

Stock Prediction Based On Annual Reports Using Large Language Model

Naveena Pokala

Dept. of Computer Science
UNC Greensboro
Greensboro, NC, U. S. A.
n_pokala@uncg.edu

Sakhawat Hossan

Dept. of Computer Science
UNC Greensboro
Greensboro, NC, U. S. A.
s_hossan@uncg.edu

Jing Deng

Dept. of Computer Science
UNC Greensboro
Greensboro, NC, U. S. A.
jing.deng@uncg.edu

Dayong Huang

Dept. of Accounting and Finance
UNC Greensboro
Greensboro, NC, U. S. A.
d_huang@uncg.edu

Abstract—As a forever goal of accurate stock market prediction, annual financial reports have been used in different ways, financial analysts use annual reports to evaluate a company’s performance and outlook, while long-term investors assess them to guide investment decisions. In this paper, we leverage these reports to predict stock prices using deep learning models. We also present a novel market integration strategy for modeling the temporal decay of information relevance in financial markets. Unlike prior studies that treat annual report information as static, our framework dynamically adjusts the predictive weight of corporate disclosures over time, reflecting how markets gradually absorb new information. By combining large language model (LLaMA-2 70B) embeddings of 10-K filings with a Market Integration Convolutional Neural Network (MI-CNN), our model predicts stock prices over short-term (1-day and 7-day) and medium-term (14-day and 30-day) horizons. This design captures both immediate reactions and sustained effects of fundamental information. Our results on S&P 500 data from year 2000–year 2025 show that the proposed MI-CNN approach performs competitively with established baseline models across all the prediction horizons.

Keywords—Machine Learning, Deep Learning, Big Data/Data Science

I. INTRODUCTION

Accurate prediction of stock prices remains a challenging problem due to the complex interplay between quantitative performance indicators and qualitative corporate disclosures. While traditional models rely primarily on structured data such as revenues and earnings, they often neglect the rich contextual information embedded in annual reports (10-K filings), which encapsulate managerial insights, strategic direction, and risk assessments that influence long-term investor sentiment.

However, most existing approaches assume the information contained in such reports retains uniform predictive power throughout the year, ignoring the dynamic nature of information absorption in real-world markets. To address this limitation, we introduce a Market Integration Strategy—a novel temporal modeling approach that quantifies how the predictive strength of annual reports diminishes as markets progressively integrate this information. Implemented through an Annual Cycle Index feature within our Market Integration deep learning framework, this strategy enables the model to account for varying relevance of fundamental disclosures across time.

Our Market Integration model combines contextual embeddings from LLaMA-2 70B with a CNN-based temporal module to forecast stock prices at four future intervals: 1-day, 7-day, 14-day and 30-day. The shorter horizon captures immediate market reactions, while the longer horizon assesses sustained market assimilation. Empirical results suggest that the proposed framework performs competitively with baseline models across all the prediction horizons.

II. RELATED WORK

The application of deep learning and large language models (LLMs) in financial prediction has gained substantial traction in recent years. These models enable nuanced analysis of both structured time series data and unstructured financial text such as SEC filings, earnings reports, and financial news.

Rao and Zhang [1] proposed a hybrid deep learning model integrating LSTM, Transformer, and CNN layers to classify AI-generated text, achieving over 99% accuracy. Although not designed specifically for stock forecasting, their architecture highlights the benefits of combining local and global feature extraction for sequential data analysis—principles highly applicable to financial modeling.

Bock [2] demonstrated the use of a neural language model to generate long-horizon “buy” signals directly from textual financial data. Similarly, Daimi and Iqbal [3] proposed a scalable LLM-based pipeline for systematic analysis of SEC 10-K filings, showing that transformer models can uncover latent signals within regulatory filings that correlate with future returns. Ni et al. [4] introduced a QLoRA-enhanced LLM fine-tuned specifically on earnings reports for stock prediction. Their work emphasizes the importance of efficient fine-tuning strategies (e.g., parameter-efficient methods) to achieve domain adaptation without the computational cost of full-scale retraining.

Papasotiriou et al. [5] evaluated the performance of GPT-based models on the equity stock rating task using multi-modal data (financial fundamentals, market prices, and news sentiment). Their framework improved consistency and performance over traditional analyst-driven approaches and highlighted LLMs’ interpretability and cost efficiency.

