Large Language Models for Financial Risk Forecasting with Multimodal and Multi-Source Data: RiskLabs

YUPENG CAO[1], ZHI CHEN[1], QINGYUN PEI[1], and YANGYANG YU[1]
[1]Stevens Institute of Technology

ABSTRACT Artificial intelligence (AI), especially large language models (LLMs), is receiving growing attention in finance. Prior work has focused largely on financial text summarization, question answering, and stock movement prediction, while the use of LLMs for financial risk forecasting remains limited. We propose RiskLabs, a framework that uses LLMs to support financial risk analysis and prediction. RiskLabs integrates multimodal financial inputs, including earnings conference call transcripts and audio, market time-series signals, and background news. Empirical results show that RiskLabs is effective for forecasting market volatility and risk measures. Comparative experiments quantify the contribution of different data sources and clarify the role of LLMs in the forecasting pipeline. We also discuss the main challenges of LLM-based financial risk forecasting and the opportunities created by multimodal integration.

INDEX TERMS large language models; financial risk forecasting; multimodal learning; volatility prediction; value at risk

I. INTRODUCTION

Financial risk forecasting with AI has long been a central topic in both academia and industry. Earlier studies applied machine learning to historical price data for risk-related prediction tasks, such as using support vector machines (SVMs) to predict credit ratings from financial statement ratios [10, 13] and tree-based methods to explore the relationship between macroeconomic indicators and asset performance [9]. More recently, researchers have increasingly extracted trading signals from publicly available information. Financial reports and media news have been used for sentiment analysis, market expectation modeling, risk-factor analysis, and early warning detection [1, 14, 16, 21, 23, 29]. With the rise of multimodal learning, unstructured multimedia signals such as earnings conference call recordings have also been incorporated into stock volatility prediction [17, 27].

Although supervised learning methods have proved effective, they are typically task-specific and lack adaptability [19]. Their predictive performance is also constrained by the amount of input data and the scale of model parameters they can handle. The emergence of LLMs represents a potential shift in this landscape. With broad world knowledge and strong zero-shot capabilities, LLMs can perform a wide range of text-related tasks without dedicated task-specific training, including summarization [32], question answering [24], and sentiment analysis [30]. In finance, LLMs have already been applied to financial text analysis [26], financial report generation [25], and agent-based stock trading systems [28, 31, 33]. These studies demonstrate the versatility of LLMs and their growing relevance to financial decision support.

Despite this progress, the role of LLMs in financial risk forecasting, particularly for volatility and value at risk (VaR), remains insufficiently understood. Moreover, because LLMs are generative models, using them directly for precise numerical regression is still difficult [20]. This paper examines the role and limits of LLMs in financial risk forecasting and asks the following research questions:

- RQ1: What role do large language models play in financial risk forecasting?
- RQ2: How does the predictive performance of LLM-based methods compare with other AI techniques for risk metrics?
- RQ3: How can different input modalities and heterogeneous data sources be effectively integrated and balanced?

To answer these questions, we propose RiskLabs, a framework designed to capture the complexity of real investment environments. RiskLabs combines multimodal inputs from multiple sources to form a unified view of the factors that shape financial markets. The framework incorporates: (1) ECC transcripts, which reflect firm performance and outlook; (2) ECC audio, which captures tone and affect beyond literal content; (3) time-series data that describe recent market dynamics; and (4) news reports. LLMs are used to extract salient information from each modality before multimodal fusion.

RiskLabs consists of four components: (a) an earnings conference call encoder that processes ECC-related data with LLM support; (b) a news-market response encoder that uses an LLM-based pipeline for news collection and interpretation; (c) a

time-series encoder for temporal signals; and (d) a multi-task prediction module that fuses these representations. By combining quantitative market data with qualitative signals, the framework produces a richer representation of the investment environment. The model is trained to jointly predict volatility and VaR over 3-, 7-, 15-, and 30-day horizons.

The contributions of this paper are fourfold. First, we propose RiskLabs, a unified framework for financial risk forecasting that extends the use of LLMs in this domain. Second, the framework integrates multimodal and multi-source financial data to improve predictive accuracy. Third, experiments validate the effectiveness of the framework on financial risk forecasting tasks. Fourth, we provide a clearer account of the role and practical value of LLMs in financial risk prediction.

II. THE RISKLABS FRAMEWORK

Figure 1 shows the overall RiskLabs architecture. The framework operates on heterogeneous financial inputs, including audio, text, and time-series data from multiple sources. It consists of four modules: (1) an earnings conference call encoder, (2) a time-series encoder, (3) a related news encoder, and (4) a multimodal fusion and multi-task prediction module. The first three modules extract modality-specific features, which are then fused for volatility and VaR forecasting at multiple horizons.

A. EARNINGS CONFERENCE CALL ENCODER

The earnings conference call encoder contains three components: audio encoding, transcript encoding, and an ECC analyzer.

1) Audio Encoding.

