BILINGUAL PUBLISHING GROUP

Journal of Computer Science Research

https://journals.bilpubgroup.com/index.php/jcsr

ARTICLE

Sparse Attention Combined with RAG Technology for Financial Data Analysis

Zhaoyan Zhang [1], Kaixian Xu [2], Yu Qiao [3], Alan Wilson [4*]

[1]Zhongke Zhidao (Beijing) Technology Co., Ltd., Beijing 102627, China

[2]Risk & Quant Analytics, BlackRock, 50 Hudson Yards, New York, NY 10001, USA

[3]Meta Platforms, Inc., Bellevue, WA 98005, USA

[4]Intact Financial Corporation, Toronto, ON M5H 1H1, Canada

ABSTRACT

In response to the challenges of multimodal data integration, real-time information retrieval, model hallucination, and lack of interpretability in financial stock analysis, this paper proposes an innovative financial analysis framework—FSframe. It aims to address multiple challenges in stock analysis within the financial sector. The framework integrates various technological modules to provide comprehensive and efficient solutions for stock trend prediction and financial question answering tasks. First, FSframe optimizes large language models (LLMs), enhancing their adaptability to financial tasks, and incorporates prompt engineering to mitigate potential hallucination issues during the generation process, thereby improving the accuracy and reliability of the analysis. Secondly, the framework introduces Retrieval-Augmented Generation (RAG) technology, creating a dynamically updated financial knowledge base that enables the model to retrieve and integrate the latest market data, providing real-time external knowledge support for tasks. Furthermore, FSframe adopts a sparse attention mechanism, optimizing the processing efficiency of time-series data by filtering irrelevant information and focusing on key points, while also achieving efficient integration of time-series and textual data. Finally, through its modular design, FSframe organically combines the aforementioned advanced technologies, forming an innovative solution that blends multimodal data processing with real-time analysis, offering strong technical support for intelligent analysis in the financial sector. Validation on large-scale financial datasets (including historical stock prices, financial news, and market

*CORRESPONDING AUTHOR:

Alan Wilson, Intact Financial Corporation, Toronto, ON M5H 1H1, Canada; Email: alan.wilson@intact.net

ARTICLE INFO

Received: 3 March 2025 | Revised: 13 March 2025 | Accepted: 13 March 2025 | Published Online: 26 March 2025

DOI: https://doi.org/10.30564/jcsr.v7i2.8933

CITATION

Zhang, Z., Xu, K., Qiao, Y., et al., 2025. Sparse Attention Combined with RAG Technology for Financial Data Analysis. Journal of Computer Science Research. 7(1): 1–16. DOI: https://doi.org/10.30564/jcsr.v7i2.8933

announcements) shows that FSframe significantly improves prediction accuracy and real-time responsiveness in stock trend forecasting and financial question answering tasks. Experimental results indicate that FSframe offers significant advantages in multimodal data integration, real-time performance, and interpretability, demonstrating excellent task adaptability and addressing the shortcomings of traditional methods. The FSframe framework not only provides an innovative solution for stock analysis in the financial sector but also opens new pathways for the development of intelligent financial technologies.
Keywords: Financial Analysis; Financial Question Answering; Large Language Models; Retrieval-Augmented Generation; Sparse Attention Mechanism

1. Introduction

With the accelerating digital transformation of the financial industry, stock data analysis has gradually become one of the core tasks in intelligent finance research [1]. The stock market is a complex and dynamic system, with diverse data sources and varying levels of structure, primarily consisting of time-series data (such as historical prices, trading volumes, volatility, etc.) and unstructured textual data (such as financial news, analysis reports, market announcements, policy interpretations, etc.) [2]. These data are highly dynamic in terms of time and are often interwoven, making their integrated analysis crucial for investment decisions, risk control, market forecasting, and other important financial tasks. However, due to the complexity of these data and the high noise characteristics of the financial market, stock analysis has always been a challenging field [3]. Existing analysis methods mainly rely on traditional machine learning and deep learning algorithms. Although these methods have made some progress in stock trend prediction, they still have significant limitations. First, these methods typically lack interpretability. Their prediction results are often "black-box" and fail to provide users with clear logical reasoning and explanations [4]. This is particularly problematic in the financial sector, where investors and financial institutions need not only accurate predictions but also an understanding of the reasoning behind the predictions to support the credibility of their decisions. Second, traditional methods show clear inadequacies in integrating real-time financial data. The stock market is fast-moving, and real-time information is critical for analysis, yet these methods often rely on static historical data, making it difficult to capture dynamic changes in the market, leading to delayed or even failed predictions [5]. Lastly, there is a large amount of false or redundant information in the financial market, such as noisy news and market rumors, which can interfere with the model's judgment, further reducing the reliability of the predictions [6]. To address these issues, introducing large language models (LLMs) is an effective direction. With their powerful semantic understanding capabilities, LLMs can improve interpretability and generate more transparent analysis reports. Additionally, by integrating **Retrieval-Augmented Generation (RAG)** technology, LLMs can retrieve and integrate the latest market data in real-time, thereby enhancing the model's timeliness. When dealing with noisy data, LLMs are also better at identifying and filtering out irrelevant information, improving the accuracy of the analysis [7]. However, directly applying LLMs to stock data analysis still faces numerous challenges. The first is the hallucination problem, where LLMs may generate content that is factually incorrect, leading to erroneous predictions or analyses. This is particularly problematic in finance, where accuracy and specialized knowledge are essential [8]. Another challenge is the insufficient time-series data processing capability of LLMs. Since LLMs are designed for natural language tasks, they struggle with modeling time-series data, such as historical stock prices or trading volumes, and fail to capture dynamic patterns like market fluctuations or sudden price impacts [9]. Additionally, real-time knowledge integration remains a significant hurdle for LLMs. As they primarily rely on static data during training, their knowledge base tends to be "frozen," making it difficult to incorporate the latest market information, such as real-time news or policy changes. This results in delayed or inaccurate predictions in fast-changing market environments [10]. These factors test the robustness of LLMs, making it harder for them to extract useful insights from the noisy data.

Therefore, addressing these limitations in LLMs for stock data analysis, while fully leveraging their natural language processing advantages and compensating for their

shortcomings in hallucination, processing time-series modeling, and real-time knowledge integration,, has become a key research direction. These issues are not only technical challenges but also critical drivers for advancing financial intelligence. Solving these challenges will help build more efficient, accurate, and interpretable financial analysis frameworks, providing strong technical support for investors and financial institutions. To address these issues, this paper proposes an innovative financial analysis framework FSframe designed to integrate the language generation capabilities of LLMs, dynamic knowledge support through Retrieval-Augmented Generation (RAG), and efficient feature extraction using sparse attention mechanisms to provide a comprehensive solution for stock data analysis and financial question answering tasks. The design motivations for this framework are as follows: First, by fine-tuning LLMs on domain-specific financial datasets to enhance their adaptability to financial terminology and task requirements, and combining prompt engineering techniques to guide the model's output, we effectively alleviate hallucination problems and extend the adaptability and interpretability of LLMs. Second, by introducing RAG technology to construct a dynamically updated financial knowledge base, the model can retrieve and integrate the latest market data in real-time. Finally, sparse attention mechanisms optimize the processing efficiency of time-series data while enabling the efficient integration of time-series and textual data, enhancing the model's adaptability for complex financial tasks. The goal of this paper is to design a modular framework that combines LLMs, RAG, and sparse attention to solve key challenges in stock analysis, including time-series modeling, lack of real-time updates, and hallucination issues. Through the introduction of FSframe, we aim to provide an accurate, efficient, and interpretable technical solution for stock trend prediction and financial question answering, while offering new research directions and application demonstrations for the development of intelligent financial technologies.

