    1. First, identify the current major wave count (Cycle, Primary, Intermediate, Minor).
    2. Look for the five-wave impulse and three-wave corrective patterns.
    3. Verify if the waves adhere to the Elliott Wave rules:
       - Wave 2 never retraces more than 100% of Wave 1
       - Wave 3 is never the shortest among Waves 1, 3, and 5
       - Wave 4 never enters the price territory of Wave 1
       - Wave 3 is often the longest and most powerful wave
    4. Identify any potential wave extensions or truncations.
    5. Look for Fibonacci relationships between waves.
    6. Consider the time factor and any potential time ratios between waves.
    7. Analyze the volume patterns accompanying the price movements.
    8. Based on the current wave count, project potential future price movements and key levels (support/resistance).

    Remember to use the RAG tool to retrieve relevant information about Elliott Wave patterns, rules, and guidelines
    whenever needed. Provide a detailed analysis with clear reasoning for your wave count and projections.

    Begin!
'''
technical_analysis_expert_agent = create_react_agent(llm, tools: [pdfs_search_tool()],
                                                     state_modifier=technical_analysis_expert_prompt_template)
```

Figure 13. Fragment of Python code, presenting prompt and ReAct agent definition.

The prompt structure represents the system's approach to agent specialization, showing how natural language instructions are transformed into specific analytical capabilities. The Technical Analysis Expert prompt outlines the scope of analysis required, incorporating both foundational Elliott wave principles and pattern recognition parameters.

The system's agent composition, detailed in Table 1, presents a structured hierarchy of seven specialized analytical units. Each agent's role is defined through specific responsibilities and access to relevant tools and knowledge bases.

Table 1. System agent roles and their responsibilities.

Role	Responsibilities
Coordinator	Serves as the central orchestrator of the system, managing workflow coordination and inter-agent communication. The agent interfaces between users and analytical components, distributes user inputs across the agent network, and ensures assembly of final outputs.
Data Engineer	Functions as the data preparation specialist, responsible for processing raw market data based on user-specified parameters. The agent fetches market historical prices through external yfinance API and prepares structured datasets for analysis.
Elliott Wave Analysts	Specializes in technical pattern recognition through identification of Elliott wave formations in historical data. This agent implements wave pattern detection algorithms and generates annotated price charts with wave overlays.
Backtester	Employs deep reinforcement learning to validate Elliott wave pattern predictions. The agent maintains a historical pattern performance database and optimizes pattern recognition accuracy through iterative learning processes.
Technical Analysis Expert	Provides advanced pattern interpretation by synthesizing backtesting results with wave patterns. The agent accesses RAG-enhanced Elliott wave theory knowledge base and determines highest probability market scenarios.
Investment Advisor	Develops actionable trading strategies through integration of technical analysis insights. The agent performs RAG-assisted market context evaluation and determines comprehensive investment strategies.
Report Writer	Produces comprehensive analysis documentation by synthesizing multi-agent analytical outputs and creating user-friendly market analysis reports.

4.3. Knowledge Integration Through RAG

In our system, we use knowledge self-reflective RAGs to enhance information retrieval and generation [11]. Knowledge graphs provide a mechanism that links diverse data points, facilitating efficient access to relevant information. By structuring data into interconnected graphs, our RAG system can better disambiguate queries and enhance response precision by leveraging these structured relationships [50]. This method improves the accuracy and relevance of generated content, allowing the model to more efficiently handle data containing a complete description of patterns and the mathematical theory behind the EWP.

4.4. Continous Learning

The ElliottAgents platform implements a continuous learning process [13,14] that enables agents to adapt and refine their knowledge over time [51]. This process is designed to enhance the system's predictive capabilities without relying on traditional fine-tuning methods. Instead, agents learn organically through their interactions and observations of the cryptocurrency market environment.

At the core of this process is the Backtester agent, which plays a crucial role in accumulating and leveraging historical knowledge. As illustrated in Figure 14, the Backtester's workflow begins with a query to determine if relevant results are already available in the backtesting knowledge base. If not, the agent initiates an analysis by fetching the necessary data, performing EWP analysis, and interpreting the results. These findings are then stored in the Neo4j v5.20 graph database for future reference.

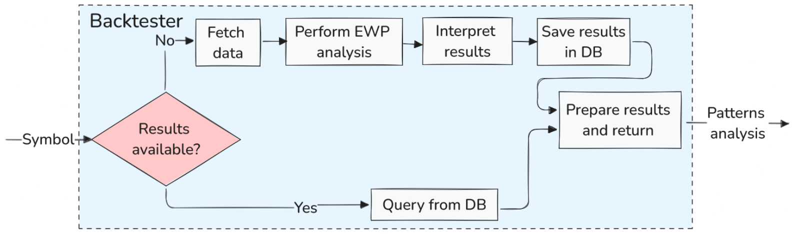


Figure 14. Logic inside Backtester agent.

This iterative process allows the system to build a repository of analyzed patterns and outcomes over time. This approach ensures that the system's predictions are grounded in a historical data and previously observed market behaviors.

The continuous learning process is further enhanced by the agents' ability to freely learn from their environment [6]. Rather than undergoing periodic fine-tuning, which can be resource-intensive and potentially disruptive, the agents incrementally update their understanding based on the outcomes of their predictions and the evolving market conditions. This adaptive learning approach allows for real-time improvements in prediction accuracy and adaptability to changing market dynamics.

5. Experimental Setup

The experimental validation of ElliottAgents was conducted through a series of experiments simulating real market conditions. Our methodology encompassed both controlled pattern recognition experiments and simulated trading scenarios, utilizing historical market data to assess the system's predictive capabilities. The experimental design focused on the accuracy of Elliott wave pattern identification, the practical utility of the generated trading signals, and the impact of the backtesting process.

