Ne Zha Buys, Tang Seng Waits: AI-Powered SEC Filing Detective

Anonymous ACL submission

Abstract

Predicting stock movements from financial disclosures remains challenging due to noisy market signals and sparse supervision. We construct a large-scale dataset of over 25,000 SEC filings (10-K and DEF 14A), aligned with daily stock prices and economic indicators for S&P 500 companies from 2000–2024. We formulate a three-class classification task (Up, Down, Stable) based on a 7-day input window, and compare model performance under two regimes: an unbalanced setting with a $\pm 2\%$ stability threshold, and a more balanced one at $\pm 0.5\%$. Deep models like GRUs and Transformers tend to collapse to the majority class, while XGBoost and SGD with RBF kernel outperform in the unbalanced and balanced settings, respectively. We also incorporate a Retrieval-Augmented Generation (RAG) chatbot for querying filings and generating grounded explanations. Our results highlight the robustness of combining traditional models with static textual features for financial trend prediction and document understanding.

1 Introduction

The S&P 500 index, which tracks 500 large-cap U.S. companies, serves as a key barometer of the U.S. economy (Reiff, 2025). Predicting stock movement often requires analyzing both historical price trends and SEC-mandated disclosures such as Form 10-K (annual financial summaries) and DEF 14A (proxy statements). These filings are long and complex, making timely analysis difficult—especially for non-experts.

This paper presents a multi-modal system that combines stock and macroeconomic time-series data with textual SEC filings to predict short-term stock movement following disclosure events. Our system also supports natural language explanations via a Retrieval-Augmented Generation (RAG) module to improve interpretability.

Our contributions are threefold:

- **Multi-Modal Dataset:** We compiled over 25,000 SEC filings from 2000–2024, aligned with daily stock prices and monthly economic indicators (CPI, inflation) for S&P 500 firms.
- **Model Benchmarking:** We evaluated stock trend prediction as a three-class classification task (Up, Down, Stable), training models on structured stock and economic features, both with and without Doc2Vec embeddings from SEC filings. We assess performance under two threshold regimes: unbalanced ($\pm 2\%$) and balanced ($\pm 0.5\%$).
- **RAG-Based Interpretability:** We developed a Retrieval-Augmented Generation chatbot that provides document-grounded answers to user queries over 10-K and DEF 14A filings.

Our system provides an interpretable, multi-source framework for understanding stock movement around financial disclosure events.

2 Related Works

Our work draws on advances in document embeddings, sequential modeling, and retrieval-based NLP. Doc2Vec (implemented via Gensim: version 4.3.3, LGPL 2.1 License, `https://radimrehurek.com/gensim/`) (Le and Mikolov, 2014) extends Word2Vec (Mikolov et al., 2013) by introducing a learned document-level embedding that captures global semantics, making it suitable for encoding long-form SEC filings. For modeling time-series data, we adopted Gated Recurrent Unit (GRU) (Chung et al., 2014), a simplified variant of Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) that offers faster training and competitive performance, particularly effective on short sequences such as 7-day stock windows. To enhance interpretability, we integrated Retrieval-Augmented Generation (RAG) (Lewis

et al., 2021), which grounds LLM outputs in retrieved document segments. Our implementation uses the LlamaIndex framework (version 0.12.19, MIT License, https://github.com/run-llama/llama_index) (Liu, 2022) to support indexing and querying of financial filings.

3 Methodology

Our framework consists of three components: (1) a document encoder that embeds 10-K and DEF 14A filings using Doc2Vec pretrained on 1,000 historical filings from 2000–2024; (2) a stock movement classifier that predicts up, down, or stable using 7-day (or 30-day) windows of stock prices, volume, CPI, and inflation data, optionally augmented with Doc2Vec embeddings; and (3) a RAG-based interpretability module that retrieves relevant filing segments and generates grounded natural language answers to user queries using a pre-trained LLM. We evaluated Gated Recurrent Units (GRU), Transformers (Vaswani et al., 2023), eXtreme Gradient Boosting (XGBoost) (version 3.0.0, Apache-2.0 License, https://github.com/dmlc/xgboost) (Chen and Guestrin, 2016), Perceptron, Multi-Layer Perceptron, Stochastic Gradient Descent with Radial Basis Function (SGD+RBF) (Rahimi and Recht, 2007), Stochastic Gradient Descent with Polynomial Kernel Approximation (SGD+Poly), Passive Aggressive Classifier (PA) (Shalev-shwartz et al., 2003), and Gaussian Naive Bayes (GNB) (Rish, 2001). We use scikit-learn library (version 1.6.1, BSD 3-Clause License, https://scikit-learn.org/stable/) (Pedregosa et al., 2018) access these algorithms.