Incorporating financial sentiment, Araci’s FinBERT [6] adapts BERT [7] for domain-specific financial text, outper-

forming generic language models on sentiment classification. FinBERT has been particularly useful when used as a preprocessing step in downstream forecasting tasks.

Brown et al. [8] introduced GPT-3, which has enabled few-shot financial reasoning, while Touvron et al. [9] developed LLaMA—a foundation model architecture known for its efficiency and performance on domain adaptation tasks.

Mehtab et al. [10] performed a comparative study of LSTM-based deep models and machine learning regressors, demonstrating that properly tuned LSTM models can yield highly accurate forecasts of NIFTY 50 index values.

Zhang et al. [11] applied a CNN model to financial news for trend prediction. These models achieve strong short-term forecasting performance but focus mainly on capturing local temporal dependencies or optimizing regression objectives.

Together, these works suggest that blending temporal modeling (e.g., CNN, LSTM) with LLM-derived contextual embeddings from regulatory and news text can lead to highly robust, interpretable, and forward-looking stock prediction systems. Our approach builds on this direction by using contextualized LLM embeddings from annual reports and aligning them with daily stock prices in a CNN pipeline for directional and magnitude-based forecasting.

III. DATASET PREPARATION AND PREPROCESSING

A. Collection of Annual Reports

We collected annual reports, specifically 10-K filings, for all S&P 500 companies spanning the period from 2000 to 2025 from the U.S. Securities and Exchange Commission’s (SEC) EDGAR database, which serves as the primary repository for corporate financial disclosures. Each report was stored in raw text form using a consistent naming convention: TICKER_YEAR.txt, where TICKER denotes the stock symbol and YEAR corresponds to the fiscal year the report is filed for, facilitating easy alignment with corresponding stock price data¹ and enabling scalable batch-level processing. To focus on semantically meaningful financial content, we implemented a sentence-level filtering pipeline using spaCy for sentence segmentation and a curated set of financial keywords (e.g., revenue, earnings, liabilities) to identify candidate sentences, which were then filtered by regular expressions matching dollar amounts, percentages, fiscal years, and quarters. Each selected sentence was cleaned of encoding artifacts and stored as a compressed version of the report, dramatically reducing document length while preserving core financial signals and improving the efficiency and relevance of downstream modeling.

To complement the textual data, we collected historical stock prices for each company using the Yahoo Finance Python library, extracting daily opening, closing, high, low prices, trading volume and volatility for the full 25-year time window. Daily returns are calculated as the percentage change between consecutive days for each price type. And volatility measure is

also calculated using rolling standard deviation of closing price returns over a 5-day window. This would allow the model to learn relative price movements rather than absolute price levels because stock prices in earlier years are much lower than those in later years.

B. Embedding Generation

To enhance the relevance of extracted representations, we applied prompt engineering before tokenization by prefixing each annual report with carefully designed prompts such as “You are a financial analyst. how this report may affect to the stock prices” to focus the model’s attention on economically meaningful content relevant to downstream predictive tasks. The reports were then tokenized using a tokenizer compatible with the LLaMA-2 70B architecture, with excessively long documents processed in chunks due to model input limitations.

Once tokenized, the reports were passed through the LLaMA-2 70B model in evaluation mode to obtain hidden state representations from the final transformer layer. Rather than using only the final token embedding, we computed the mean of hidden states across all tokens to generate a robust, fixed-length document representation, yielding a dense 8192-dimensional vector that encapsulates the overall semantic content while minimizing individual token-level noise. These embeddings served as the primary textual input to our CNN-based prediction model.

IV. MARKET INTEGRATION STRATEGY: A NOVEL APPROACH TO TEMPORAL INFORMATION DECAY IN FINANCIAL PREDICTION

A. Market Integration Strategy

The Market Integration Strategy represents a fundamental innovation in modeling the temporal decay of information relevance in financial markets. Traditional approaches to stock prediction, such as those grounded in Fama [12] Efficient Market Hypothesis (EMH), assume instantaneous and complete absorption of public information into asset prices, leaving no room for delayed predictive effects. This assumption fails to account for the dynamic nature of information processing in real-world markets. Our strategy is inspired by behavioral critiques of EMH, articulated by Shiller [13], which suggest that information is not instantly absorbed due to factors such as investor psychology, media-driven narrative diffusion, and social contagion. These behavioral dynamics imply that markets integrate complex disclosures like annual reports gradually, as investor attention, interpretation, and sentiment evolve over time. By incorporating a time-sensitive weighting mechanism, our framework captures this evolving relevance and aligns more closely with behavioral finance than with the static assumptions of classical EMH.