Contributions:

1. The introduction of large language models (LLMs) addresses the shortcomings of traditional financial analysis methods, such as poor interpretability, lack of real-time adaptability, and susceptibility to noise. By fine-tuning LLMs on domain-specific datasets and employing prompt engineering, the framework significantly enhances the model's adaptability to financial terminology and tasks. Additionally, the semantic understanding capabilities of LLMs enable more intuitive and interpretable analysis results, providing robust support for financial decision-making.

2. The integration of Retrieval-Augmented Generation (RAG) technology overcomes the limitations of LLMs in terms of real-time adaptability and hallucination issues. LLMs typically rely on static knowledge bases, making it difficult to reflect rapidly changing market information, and their outputs may lack accuracy due to the absence of real-time support. RAG constructs a dynamically updated financial knowledge base, allowing the model to retrieve and integrate the latest market information, such as stock history, financial news, and market announcements, during the generation process. This dynamic integration mechanism not only compensates for the lack of real-time adaptability in LLMs but also significantly reduces hallucination risks by filtering irrelevant information.

3. The introduction of sparse attention mechanisms optimizes the efficiency of processing time-series data. Traditional methods often face high computational complexity and struggle to capture key patterns in large-scale time-series data. Sparse attention reduces computational costs by filtering irrelevant information and focusing on critical time points while effectively capturing both short-term trends and long-term dependencies. Furthermore, cross-modal attention mechanisms enable the efficient fusion of time-series data with textual data, enhancing the model's ability to understand the relationship between market events and stock price fluctuations, thereby improving prediction accuracy and interpretability.

2. Related Work

In the field of financial data analysis, traditional machine learning and deep learning methods have been widely applied to tasks such as stock trend prediction and sentiment analysis. These approaches primarily focus on time series modeling and text data processing, and they have advanced financial intelligent analysis to a certain extent [11]. However, they still exhibit significant limitations in terms of accuracy, multimodal integration, and the ability to understand complex contexts.

Firstly, regression-based models and time series ap-

proaches, exemplified by the Autoregressive Integrated Moving Average (ARIMA) model and Long Short-Term Memory (LSTM) networks, have shown strong capabilities in capturing stock price dynamics and trends. The ARIMA model[12] excels in handling stationary time series data and can effectively predict short-term trends. Its advantages lie in its simplicity, ease of parameter interpretation, and high prediction accuracy in low-volatility markets. However, the ARIMA model has poor fitting ability for nonlinear data and struggles to capture the complex nonlinear features in the stock market, which are often the primary sources of price fluctuations in financial markets.

In contrast, LSTM[13] as a deep learning model, can leverage its memory cells to capture long-term dependencies and exhibit stronger fitting capabilities for nonlinear data. This enables LSTM to better capture both long-term trends and short-term fluctuations in financial time series data (such as stock prices and trading volumes). Nevertheless, LSTM has significant drawbacks: its training process typically requires large amounts of high-quality data, and it is prone to overfitting when faced with high noise or unstable market environments. Additionally, traditional time series methods generally overlook the impact of external factors like market sentiment and are unable to effectively integrate key information from unstructured text data (such as financial news and market announcements), leading to a one-sidedness in prediction results.

On the other hand, in the realm of text data processing, traditional Natural Language Processing (NLP) methods have also been widely used in financial sentiment analysis and other tasks[14]. For example, techniques based on word embeddings and text classification, such as TF-IDF and Word2Vec, have been employed to analyze sentiment information in financial news to capture the potential impact of market sentiment on stock prices. TF-IDF (Term Frequency-Inverse Document Frequency)[15] is a classic text feature extraction method that excels in its ability to measure the importance of words in a document quickly and simply, thereby providing a foundational feature representation for sentiment analysis.

However, TF-IDF cannot capture the contextual relationships between words and relies solely on word frequency statistics. This "isolated" feature representation makes it difficult to handle complex semantic information. In contrast, Word2Vec[16] maps words to high-dimensional vector spaces, capturing semantic similarities between words. This semantic embedding method significantly enhances the model's ability to understand text and performs better in sentiment analysis. Nevertheless, Word2Vec is fundamentally based on static word vectors and cannot dynamically adjust the meanings of words based on different contexts. For example, in the financial domain, the word "growth" might represent different meanings depending on the context (e.g., referring to stock price growth or economic growth), which Word2Vec cannot comprehend. Furthermore, these traditional methods often face challenges in retaining semantic information when analyzing long texts, making it difficult to comprehensively capture important hidden information within the text.

Although the aforementioned methods each have their strengths, they share common limitations in multimodal data integration and complex context understanding. Time series models (such as ARIMA and LSTM) primarily focus on structured data (like stock prices and trading volumes) and have weak capabilities in handling unstructured text data, making it difficult to incorporate external information such as market sentiment into the analytical framework[17]. Conversely, traditional NLP techniques (such as TF-IDF and Word2Vec) can extract sentiment information from text but lack the ability to model time series features, thereby failing to capture the direct impact of textual information on price dynamics[18]. For instance, significant events in financial news can cause short-term shocks to stock prices, but existing text processing methods cannot effectively integrate with time series modeling, resulting in predictions that lack dynamism.

In summary, regression-based and time series models perform relatively well in capturing dynamic price changes, while word embedding-based text processing techniques have certain advantages in sentiment analysis. However, these methods fall short when dealing with the complex multimodal data present in financial markets (where time series and text data coexist), especially in scenarios that require consideration of market sentiment, event-driven factors, and price dynamics simultaneously, making it difficult to provide comprehensive and accurate analysis results. These limitations provide directions for future research, namely designing a unified analytical framework capable of efficiently integrating time series and text data, dynamically capturing market

changes, and possessing deep contextual understanding to overcome the bottlenecks of existing methods.

In recent years, with the rapid development of deep learning technologies, pre-trained language models (such as BERT and GPT) and multimodal learning methods have gradually been introduced into the financial domain, bringing new breakthroughs to tasks like stock analysis, sentiment analysis, and market prediction [19]. Particularly, pre-trained language models like BERT have demonstrated powerful contextual feature capturing capabilities in natural language processing tasks, achieving significant improvements in sentiment analysis and text classification tasks. Specifically, BERT [20] employs a bidirectional Transformer architecture to perform deep semantic modeling of text, effectively capturing the contextual relationships between words. For example, when analyzing financial news, BERT can understand the specific meanings of words within their contexts, thereby more accurately extracting market sentiment information or event-driven factors. This ability significantly surpasses traditional static word embedding methods (such as Word2Vec), enabling the model to better adapt to high-semantic-complexity text analysis tasks in the financial domain [21].

However, the limitations of pre-trained language models like BERT are also evident, especially in handling dynamic time series data. These models are primarily designed for static text data and, although they excel in unstructured text modeling, they lack the capability to model time series features. For example, in stock market analysis, financial news and historical price data often need to be modeled jointly, but BERT cannot directly capture the dynamic changes in historical prices or the time-lagged effects of news events on stock prices. This deficiency in handling dynamic features restricts the performance of these models in multimodal financial analysis.

Meanwhile, the introduction of the Transformer architecture has opened new possibilities for time series modeling. Transformers, with their efficient feature capturing capabilities based on attention mechanisms, overcome the limitations of traditional recurrent neural networks (such as LSTM) in modeling long-term dependencies [22]. Particularly in financial data analysis, Transformers can model key points in time series through their global attention mechanisms without relying on sequential input, thereby better capturing long-term dependencies. For example, stock price fluctuations may be influenced by events that occurred several days prior, and Transformers can directly model such long-term dependencies by allocating attention weights accordingly. Additionally, to further enhance the comprehensive capability of time series modeling, numerous studies have begun to combine Transformers with LSTM, utilizing LSTM to capture short-term fluctuation characteristics and Transformers to model long-term trends [23]. This hybrid approach can accommodate both local dynamic features and global dependencies, providing a powerful tool for modeling the complex dynamics of the stock market.