4 Evaluation

4.1 Dataset

We constructed a multi-source dataset (2000–2024) comprising ∼25,000 SEC filings (10-K and DEF 14A) (via yfinance: version 0.2.59, Apache License, https://github.com/ranaroussi/yfinance), daily stock data for ∼500 S&P 500 companies (via sec-api: version 1.0.32, MIT License, https://github.com/janlukasschroeder/sec-api-python), and monthly CPI and inflation data. Filings were aligned to stock timelines by tagging each day with available document paths. We used ∼1,000 filings to pretrain a Doc2Vec model (size=384, epochs=20) due to compute constraints. 347 stock CSV files are used for training, 150 stock CSV files are used for testing.

4.2 Experimental Setup

We evaluated models under two regimes based on 1-day change of closing price: an **unbalanced setting** using a $\pm 2\%$ threshold to define up, down, or stable classes, and a **balanced setting** using a tighter $\pm 0.5\%$ threshold to reduce class imbalance. In the unbalanced setting, we test XGBoost, GRU, and Transformer models, both with and without Doc2Vec embeddings (note: Transformer is only tested with Doc2Vec). In the balanced setting, we evaluated Perceptron, Multi-Layer Perceptron, Stochastic Gradient Descent with Radial Basis Function (SGD+RBF), Stochastic Gradient Descent with Polynomial Kernel Approximation (SGD+Poly), Passive Aggressive Classifier (PA), and Gaussian Naive Bayes (GNB).

4.3 Model Hyperparameters & Training Details

Table 1 summarizes the key hyperparameters used in our experiments. *Due to resource constraints, we did not perform extensive hyperparameter tuning. Values (e.g., GRU hidden size, XGBoost depth, RBF/Poly kernel parameters) were selected based on common practice and prior experience with time-series and classification tasks.* For example, we use a larger GRU hidden size when Doc2Vec embeddings are present. For SGD-based kernel approximations, we follow conventional settings such as $\gamma = 0.1$.

Model	Key Parameters
GRU	num_layers=2, batch=32, hidden_size=64 / 128
Transformer	nhead=4, num_layers=2, hidden_size=64, batch=32
XGBoost	max_depth=6, eta=0.1, n_estimators=100
SGD + RBF	gamma=0.1, loss="log_loss"
SGD + Poly	degree=3, coef0=1, loss="log_loss"
Perceptron	max_iter=1000, warm_start=True
PassiveAggressive	loss="hinge", max_iter=1000, warm_start=True
GNB	default scikit-learn settings

Table 1: Model configurations used in our experiments.

We used torch (version 2.6.0) (Paszke et al., 2019) for model implementation and training, pandas (version 2.2.3) (Mckinney, 2011) for data preprocessing, and scikit-learn (version 1.6.1), (Pedregosa et al., 2018), for model evaluation and dataset spliting.

All experiments were conducted on a MacBook with an M1 GPU. Most training runs (e.g., XGBoost, SGD, MLP etc.) completed in under one hour. Training GRU and Transformer models without Doc2Vec embeddings took approximately 6–8 hours each. When Doc2Vec embeddings were included, GRU training extended to nearly one week due to increased input dimensionality and data sparsity.

Model	Class	Precision	Recall	F1-Score
GRU	Stable	0.76	1.00	0.86
	Up	0.00	0.00	0.00
	Down	0.19	0.00	0.00
GRU + Doc2Vec	Stable	0.76	1.00	0.86
	Up	0.00	0.00	0.00
	Down	0.00	0.00	0.00
XGBoost	Stable	0.77	1.00	0.87
	Up	0.54	0.02	0.05
	Down	0.58	0.04	0.08
XGBoost + Doc2Vec	Stable	0.80	0.88	0.84
	Up	0.26	0.16	0.20
	Down	0.25	0.18	0.21
Transformer	Stable	0.76	1.00	0.86
	Up	0.00	0.00	0.00
	Down	0.00	0.00	0.00

Table 2: Per-class precision, recall, and F1 scores on the **unbalanced (±2%) setting**.