Audio embeddings are first extracted using Wav2Vec2 [3], and then processed by a multi-head self-attention (MHSA) layer with pooling to obtain the audio feature vector T_a. Let the raw audio input be denoted as $A_c = \{a_1^c, a_2^c, ..., a_n^c\}$, where a_i^c is the ith audio frame in a sample. Each frame is converted into a vector representation:

$$e_i^{ac} = \text{Wav2Vec2}(a_i^c) \tag{1}$$

This yields the audio embedding sequence $E_{ac} = \{e_1^{ac}, e_2^{ac}, ..., e_n^{ac}\}$. The shape of E_{ac} is 520×512, where 520 is the maximum number of audio frames across all firms and 512 is the feature dimension produced by the model. ECCs with fewer than 520 frames ($n < 520$) are zero-padded to ensure a fixed input size.

The embedding sequence E_{ac} is then fed into MHSA to extract audio features. The MHSA block contains multi-head attention, normalization, and an MLP block. The MLP is implemented as a two-layer feed-forward network with ReLU activation. MHSA is a basic building block used throughout the framework. The multi-head self-attention operation is defined as:

$$\text{Multihead} = \text{Concat}(\text{head}_1, ..., \text{head}_h)W^O \tag{2}$$

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{3}$$

where the dimensions of query Q and key K are d_k, and the dimension of value V is d_v. The weight matrices satisfy $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$, and $W^O \in \mathbb{R}^{d_v \times d_{model}}$. Attention weights are computed by normalizing the query-key dot products and applying softmax:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{4}$$

Applying Eqs. (2)-(4) to E_{ac} gives:

$$T_{ac} = \text{MHSA}(E_{ac}) \tag{5}$$

where $T_{ac} = \{t_1^{ac}, t_2^{ac}, ..., t_n^{ac}\}$ has size 520×512. Average pooling is then applied:

$$T_a = \text{AveragePooling}(T_{ac}) \tag{6}$$

where T_a is the final 512-dimensional audio feature.

2) Transcript Encoding.

Similar to the audio branch, the transcript encoder first uses SimCSE [7] to generate vector representations for each sentence in the ECC transcript. These embeddings are then processed by an MHSA layer with pooling to obtain the text feature T_t. Let the raw transcript be $T_c = \{t_1^c, t_2^c, ..., t_n^c\}$, where t_i^c denotes the ith sentence. Each sentence is embedded as:

$$e_i^{tc} = \text{SimCSE}(t_i^c) \tag{7}$$

This produces $E_{tc} = \{e_1^{tc}, e_2^{tc}, ..., e_n^{tc}\}$ with shape 520×768, where 520 is the maximum number of sentences across all samples and 768 is the output dimension of SimCSE. ECCs with fewer than 520 sentences are zero-padded. Applying MHSA to E_{tc} yields $T_{tc} = \{t_1^{tc}, t_2^{tc}, ..., t_n^{tc}\}$, and average pooling gives:

$$T_t = \text{AveragePooling}(T_{tc}) \tag{8}$$

where T_t is the final 768-dimensional text feature.

3) ECC Analyzer.

The ECC analyzer uses an LLM to summarize earnings conference calls, extract salient information, and convert it into a text feature. The process is illustrated in Figure 2.

To summarize long ECC transcripts, we adopt a hierarchical strategy. The full document is first divided into chunks, and the LLM generates a summary for each chunk. These chunk summaries are then summarized again to form a document-level summary. This two-stage design preserves both local detail and global context. We then use OpenAI's `text-embedding-3-small` model to generate a 1024-dimensional embedding T_s for the summarized content:

$$T_s = \text{Embedding}(\text{Concatenated}[\text{chunk summaries}; \text{summary}]) \tag{9}$$

We further extract the most informative key sentences from the complete transcript. The processed text chunks are first embedded into a vector space. A finance-oriented question bank is then constructed with expert input and used as a set of queries. For each question, relevant chunks are retrieved from the vector space. A contextual compressor and the LLM are then used to filter these retrieved chunks and retain only the sentences most useful for answering the query. The selected sentences are organized into coherent prompts, and the LLM generates the corresponding answers, which are treated as distilled key statements. These generated sentences are concatenated and embedded to produce the final 1024-dimensional feature T_f:

$$T_f = \text{Embedding}(\text{Concatenated}[\text{selected sentences}]) \tag{10}$$

B. TIME-SERIES ENCODER

For the time-series branch, we use the Volatility Index (VIX) from the 30 days preceding the ECC release date. To extract informative temporal features from this sequence, we employ a bidirectional long short-term memory network (BiLSTM) [18]:

$$T_v = \text{BiLSTM}(\text{VIX})$$

The BiLSTM uses 64 hidden states, resulting in a 128-dimensional output feature T_v due to bidirectional modeling.

C. RELATED NEWS ENCODER

News is an important driver of stock-price movement and a key signal of market trends. It covers macroeconomic indicators, industry developments, and firm-specific events such as earnings releases, mergers and acquisitions, regulatory changes, and leadership turnover. For each stock, related news items are aggregated into a single text sequence and processed by an LLM to generate a news feature T_n.