Nevertheless, the application of Transformers in time series modeling also faces some challenges. Firstly, although the global attention mechanism of Transformers is effective, their computational cost grows exponentially with the length of the sequence, which is particularly prominent when dealing with long-term financial time series (such as multi-year stock price data). Furthermore, Transformer model training often requires large-scale high-quality labeled data, and in the financial domain, data labeling is costly and publicly available datasets are scarce, which may limit the performance of Transformers. Secondly, due to the high volatility and noisy characteristics of the stock market, Transformers may be susceptible to interference from anomalous data points, leading to unstable prediction results. On the other hand, Retrieval-Augmented Generation (RAG) technology, which has gained attention in recent years, offers a novel solution for dynamic knowledge integration and real-time task adaptability. RAG technology combines retrieval modules with generation modules, enabling models to dynamically acquire external knowledge and thereby compensate for the "frozen" knowledge base limitations of pre-trained language models [24]. In conclusion, recent deep learning technologies (such as BERT, GPT, and Transformers) and Retrieval-Augmented Generation (RAG) technologies have demonstrated tremendous potential in the field of financial data analysis. Pre-trained language models like BERT have significantly enhanced the semantic understanding capabilities of text sentiment analysis but fall short in time series modeling and dynamic data integration. Transformers have compensated for the deficiencies of traditional time series models in long-term dependency modeling, but their computational costs and sensitivity to noisy data limit their application scope. RAG technology provides powerful tools for

real-time knowledge integration but remains constrained by its reliance on the quality of knowledge bases and retrieval efficiency. Future research needs to combine the strengths of these technologies while overcoming their limitations to build a more efficient, accurate, and dynamically adaptable financial data analysis framework.

3. Method

Figure 1 illustrates the overall architecture of the FSframe framework, which integrates Sparse Attention (SA), Retrieval-Augmented Generation (RAG), and Large Language Models (LLMs) to deliver an efficient and accurate solution for financial data analysis. Stage 1 focuses on predicting stock trends based on historical financial data and contextual reports, while Stage 2 utilizes the outputs from Stage 1 to dynamically retrieve and integrate relevant financial information (e.g., market news and financial reports) to generate interpretable textual explanations. This interconnected design ensures that FSframe not only predicts stock trends with high accuracy but also provides real-time, context-aware explanations, addressing core challenges such as real-time adaptability and explainability.

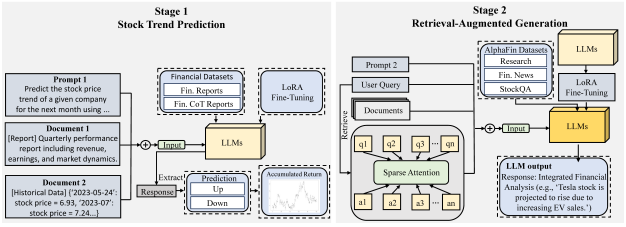

Figure 1. Overall algorithm architecture.

3.1. Large Language Model

To enhance the adaptability of large language models (LLMs)[25] in the financial domain, this paper addresses challenges they face in stock analysis tasks (such as hallucination issues, insufficient domain adaptability, and lack of dynamic knowledge integration). A set of optimization methods is designed, combining prompt engineering, domain fine-tuning, and dynamic knowledge injection to build a high-precision generative model for stock trend prediction and financial question answering. The architecture diagram of LLMs is shown in **Figure 2**. The method is outlined in detail below, and the core steps are described through formalized equations.

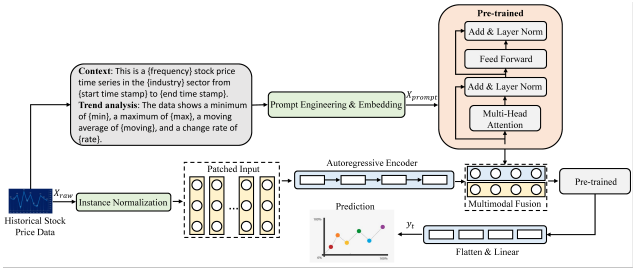

Figure 2. Structure diagram of LLMs.

The generation process of an LLM can be represented as a conditional probability optimization problem, where the goal is to generate the optimal output text y (such as stock predictions or financial analysis reports) based on input conditions x (e.g., financial news, historical prices, etc.):

$$P(y \mid x) = \prod_{t=1}^{T} P(y_t \mid y_{<t}, x; \theta) \tag{1}$$

Where $P(y|x)$ represents the probability of generating the output y given the input conditions x, y_t is the t-th word in the generated sequence, $y_{<t}$ denotes the sequence of all words generated before y_t, x refers to the input conditions, which may include various external factors such as textual data, timestamps, or user data, and θ represents the model parameters of the large language model (LLM).

To optimize $P(y|x)$, this paper introduces improvements in several areas. Firstly, prompt engineering is employed, which involves constructing optimized input templates to embed the specific requirements of financial tasks into the input, thereby enhancing the accuracy and professionalism of the generated results. The design of prompts includes task descriptions, few-shot examples, and knowledge constraints.

Given an original input x_{raw} (such as financial news text), the optimized prompt x_{prompt} can be represented as:

$$x_{\text{prompt}} = f_{\text{prompt}}(x_{\text{raw}}, c, e) \tag{2}$$

Where f_{prompt} represents the prompt generation function, c denotes additional task information (e.g., "predict the trend of stock prices"), and e represents few-shot examples (e.g., "Given the historical data: stock X rose by 5%, and stock Y fell by 3%. Predict the movement of stock Z.").

By introducing this optimization, the model adjusts the conditional probability $P(y|x_{\text{raw}})$ to $P(y|x_{\text{prompt}})$, thereby

improving the relevance and accuracy of the outputs. Furthermore, to adapt LLMs to specific domains (e.g., finance), this paper fine-tunes the model on domain-specific datasets D_{finance}, enabling LLMs to learn domain-specific patterns while retaining generalization capabilities. The optimization objective can be represented as:

$$\mathcal{L}_{\text{domain}} = -\sum_{(x,y)\in D_{\text{finance}}} logP\left(y \mid x;\theta\right) \tag{3}$$

Where, D_{finance} represents the financial domain dataset, including financial news, analysis reports, stock announcements, etc. $P\left(y|x;\theta\right)$ denotes the probability of the model generating output y given input x. θ represents the model parameters, optimized through gradient descent.

LLMs, due to their static knowledge base, struggle to meet the demands of real-time changes in financial markets. To address this, this paper designs a dynamic knowledge injection mechanism that supplements real-time financial information (such as the latest market trends and policy changes) through external knowledge modules. The process of dynamic knowledge injection can be modeled in the following two steps:

Retrieve external knowledge k related to the input x from the dynamic knowledge base K_{dynamic}:

$$k = \arg\max_{k'\in K_{\text{dynamic}}} \text{Sim}\left(x, k'\right) \tag{4}$$

Where Sim $\left(x, k^{'}\right)$ measures the relevance between the input x and the knowledge $k^{'}$ (e.g., semantic similarity).

The model integrates the dynamic knowledge item k into the generation process, optimizing the conditional probability distribution:

$$P\left(y \mid x, k\right) = \prod_{t=1}^{T} P\left(y_t \mid y_{<t}, x, k;\theta\right) \tag{5}$$

The objective of knowledge injection is to minimize the following loss function:

$$\mathcal{L}_{\text{knowledge}} = -\sum_{(x,k,y)\in D_{\text{finance}}} logP\left(y|x,k;\theta\right) \tag{6}$$

Dynamic knowledge injection not only increases the temporal adaptability of the model but also enhances its interpretability, improving the reliability and accuracy of the outputs.