B. Annual Cycle Index

Our Market Integration Strategy is implemented through a sophisticated neural network architecture that dynamically adjusts the weighting of annual report information based on temporal positioning within the prediction year. The strategy

¹Here we focus on closing price only. Opening price and combination of both would fall into our future work.

introduces a novel feature called ρ , represents the normalized day index, indicating the relative position of a day in the fiscal year. It is computed as:

$$\rho = \frac{d}{D} \quad (1)$$

where d is the day index and D represents the total number of days in the fiscal year.

C. Document Encoder

The Document Encoder processes high-dimensional annual report embeddings while simultaneously modeling market integration effects. The encoder transforms annual report embeddings $\mathbf{e} \in \mathbb{R}^{8192}$ from LLaMA-2 through a multi-layer perceptron to extract document features $\mathbf{f}_d$:

$$\mathbf{f}_d = \text{ReLU}(\mathbf{W}_2 \cdot \text{ReLU}(\mathbf{W}_1 \mathbf{e} + \mathbf{b}_1) + \mathbf{b}_2) \quad (2)$$

Simultaneously, a parallel pathway processes the temporal information to determine the degree of market integration:

$$w = \text{ReLU}(W_4 \cdot \text{ReLU}(W_3 \rho + b_3) + b_4) \quad (3)$$

The market integration factor is computed as:

$$\alpha = \sigma(w) \quad (4)$$

where σ is the sigmoid function ensuring $0 < \alpha < 1$.

The remaining predictive power is calculated as:

$$\beta = 1 - \alpha \quad (5)$$

The final document representation incorporates the temporal weighting:

$$f = f_d \odot \beta \quad (6)$$

where $\odot$ denotes element-wise multiplication. This formulation elegantly captures the intuition that as market integration approaches 1 (complete absorption), the remaining predictive power approaches 0 (minimal additional information value).

D. Market Integration Dynamics

The temporal dynamics of our Market Integration Strategy reflect realistic information flow patterns in financial markets. In the first quarter following annual report release, when $\rho \approx 0.08 - 0.33$, market integration is minimal, preserving substantial predictive power of the annual report. During this period, the market is actively processing forward-looking guidance, strategic initiatives, and financial health indicators disclosed in the annual report.

As the prediction year progresses into the second and third quarters ($\rho \approx 0.33 - 0.75$), market integration increases as quarterly earnings reports provide updated performance data, management guidance is refined, and the annual report's strategic outlook becomes partially validated or contradicted by emerging results. The model automatically reduces the weight assigned to annual report information during this period.

In the fourth quarter ($\rho \approx 0.75 - 1.0$), when three quarters of actual performance data are available and the market anticipates the next annual report, integration is substantial, significantly diminishing the predictive relevance of the previous year's disclosures.