D. MULTIMODAL FUSION AND MULTI-TASK LEARNING

We use additive fusion to learn a joint representation:

$$E = w_0 + w_1 \cdot T_a + w_2 \cdot T_t + w_3 \cdot T_f + T_v + T_n + \varepsilon \tag{11}$$

RiskLabs jointly models volatility prediction and VaR prediction in a multi-task framework. The prediction module contains two independent single-layer feed-forward networks, one for volatility (`vol`) and one for VaR (`var`). The framework predicts volatility and VaR across multiple horizons. The volatility of a stock is defined as the natural logarithm of the standard deviation of returns r within a τ-day window. The 3-day, 7-day, 15-day, and 30-day volatility targets are computed as:

$$v[d-\tau,d] = \ln\left(\sqrt{\frac{\sum_{i=0}^{\tau}(r_{d-i}-\bar{r})^2}{\tau}}\right) \tag{12}$$

Let $s \in S$ denote a stock and $c \in C$ an earnings conference call associated with stock s. Each conference call c is represented by a set of audio segments $a_i^c \in A_c$ and corresponding transcript sentences $t_i^c \in T_c$, where $i \in [1,N]$ and N is the maximum number of audio segments per call. For each stock, daily return is defined as $r_i = \frac{p_i - p_{i-1}}{p_{i-1}}$, where p_i is the adjusted closing price and $\bar{r}$ is the average return over the τ-day window. The regression goal is to learn $f(c) \to v[d-\tau,d]$.

The second task is to predict the one-day VaR of the target stock from multi-source inputs. VaR is defined as:

$$v = F^{-1}(p)$$

where $F(\cdot)$ is the cumulative loss distribution, p is the prespecified quantile, and v denotes VaR. Under quantile regression, we have:

$$L_\tau(y,\hat{y}) = \begin{cases} \tau \cdot (y-\hat{y}) & \text{if } y \geq \hat{y} \\ (1-\tau)\cdot(\hat{y}-y) & \text{if } y < \hat{y} \end{cases}$$

RiskLabs is trained by optimizing the multi-task loss:

$$L = \mu\left(\sum_i (\hat{y}_i - y_i)^2\right) + (1-\mu)\max(q\times(v-\hat{v}),(1-q)(\hat{v}-v)) \tag{13}$$

III. EXPERIMENTS AND DISCUSSION

A. EXPERIMENTAL SETUP

We compare the proposed volatility forecasting method against several representative baselines, including Multi-Task LSTM with Attention (MT-LSTM+ATT), Hierarchical Attention Networks (HAN), Multimodal Deep Regression Models (MRDM), and Hierarchical Transformer-based Multi-task Learning (HTML). The ECC analyzer and the news-market response encoder both use GPT-4. The temperature is fixed at 0 throughout the experiments to ensure stable LLM outputs and reproducible behavior. The overall training code is implemented in PyTorch. Each multi-head attention layer contains 8 attention heads, the batch size is selected from $b \in \{2,4,8\}$, and the learning rate of the Adam optimizer is selected from $\{1e\text{-}3, 1e\text{-}5, 1e\text{-}6, 1e\text{-}7\}$ via grid search. Except for the task-balancing parameter μ, the best hyperparameters are kept consistent across experiments.

a: Dataset.

The dataset is split into training and test sets with an 8:2 ratio. Importantly, the split is performed chronologically, so all dates in the training set precede those in the test set. This temporal isolation ensures that the forecasting problem is realistic and that the model always predicts future risk from past information.

b: Baselines.

We compare RiskLabs with the following baselines.

- **Classical methods.** We include GARCH and related variants [6, 11], which are classic autoregressive models for volatility forecasting. These methods are widely used for short-horizon volatility prediction, but may perform poorly when predicting average volatility over longer horizons.
- **LSTM [8].** A standard LSTM is used as a baseline because of its effectiveness in modeling sequential financial data.
- **MT-LSTM+ATT [15].** This model combines the prediction of average n-day volatility with single-day volatility forecasting and uses an attention-enhanced LSTM as the backbone.
- **HAN (GloVe).** This baseline implements a hierarchical attention network with both word-level and sentence-level attention. Words are first mapped into 300-dimensional pretrained GloVe embeddings, sentence representations are encoded with a Bi-GRU [4], and a second Bi-GRU is used to obtain document-level representations before regression.
- **MRDM [17].** MRDM introduces multimodal deep regression for volatility forecasting by combining pretrained GloVe embeddings with custom acoustic features. The two modalities are encoded separately by BiLSTMs and then fused through a two-layer dense network.
- **HTML [27].** HTML is a strong baseline that uses Whole Word Masking BERT (WWM-BERT) for transcript encoding. As in MRDM, the same audio features are used, and the unimodal features are further processed by a sentence-level Transformer to produce a multimodal representation for each call.
- **GPT-3.5-Turbo.** This baseline evaluates whether a general-purpose LLM can directly perform financial risk prediction. ECC transcripts are used as prompts, and GPT-3.5-Turbo is instructed to generate numerical risk forecasts with temperature fixed at 0.