5.1. Data Sources

Our experiments utilized historical market data obtained through the yfinance library, integrating real-time market data. The dataset comprises two distinct categories of financial instruments:

- Traditional Equity Securities:
  - Oracle Corporation (ORCL): Representative of established technology sector;
  - Taiwan Semiconductor Manufacturing Company (TSM): Major semiconductor industry constituent;
  - JPMorgan Chase & Co (JPM): Leading financial services institution.
- Cryptocurrency Assets:
  - Bitcoin/USD (BTC/USD): Primary cryptocurrency benchmark;
  - Ethereum/USD (ETH/USD): Leading smart contract platform;
  - Solana/USD (SOL/USD): Emerging blockchain protocol.

The analysis utilizes both daily and hourly intervals to capture fractal formations and detailed wave structure evolution. The historical data span approximately four years (2020–2024) for the daily interval and approximately 4 months for the hourly interval (June–September 2024). This extended temporal scope ensures sufficient data density for pattern validation while capturing the various market phases.

We used OpenAI's GPT-4o-mini model [20], with default settings for temperature parameter, as the core language model for the agent system. It enables natural language processing capabilities across the multi-agent architecture. This choice was based on evaluation of contemporary language models, including Mixtral, Claude, and other GPT variants. While all tested models demonstrated competent natural language processing capabilities, GPT-4o-mini emerged as the optimal choice due to two primary factors: its context length management and favorable cost–efficiency ratio.

5.2. Evaluation Metrics

The quality of ElliottAgents outputs was assessed using a framework focused on its pattern recognition capabilities, predictive accuracy, and usability for end-users. Our evaluation methodology revolved around the system's ability to deliver actionable insights based on identified Elliott wave patterns while maintaining alignment with theoretical constructs like Fibonacci ratios [4]. We designed the evaluation to measure:

- Pattern recognition: Verifying the system accuracy in detecting Elliott wave structures such as impulsive and corrective waves.
- Actionable insights: Evaluating the relevance and clarity of trading recommendations, including specific entry, exit, stop-loss, and take-profit levels and visualization of current market situation on a chart.
- Predictive accuracy: Comparing forecasted price movements and targets to actual outcomes using historical data.
- Adaptability: Testing the system across diverse financial instruments and market conditions.
- Backtesting impact: Quantifying the effect of DRL on improving pattern identification and subsequent predictions.

The experimental framework focused on validating ElliottAgents' performance through a set of specific use cases. Each use case represented a common scenario in financial analysis. The Table 2 summarizes these use cases and their objectives.

Table 2. Summary of use cases.

Use Case	**Description**
Impulse wave detection	Identify five-wave sequences that signify major market trends.
Corrective wave recognition	Detect three-wave corrections following impulsive sequences, predicting potential trend reversals.
Full-cycle analysis	Analyze complete wave cycles (impulsive and corrective) to identify broader market trends.
Wave extension analysis	Identify extended waves exceeding standard Fibonacci ratios, highlighting significant trends.
Support and resistance levels	Calculate critical price levels based on wave endings to assist in risk management.

Although the EWP identifies thirteen distinct wave patterns in market movements, including various complex formations such as triangles, flats, and zigzags [4], our experimental framework concentrated on the most fundamental and frequently occurring patterns.

5.3. Experiments Design

Experimental validation of ElliottAgents was conducted through a three-phase testing, designed to evaluate the system performance in real-world market conditions. Each phase was structured to assess specific aspects of the system's capabilities, from fundamental pattern recognition to advanced predictive analytics.

5.3.1. Pattern Recognition Validation

The initial phase focused on evaluating the system's pattern recognition capabilities across the previously defined wave categories. This validation process involved identification of established wave patterns in historical data, coupled with verification of agents' analyses.

5.3.2. Historical Market Analysis

The second phase encompassed testing using historical market data from Alphabet Inc. (GOOGL) stock and the BTC-USD cryptocurrency pair. The analysis utilized two years of daily price movements, implementing an iterative testing methodology to thoroughly evaluate system performance. This approach enabled progressive data addition to assess the system's adaptability to evolving market conditions. The testing protocol focused on pattern recognition capabilities in live market conditions, generation of actionable trading signals, and overall system adaptability to different market scenarios.

5.3.3. Cross-Validation and DRL Impact Assessment

The final phase implemented cross-validation testing to evaluate pattern recognition accuracy and quantify the impact of DRL on prediction performance. This phase analyzed 1000 candlesticks across both daily and hourly intervals, utilizing ten years of historical data for DRL training. The analysis focused on two primary pattern types: incomplete (1-2-3-4) and complete (1-2-3-4-5) impulsive waves.

The validation criteria were tailored to each pattern type. For incomplete waves (1-2-3-4), predictions were evaluated based on the directional accuracy of subsequent price movements over a period calculated as 1.62× the length of the first wave. This multiplier, derived from Fibonacci relationships integral to Elliott wave theory, provided a standardized timeframe for validation. For complete waves (1-2-3-4-5), the validation process examined the correlation between wave 5 and the subsequent corrective wave A, focusing on both directional alignment and magnitude comparison. Incorporated comparative analysis between standard pattern recognition and DRL-enhanced prediction, enabling assessment of the DRL component's contribution to prediction accuracy.

6. Results

In this section, we summarize the outcomes of experimental validations conducted across various financial instruments. The results illustrate the accuracy and reliability of the multi-agent framework in detecting complex market patterns and generating actionable trading insights.

6.1. Detecting Different Waves Types

In the first part of the experiment, we evaluated ElliottAgents' ability to accurately identify and classify Elliott wave formations across diverse market conditions. This analysis focused on four fundamental pattern categories: impulsive waves, corrective waves, full wave cycles, and wave extensions. The process of validating the patterns found consisted of checking the result of the analysis with historical data and assessing whether the recommendation made is correct and takes into account the principles described by Elliott.