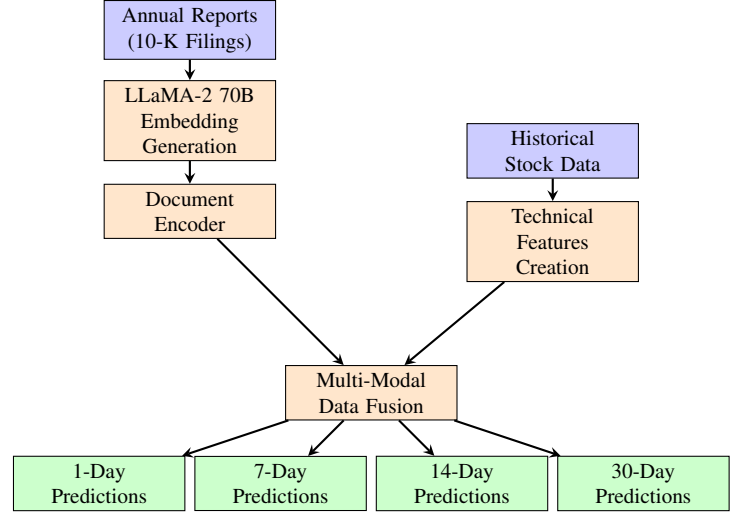


Fig. 1. Model's Flow Chart

V. MARKET INTEGRATION CNN (MI-CNN) ARCHITECTURE FOR MULTI-MODAL FINANCIAL PREDICTION

A. Architecture Overview

Our Market Integration CNN (MI-CNN) addresses a key limitation of traditional financial prediction models, which often rely solely on either textual analysis or technical indicators. By combining fundamental information from annual reports with historical stock price patterns, our hybrid architecture integrates three components: a Document Encoder with market integration modeling, a Price Sequence Encoder for technical feature extraction, and a Fusion Network for multi-modal prediction.

Figure 1 presents the overall architecture of our proposed framework, illustrating the complete pipeline from data inputs to final stock price predictions. The framework processes two parallel streams of information: (1) annual reports (10-K filings) processed through LLaMA-2 70B embedding generation, a document encoder, and a market integration strategy, and (2) historical stock data processed through technical feature creation. These two streams converge at the multi-modal data fusion stage, which feeds into the MI-CNN model to generate 1-day, 7-day, 14-day and 30-day predictions, ultimately producing stock price forecasts. Figure 2 provides a detailed view of the MI-CNN architecture, showcasing the internal processing mechanisms.

B. Technical Pattern Extraction

The Price Sequence Encoder component extracted meaningful patterns from sequences of technical indicators derived

from 60 days of historical stock price data. The technical features included returns for open, high, low, and close prices, normalized trading volume, and rolling volatility. The CNN architecture used one-dimensional convolutions to capture short-term dependencies in price sequences, producing a compact technical representation $\mathbf{f}_{price} \in \mathbb{R}^{32}$.

The CNN employed two convolutional layers with increasing filter dimensions (32 to 64) to learn increasingly abstract representations of technical patterns. Adaptive average pooling reduced the temporal dimension while preserving salient features, enabling the model to capture both local technical patterns and global trends in price sequences.

C. Multi-Modal Fusion and Prediction

The Fusion Network combined the weighted document features from market integration strategy and technical features through concatenation:

$$\mathbf{f}_{combined} = [\mathbf{f}_{doc}^{weighted}; \mathbf{f}_{price}] \in \mathbb{R}^{96} \quad (7)$$

where the semicolon denotes concatenation, resulting in a 96-dimensional combined feature vector. The combined features were processed through multiple fully connected layers with decreasing dimensionality to produce the final fused representation:

$$\mathbf{f}_{fused} = \text{MLP}(\mathbf{f}_{combined}) \in \mathbb{R}^{32} \quad (8)$$

This fusion approach preserved the distinct characteristics of both modalities while enabling the network to learn complex interactions between fundamental and technical factors. The architecture enabled the model to learn when technical patterns should override fundamental signals, when fundamental factors should dominate, and how to optimally combine both sources depending on market conditions.

The model generated predictions for multiple horizons simultaneously through specialized prediction heads:

$$\hat{r}_{1d} = \mathbf{w}_{1d}^T \mathbf{f}_{fused} + b_{1d} \quad (9)$$

$$\hat{r}_{7d} = \mathbf{w}_{7d}^T \mathbf{f}_{fused} + b_{7d} \quad (10)$$

$$\hat{r}_{14d} = \mathbf{w}_{14d}^T \mathbf{f}_{fused} + b_{14d} \quad (11)$$

$$\hat{r}_{30d} = \mathbf{w}_{30d}^T \mathbf{f}_{fused} + b_{30d} \quad (12)$$

where $\hat{r}_h$ denotes the predicted percentage return for horizon h , computed as a linear transformation of the fused feature representation with learnable weight vector $\mathbf{w}_h$ and bias term b_h . The predicted returns were converted to absolute price predictions using:

$$\hat{P}_{t+h} = P_t \times \left(1 + \frac{\hat{r}_h}{100}\right) \quad (13)$$

where P_t was the current stock price.

D. Training Configuration

Training configuration used for the model optimization is as shown in Table I.