B. PERFORMANCE COMPARISON (RQ1 AND RQ2)

Table 1 compares the models across 3-day, 7-day, 15-day, and 30-day forecasting horizons. In addition to predictive accuracy, the table reports VaR estimation and whether each method supports multi-task learning. RiskLabs achieves the best overall performance, particularly on short- and medium-horizon forecasting, as shown by its lower mean squared error (MSE). Its advantage is most pronounced when ECCs, time-series signals, and news are modeled jointly, where it outperforms the strong HTML baseline. RiskLabs also performs well on VaR prediction, although its 30-day volatility forecasts remain slightly worse than those of HTML, suggesting that LLM-assisted methods still face limitations at longer horizons.

Table 1. Performance comparison between representative baselines and the proposed RiskLabs framework

Model	MSE	3-day MSE	7-day MSE	15-day MSE	30-day MSE	VaR	Multi-task
Classical Methods	0.713	1.710	0.526	0.330	0.284	/	No
LSTM	0.746	1.970	0.459	0.320	0.235	/	No
MT-LSTM-ATT	0.739	1.983	0.435	0.304	0.233	/	No
HAN	0.598	1.426	0.461	0.308	0.198	/	No
MRDM	0.577	1.371	0.420	0.300	0.217	/	No
HTML	0.401	0.845	0.349	0.251	0.158	/	Yes
GPT-3.5-Turbo	2.198	2.152	1.793	2.514	2.332	0.371	Yes
RiskLabs	0.324	0.585	0.317	0.233	0.171	0.049	Yes

Table 2 further compares AI methods with traditional approaches on VaR estimation. With the target VaR set to 0.05, predictions closer to 0.05 are better. RiskLabs yields 0.049, the closest value among all methods. By contrast, the historical method produces 0.016, which departs markedly from the 95% VaR benchmark and suggests that purely historical approaches may overreact to earlier high-volatility periods. Two conclusions follow. First, directly prompting an LLM to output numerical risk forecasts is ineffective and may be unsafe in practice. Second, LLMs are better viewed as analytical assistants than standalone predictors: they help organize and interpret heterogeneous financial information, which in turn improves downstream deep learning models.

Table 2. VaR prediction comparison between AI methods and traditional approaches. The target VaR is 0.05

Method	Predicted VaR
Historical Method	0.016
Fully Connected Neural Network	0.044
LSTM	0.056
RiskLabs	0.049

C. ABLATION STUDY (RQ3)

Beyond overall model comparison, we also evaluate the contribution of different data combinations to prediction performance. The following ablation settings are considered:

- **Audio + Text.** Only ECC information is used. Audio and transcript signals are processed by multi-head self-attention to extract key features, which are then used directly for forecasting.
- **Audio + Text + Analysis.** On top of the previous setting, this variant adds the ECC analysis output produced by the full conference call encoder.
- **Audio + Text + Analysis + VIX.** This version further incorporates time-series information and evaluates how much the additional temporal signal improves performance.

Table 3. Ablation results for different module combinations

Modules	MSE	3-day MSE	7-day MSE	15-day MSE	30-day MSE	VaR
Audio + Text	0.373	0.645	0.362	0.280	0.204	0.131
Audio + Text + Analysis	0.357	0.627	0.335	0.267	0.199	0.057
Audio + Text + Analysis + VIX	0.324	0.585	0.317	0.233	0.171	0.049

As shown in Table 3, even the `Audio + Text` configuration outperforms HTML on the 3-day task. At the 7-, 15-, and 30-day horizons, it remains competitive with HTML and outperforms the other baselines. This suggests that large pretrained models already provide strong representations for textual and acoustic signals and that MHSA can extract useful predictive information

from them. Adding ECC analysis and time-series features yields further gains, especially at medium and long horizons. The results indicate that ECC information is more directly tied to short-term risk fluctuations, whereas multi-source integration matters more for longer-horizon forecasting.

D. CHALLENGES AND OPPORTUNITIES

The results indicate that LLMs can improve financial risk forecasting when used to integrate diverse information sources. This motivates extending the framework to additional signals that may affect market volatility, such as news disseminated through social media. Preliminary small-scale experiments suggest that such information may improve performance. Two challenges remain, however: news quality is highly uneven, and the relevant modules still require broader validation on larger datasets.

To address these issues, we focus on two directions: more selective filtering of low-quality news and continued data collection to expand the dataset. We also discuss several mechanisms that could strengthen the framework: (a) a Bayesian VaR method that updates estimates under a probabilistic prior; (b) a news-market response encoder; (c) dynamic rolling-window training; and (d) a time-decay hyperparameter in RiskLabs. Together, these additions could make daily training and forecasting more adaptive.

1) Modeling Relationships Among Multiple Response Variables with VAR.

RiskLabs predicts multiple risk indicators simultaneously, and these indicators are often intrinsically related. Modeling such interactions can further improve predictive performance by treating the indicators as parts of one coupled system rather than as isolated targets.

To capture these cross-horizon dependencies, we use a vector autoregression (VAR) model. Specifically, we consider four future volatility measures: 3-day, 7-day, 15-day, and 30-day volatility. Let $\sigma_{-m,t}$ denote the future m-day volatility observed at time $t-m$, computed by tracing back m days from the current time and calculating the standard deviation of returns over that period.