6.1.1. Impulse Waves

Impulse waves, as defined by the EWP, often signal the continuation of a dominant trend. However, after the completion of the fifth wave, a corrective phase is typically expected. Figure 15 shows an analysis of Amazon's stock over 1 year period using daily intervals. After identifying the completion of the fifth wave at USD 210 per share, ElliottAgents issued a sell signal, anticipating a corrective phase. The target price was set at USD 198, corresponding to the end of the third wave within the correction. While the price briefly spiked above USD 210 for one day, it subsequently reversed and adhered to the predicted corrective path. The stock reached the USD 198 target within ten days, resulting in a theoretical profit of USD 12 per share, representing a 6% gain.

Figure 15. Impulse wave (labeled 1-2-3-4-5) recognized on AMZN 1d chart.

6.1.2. Corrective Waves

Recognizing corrective waves can help in understanding when charts will reverse and start follwoing the main chart direction. Figure 16 presents the result of analysis performed on Bitcoin/USD over a 1-year peroid. The primary recommendation was to buy, with a projected bullish momentum leading to higher prices. Specific price levels were identified for risk management: support level at USD 54,948, resistance level at USD 73,330, entry price for a long position at USD 63,329, and recommended stop-loss level at USD 54,948. A take-profit target was set at USD 80,000, aligning with the expected continuation of the identified impulse waves. The system also provided a timeframe for the anticipated price movement, expecting it to occur within 1 to 3 months. Additionally, it offered an alternative scenario, warning that a break below USD 54,948 would necessitate a reassessment of positions and could indicate a deeper correction.

Figure 16. Partial corrective wave (labeled A-B) found on the Bitcoin 1d chart.

6.2. Historical Market Data Analysis

In the second part of the experiment, the system tests were conducted using historical data at the daily interval. The system was run on limited historical data from Alphabet Inc. and the BTC-USD pair, with data from the past two years, to recognize specific waves pattern and identify possible buy or sell signals in the charts using knowledge from the backtesting process. When the system issued such a signal, we iteratively added additional historical data, allowing the system to detect other patterns and issue other signals. This approach enabled us to evaluate its effectiveness in simulated but realistic market conditions.

Figure 17 illustrates the pattern recognition capabilities of ElliottAgents applied to the Alphabet Inc. stock price data over a two-year period from October 2022 to October 2024. The system identified multiple Elliott wave patterns and corresponding trading signals across varying market conditions. Vertical annotations indicate buy and sell signals generated by the system based on pattern completion and wave sequence analysis. The chart demonstrates the system's ability to recognize both impulsive and corrective wave patterns across different market phases.

Figure 17. All waves (marked with blue lines) detected by ElliottAgents for Alphabet stock over a 2-year period applied in one chart.

Figure 18 presents the application of ElliottAgents pattern recognition algorithms to Bitcoin/USD price data spanning October 2022 to September 2024. The analysis captured significant price movements from USD 20,000 to USD 70,000, demonstrating the system's adaptability to cryptocurrency market volatility. The vertical indicators denote algorithmic buy and sell signals generated through wave pattern identification. The system successfully identified multiple trading opportunities across varying market conditions, including both consolidation periods (as seen in mid-2023) and trending phases. Of particular interest is the system's performance during high-volatility periods characteristic of cryptocurrency markets, suggesting robust pattern recognition capabilities across different market states.

Figure 18. All waves (marked with blue lines) detected by ElliottAgents for BTC-USD over a 2-year period applied in one chart.

The recurring identification of tradeable patterns, marked by alternating buy and sell signals, suggests consistent pattern recognition performance across different market conditions and price levels. This long-term analysis validates the system's capability to maintain reliable pattern detection across extended time periods while adapting to changing market dynamics.

6.3. Quantity Tests for Pattern Recognition

The final phase of experimentation focused on quantitative validation of pattern detection accuracy and the impact of DRL on prediction reliability. This phase implemented a rigorous cross-validation methodology analyzing 1000 temporal data points (candlesticks) at daily intervals across previously examined equity instruments and the BTC-USD cryptocurrency pair. The experiments examined two distinct pattern categories: incomplete impulsive waves (1-2-3-4) and complete impulsive waves (1-2-3-4-5), and in each case waves could not overlap.

The methodological approach incorporated comparative analysis between standard pattern recognition and DRL-enhanced prediction capabilities. Pattern validation was validated on price direction prediction (upward or downward price movement). The validation window, denoted as n periods, was determined through theoretical relationships established in the EWP: for incomplete patterns (1-2-3-4), n was calculated as $1.62\times$ the first wave length, corresponding to the theoretical extent of wave five. For complete patterns (1-2-3-4-5), validation examined the relationship between wave 5 and the subsequent corrective wave A, requiring that waves have the same length.

The experimental results, presented in Table 3, demonstrate several findings from the cross-validation analysis conducted across 1000 data samples in dual time intervals. Analysis shows that complete impulsive wave patterns consistently achieved higher predictive accuracy compared to their incomplete counterparts. This differential in predictive capability suggests that fully formed wave structures provide more reliable indicators for future price movements, aligning with theoretical expectations from the EWP. Analysis conducted at hourly intervals reveals a reduction in pattern identification frequency. This can be attributed to the increased granularity of price movements at shorter time scales, where market noise may obscure underlying wave formations. The integration of DRL demonstrated substantial impact on prediction accuracy across all tested scenarios. This enhancement suggests successful implementation of the learning process.

Table 3. Comparison of pattern recognition with and without backtesting for stock market.

Interval	Stock	1-2-3-4 Patterns			1-2-3-4-5 Patterns		
		N	Without Backtesting	With Backtesting	N	Without Backtesting	With Backtesting
Daily Interval	ORCL	19	57.89%	73.68%	15	60.00%	73.34%
	JPM	24	58.34%	66.67%	17	52.94%	64.70%
	TSM	22	59.09%	63.64%	23	65.22%	82.61%
Hourly Interval	ORCL	12	58.34%	66.67%	9	66.67%	88.89%
	JPM	10	50.00%	70.00%	8	62.50%	75.00%
	TSM	11	45.45%	58.33%	9	77.78%	77.78%

N: number of patterns found.