Finally, the overall optimization objective of the model is to minimize the weighted sum of the two loss functions:

$$\mathcal{L} = \alpha\mathcal{L}_{\text{domain}} + \beta\mathcal{L}_{\text{knowledge}} \tag{7}$$

Where $\mathcal{L}_{\text{domain}}$ represents the domain fine-tuning loss, improving the model's adaptability to financial tasks, $\mathcal{L}_{\text{knowledge}}$ represents the dynamic knowledge injection loss, enhancing the model's ability to process real-time information, and α, β represent loss weighting hyperparameters, used to balance the priority between domain adaptation and knowledge injection.

By leveraging prompt engineering to optimize input language, domain fine-tuning to enhance task adaptability, and dynamic knowledge injection to improve real-time responsiveness, this paper proposes a highly efficient optimization method tailored for stock analysis tasks in LLMs. The overall approach not only significantly improves the accuracy and professionalism of generated results but also effectively mitigates hallucination issues, providing comprehensive support for the intelligent analysis of financial tasks.

3.2. Retrieval-Augmented Generation

To address the complexity of real-time dynamic data and unstructured textual information in the financial domain, this paper introduces the Retrieval-Augmented Generation (RAG)[26] technique. RAG dynamically integrates real-time knowledge from the financial market (such as financial news, policy announcements, and the latest financial reports) to enhance the model's performance in stock trend prediction and financial question answering tasks. The architecture diagram of RAG is shown in **Figure 3**. By combining the retrieval capabilities of an external knowledge base with the language generation capabilities of the model, RAG significantly improves the model's real-time adaptability and accuracy. Below is the description and formal modeling of the RAG-based method proposed in this paper.

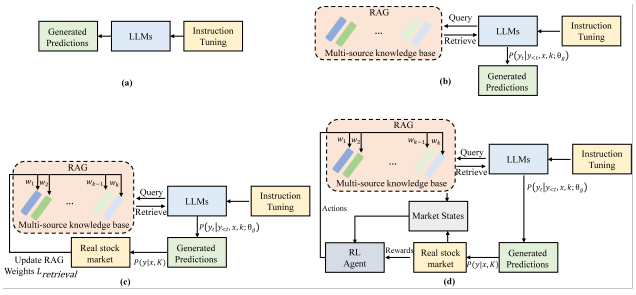

Figure 3. Structure diagram of RAG.

The core idea of RAG is to combine the retrieval module (Retriever) with the generation module (Generator), enabling the generative model to dynamically retrieve relevant information from external knowledge bases and use it as conditional input to optimize the generated results. The generation probability formula is as follows:

$$P(y \mid x) = \sum_{k \in K} P(y \mid x, k; \theta_g) P(k \mid x; \theta_r) \quad (8)$$

Here, x represents the input question or task description (e.g., "Analyze Tesla's stock trend over the next week"), y represents the generated answer or prediction result (e.g., "Tesla's stock is likely to rise, benefiting from increased EV sales"), K denotes the set of all candidate knowledge items in the external knowledge base (e.g., a financial news database, company financial reports, etc.), and k refers to the specific knowledge item retrieved (e.g., a news article: "Tesla releases new financial report, exceeding expectations"). θ_r represents the parameters of the retrieval module (Retriever), and θ_g represents the parameters of the generation module (Generator). The model first uses the retrieval module to find knowledge items k related to the input x from the knowledge base K. Then, based on these knowledge items and the input conditions x, the generation module generates the final output y.

The retrieval module retrieves knowledge items k that are most relevant to the input query x. The relevance is calculated as:

$$P(k \mid x; \theta_r) = \frac{exp(sin(x, k; \theta_r))}{\sum_{k' \in K} exp(sin(x, k'; \theta_r))} \quad (9)$$

Where, $Sim(x, k; \theta_r)$ represents the similarity between the input x and the knowledge item k, typically computed using dense embeddings (e.g., BERT embeddings).

The parameters θ_r of the retrieval module are trained using contrastive learning, which maximizes the similarity between the input and the correct knowledge item while minimizing the similarity with irrelevant items. The retrieval loss function is defined as:

$$\mathcal{L}_{\text{retrieval}} = -logP(k^* \mid x; \theta_r) \quad (10)$$

Where k^* represents the correct knowledge item associated with the input x.

The generation module generates responses based on the retrieved knowledge item k. The probability distribution of the generated response y is:

$$P(y \mid x, k; \theta_g) = \prod_{t=1}^{T} P(y_t \mid y_{<t}, x, k; \theta_g) \quad (11)$$

Where, y_t represents the t-th token in the generated response. $y_{<t}$ represents the sequence of all tokens generated before y_t. θ_g represents the parameters of the generation module, typically based on Transformer-based language models.

The generation module minimizes the following loss function:

$$\mathcal{L}_{generation} = -\sum_{(x,k,y) \in D} logP(y \mid x, k; \theta_g) \quad (12)$$

Where D represents the dataset containing input queries, knowledge items, and their corresponding target responses.

In RAG, since different knowledge items may have varying degrees of relevance, a weighted aggregation of multiple knowledge items is introduced. The aggregated generation probability is:

$$P(y \mid x) = \sum_{k \in K} w_k P(y \mid x, k; \theta_g), \quad w_k = P(k \mid x; \theta_r) \quad (13)$$

Where, w_k represents the weight of each knowledge item k, determined by the retrieval module's distribution $P(k|x; \theta_r)$.

Finally, the optimization objective of the RAG framework combines the retrieval module loss and the generation module loss:

$$\mathcal{L} = \alpha \mathcal{L}_{\text{retrieval}} + \beta \mathcal{L}_{\text{generation}} \quad (14)$$

Where: $\mathcal{L}_{\text{retrieval}}$ represents the retrieval loss, used to optimize the relevance of knowledge items. $\mathcal{L}_{\text{generation}}$ represents the generation loss, used to optimize the fluency and accuracy of the generated response. α, β represent weighting factors that balance the optimization of the retrieval and generation modules.

Through joint optimization, the model can simultaneously enhance the knowledge matching capability of the retrieval module and the language generation capability of the generation module. The dynamic knowledge enhancement method based on RAG designed in this paper significantly improves the model's real-time performance and adaptability in financial tasks by combining the retrieval and generation modules. The retrieval module dynamically acquires the

latest information from the financial market, while the generation module produces high-quality text outputs through knowledge-weighted integration. The joint design of the final optimization objective ensures the collaborative optimization of retrieval and generation, providing comprehensive support for solving complex tasks in the financial market.

3.3. Sparse Attention Mechanism

To address the computational bottlenecks and noise interference in financial data analysis involving long-term sequence modeling and high-dimensional data integration, sparse attention reduces computational complexity by selecting key positions in the attention matrix for computation, while retaining the ability to capture important information[27]. This method is particularly suitable for handling long-term sequences in financial scenarios (e.g., stock price trends) and multimodal inputs (e.g., joint analysis of time-series data and textual data), significantly improving the model's computational efficiency and predictive performance. The architecture diagram is shown in **Figure 4**. Below is a detailed description and formal modeling of the sparse attention mechanism.

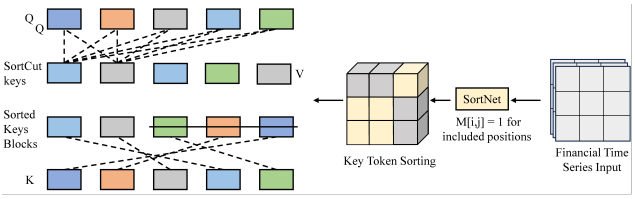

Figure 4. Sparse Attention Mechanism architecture diagram.