Volatility typically exhibits clustering, meaning that the fluctuation pattern in one period often extends into the next. This motivates the use of recent historical outcomes as predictors for future periods. We further assume that volatility correlations across different time scales interact with one another, and that these relationships can be quantified through a coefficient matrix estimated from historical data.

To estimate the model coefficients, we adopt a Bayesian approach and infer the posterior distribution of the parameters. For a generic linear regression $Y = X\theta$, where θ denotes the coefficient vector, the posterior is:

$$P(\theta|Y,X) = \frac{P(Y|\theta,X)P(\theta)}{P(Y|X)} = \frac{P(Y|\theta,X)P(\theta|X)}{\int_{-\infty}^{\infty} P(Y|X,\theta)p(\theta)d\theta} \tag{14}$$

where $P(\theta)$ or $P(\theta|X)$ is the prior distribution, which can be specified from previous studies or domain knowledge, and $P(Y|\theta,X)$ is the likelihood. Since $P(Y|X)$ is the marginal likelihood, Eq. (14) can be simplified to:

$$P(\theta|Y,X) \propto P(Y|\theta,X)P(\theta|X) \tag{15}$$

Once the prior and likelihood are specified, the posterior can be estimated. We assume a Gaussian prior for θ under simple linear regression. Empirical evidence shows that realized volatility is right-skewed [2], and earlier work has found that historical S&P 500 volatility follows a lognormal distribution [5]. We therefore model the logarithm of the original data and obtain the following VAR system:

$$\begin{cases} \log(\sigma_{3,t}) = \alpha_3 + \beta_{1,1}\log(\sigma_{-3,t}) + \beta_{1,2}\log(\sigma_{-7,t}) + \beta_{1,3}\log(\sigma_{-15,t}) + \beta_{1,4}\log(\sigma_{-30,t}) + u_{3,t} \\ \log(\sigma_{7,t}) = \alpha_7 + \beta_{2,1}\log(\sigma_{-3,t}) + \beta_{2,2}\log(\sigma_{-7,t}) + \beta_{2,3}\log(\sigma_{-15,t}) + \beta_{2,4}\log(\sigma_{-30,t}) + u_{7,t} \\ \log(\sigma_{15,t}) = \alpha_{15} + \beta_{3,1}\log(\sigma_{-3,t}) + \beta_{3,2}\log(\sigma_{-7,t}) + \beta_{3,3}\log(\sigma_{-15,t}) + \beta_{3,4}\log(\sigma_{-30,t}) + u_{15,t} \\ \log(\sigma_{30,t}) = \alpha_{30} + \beta_{4,1}\log(\sigma_{-3,t}) + \beta_{4,2}\log(\sigma_{-7,t}) + \beta_{4,3}\log(\sigma_{-15,t}) + \beta_{4,4}\log(\sigma_{-30,t}) + u_{30,t} \end{cases} \tag{16}$$

Here, $\sigma_{m,t}$ is the future m-day volatility at time t, $u_{m,t}$ is white noise with $u_{m,t} \sim N(0,1)$, $\beta_{i,j}$ denotes the linear dependency coefficients, and α_m is the intercept term.

We estimate the posterior by Markov Chain Monte Carlo (MCMC). MCMC approximates the posterior through sampling in probability space. A Markov chain satisfies:

$$P(X_{t+1} \mid X_1, X_2, \ldots, X_t) = P(X_{t+1} \mid X_t) \tag{17}$$

which means that the transition probability depends only on the current state. To compute posterior expectations when direct evaluation is difficult, we construct a Markov chain whose stationary distribution matches the target posterior. Metropolis-Hastings provides a general way to construct such a chain, while Gibbs sampling can be used for models with many parameters.

To evaluate the MCMC process, we use several diagnostics, including the Monte Carlo standard error (MCSE), effective sample size (ESS), and the Gelman-Rubin statistic $\hat{R}$. The latter is defined as:

$$\hat{R} = \sqrt{\frac{\text{Variance between Chains}}{\text{Variance within Chains}}} \tag{18}$$

In practice, $\hat{R}$ should be close to 1 and below 1.01.

To validate the method, we conduct a case study on the stock `TWTR` using a 250-day training window from 2016-02-22 to 2017-02-15. The Bayesian regression results are shown in Table 4. All parameters have $\hat{R} = 1.0$, and both `ess_tail` and `ess_bulk` exceed 7000, indicating stable sampling. Although there is no universally fixed ESS threshold, previous work suggests that ESS above 400 is generally sufficient for stable Monte Carlo standard errors [12, 22]. The results therefore support the validity of the sampling procedure.