Table 4 presents the results of the cross-validation experiments for 1000 data samples in two time intervals on Bitcoin cryptocurrency. Again, the identification of a complete impulsive wave pattern contributes to better predictions of subsequent price movements than an incomplete impulse wave pattern even in intensive cryptocurrency markets. The use of DRL resulted in a improvement in prediction, similar to the one that we display in Table 3, showing that agents are able to use the learning process with historical data in better interpretation of patterns.

Table 4. Comparison of pattern recognition with and without backtesting for cryptocurrency market.

Interval	Cryptocurrency	1-2-3-4 Patterns			1-2-3-4-5 Patterns		
		N	Without Backtesting	With Backtesting	N	Without Backtesting	With Backtesting
Daily Interval	BTC/USD	12	58.34%	66.67%	9	66.67%	88.89%
	ETH/USD	13	53.84%	61.54%	9	77.78%	77.78%
	SOL/USD	10	50.00%	70.00%	8	62.50%	75.00%

Table 4. *Cont.*

Interval	Cryptocurrency	1-2-3-4 Patterns			1-2-3-4-5 Patterns		
		N	Without Backtesting	With Backtesting	N	Without Backtesting	With Backtesting
Hourly Interval	BTC/USD	8	62.50%	62.50%	6	50.00%	66.67%
	ETH/USD	9	44.44%	66.67%	6	66.67%	66.67%
	SOL/USD	12	66.67%	83.34%	9	66.67%	77.78%

N: number of patterns found.

6.4. Result Analysis

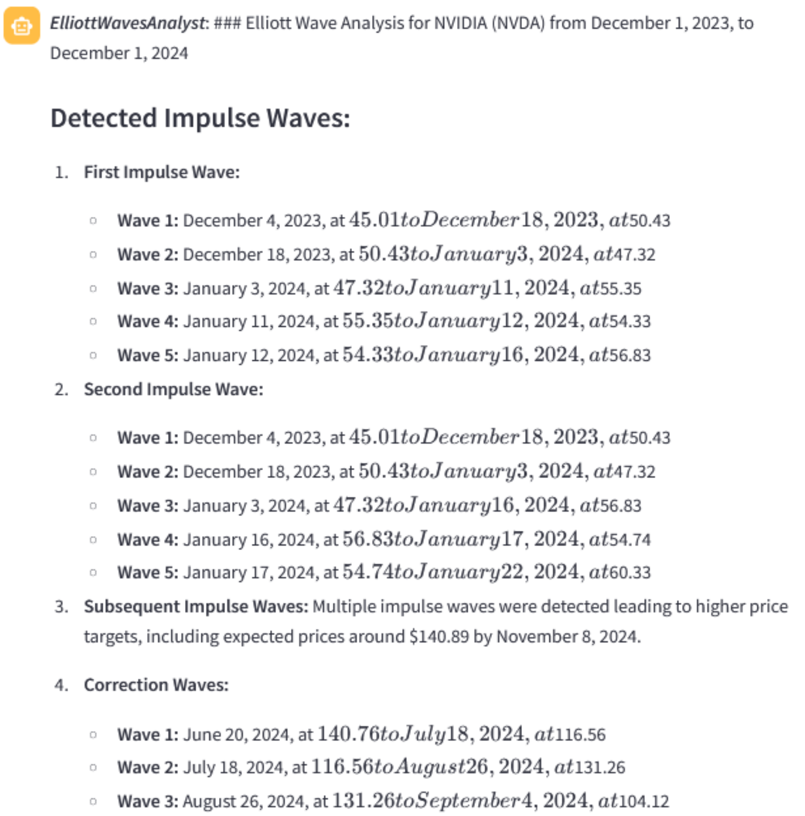
Results demonstrate the ElliottAgents platform's ability to not only identify Elliott wave patterns but also to translate these patterns into practical trading strategies with specific price targets and risk management guidelines. Our agents can communicate using natural language, and Figures 19 and 20 demonstrate a single message returned by an agent, used for communication in order to perform analysis. The initial phase of experimentation demonstrated the system's capability to detect and interpret various Elliott wave formations with high accuracy. The analyses generated by the platform proved accessible to end-users without requiring advanced financial knowledge, suggesting potential applications for both individual investors and professional traders.

Cross-validation testing revealed insights into the system's pattern recognition capabilities. The analysis of 1000 samples across multiple securities demonstrated that complete impulsive wave patterns (1-2-3-4-5) consistently yielded higher prediction accuracy compared to incomplete patterns (1-2-3-4). The integration of deep reinforcement learning through backtesting substantially enhanced pattern recognition accuracy, with improvements of up to 16% observed in daily interval analysis. This improvement was consistent across both traditional equity markets and cryptocurrency trading pairs, suggesting robust adaptability to different market conditions.

The platform demonstrated particular strength in daily interval analysis compared to hourly intervals, indicating enhanced effectiveness in identifying and analyzing longer-term market movements. This finding aligns with the Elliott wave principle's emphasis on fractal pattern recognition across different time scales. The system's performance in cryptocurrency markets proved especially noteworthy, maintaining consistent accuracy despite the characteristically higher volatility of these markets.

The small number of detected patterns in certain test phases can be attributed to the strict non-overlapping criteria implemented in the pattern recognition algorithm, particularly relevant in daily interval analysis where wave formations often extend over multiple trading sessions. This methodological constraint, while potentially limiting the quantity of identified patterns, contributed to the high quality and reliability of the pattern recognition results.

These experimental outcomes validate the efficacy of combining traditional Elliott wave analysis with modern artificial intelligence techniques through a multi-agent architecture. The system's demonstrated ability to maintain consistent performance across varying market conditions, timeframes, and asset classes suggests its potential utility as a practical tool for market analysis and trading decision support. The integration of backtesting capabilities through DRL proved particularly valuable in enhancing the system's predictive accuracy, indicating the potential for continued improvement through learning from the past.