Model	Class	Precision	Recall	F1-Score
MLP + Doc2Vec	Stable	0.00	0.00	0.00
	Up	0.38	1.00	0.55
	Down	0.00	0.00	0.00
Perceptron + Doc2Vec	Stable	0.28	0.91	0.43
	Up	0.39	0.03	0.06
	Down	0.35	0.07	0.11
SGD + RBF + Doc2Vec	Stable	0.28	0.08	0.12
	Up	0.37	0.66	0.48
	Down	0.34	0.26	0.30
SGD + Poly + Doc2Vec	Stable	0.28	0.13	0.17
	Up	0.38	0.25	0.30
	Down	0.34	0.62	0.44
GNB + Doc2Vec	Stable	0.28	0.99	0.44
	Up	0.39	0.00	0.01
	Down	0.38	0.01	0.01
PA + Doc2Vec	Stable	0.29	0.80	0.42
	Up	0.38	0.18	0.24
	Down	0.36	0.04	0.08

Table 3: Per-class precision, recall, and F1 scores on the **balanced (±0.5%) setting**, using Doc2Vec features.

4.4 Analysis

Our evaluation results (based on a single run with seed 42), as summarized in Tables 2 and 3, reveal several key findings across both unbalanced and balanced settings:

- **Class Imbalance Remains a Major Challenge:** Under the unbalanced regime (±2% threshold), sequence-based models like GRU and Transformer completely collapse to predicting the dominant *Stable* class. We tested multiple thresholds (0.5%, 1%, 2%) and weighted losses for GRU, but none improved minority-class recall. Transformer was only evaluated in its default configuration and already exhibited mode collapse.

- **Doc2Vec Can Hurt Sequential Models:** Adding Doc2Vec embeddings to GRU exacerbates its failure by fully collapsing to the *Stable* class. This suggests that injecting high-dimensional, unfiltered document representations may overwhelm the GRU's capacity to learn temporal patterns. Due to Transformer's weak base performance, we did not test it with Doc2Vec.

- **XGBoost Performs Best Under Imbalance:** Among all models tested in the unbalanced regime, XGBoost stands out for its robust handling of class imbalance. Although recall for minority classes is still low, its precision on *Up* (0.54) and *Down* (0.58) indicates that when the model does make such predictions, they tend to be correct.

- **XGBoost + Doc2Vec Enhances Recall:** Incorporating Doc2Vec improves XGBoost's recall on minority classes substantially (*Up*: 2% $\rightarrow$ 16%, *Down*: 4% $\rightarrow$ 18%), though precision decreases. This trade-off leads to higher overall F1-scores, validating the benefit of document embeddings for capturing rare but meaningful events.

- **SGD + RBF Remains Strongest in Balanced Regime:** In the balanced regime (±0.5% threshold), we switch to a 30-day lookback window to provide more historical context. Here, Stochastic Gradient Descent with RBF kernel approximation achieves the highest per-class F1-scores and maintains a solid balance between precision and recall across all classes.

It demonstrates strong generalization despite using static embeddings and partial-fit training.

- **SGD + Poly Offers Competitive Alternative:** While slightly trailing SGD + RBF, the polynomial kernel variant of SGD performs well, especially on the *Down* class (F1 = 0.44), highlighting the potential of non-linear feature transformations when combined with document-based features.
- **PA and Perceptron Are Moderately Competitive:** Passive-Aggressive (PA) and Perceptron classifiers both outperform GNB and MLP. While their F1-scores are lower than those of SGD variants, they still demonstrate non-trivial recall on minority classes. Notably, Perceptron shows moderate precision across all classes, though its recall remains limited.
- **GNB Fails to Generalize Despite High Precision:** Gaussian Naive Bayes achieves high recall on the *Stable* class (99%) and moderate precision on *Up* (0.39) and *Down* (0.38), yet its overall F1-score on minority classes remains near zero. This suggests over-reliance on strong priors and inadequate capacity to generalize under label imbalance.
- **MLP Still Underperforms Despite Extensive Tuning:** Despite testing various hidden sizes (256–1024), layer depths (2–4), dropout, batch normalization, and weighted loss, MLP consistently collapses to predicting *Up* only, and entirely fails on other classes. This suggests either overfitting or a lack of sufficient inductive bias to extract patterns from the input space.
- **Limited Temporal Signal in Stock Prices:** Sequential models such as GRU and Transformer show no significant advantage, reinforcing the hypothesis that short-term stock price movements are dominated by exogenous market events. This supports the use of static, feature-based models over temporal architectures for this particular prediction task.

These insights highlight the difficulty of modeling financial time series using sparse and imbalanced labels, and emphasize the importance of partial-fit support, feature engineering, and embedding quality when working with real-world financial disclosures.