Table 4. Bayesian VAR results obtained with MCMC

Dependent Variable	Independent Variable	N	Mean	Sd	HDI_3%	HDI_97%	mcse_mean	mcse_sd	ess_bulk	ess_tail	r_hat
3-Day	Intercept	250	-2.571	0.364	-3.275	-1.908	0.004	0.003	8342.0	7461.0	1.0
3-Day	3-Day	250	0.070	0.058	-0.044	0.174	0.001	0.000	9345.0	8076.0	1.0
3-Day	7-Day	250	0.118	0.116	-0.107	0.329	0.001	0.001	8512.0	7501.0	1.0
3-Day	15-Day	250	0.017	0.146	-0.249	0.297	0.002	0.001	7259.0	7407.0	1.0
3-Day	30-Day	250	0.203	0.143	-0.079	0.465	0.002	0.001	7072.0	7532.0	1.0
7-Day	Intercept	250	-2.748	0.251	-3.207	-2.265	0.003	0.002	8684.0	7809.0	1.0
7-Day	3-Day	250	0.042	0.040	-0.033	0.118	0.000	0.000	8600.0	7890.0	1.0
7-Day	7-Day	250	0.181	0.079	0.033	0.330	0.001	0.001	7281.0	7845.0	1.0
7-Day	15-Day	250	-0.078	0.100	-0.270	0.104	0.001	0.001	7428.0	7426.0	1.0
7-Day	30-Day	250	0.142	0.098	-0.035	0.333	0.001	0.001	7308.0	7386.0	1.0
15-Day	Intercept	250	-3.350	0.207	-3.729	-2.960	0.002	0.002	9410.0	8082.0	1.0
15-Day	3-Day	250	0.039	0.034	-0.025	0.103	0.000	0.000	10471.0	8203.0	1.0
15-Day	7-Day	250	0.068	0.067	-0.058	0.194	0.001	0.001	8018.0	7923.0	1.0
15-Day	15-Day	250	-0.026	0.085	-0.182	0.134	0.001	0.001	7053.0	7467.0	1.0
15-Day	30-Day	250	0.006	0.083	-0.149	0.165	0.001	0.001	7373.0	8124.0	1.0
30-Day	Intercept	250	-3.604	0.164	-3.904	-3.280	0.002	0.001	9388.0	8508.0	1.0
30-Day	3-Day	250	0.015	0.027	-0.034	0.067	0.000	0.000	9676.0	8515.0	1.0
30-Day	7-Day	250	0.108	0.052	0.006	0.200	0.001	0.000	7919.0	7510.0	1.0
30-Day	15-Day	250	0.005	0.065	-0.115	0.129	0.001	0.001	7004.0	7650.0	1.0
30-Day	30-Day	250	-0.140	0.065	-0.263	-0.020	0.001	0.001	7145.0	7494.0	1.0

We also conduct a rolling-window simulation based only on historical volatility information. As shown in Figure 3, each next-day forecast is trained on an approximately 250-day window, and the procedure is iterated 100 times. Table 5 reports the distribution of the absolute error percentage (AEP), defined as:

Table 5. Distribution statistics of the absolute error percentage (AEP)

Horizon	Sample Size	Mean	Std	Skewness	Kurtosis	5% Quantile	25% Quantile	50% Quantile	75% Quantile	95% Quantile
3 Days	100	-0.13167	0.08377	-0.42741	-0.70133	-0.26854	-0.19993	-0.13011	-0.07004	-0.01447
7 Days	100	-0.08501	0.05202	-0.37466	-0.43682	-0.17094	-0.11859	-0.08499	-0.04545	-0.00960
15 Days	100	-0.07950	0.05310	-0.23159	-1.37432	-0.16238	-0.11840	-0.07607	-0.03119	-0.00952
30 Days	100	-0.04195	0.03775	-1.75213	3.12077	-0.11573	-0.05777	-0.03011	-0.01555	-0.00600

$$AEP = \frac{|\hat{y} - y|}{y} \tag{19}$$

where $\hat{y}$ is the estimate from the Bayesian VAR model and y is the ground-truth value. As the prediction horizon increases, both the average estimation bias and the standard deviation of AEP decrease. The negative skewness is consistent with the negative mean values. These results suggest that a historical-data-based VAR remains a viable approach for multi-horizon volatility forecasting.

2) News-Market Response Encoder.

News is an important indicator of market trends and a major driver of stock-price movements. It spans macroeconomic indicators, industry developments, and firm-specific events such as earnings announcements, mergers and acquisitions, regulatory changes, and executive turnover.

Several properties of news shape its market impact. First, news is most influential when it is recent. Current information is more likely than stale information to affect decisions and beliefs. Second, similar news items often trigger similar market reactions. When two pieces of news share key characteristics such as topic, sentiment, and investor relevance, they often produce comparable responses. For example, an unexpected earnings beat usually pushes prices upward, whereas regulatory setbacks or legal disputes tend to drive them downward. Such patterned responses arise because investors interpret new information through past experience and established market precedents.

Analyzing similarity among news items is therefore useful for understanding potential market responses. However, a company may receive many news items on a single day, each contributing only partially to future price movements. We therefore treat all news related to the same company on the same day as a single analysis unit and evaluate the cumulative effect at the group level.

We collect all news related to a company within the three days preceding the target trading day and record the corresponding daily market reactions. This setup allows the LLM to analyze the news content and infer its likely effect on stock performance over the following days. We then search for historical dates whose news profiles resemble that of the day before the target date. These historical analogs provide additional evidence for inferring potential market reactions.