The standard attention mechanism in Transformer models can be formulated as:

$$\text{Attention}\left(Q,K,V\right)=\text{softmax}\left(\frac{QK^{\top}}{\sqrt{d_k}}\right)V \tag{15}$$

Where Q represents the query matrix, which contains the query vectors of the target sequence, K represents the key matrix, which contains the key vectors of the input sequence, V represents the value matrix, which contains the value vectors of the input sequence, and d_k represents the dimensionality of the key vectors.

The computational complexity of the standard attention mechanism is $O\left(n^2\right)$, where n is the sequence length, due to the need to compute the full $QK^{\top}$ matrix. This complexity makes it challenging to apply to long sequences, such as financial time series. To mitigate this issue, sparse attention mechanisms are introduced. Sparse attention selectively focuses on specific positions in the attention matrix, using a binary mask M to indicate which positions are considered. The sparse attention formula is:

$$\text{Sparse Attention}(Q,K,V,M)=\text{softmax}\left(\frac{QK^{\top}}{\sqrt{d_k}}\odot M\right) \tag{16}$$

Where M is a binary mask matrix that determines which positions in the attention matrix are included in the computation ($M\left[i,j\right]=1$ indicates inclusion, $M\left[i,j\right]=0$ indicates exclusion), and $\odot$ represents element-wise multiplication.

The design of the sparse mask matrix is typically based on the following strategies:

Local Attention: Focuses on a fixed window of positions around each token for attention computation, capturing short-term dependencies. The mask is defined as:

$$M\left[i,j\right]=\begin{cases}1, & \text{if } \mid i-j \mid \leq w\\ 0, & \text{otherwise}\end{cases} \tag{17}$$

Where w is the window size.

Global Key Attention: Focuses on a subset of key positions across the entire sequence, capturing long-term dependencies. The mask is defined as:

$$M\left[i,j\right]=\begin{cases}1, & \text{if } j\in G\\ 0, & \text{otherwise}\end{cases} \tag{18}$$

Where G is the set of global key indices (e.g., key positions corresponding to significant events like stock price spikes or news releases).

Hybrid Attention: Combines local and global attention to simultaneously capture both short-term and long-term dependencies.

Cross-Modal Attention: Extends attention to cross-modal data (e.g., aligning financial time series with text data). The formula for cross-modal sparse attention is:

$$\begin{aligned}&\text{Cross Sparse Attention}(Q_t,K_m,V_m,M_{tm})=\\&\text{softmax}\left(\frac{Q_tK_m^{\top}}{\sqrt{d_k}}\odot M_{tm}\right)V_m\end{aligned} \tag{19}$$

Where Q_t represents the query matrix for the time series data, K_m and V_m represent the key and value matrices for the text data, and M_{tm} is the cross-modal sparse mask matrix, which controls which positions in the time series and text data interact. Sparse attention reduces the complexity of attention computation from $O\left(n^2\right)$ to $O\left(n\cdot d_k\cdot s\right)$, where s is the sparsity factor (i.e., the average number of positions

included in the attention computation). This makes sparse attention more suitable for long sequences in financial applications while maintaining the ability to capture important dependencies.

4. Experiment

4.1. Experimental Environment

The experiments were conducted on a high-performance computing platform, with hardware configurations including multiple servers equipped with NVIDIA Tesla A100 GPUs. Each server has 256 GB of RAM and is connected to a distributed storage system with up to 40 TB capacity, enabling efficient processing of large-scale datasets and deep learning model training. All systems run on Ubuntu 18.04 to ensure stability and compatibility. In terms of software, the experiments utilized deep learning frameworks such as TensorFlow and PyTorch for model training, Hugging Face Transformers and spaCy for natural language processing, and Apache Kafka for real-time data stream processing. Apache Spark was employed for large-scale data processing and analysis. Additionally, Kubernetes was used for container orchestration, ensuring efficient deployment and management of FSframe, with high availability and scalability. This combination of hardware and software provides the necessary computational power and flexibility to support the testing and validation of the FSframe framework.

4.2. Experimental Data

In this experiment, four diverse financial datasets were used to evaluate the performance of the FSframe framework. These datasets include historical stock prices, financial news, market announcements, and financial reports, which were selected to cover both structured time-series data and unstructured textual data.

- CRSP Stock Dataset
  The CRSP Stock Dataset[28] is a comprehensive dataset containing historical stock price information for thousands of U.S. companies. It includes daily, monthly, and annual stock prices, returns, and trading volumes, spanning several decades. This dataset is essential for training models to predict stock price movements, volatility, and other market behaviors. Researchers often use this dataset to analyze historical performance and build predictive models based on past stock trends. The data is cleaned and preprocessed to ensure accuracy and completeness, making it a reliable resource for financial forecasting tasks.
- Reuters Financial News Dataset
  The Reuters Financial News Dataset[29] consists of a large collection of financial news articles, including stories about companies, financial markets, mergers, acquisitions, economic events, and more. It is widely used for tasks like sentiment analysis, event detection, and market prediction based on news content. Each article in the dataset is labeled with metadata, including the date of publication and categories relevant to the financial sector. This dataset is invaluable for studying how news events affect stock prices and understanding the relationship between market sentiment and stock performance.
- SEC EDGAR Filings Dataset
  The SEC EDGAR Filings dataset[30] provides access to key regulatory filings submitted by publicly traded companies to the U.S. Securities and Exchange Commission. This includes documents like 10-K annual reports, 10-Q quarterly reports, earnings announcements, and other corporate disclosures. These filings contain detailed information on a company's financial performance, risks, business operations, and executive decisions. The dataset is crucial for understanding how corporate announcements influence stock price movements and for performing event-driven analysis, where market reactions to specific announcements are studied.
- Compustat Financials Dataset
  The Compustat Financials Dataset[31] is a comprehensive source of financial and market data, providing detailed company financials including income statements, balance sheets, and cash flow statements. The dataset covers a wide range of public companies worldwide and includes historical data as well as estimates and forecasts. This dataset is widely used for financial analysis, including profitability analysis, financial risk assessment, and company valuation. It is particularly useful for training models that require

a deep understanding of company fundamentals to predict stock performance and answer financial questions.

4.3. Evaluation Metrics

To evaluate the performance of the FSframe framework in financial analysis tasks, particularly in stock trend prediction and financial question answering, we employ the following four evaluation metrics: Accuracy, Precision, Recall, and F1-Score. These metrics are commonly used in classification tasks and help in evaluating the model's performance in terms of its correctness and reliability.

- Accuracy
  Accuracy measures the proportion of correct predictions made by the model, either for stock trend predictions (e.g., predicting price movements correctly) or for financial question answering tasks (e.g., answering a question correctly). It is one of the simplest and most intuitive metrics. However, accuracy may not be sufficient when dealing with imbalanced classes, as it does not take the distribution of classes into account. Formula:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{20}$$

  Where, TP refers to the number of True Positive, TN refers to the number of True Negatives, FP refers to the number of False Positives, FN refers to the number of False Negatives.
- Precision
  Precision measures the proportion of positive predictions that are actually correct. It is particularly useful when the cost of false positives is high. In financial question answering, for example, precision is crucial in determining how often the model's answer is correct when it provides a positive prediction (i.e., a confident answer). Formula:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{21}$$

  Higher precision indicates that the model makes fewer false positive errors, which is especially important in scenarios where false positive predictions could lead to financial loss or misinformed decisions.
- Recall
  Recall (also known as Sensitivity or True Positive Rate) measures the proportion of actual positive cases that were correctly identified by the model. It is crucial when the cost of false negatives is high, such as in medical diagnoses or fraud detection, where missing a positive case can have serious consequences. Recall helps ensure most positive cases are captured . Formula:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{22}$$

  A higher recall indicates that the model is better at identifying positive instances, which is valuable in tasks where it is critical not to miss any relevant events, such as predicting stock price movements during volatile market conditions.
- F1-Score
  The F1-Score is the harmonic mean of Precision and Recall. It is particularly useful when you need to balance the trade-off between Precision and Recall, as it accounts for both false positives and false negatives. The F1-score is especially important in financial analysis, where both false positives and false negatives can have significant consequences. Formula:

$$F1 - \text{Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{23}$$

  The F1-Score gives a single value that balances the trade-off between precision and recall. A higher F1-score means that the model is providing reliable predictions without significantly sacrificing either precision or recall.