ElliottWavesAnalyst: ### Elliott Wave Analysis for NVIDIA (NVDA) from December 1, 2023, to December 1, 2024

Detected Impulse Waves:

1. **First Impulse Wave:**
    - **Wave 1:** December 4, 2023, at $45.01 to December 18, 2023, at$50.43
    - **Wave 2:** December 18, 2023, at $50.43 to January 3, 2024, at$47.32
    - **Wave 3:** January 3, 2024, at $47.32 to January 11, 2024, at$55.35
    - **Wave 4:** January 11, 2024, at $55.35 to January 12, 2024, at$54.33
    - **Wave 5:** January 12, 2024, at $54.33 to January 16, 2024, at$56.83
2. **Second Impulse Wave:**
    - **Wave 1:** December 4, 2023, at $45.01 to December 18, 2023, at$50.43
    - **Wave 2:** December 18, 2023, at $50.43 to January 3, 2024, at$47.32
    - **Wave 3:** January 3, 2024, at $47.32 to January 16, 2024, at$56.83
    - **Wave 4:** January 16, 2024, at $56.83 to January 17, 2024, at$54.74
    - **Wave 5:** January 17, 2024, at $54.74 to January 22, 2024, at$60.33
3. **Subsequent Impulse Waves:** Multiple impulse waves were detected leading to higher price targets, including expected prices around $140.89 by November 8, 2024.
4. **Correction Waves:**
    - **Wave 1:** June 20, 2024, at $140.76 to July 18, 2024, at$116.56
    - **Wave 2:** July 18, 2024, at $116.56 to August 26, 2024, at$131.26
    - **Wave 3:** August 26, 2024, at $131.26 to September 4, 2024, at$104.12

Figure 19. Response from Investment Advisor agent.

InvestmentAdvisor: ### Comprehensive Investment Strategy for NVIDIA (NVDA)

1. Elliott Wave Analysis Recap:

- **Detected Impulse Waves:**
    - **Initial Wave Structure:** The analysis has identified several impulse waves indicating a strong bullish trend.
        - First Impulse Wave: Notable movements from $45.01 to$56.83 from December 2023 to mid-January 2024.
        - Subsequent waves indicate continued upward momentum with a target price around $149.77 by November 2024.
- **Correction Waves:** Periods of price retracement have been identified, suggesting strategic entry points.

2. Investment Recommendations:

- **Buy Recommendation:**
    - **Entry Point:** Consider entering long positions at around $116.00, following any corrective wave stabilization.
    - **Target Price:** Aim for a price target of $149.77 by November 8, 2024.
- **Sell Recommendation:**
    - **Exit Point:** Consider taking profits if the price approaches $149.77 or shows signs of reversal.
- **Stop-Loss Level:**
    - Set stop-loss orders around $110.00 to mitigate potential losses, particularly in the event of unforeseen market corrections.

Figure 20. Response from Elliott Wave Analyst agent.

7. Real-World Applications and Challenges

The integration of the Elliott wave principle with AI-driven multi-agent systems has demonstrated promising results in stock market forecasting, as evidenced by the experiments conducted using the ElliottAgents platform. The ability to accurately recognize complex wave patterns and generate actionable insights has the potential to significantly impact various real-world applications, particularly in the domain of financial analysis and trading. However, the deployment of such systems in practice poses distinct challenges that warrant further investigation.

7.1. Adaptation in Dynamic Market Conditions

While ElliottAgents demonstrates reliability under standard market conditions, significant limitations emerge during periods of market dislocation or unprecedented external shocks [52].

This limitation became particularly evident during the COVID-19 pandemic-induced market turbulence of March 2020, where traditional price formation mechanisms experienced substantial disruption. During such periods of extreme market stress, several critical constraints emerge in the system's analytical framework.

First, the pattern recognition algorithms, while effective in standard market conditions, demonstrate reduced efficacy when confronted with unprecedented volatility regimes. The system's Elliott wave identification mechanisms, calibrated on historical market behavior, may fail to adapt sufficiently rapidly to fundamental shifts in market structure. This temporal lag in pattern adaptation can result in decreased prediction accuracy during the initial phases of market dislocations.

Second, the backtesting process requires substantial data accumulation within new market regimes before achieving optimal recalibration. This inherent latency in the learning process potentially compromises the system's predictive capabilities during the acute phase of market disruptions. While the continuous learning framework eventually adapts to novel market conditions, the intermediate period of adjustment may result in suboptimal pattern recognition and wave identification.

Third, the system's current implementation lacks integration of mechanisms for incorporating sudden macroeconomic shocks or global event-driven market dislocations. While the RAG component provides some capability for processing real-time market information, the translation of unprecedented external events into actionable pattern recognition parameters remains a significant challenge. In our future research, we want to resolve this limitation with integration of an agent that could analyze live macroeconomic factors.

The identified constraints underscore the inherent challenges in developing fully adaptive market analysis systems capable of maintaining performance across all market conditions. While ElliottAgents represents a significant advance in automated technical analysis, its current implementation remains bounded by these fundamental limitations in extreme market scenarios.

7.2. Limitations and Open Research Questions

Despite ElliottAgents demonstrated effectiveness in pattern recognition and market analysis, several limitations and areas for future investigation warrant consideration. The current implementation focuses on a subset of Elliott wave patterns, primarily addressing impulsive waves, corrective waves, and wave extensions. While this approach has proven effective, it represents only a portion of the thirteen distinct patterns identified in Elliott wave theory. This limitation presents an opportunity for expanding the system's pattern recognition capabilities to include more complex formations such as triangles, flats, and zigzags.

The reliance on historical data for DRL, while effective, raises questions about the system's adaptability to unprecedented market conditions. Although backtesting has demonstrated significant improvements in pattern recognition accuracy, the approach may be limited by the assumption that future market behavior will reflect historical patterns.

This limitation becomes particularly relevant during periods of market stress or structural changes in trading dynamics such as politcal events.