The dataset was divided using a two-level time-aware strategy to prevent data leakage and ensure realistic forecasting

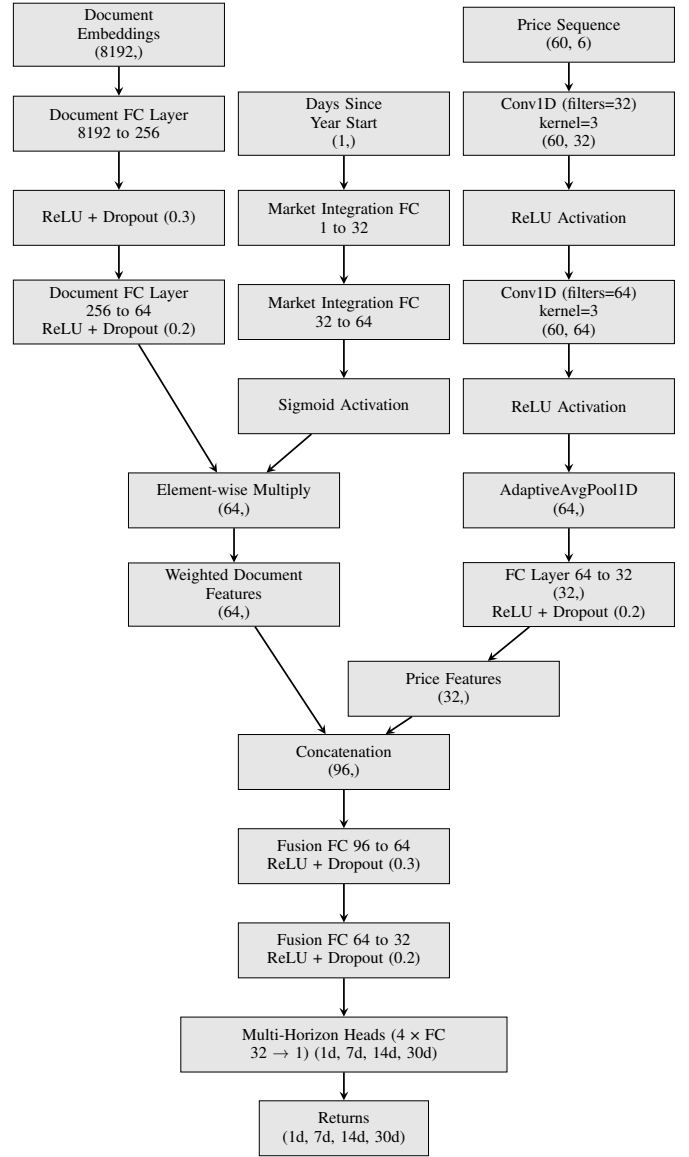


Fig. 2. Market Integration CNN (MI-CNN) architecture

evaluation. First, all available annual reports were split chronologically into separate training and testing folders, where the model was trained on earlier years and evaluated on unseen future years, preserving temporal consistency. Within the training folder, an 80/20 chronological split was applied to create the training and validation subsets used for hyperparameter tuning and early stopping, ensuring that validation data always represented later periods within the training range. The testing folder was used only for the final evaluation of the trained model's predictive performance across multiple metrics (e.g., MAPE, RMSE).

E. Training Procedure

Our prediction framework employed a multi-modal approach that combined historical price patterns with fundamental analysis to forecast future stock movements over multiple

TABLE I
TRAINING CONFIGURATION

Parameter	Value
Optimizer	AdamW
Activation Function	ReLU
Learning Rate	0.0001
Batch Size	32
Weight Decay	0.01
Loss Function	L1 Loss
Max Epochs	50
Early Stopping Patience	5 Epochs

horizons. Specifically, the model took input as a sequence of 60 consecutive trading days of historical stock price data, encompassing technical indicators such as returns, trading volume, and volatility measures, alongside the annual report embeddings that captured comprehensive fundamental information about the company’s financial health and strategic outlook. Using this combined information, the model generated predictions for stock price movements at four distinct future horizons: 1-day, 7-day, 14-day and 30-day ahead of the current prediction point. This temporal structure enabled the model to capture both short-term market dynamics influenced by technical patterns and longer-term fundamental factors derived from annual report disclosures (Note : All the prediction horizons are in trading days). The 60-day historical window provided sufficient context for identifying technical trends and momentum patterns, while the prediction horizons allowed for comprehensive assessment of how fundamental information affected stock prices over different time scales, with 1-day and 7-day predictions capturing immediate market reactions and 14-day and 30-day predictions reflecting more sustained fundamental impacts.