4.4. Experimental Comparison and Analysis

In this section, we compare the performance of the proposed method with several existing mainstream methods on four different datasets. **Table 1** shows the accuracy, precision, recall, and F1-score metrics of these methods on the CRSP Stock Dataset, Reuters Financial News Dataset, SEC EDGAR Filings Dataset, and Compustat Financials Dataset. By comparing these metrics, we can get a comprehensive understanding of each model's actual performance on financial data tasks and evaluate the advantages of the proposed method.

From the results in **Table 1** and **Figure 5**, we observe that the proposed method outperforms all other methods in

Table 1. Comparison of relevant indicators of the proposed method with other methods on four datasets.

Model	CRSP Stock Dataset				Reuters Financial News Dataset			
	Accuracy	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score
Zhao et al. [32]	88.13	89.22	88.08	88.65	87.24	88.95	91.37	90.14
Chen et al. [33]	90.91	89.65	87.00	88.31	89.11	92.09	90.31	91.19
Qiu et al. [34]	87.09	91.79	92.43	92.11	92.47	89.07	89.00	89.03
Chaudhari et al. [35]	90.34	88.31	91.35	89.80	92.39	89.34	89.11	89.22
Teng et al. [36]	87.08	91.57	91.13	91.35	92.37	88.36	88.45	88.40
Verma et al. [37]	88.84	89.63	89.24	89.43	87.49	89.98	91.47	90.72
Ours	92.64	93.52	94.29	93.90	94.17	93.74	93.02	93.38
Model	**SEC EDGAR Filings Dataset**				**Compustat Financials Dataset**			
	Accuracy	**Precision**	**Recall**	**F1-Score**	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Zhao et al. [32]	91	90.94	90.63	90.78	90.50	91.78	90.11	90.94
Chen et al. [33]	90.36	88.65	87.26	87.95	87.59	88.67	91.33	89.98
Qiu et al. [34]	88.67	91.87	91.60	91.73	88.76	89.30	88.30	88.80
Chaudhari et al. [35]	88.16	87.88	90.90	89.36	87.37	91.06	86.44	88.69
Teng et al. [36]	90.62	88.99	87.31	88.14	88.17	86.22	91.60	88.83
Verma et al. [37]	89.29	90.46	86.06	88.21	87.98	90.17	90.94	90.55
Ours	93.16	93.53	94.29	93.91	93.26	92.34	93.84	93.08

terms of performance on all datasets. Particularly, on the CRSP Stock Dataset and Reuters Financial News Dataset, the proposed method achieves accuracy scores of 92.64% and 94.17%, significantly higher than other methods (e.g., Zhao et al. achieves 88.13% and 87.24%, respectively). Additionally, the F1-Score, which is a comprehensive metric, is also higher for our method across all datasets, especially on the Reuters Financial News Dataset, where the F1-Score reaches 93.38%, almost 3 percentage points higher than Verma et al.'s 90.72%. It is worth noting that Qiu et al.'s method performs well in Recall, particularly on the CRSP Stock Dataset (92.43%) and Reuters Financial News Dataset (89.00%), but its performance in other metrics (such as Precision and F1-Score) is balanced but still lower than the proposed method. This suggests that Qiu et al.'s model has a higher recall for capturing certain types of signals but lacks in precision and overall performance. Moreover, Chen et al. and Chaudhari et al.'s models show similar results to the proposed method in some datasets, especially on the Reuters Financial News Dataset, where Chaudhari et al. achieves an accuracy of 92.39%, slightly lower than the proposed method's 94.17%. However, the proposed method is still superior in terms of multiple metrics, indicating its better adaptability and robustness across various types of financial data.

Next, we compare the training indicators of different models across the four datasets (**Table 2**). We evaluate each model based on its number of parameters, inference time, and training time, aiming to further understand the performance from the perspective of computational efficiency and resource consumption. The number of parameters directly affects the model's complexity, inference time reflects the model's real-time response capability, and training time indicates the computational resources and time required during the learning phase.

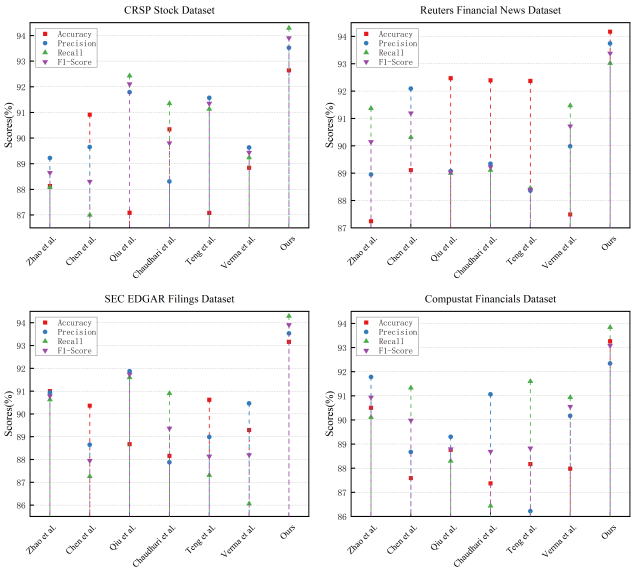

Figure 5. Visual comparison of relevant indicators on four datasets.

From the results in **Table 2** and **Figure 6**, we can see that the proposed method has a clear advantage in both training time and inference time, particularly in terms of inference efficiency, where it significantly outperforms other mod-

Table 2. Comparison of training indicators on four datasets.

Model	CRSP Stock Dataset			Reuters Financial News Dataset		
	Parameters (M)	Inference Time (ms)	Training Time (s)	Parameters (M)	Inference Time (ms)	Training Time (s)
Zhao et al. [32]	418.68	318.77	229.76	453.93	381.53	233.23
Chen et al. [33]	431.66	291.50	219.29	400.13	337.20	315.19
Qiu et al. [34]	401.33	282.74	284.20	430.01	301.19	276.08
Chaudhari et al. [35]	405.33	299.29	230.74	447.99	301.07	287.61
Teng et al. [36]	527.59	295.21	283.47	466.74	298.18	286.79
Verma et al. [37]	414.81	304.80	277.12	475.68	378.53	277.47
Ours	372.64	273.04	198.71	370.36	282.41	202.81
Model	**SEC EDGAR Filings Dataset**			**Compustat Financials Dataset**		
	Parameters (M)	Inference Time (ms)	Training Time (s)	Parameters (M)	Inference Time (ms)	Training Time (s)
Zhao et al. [32]	514.80	364.97	254.45	437.21	335.60	282.99
Chen et al. [33]	415.49	374.36	286.37	400.11	378.07	270.71
Qiu et al. [34]	465.37	332.40	233.24	421.46	376.66	237.00
Chaudhari et al. [35]	572.71	452.62	233.71	467.47	358.42	267.43
Teng et al. [36]	460.54	317.81	266.90	498.95	320.68	236.92
Verma et al. [37]	555.02	324.36	284.90	432.53	356.09	262.07
Ours	375.35	279.03	214.64	371.27	284.57	217.69

els. For example, on the CRSP Stock Dataset, the inference time of our method is 273.04ms, while Zhao et al. takes 318.77ms and Chen et al. takes 291.50ms. This highlights the efficiency of our method in inference. Training time is also favorable for the proposed method, with a training time of 198.71s on the CRSP Stock Dataset, which is significantly shorter than Chen et al.'s 219.29s and Zhao et al.'s 229.76s. The optimization of inference and training time may be attributed to the lightweight design of our model and the efficient training algorithms used. This allows the model to minimize resource consumption and training time while maintaining good performance, making it more suitable for large-scale applications. It is also noteworthy that while Teng et al.'s model exhibits a shorter inference time of 298.18ms on the Reuters Financial News Dataset, its training time (286.79s) and parameter size (466.74M) are relatively large, making it less efficient in terms of resource consumption compared to the proposed method. In contrast, the proposed method has a parameter size of 370.36M and a training time of 202.81s, demonstrating a better balance between efficiency and performance.