7.3. Future Enhancements

The current implementation of ElliottAgents, while demonstrating significant capabilities in pattern recognition and market analysis, presents several opportunities for future enhancement and expansion. Currently, we are focusing on fundamental Elliott wave patterns, and significant potential exists for incorporating advanced wave formations, including truncations, zigzags, flat corrections, and complex triangular patterns [4]. This expansion would enable more comprehensive market analysis, potentially leading to improved predictive accuracy in diverse market scenarios.

Technical analysis integration represents another significant development opportunity. Future iterations could incorporate additional methodologies [1] to complement the Elliott wave framework. The integration of moving averages, momentum indicators, and volume analysis could provide supplementary validation metrics for wave pattern identification. This multi-faceted analytical approach would improve signal reliability and enhance prediction accuracy through corroborating indicators.

A particularly transformative advancement opportunity involves the integration of Anthropic's Agentic Computer Interface (ACI) [53], which could enable autonomous trading capabilities. The ACI implementation would allow agents to directly execute trades on real markets without requiring human intervention in the transaction process. Through LLMs, agents could navigate trading platforms, submit orders, and manage positions autonomously while maintaining adherence to predefined risk parameters. This enhancement would represent a significant step toward fully automated trading systems, where agents not only analyze market conditions but also execute trades based on their findings. The system would maintain human oversight capabilities while eliminating the latency and potential errors associated with manual trade execution.

The modular nature of the multi-agent architecture presents opportunities for system expansion through the introduction of specialized agents. Future development could incorporate sentiment analysis capabilities for news and social media integration and portfolio optimization algorithms [54]. These proposed enhancements, particularly the integration of ACI technology for autonomous trading, would significantly expand the system's practical utility in real-world trading environments.

8. Conclusions

As financial market complexity intensifies, the integration of AI with traditional analysis models presents a promising approach to improve market prediction. In the introduction, we addressed the critical question: Can a multi-agent platform, combining the EWP, LLMs, and DRL, offer a solution that meets the requirements of dynamic and data-intensive financial markets? Our findings with ElliottAgents indicate that a structured, multi-agent framework that synthesizes classical financial theories with advanced AI provides an answer to this question, offering analytical capabilities for real-time market prediction and adaptable insights.

The integration of the EWP within the multi-agent framework demonstrated that traditional technical analysis methods could be enhanced when combined with AI-driven tools. The EWP fractal patterns in market behavior, supported by LLMs and DRL, allowed ElliottAgents to identify significant market trends accurately and to generate actionable insights based on past behaviors. LLMs enabled the system to contextualize and interpret complex patterns within a broader dataset, increasing its ability to process diverse market scenarios. Additionally, DRL contributed a layer of continuous improvement, allowing the system to learn from historical data and refine its analytical models, which proved crucial for the system's adaptability in rapidly changing market conditions.

The system's practical efficacy was demonstrated through validation across multiple market scenarios. In our extended testing period, ElliottAgents achieved notable performance metrics, including a 17.4% return on investment in long-term position trading for NVDA stock and a 13.3% gain in medium-term trading for GOOGL. These results were achieved through identification of wave patterns and strategic entry/exit points. Furthermore, the system demonstrated robust risk management capabilities by accurately identifying support and resistance levels, with stop-loss recommendations maintaining an average drawdown limitation of 5% across tested scenarios. While these results highlight the system's practical value, direct comparisons to baseline models are challenging due to ElliottAgents' unique combination of visual and numerical outputs, which enhance interpretability. Future research will focus on developing benchmarking frameworks that can evaluate such multi-faceted systems effectively.

ElliottAgents further leverages a multi-agent structure to address specific analytic tasks efficiently [33,37]. Each agent within the system handles a dedicated aspect of financial analysis—data retrieval, pattern recognition, report generation—coordinated to work toward a cohesive output. This modularity allowed agents to interact and learn iteratively, combining their specialized knowledge to enhance overall predictive accuracy. The system's adaptability stems from its ability to dynamically adjust agent tasks based on real-time analysis needs, while each agent's discrete role provides transparency and interpretability, essential for human analysts needing insight into each step of the decision-making process. Potential industrial applications of ElliottAgents include:

- Automated risk assessment through real-time pattern monitoring;
- Market trend change detection for strategic position adjustment;
- Trade execution timing optimization.

The practical implications of ElliottAgents underscore its potential for both individual and institutional investors. By consolidating diverse AI functionalities—such as RAG [12,55] for real-time data accuracy and ReAct [40] models for integrated reasoning and action—ElliottAgents offers a versatile, user-friendly tool capable of producing reliable and interpretable insights. Notably, this approach addresses the common limitations of conventional AI systems [28] that often lack interpretability and adaptability in complex, data-heavy domains like financial markets. ElliottAgents stands out by balancing AI's computational strength with structured interpretability, enhancing user confidence in the system's outputs.

As a direct answer to our initial question, the results affirm that ElliottAgents effectively combines traditional and AI-driven techniques within a multi-agent framework. By doing so, it achieves the desired goals of interpretability, adaptability, and accuracy. Moreover, the modular, scalable design of ElliottAgents means that future developments and additional functionalities, such as the inclusion of new analytical tools or other financial models, can be incorporated without overhauling the entire system. This design provides a roadmap for how AI can be incrementally integrated into classical analysis, expanding its potential for practical, scalable applications in finance.

This research has made several contributions to the field of AI-driven financial analysis. First, we demonstrated the successful integration of the EWP with modern AI technologies through the ElliottAgents platform, showing how traditional technical analysis methods can be enhanced by computational intelligence. Second, we developed an innovative pattern recognition system that leverages LLMs, RAG, and DRL to automatically identify and classify Elliott wave formations, significantly improving the accuracy and reliability of technical analysis in high-speed trading environments. Our experimental results validate these contributions, showing consistent improvement in pattern recognition accuracy across different market conditions and timeframes. As financial markets continue to evolve in complexity, this integration of established analytical frameworks with advanced AI capabilities provides a promising direction for future development of automated market analysis systems.