Based on these observations, we build a news-market response encoder with two functions. First, it collects company-specific news and daily market reactions from the three days before the target trading day and uses an LLM to analyze their likely effect on subsequent stock performance. Second, it retrieves historical dates with news similar to that observed on the target date. The main challenge is similarity assessment. Because the comparison is performed between sets of news rather than single articles, exactly repeated combinations are rare and direct retrieval is often inaccurate.

To address this issue, we design a news enhancement pipeline that extracts relevant attributes from each news group and stores them as metadata. These metadata are central to retrieving historically similar news groups. Rather than comparing raw news content directly, we first compare metadata and retrieve the top-k historical news groups that share similar attributes with the target. We then perform a finer-grained similarity assessment on this reduced candidate set.

This approach simplifies similarity retrieval and makes event-based comparison more efficient. Once similar historical news groups are identified, their associated market reactions can be used to infer likely market behavior after the target trading day.

3) Time-Decay Hyperparameter and Dynamic Rolling-Window Training.

The main inputs to RiskLabs are: (1) earnings conference calls, (2) historical time-series data, and (3) news articles. However, these inputs are not available at the same frequency. News and time-series data can typically be collected daily, whereas earnings conference calls occur only on discrete reporting dates.

As a result, some training days may not contain a newly released conference call. If ECC information is simply omitted on those days, the model cannot capture the continuing effect that a recent conference call may still exert on stock prices. To preserve this lingering influence, we introduce a hyperparameter that controls the decay of ECC relevance over time via an exponential decay function:

$$I(t) = I(0)e^{-\lambda t} \tag{20}$$

where λ controls the decay rate and t is the elapsed time since the most recent earnings conference call. For a fixed λ, the influence of the conference call gradually weakens as t increases, which is consistent with common financial intuition. Figure 6 illustrates this daily-input setting. When no new ECC is available, the model still uses the most recent conference call and modulates its diminishing impact through λ.

Another time-related issue is that the effects of different inputs evolve over time. RiskLabs uses both time-series data and news, and the relationship between these inputs and the response variables is often highly time-sensitive. If the model is trained once and then applied far beyond the training period, performance may deteriorate because the model no longer matches later market conditions.

To remain sensitive to current trends and volatility, model parameters should be updated frequently. We address this with dynamic rolling-window training. A model is trained on a fixed historical window up to the target trading day and used only for that day. The window then advances by one day and a new model is trained for the next target date. This iterative process keeps the system aligned with the latest market data.

a: Detailed VaR Analysis.

Figure 7 compares VaR predictions against realized asset returns. The historical method produces a relatively flat curve, suggesting stable but less responsive forecasts. In contrast, the fully connected neural network shows stronger day-to-day variation, indicating greater responsiveness to new information. This difference suggests that AI-based methods, including neural networks and LLM-assisted models, are better able to absorb daily information than methods that rely mainly on static historical scenarios.

IV. CONCLUSION

This paper proposes RiskLabs, a framework for financial risk forecasting that incorporates LLMs into a multimodal, multi-source prediction pipeline. By organizing and analyzing heterogeneous financial data with LLM support, RiskLabs improves the performance of deep learning models on risk forecasting tasks. The framework combines an ECC encoder, a time-series encoder, and a news-market response encoder, whose outputs are fused for multi-task prediction of volatility and VaR.

The empirical results support three conclusions. First, RiskLabs is effective for financial risk forecasting. Second, LLMs are not reliable standalone numerical forecasters, but they are useful for organizing, analyzing, and representing financial information in ways that improve downstream prediction models. Third, the ablation study confirms that each major component of RiskLabs contributes to final performance. Overall, the results suggest that LLMs have significant potential in financial risk assessment and open new directions for AI-driven financial modeling.

REFERENCES

[1] Noujoud Ahbali, Xinyuan Liu, Albert Nanda, Jamie Stark, Ashit Talukder, and Rupinder Paul Khandpur. 2022. Identifying corporate credit risk sentiments from financial news. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*. 362-370.

[2] Torben G Andersen, Tim Bollerslev, Francis X Diebold, and Heiko Ebens. 2001. The distribution of realized stock return volatility. *Journal of Financial Economics* 61, 1 (2001), 43-76.

[3] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems* 33 (2020), 12449-12460.

[4] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

[5] Pierre Cizeau, Yanhui Liu, Martin Meyer, C-K Peng, and H Eugene Stanley. 1997. Volatility distribution in the S&P500 stock index. *Physica A: Statistical Mechanics and its Applications* 245, 3-4 (1997), 441-445.

[6] Philip Hans Franses and Dick Van Dijk. 1996. Forecasting stock market volatility using (non-linear) GARCH models. *Journal of Forecasting* 15, 3 (1996), 229-236.

[7] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821* (2021).

[8] Felix A Gers, Jurgen Schmidhuber, and Fred Cummins. 2000. Learning to forget: Continual prediction with LSTM. *Neural Computation* 12, 10 (2000), 2451-2467.

[9] Shihao Gu, Bryan Kelly, and Dacheng Xiu. 2020. Empirical asset pricing via machine learning. *The Review of Financial Studies* 33, 5 (2020), 2223-2273.