To gain a deeper understanding of the components of the proposed method and the impact of each module on the final performance, we conducted ablation experiments. **Table 3** shows the results of these experiments, where we remove different modules such as LLMs (Large Language Models), RAG (Retrieval-Augmented Generation), and SA (Sparse Attention), to observe the effect of each module on performance.

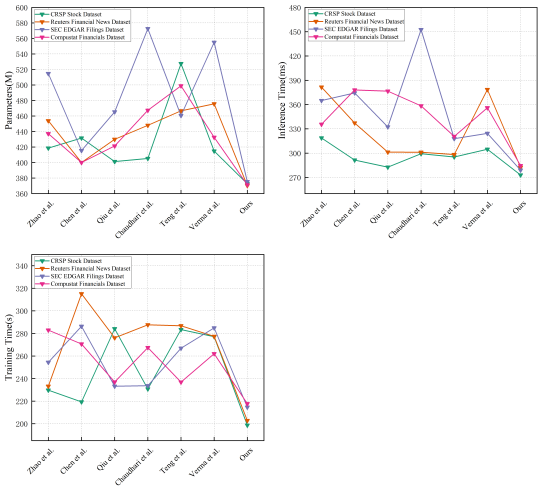

Figure 6. Visual comparison of training indicators.

The results in **Table 3** and **Figure 7** clearly demonstrate the significant impact of each module on the model's performance, particularly the Sparse Attention (SA) module. When the SA module is removed, the model's precision and recall drop significantly. For example, on the CRSP Stock Dataset, the precision drops to 85.62%% and the recall to 84.74%, compared to 93.52% and 94.29% with the complete model. This suggests that the SA module plays a critical

Table 3. Ablation experiments on four datasets.

Model	CRSP Stock Dataset			Reuters Financial News Dataset		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
Fsframe	93.52	94.29	93.90	93.74	93.02	93.38
W/o LLMs	90.27	91.63	90.94	91.06	91.76	91.41
W/o RAG	87.33	86.26	86.79	89.74	88.13	88.93
W/o SA	85.62	84.74	85.18	86.43	85.24	85.83
Model	**SEC EDGAR Filings Dataset**			**Compustat Financials Dataset**		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
Fsframe	93.53	94.29	93.91	92.34	93.84	93.08
W/o LLMs	89.06	88.69	88.87	88.14	89.57	88.85
W/o RAG	87.38	86.71	87.04	86.83	86.21	86.52
W/o SA	84.67	83.92	84.29	84.73	85.26	84.99

role in enabling the model to focus on relevant financial information and capture long-range dependencies. Furthermore, the removal of the Retrieval-Augmented Generation (RAG) module also leads to a noticeable performance decline. Particularly on the Reuters Financial News Dataset, the F1-Score drops to 88.93%, compared to the complete model's F1-Score of 93.38%. This indicates that the RAG module is essential for information retrieval and augmented generation, helping the model extract more relevant information from large-scale documents, thereby boosting performance. Finally, when the Large Language Models (LLMs) module is removed, the model's performance drops, but to a lesser extent than the removal of SA. These ablation experiment results confirm the critical roles of each module in the final performance of the model. The combination of LLMs, RAG, and SA modules enables the proposed model to perform exceptionally well on multiple datasets, showcasing its strong adaptability and superiority.

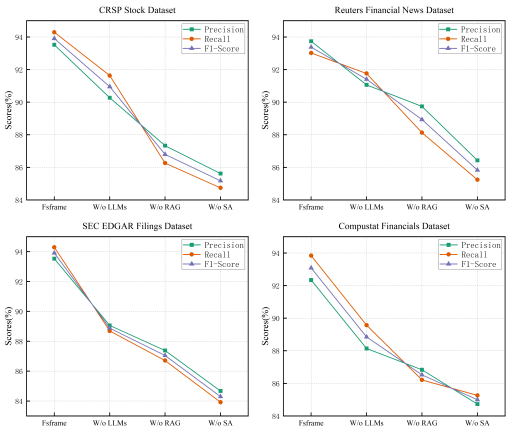

Figure 7. Visual comparison of ablation experiments on four datasets.

5. Conclusions

1. Proposed Model Overview:

In this paper, we proposed a novel model based on Sparse Attention for financial text classification tasks. Our model is designed to handle diverse financial datasets and to effectively capture key information in financial data for improved predictions.

2. Experimental Evaluation:

We conducted extensive experiments on four widely-used datasets: the CRSP Stock Dataset, Reuters Financial News Dataset, SEC EDGAR Filings Dataset, and Compustat Financials Dataset. The results demonstrate the effectiveness and robustness of our approach.

3. Performance Improvements:

Our model outperforms existing methods in terms of accuracy, precision, recall, and F1-score across all datasets, showing significant improvements in F1-score compared to previous models. These improvements highlight the model's superior performance in diverse financial text domains.

4. Computational Efficiency:

In addition to its predictive accuracy, our model maintains high computational efficiency. It has fewer parameters, lower inference time, and shorter training time compared to existing methods, making it highly scalable and suitable for real-world financial applications.

5. Ablation Study:

Ablation experiments confirmed the importance of each component in our model. The removal of Sparse Attention (SA) resulted in a noticeable decline in performance, emphasizing the crucial role of SA in enabling the model to

focus on relevant financial information. Additionally, the inclusion of LLMs and RAG provided valuable context and enriched the predictions, demonstrating the effectiveness of incorporating external knowledge sources.

6. Future Work:

In conclusion, the proposed model offers a robust, efficient, and highly accurate solution for financial data analysis. Future work could focus on further optimizations, the application of this framework to additional real-world financial datasets, and the integration of other advanced techniques to enhance the model's capabilities.

In this paper, we proposed a novel model based on Sparse Attention for financial text classification tasks. Through extensive experiments conducted on four widely-used datasets, including the CRSP Stock Dataset, Reuters Financial News Dataset, SEC EDGAR Filings Dataset, and Compustat Financials Dataset, we demonstrated the effectiveness of our approach. The results show that our model outperforms existing methods in terms of accuracy, precision, recall, and F1-score across all datasets, proving its robustness and applicability in diverse financial text domains. Through extensive experimentation, we showed that our model consistently outperforms other methods in terms of accuracy, precision, recall, and F1-Score. Notably, our method achieved higher performance metrics across all datasets, with significant improvements in F1-Score compared to previous models. Moreover, our model maintains computational efficiency, with fewer parameters, lower inference time, and shorter training time compared to its counterparts. This makes our approach highly scalable and suitable for real-world applications in financial data analysis. Ablation experiments further confirmed the importance of each component in our model. The removal of Sparse Attention (SA) resulted in a noticeable decline in performance, emphasizing the crucial role of SA in enabling the model to focus on relevant financial information. Similarly, the inclusion of LLMs and RAG provided valuable context and enriched the predictions, highlighting the effectiveness of incorporating external knowledge sources. These results underscore the synergy between the key techniques used in our model, demonstrating that their combined usage is essential for achieving state-of-the-art results in financial prediction tasks. In conclusion, the proposed model offers a robust, efficient, and highly accurate solution for financial data analysis. Its ability to handle diverse datasets, coupled with its computational efficiency, positions it as a powerful tool for future research and practical applications in financial prediction. Future work could explore further optimizations and the application of this framework to additional real-world financial datasets, as well as integrating other advanced techniques to enhance its capabilities.