Author Contributions: Conceptualization, M.W., J.A.C. and E.N.-S.; Methodology, M.W., J.A.C. and E.N.-S.; Software, M.W.; Validation, M.W. and J.A.C.; Formal analysis, M.W. and J.A.C.; Investigation, M.W. and J.A.C.; Resources, M.W.; Data curation, M.W.; Writing—original draft, M.W. and J.A.C.; Writing—review & editing, M.W., J.A.C. and E.N.-S.; Visualization, M.W.; Supervision, J.A.C.; Project administration, E.N.-S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Murphy, J.J. *Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications*; New York Institute of Finance: New York, NY, USA, 1999.
2. Vantuch, T.; Zelinka, I.; Vasant, P. Market prices trend forecasting supported by elliott wave's theory. In Proceedings of the First EAI International Conference on Computer Science and Engineering, Penang, Malaysia, 11–12 November 2016.
3. Graham, B. *The Intelligent Investor: The Definitive Book on Value Investing*; Harper Collins Publ.: New York, NY, USA, 2006.
4. Frost, A.J.; Robert R.P., Jr.; Charles, J.C. *Elliott Wave Principle: Key to Market Behavior*; Wiley: Hoboken, NJ, USA, 2001.
5. TradingView. Available online: https://www.tradingview.com/ (accessed on 10 November 2024).
6. Cheng, Y.; Zhang, C.; Zhang, Z.; Meng, X.; Hong, S.; Li, W.; Wang, Z.; Wang, Z.; Yin, F.; Zhao, J.; et al. Exploring large language model based intelligent agents: Definitions, methods, and prospects. *arXiv* **2024**, arXiv:2401.03428.
7. Scalamogna, G. Ai Agent Capabilities Engineering. 2024. Available online: https://towardsdatascience.com/ai-agent-capabilities-engineering-34c7785f413e (accessed on 15 August 2024).
8. Zhao, P.; Jin, Z.; Cheng, N. An in-depth survey of large language model-based artificial intelligence agents. *arXiv* **2023**, arXiv:2309.14365v1.
9. Gamil, A.A.; Elfouly, R.S.; Darwish, N.M. Stock technical analysis using multi agent and fuzzy logic. In Proceedings of the World Congress on Engineering, WCE 2007, London, UK, 2–4 July 2007.
10. Luo, Y.; Liu, K.; Davis, D.N. A multi-agent decision support system for stock trading. *IEEE Netw.* **2002**, *16*, 20–27.
11. Asai, A.; Wu, Z.; Wang, Y.; Sil, A.; Hajishirzi, H. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv* **2023**, arXiv:2310.11511.
12. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.; Rocktäschel, T.; et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv* **2021**, arXiv:2005.11401v4.
13. Wang, L.; Zhang, X.; Su, H.; Zhu, J. A comprehensive survey of continual learning: Theory, method and application. *arXiv* **2024**, arXiv:2302.00487. [CrossRef]
14. Wickramasinghe, B.; Saha, G.; Roy, K. Continual learning: A review of techniques, challenges, and future directions. *IEEE Trans. Artif. Intell.* **2024**, *5*, 2526–2546. [CrossRef]
15. Suresh, A.S.; Naveen Prasath, S. A study to understand elliott wave principle. *Int. J. Eng. Res. Gen. Sci.* **2016**, *4*, 352–363.
16. Boroden, C. *Fibonacci Trading: How to Master the Time and Price Advantage*; McGraw Hill: New York, NY, USA, 2008.
17. van Ginneken, K. Profitability of Elliott Waves and Fibonacci Retracement Levels in the Foreign Exchange Market. Master's Thesis, Tilburg School of Economics and Management, Tilburg University, Tilburg, The Netherlands, 2013.
18. Naveed, H.; Khan, A.U.; Qiu, S.; Saqib, M.; Anwar, S.; Usman, M.; Akhtar, N.; Barnes, N.; Mian, A. A comprehensive overview of large language models. *arXiv* **2024**, arXiv:2307.06435.
19. Raiaan, M.A.K.; Mukta, M.S.H.; Fatema, K.; Fahad, N.M.; Sakib, S.; Mim, M.M.J.; Ahmad, J.; Ali, M.E.; Azam, S. A review on large language models: Architectures, applications, taxonomies, open issues and challenges. *IEEE Access* **2024**, *12*, 26839–26874. [CrossRef]
20. OpenAI. Gpt-4o system card. *arXiv* **2024**, arXiv:2410.21276.
21. Team, G.; Anil, R.; Borgeaud, S.; Alayrac, J.B.; Yu, J.; Soricut, R.; Schalkwyk, J.; Dai, A.M.; Hauth, A.; Millican, K.; et al. Gemini: A family of highly capable multimodal models. *arXiv* **2024**, arXiv:2312.11805.
22. Bouchard, L.-F.; Louie, P. *Building LLMs for Production: Enhancing LLM Abilities and Reliability with Prompting, Fine-Tuning, and RAG*; Towards AI: Wilmington, DE, USA, 2024.
23. Tan, M.; Merrill, M.A.; Gupta, V.; Althoff, T.; Hartvigsen, T. Are language models actually useful for time series forecasting? *arXiv* **2024**, arXiv:2406.16964v1.
24. Tsay, R.S. *Analysis of Financial Time Series*, 3rd ed.; Wiley: Hoboken, NJ, USA, 2010.
25. Tang, H.; Zhang, C.; Jin, M.; Yu, Q.; Wang, Z.; Jin, X.; Zhang, Y.; Du, M. Time series forecasting with llms: Understanding and enhancing model capabilities. *arXiv* **2024**, arXiv:2402.10835v2.

26. Szydlowski, K.L.; Chudziak, J.A. Transformer-style neural network in stock price forecasting. In Proceedings of the 21th International Conference on Modeling Decisions for Artificial Intelligence (MDAI 2024), Tokyo, Japan, 27–31 August 2024.
27. Cinkusz, K.; Chudziak, J.A. Towards llm-augmented multiagent systems for agile software engineering. In Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering (ASE 2024), Sacramento, CA, USA, 27 October–1 November 2024.