[10] Luckyson Khaidem, Snehanshu Saha, and Sudeepa Roy Dey. 2016. Predicting the direction of stock market prices using random forest. *arXiv preprint arXiv:1605.00003* (2016).

[11] Ha Young Kim and Chang Hyun Won. 2018. Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models. *Expert Systems with Applications* 103 (2018), 25-37.

[12] John Kruschke. 2014. *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan*. Academic Press.

[13] Young-Chan Lee. 2007. Application of support vector machines to corporate credit rating prediction. *Expert Systems with Applications* 33, 1 (2007), 67-74.

[14] Charalampos M Liapis, Aikaterini Karanikola, and Sotiris Kotsiantis. 2023. Investigating deep stock market forecasting with sentiment analysis. *Entropy* 25, 2 (2023), 219.

[15] Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114* (2015).

[16] Saloni Mohan, Sahitya Mullapudi, Sudheer Sammeta, Parag Vijayvergia, and David C Anastasiu. 2019. Stock price prediction using news sentiment analysis. In *2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService)*. IEEE, 205-208.

[17] Yu Qin and Yi Yang. 2019. What you say and how you say it matters: Predicting stock volatility using verbal and vocal cues. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 390-401.

[18] Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. 2019. The performance of LSTM and BiLSTM in forecasting time series. In *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 3285-3292.

[19] Amanpreet Singh, Narina Thakur, and Aakanksha Sharma. 2016. A review of supervised machine learning algorithms. In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*. IEEE, 1310-1315.

[20] Xingyou Song, Oscar Li, Chansoo Lee, Daiyi Peng, Sagi Perel, Yutian Chen, et al. 2024. OmniPred: Language models as universal regressors. *arXiv preprint arXiv:2402.14547* (2024).

[21] Wataru Souma, Irena Vodenska, and Hideaki Aoyama. 2019. Enhanced news sentiment analysis using deep learning methods. *Journal of Computational Social Science* 2, 1 (2019), 33-46.

[22] Aki Vehtari, Andrew Gelman, Daniel Simpson, Bob Carpenter, and Paul-Christian Burkner. 2021. Rank-normalization, folding, and localization: An improved $\hat{R}$ for assessing convergence of MCMC (with discussion). *Bayesian Analysis* 16, 2 (2021), 667-718.

[23] Dan Wang, Zhi Chen, Ionut Florescu, and Bingyang Wen. 2023. A sparsity algorithm for finding optimal counterfactual explanations: Application to corporate credit rating. *Research in International Business and Finance* 64 (2023), 101869.

[24] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems* 35 (2022), 24824-24837.

[25] Qianqian Xie, Weiguang Han, Zhengyu Chen, Ruoyu Xiang, Xiao Zhang, Yueru He, Mengxi Xiao, Dong Li, Yongfu Dai, Duanyu Feng, et al. 2024. The FinBen: A holistic financial benchmark for large language models. *arXiv preprint arXiv:2402.12659* (2024).

[26] Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. 2023. FinGPT: Open-source financial large language models. *arXiv preprint arXiv:2306.06031* (2023).

[27] Linyi Yang, Tin Lok James Ng, Barry Smyth, and Riuhai Dong. 2020. HTML: Hierarchical transformer-based multi-task learning for volatility prediction. In *Proceedings of The Web Conference 2020*. 441-451.

[28] Yangyang Yu, Haohang Li, Zhi Chen, Yuechen Jiang, Yang Li, Denghui Zhang, Rong Liu, Jordan W Suchow, and Khaldoun Khashanah. 2023. FinMem: A performance-enhanced LLM trading agent with layered memory and character design. *arXiv preprint arXiv:2311.13743* (2023).

[29] Beichen Zhang. 2020. *Financial Risk Disclosure Return Premium: A Topic Modeling Approach*. Master's thesis. Stevens Institute of Technology.

[30] Boyu Zhang, Hongyang Yang, Tianyu Zhou, Muhammad Ali Babar, and Xiao-Yang Liu. 2023. Enhancing financial sentiment analysis via retrieval augmented large language models. In *Proceedings of the Fourth ACM International Conference on AI in Finance*. 349-356.

[31] Chong Zhang, Xinyi Liu, Mingyu Jin, Zhongmou Zhang, Lingyao Li, Zhengting Wang, Wenyue Hua, Dong Shu, Suiyuan Zhu, Xiaobo Jin, et al. 2024. When AI Meets Finance (StockAgent): Large language model-based stock trading in simulated real-world environments. *arXiv preprint arXiv:2407.18957* (2024).

[32] Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B Hashimoto. 2024. Benchmarking large language models for news summarization. *Transactions of the Association for Computational Linguistics* 12 (2024), 39-57.

[33] Wentao Zhang, Lingxuan Zhao, Haochong Xia, Shuo Sun, Jiaze Sun, Molei Qin, Xinyi Li, Yuqing Zhao, Yilei Zhao, Xinyu Cai, et al. 2024. FinAgent: A multi-modal foundation agent for financial trading: Tool-augmented, diversified, and generalist. *arXiv preprint arXiv:2402.18485* (2024).