VI. RESULTS AND DISCUSSIONS

We considered ARIMA and LSTM as our baseline model. To better understand the contribution of annual report, ARIMA and LSTM model are trained only on numerical data.

TABLE II
PERFORMANCE COMPARISON OF ARIMA, LSTM, AND MI-CNN
ACROSS MULTIPLE PREDICTION HORIZONS

Horizon	Model	MAE	RMSE	MAPE (%)	Dir. Hit Rate
1-Day	ARIMA	2.36	3.25	1.40	49.45
	LSTM	2.36	3.26	1.41	50.54
	MI-CNN	2.37	3.27	1.42	50.31
7-Day	ARIMA	6.63	8.54	3.95	47.80
	LSTM	6.64	8.55	3.95	51.66
	MI-CNN	6.67	8.60	3.98	52.05
14-Day	ARIMA	9.35	11.79	5.59	46.90
	LSTM	9.33	11.76	5.60	52.54
	MI-CNN	9.42	11.86	5.66	52.78
30-Day	ARIMA	14.04	17.16	8.31	45.88
	LSTM	13.97	17.05	8.32	53.21
	MI-CNN	14.19	17.27	8.51	53.14

A. Discussion of Results

The performance comparison between ARIMA, LSTM, and the proposed MI-CNN in Table II shows that all three models performed similarly with respect to every metric evaluated across all the prediction horizon. Even though ARIMA produced the low error values for 1-day and 7-day prediction horizons, those differences were minimal when compared to both LSTM and MI-CNN. Therefore, it appears that simple statistical methods can be equally effective as sophisticated deep learning techniques for short term forecasts.

For 1-day prediction horizon, MI-CNN and LSTM achieved almost same direction hit rate, while ARIMA is slightly behind at 49.45%. Although, all three values are close to 50%, even small improvements above this threshold can be meaningful in financial prediction.

For 7-day prediction horizon, ARIMA performed competitively across all the metrics but failed to improve in direction hit rate. But LSTM and MI-CNN model shows improvement in direction hit rate. This indicates that while traditional models may perform well in minimizing numerical error, they are less effective in capturing the directional movement of stock prices.

Overall, these results suggest that increasing model complexity may not significantly improve performance for short-term forecasting tasks. Short-term price movements appear to be dominated by market noise, limiting the advantage of more sophisticated models.

For the 14-day and 30-day prediction horizons, all models again show similar performance in terms of error metrics. However, LSTM and MI-CNN achieve higher direction hit rates compared to ARIMA. This suggests that deep learning models are better at capturing directional trends over longer horizons. Additionally, direction hit rate improves for LSTM and MI-CNN as the prediction horizon increases, while it decreases for ARIMA. This indicates that ARIMA struggles to capture longer-term market dynamics.

B. Analysis of Results

In comparison across multiple horizons there are many similar conclusions to be drawn about the performance of these various methods:

- All three models perform almost same for 1-day prediction horizon. This highlight the difficulty of modeling short-term stock movements due to high market noise and volatility.
- Although traditional time series forecasting techniques such as ARIMA have been effective at reducing error in terms of magnitude they tend to do poorly when it comes to identifying direction or trend. And the error in direction hit rate has decreased with increase in horizons.
- LSTM and MI-CNN exhibit very similar performance across all metrics and horizons. This suggests that the inclusion of annual report (10-K) information alone does not provide a significant advantage over models based solely on historical price data.
- One possible explanation is that annual reports are low-frequency disclosures and may not capture short-term

market dynamics effectively. As a result, their predictive signal for short- and medium-term price movements may be limited or already incorporated into market prices.

- These findings motivate the need to incorporate higher-frequency and event-driven financial disclosures, such as quarterly reports (10-Q) and 8-K filings, which provide more timely updates about company performance and significant events. Such information may offer additional predictive value that is not captured by annual reports alone.