28. Chudziak, A. Predictability of stock returns using neural networks: Elusive in the long term. *Expert Syst. Appl.* **2023**, *213*, 119203. [CrossRef]
29. Amaratunga, T. *Understanding Large Language Models Learning Their Underlying Concepts and Technologies*; Apress: New York, NY, USA, 2023.
30. Rahali, A.; Akhloufi, M.A. End-to-end transformer-based models in textual-based nlp. *AI* **2023**, *4*, 54–110. [CrossRef]
31. Together AI. Building Your Own Rag Application Using Together AI and Langchain. 2024. Available online: https://www.together.ai/blog/rag-tutorial-langchain (accessed on 30 October 2024).
32. Guo, T.; Chen, X.; Wang, Y.; Chang, R.; Pei, S.; Chawla, N.V.; Wiest, O.; Zhang, X. Large language model based multi-agents: A survey of progress and challenges. *arXiv* **2024**, arXiv:2402.01680v2.
33. Li, J.; Zhang, Q.; Yu, Y.; Fu, Q.; Ye, D. More agents is all you need. *arXiv* **2024**, arXiv:2402.05120v1.
34. LangChain. Langgraph: Multi-Agent Workflows. Available online: https://blog.langchain.dev/langgraph-multi-agent-workflows/ (accessed on 30 October 2024).
35. Chen, H. Understand the Llm Agent Orchestration. 2024. Available online: https://medium.com/scisharp/understand-the-llm-agent-orchestration-043ebfaead1f (accessed on 1 June 2024).
36. Kapoor, S.; Stroebl, B.; Siegel, Z.S.; Nadgir, N.; Narayanan, A. Ai agents that matter. *arXiv* **2024**, arXiv:2407.01502v1.
37. Weng, L. Llm-Powered Autonomous Agents. 2023. Available online: https://lilianweng.github.io/ (accessed on 10 September 2024).
38. Kabbani, T.; Duman, E. Deep reinforcement learning approach for trading automation in the stock market. *IEEE Access* **2022**, *10*, 93564–93574. [CrossRef]
39. Chan, C.-M.; Chen, W.; Su, Y.; Yu, J.; Xue, W.; Zhang, S.; Fu, J.; Liu, Z. Chateval: Towards better llm-based evaluators through multi-agent debate. *arXiv* **2023**, arXiv:2308.07201.
40. Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; Cao, Y. React: Synergizing reasoning and acting in language models. *arXiv* **2023**, arXiv:2210.03629.
41. Shankar, A. Building React Agents from Scratch: A Hands-on Guide Using Gemini. 2024. Available online: https://medium.com/google-cloud/building-react-agents-from-scratch-a-hands-on-guide-using-gemini-ffe4621d90ae (accessed on 13 October 2024).
42. Lapan, M. *Deep Reinforcement Learning Hands-On*, 2nd ed.; Packt: Birmingham, UK, 2020.
43. Wei, Z.; Xu, J.; Lan, Y.; Guo, J.; Cheng, X. *Reinforcement Learning to Rank with Markov Decision Process*; Association for Computing Machinery: New York, NY, USA, 2017; pp. 945–948. [CrossRef]
44. Lussange, J.; Lazarevich, I.; Bourgeois-Gironde, S.; Palminteri, S.; Gutkin, B. Modelling stock markets by multi-agent reinforcement learning. *Comput. Econ.* **2020**, *57*, 113–147. [CrossRef]
45. Akintola, K.G.; Oyetunji, O.E. Development of an agent-based framework for stock market trading. *IRE J.* **2021**, *4*, 98–106.
46. Chudziak, J.A.; Wawer, M. Elliottagents: A natural language-driven multi-agent system for stock market analysis and prediction. In Proceedings of the 38th Pacific Asia Conference on Language, Information and Computation, Tokyo, Japan, 7–9 December 2024.
47. Tirea, M.; Tandau, I.; Negru, V. Stock market multi-agent recommendation system based on the elliott wave principle. In Proceedings of the International Conference on Availability, Reliability, and Security, Prague, Czech Republic, 20–24 August 2012.
48. Ewa, N.-S. Application of parallel price method to optimal portfolio selection. In Proceedings of the IEEE International Conference on Methods and Models in Automation and Robotics (MMAR2003), Międzyzdroje, Poland, 25–28 August 2003; Volume 1.
49. Szydlowski, K.L.; Chudziak, J.A. Toward predictive stock trading with hidformer integrated into reinforcement learning strategy. In Proceedings of the 36th International Conference on Tools for Artificial Intelligence (ICTAI 2024), Herndon, VA, USA, 30 October–1 November 2024.
50. Larson, J.; Truitt, S. Graphrag: Unlocking Llm Discovery on Narrative Private Data. 2024. Available online: https://www.microsoft.com/en-us/research/blog/graphrag-unlocking-llm-discovery-on-narrative-private-data (accessed on 10 May 2024).
51. Hu, B.; Zhao, C.; Zhang, P.; Zhou, Z.; Yang, Y.; Xu, Z.; Liu, B. Enabling intelligent interactions between an agent and an llm: A reinforcement learning approach. *arXiv* **2024**, arXiv:2306.03604v8.
52. Taleb, N.N. *The Black Swan: The Impact of the Highly Improbable*; Random House and Penguin: New York, NY, USA, 2007.
53. Greyling, C. Anthropic Aci (AI Agent Computer Interface). 2024. Available online: https://cobusgreyling.substack.com/p/anthropic-aci-ai-agent-computer-interface (accessed on 10 November 2024).
54. Li, Y.; Yu, Y.; Li, H.; Chen, Z.; Khashanah, K. Tradinggpt: Multi-agent system with layered memory and distinct characters for enhanced financial trading performance. *arXiv* **2023**, arXiv:2309.03736.
55. Gao, Y.; Xiong, Y.; Gao, X.; Jia, K.; Pan, J.; Bi, Y.; Dai, Y.; Sun, J.; Wang, H. Retrieval-augmented generation for large language models: A survey. *arXiv* **2024**, arXiv:2312.10997v5.