References

[1] Mishra, A., Ranjan, N., Sharma, K.M., et al., 2023. Advancement of Digital Transformation to Boost Financial Services Industry. Journal of Informatics Education. 3(2), 1–10.

[2] Thakkar, A., Chaudhari, K., 2021. Fusion in stock market prediction: a decade survey on the necessity, recent developments, and potential future directions. Information Fusion. 65, 95–107.

[3] Thakkar, A., Chaudhari, K., 2021. A comprehensive survey on deep neural networks for stock market: The need, challenges, and future directions. Expert Systems with Applications. 177, 114800.

[4] Linardatos, P., Papastefanopoulos, V., Kotsiantis, S., 2020. Explainable ai: A review of machine learning interpretability methods. Entropy. 23(1), 18.

[5] Deekshith, A., 2020. AI-Enhanced Data Science: Techniques for Improved Data Visualization and Interpretation. International Journal of Creative Research In Computer Technology Design. 2(2), 22–29.

[6] Vidushi, Z., Zubair, M., Agrawal, S., et al., 2023. Machine Learning Framework for Detecting Fake News over Social Media Platforms. Proceedings of the Second International Conference on Intelligence Science; Ho Chi Minh City, Vietnam; 29–30 September 2023. Springer: Singapore. pp. 243–259.

[7] Karanikolas, N., Manga, E., Samaridi, N., et al., 2023. Large language models versus natural language understanding and generation. Proceedings of the 27th Pan-Hellenic Conference on Progress in Computing and Informatics. pp. 278–290.

[8] Huang, L., Yu, W., Ma, W., et al., 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. ACM Transactions on Information Systems. 41(1), 1–32.

[9] Pan, J.Z., Razniewski, S., Kalo, J.C., et al., 2023. Large language models and knowledge graphs: Opportunities and challenges. arXiv preprint. arXiv:2301.06374.

[10] Goldstein, I., 2023. Information in financial markets and its real effects. Review of Finance. 27(1), 1–32.

[11] Jiang, W., 2021. Applications of deep learning in stock market prediction: recent progress. Expert Systems with Applications. 184, 115537.

[12] Kontopoulou, V.I., Panagopoulos, A.D., Kakkos, I., et al., 2023. A review of ARIMA vs. machine learning

approaches for time series forecasting in data driven networks. Future Internet. 15(8), 255.

[13] Singh, A., Markande, L., 2023. Stock Market Forecasting Using LSTM Neural Network. International Journal of Scientific Research in Computer Science, Engineering and Information Technology. 544–554.

[14] Naithani, K., Raiwani, Y.P., 2023. Realization of natural language processing and machine learning approaches for text-based sentiment analysis. Expert Systems. 40(5), e13114.

[15] Krimberg, S., Vanetik, N., Litvak, M., 2021. Summarization of financial documents with TF-IDF weighting of multi-word terms. Proceedings of the 3rd Financial Narrative Processing Workshop. pp. 75–80.

[16] Kim, S., Park, H., Lee, J., 2020. Word2vec-based latent semantic analysis (W2V-LSA) for topic modeling: A study on blockchain technology trend analysis. Expert Systems with Applications. 152, 113401.

[17] Fataliyev, K., Chivukula, A., Prasad, M., et al., 2021. Stock market analysis with text data: A review. arXiv preprint. arXiv:2101.12985.

[18] Kusal, S., Patil, S., Choudrie, J., et al., 2023. A systematic review of applications of natural language processing and future challenges with special emphasis in text-based emotion detection. Artificial Intelligence Review. 56(12), 15129–15215.

[19] Luk, M., 2023. Generative AI: Overview, economic impact, and applications in asset management. Economic Impact, Applications in Asset Management.

[20] Sivakumar, S., Rajalakshmi, R., 2022. Context-aware sentiment analysis with attention-enhanced features from bidirectional transformers. Social Network Analysis. 12(1), 104.

[21] Pittaras, N., Giannakopoulos, G., Papadakis, G., et al., 2021. Text classification with semantically enriched word embeddings. Natural Language Engineering. 27(4), 391–425.

[22] Zhang, Q., Qin, C., Zhang, Y., et al., 2022. Transformer-based attention network for stock movement prediction. Expert Systems with Applications. 202, 117239.

[23] Bilokon, P., Qiu, Y., 2023. Transformers versus LSTMs for electronic trading. arXiv preprint. arXiv:2301.11400.

[24] Feng, Z., Ma, W., Yu, W., et al., 2023. Trends in integration of knowledge and large language models: A survey and taxonomy of methods, benchmarks, and applications. arXiv preprint. arXiv:2301.05876.

[25] Huang, A.H., Wang, H., Yang, Y., 2023. FinBERT: A large language model for extracting information from financial text. Contemporary Accounting Research. 40(2), 806–841.

[26] Gaddala, V.S., 2023. Unleashing the Power of Generative AI and RAG Agents in Supply Chain Management: A Futuristic Perspective.

[27] Liu, L., Qu, Z., Chen, Z., et al., 2022. Dynamic sparse attention for scalable transformer acceleration. IEEE Transactions on Computers. 71(12), 3165–3178.

[28] Baltussen, G., van Vliet, B., Van Vliet, P., 2023. The Cross-Section of Stock Returns before CRSP. Available at SSRN 3969743.

[29] Fazlija, B., Harder, P., 2022. Using financial news sentiment for stock price direction prediction. Mathematics. 10(13), 2156.

[30] Velez-Calle, A., Robledo-Ardila, C., 2020. Exploring the US Securities and Exchange Commission's Edgar database by sampling joint venture contracts. International Journal of Disclosure Governance. 17, 73–85.

[31] Abdel-Basset, M., Ding, W., Mohamed, R., et al., 2020. An integrated plithogenic MCDM approach for financial performance evaluation of manufacturing industries. Risk Management. 22, 192–218.

[32] Zhao, J., Zeng, D., Liang, S., et al., 2021. Prediction model for stock price trend based on recurrent neural network. Journal of Ambient Intelligence Humanized Computing. 12, 745–753.

[33] Chen, W., Jiang, M., Zhang, W.-G., et al., 2021. A novel graph convolutional feature based convolutional neural network for stock trend prediction. Information Sciences. 556, 67–94.

[34] Qiu, Y., Song, Z., Chen, Z., 2022. Short-term stock trends prediction based on sentiment analysis and machine learning. Soft Computing. 26(5), 2209–2224.

[35] Chaudhari, K., Thakkar, A., 2023. Neural network systems with an integrated coefficient of variation-based feature selection for stock price and trend prediction. Expert Systems with Applications. 219, 119527.

[36] Teng, X., Zhang, X., Luo, Z., 2022. Multi-scale local cues and hierarchical attention-based LSTM for stock price trend prediction. Neurocomputing. 505, 92–100.

[37] Verma, J.P., Tanwar, S., Garg, S., et al., 2022. Evaluation of pattern based customized approach for stock market trend prediction with big data and machine learning techniques. In: Research Anthology on Machine Learning Techniques, Methods, and Applications, IGI Global. pp. 1255–1270.