VII. LIMITATIONS AND FUTURE WORK

Our Market Integration framework is mainly based on pre-computed annual report embeddings, omitting higher-frequency signals such as quarterly earnings, management guidance, or analyst revisions. This leaves temporal gaps where information decay and sentiment shift during major corporate or sectoral events may not be fully captured. Additionally, our analysis focuses on a specific set of firms and periods, and market integration patterns may vary across regulatory frameworks, reporting standards, and market conditions, e.g. bull vs. bear market conditions, limiting its impacts. Another limitation is that the embeddings are generated using a pre-trained language model without domain-specific fine-tuning. While these models can provide good representations, they can fail to represent the nuances of financial semantics for more precise prediction.

Future studies should investigate the utilization of domain specific financial language models, e.g., FinBERT and FinGPT, along with examining various methods for fine tuning on LLM prompts for better embeddings.

VIII. CONCLUSION

This work presents a systematic deep learning-based investigation of market integration theory, offering both theoretical insight and practical value for financial prediction based on corporate financial reports. Across all prediction timelines, the proposed MI-CNN model demonstrates strong performance compared to baseline models in terms of error-based metrics, maintaining comparable MAE, RMSE, MAPE, and especially direction hit rate.

These findings illustrate that Market Integration Strategy offers an interesting approach in modelling the temporal decay of annual reports. By dynamically adjusting the relevance of annual report embeddings over time, the model aligns more closely with real-world market behavior, where annual report information is gradually absorbed rather than instantaneously, sometimes even before the reports are published.

Overall, this study highlights both the potential and limitations of incorporating low-frequency financial disclosures into stock prediction models. The findings suggest that while annual reports contain valuable contextual information, they are limited by their inability to capture high frequency changes. This support the need for incorporating higher-frequency and event-driven data sources like quarterly reports and 8-K reports to fully capture market dynamics.

REFERENCES

- [1] J. Rao and Q. Zhang, "Deep learning with llm: A new paradigm for financial market prediction and analysis," *International Journal of Multidisciplinary Research and Growth Evaluation*, vol. 6, no. 2, pp. 211–215, 2025.
- [2] J. R. Bock, "Generating long-horizon stock "buy" signals with a neural language model," 2024. [Online]. Available: <https://arxiv.org/abs/2410.18988>
- [3] S. A. Daimi and A. Iqbal, "A scalable data-driven framework for systematic analysis of sec 10-k filings using large language models," 2024. [Online]. Available: <https://arxiv.org/abs/2409.17581>
- [4] H. Ni, S. Meng, X. Chen, Z. Zhao, A. Chen, P. Li, S. Zhang, Q. Yin, Y. Wang, and Y. Chan, "Harnessing earnings reports for stock predictions: A glora-enhanced llm approach," in *2024 6th International Conference on Data-driven Optimization of Complex Systems (DOCS)*, 2024, pp. 909–915.
- [5] K. Papasotiriou, S. Sood, S. Reynolds, and T. Balch, "Ai in investment analysis: Llms for equity stock ratings," in *Proceedings of the 5th ACM International Conference on AI in Finance*, ser. ICAIF '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 419–427. [Online]. Available: <https://doi.org/10.1145/3677052.3698694>
- [6] A. Araci, "Finbert: Financial sentiment analysis with pre-trained language models," *arXiv preprint arXiv:2006.08097*, 2020.
- [7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.
- [8] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 1877–1901.
- [9] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.
- [10] S. Mehtab, J. Sen, and A. Dutta, "Stock price prediction using machine learning and lstm-based deep learning models," in *Machine Learning and Metaheuristics Algorithms, and Applications*, S. M. Thampi, S. Piramuthu, K.-C. Li, S. Berretti, M. Wozniak, and D. Singh, Eds. Singapore: Springer Singapore, 2021, pp. 88–106.
- [11] X. Zhang, Y. Li, and M. Zhang, "Financial news predicting stock trends using cnn," in *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2019, pp. 1398–1403.
- [12] E. F. Fama, "Efficient capital markets: A review of theory and empirical work," *The Journal of Finance*, vol. 25, no. 2, pp. 383–417, 1970.
- [13] R. J. Shiller, "From efficient markets theory to behavioral finance," *Journal of Economic Perspectives*, vol. 17, no. 1, pp. 83–104, 2003.