4.5 RAG Demonstration

To support interpretability, we developed a Retrieval-Augmented Generation (RAG) prototype using LlamaIndex and LLaMA 3. The system enables users to query SEC filings and receive grounded natural language responses. While it supports both 10-K and DEF 14A filings, we present one example based on a 10-K:

> *Analyze this 10-K filing for financial signals that may affect stock movement. Consider revenue, profitability, debt, risks, strategies, and industry trends. Provide a sentiment score between -1 and 1 with justification.*

The RAG system returned (abridged):

> **Positives:** Holding gains suggest revenue stability; EPS reflects profitability.
> **Negatives:** $250M sales drop; EPS down 10 cents; risk disclosures raise concerns.
> **Sentiment Score:** 0.2 — slightly positive due to stronger financial indicators.

Though not used during training due to time constraints, this RAG module complements our classifiers by offering interpretable, document-grounded feedback for end-users.

5 Conclusion and Future Work

We proposed a hybrid framework for financial trend prediction that integrates time-series stock data, economic indicators, and document embeddings from SEC filings. Experimental results showed that deep sequence models (GRU, Transformer) struggled to generalize across class imbalance, while non-deep models like XGBoost and Stochastic Gradient Descent (SGD) with kernel approximations yielded stronger performance—especially when combined with Doc2Vec features. We also implemented a RAG-based chatbot to enhance interpretability via grounded responses to filing queries.

Looking ahead, future work could explore hybrid agentic architectures that delegate time-series prediction, document retrieval, and decision-making to specialized modules. Additionally, incorporating more frequent textual sources (e.g., financial news, earnings calls) may mitigate the sparsity introduced by infrequent SEC filings. Finally, class imbalance remains a major challenge—calling for techniques like focal loss, class reweighting, or resampling to improve generalization on minority classes.

Ethical Considerations and Societal Impact

This work is designed to enhance interpretability in financial decision-making by providing both statistical predictions and document-grounded explanations. However, several ethical concerns arise. First, predictive systems may inadvertently reinforce existing inequalities. Second, retail investors might over-rely on model outputs without fully understanding the underlying uncertainties. Third, large-scale deployment could increase asymmetries between institutional and individual actors. We urge that such models be used as support tools—not decision-makers—and that users remain informed about their limitations.

Limitations

Despite the breadth of our dataset and methods, several important limitations remain:

1. **Sparse Filing Coverage:** Most trading days do not coincide with an SEC filing release, resulting in frequent missing (zero) embeddings. This limits the utility of document information in the classification task.

2. **Lossy Document Encoding:** We use Doc2Vec to encode entire filings into a single vector. Given the complexity of SEC documents, this inevitably compresses and potentially omits important semantic information.

3. **Unreliable Temporal Modeling:** Our experiments show that models such as GRU and Transformer fail to learn meaningful temporal patterns, likely due to the inherent randomness and volatility in stock price movements. This questions the effectiveness of sequential modeling for such financial tasks.

4. **Disjoint RAG and Prediction Models:** The RAG component, while useful for interpretability, operates independently from the predictive models. There is no mechanism to reconcile contradictions between them, nor to use retrieved insights during classification.

5. **Limited Model Exploration:** For some models (e.g., GNB, Transformer), we did not explore advanced tuning due to early poor performance or resource constraints. The possibility remains that better configurations could improve their results.

These limitations point to the need for deeper integration across modalities, more robust embedding techniques, and greater alignment between interpretability modules and prediction pipelines.

References

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794. ACM.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *Preprint*, arXiv:1412.3555.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9:1735–1780.

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. *Preprint*, arXiv:1405.4053.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Preprint*, arXiv:2005.11401.

Jerry Liu. 2022. LlamaIndex.

Wes Mckinney. 2011. pandas: a foundational python library for data analysis and statistics. *Python High Performance Science Computer*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *Preprint*, arXiv:1301.3781.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, and 2 others. 2019. Pytorch: An imperative style, high-performance deep learning library. *Preprint*, arXiv:1912.01703.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Andreas Müller, Joel Nothman, Gilles Louppe, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2018. Scikit-learn: Machine learning in python. *Preprint*, arXiv:1201.0490.

Ali Rahimi and Benjamin Recht. 2007. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc.

Nathan Reiff. 2025. The top 25 stocks in the s&p 500. https://www.investopedia.com/ask/answers/08/find-stocks-in-sp500.asp. Accessed: 2025-04-04.

Irina Rish. 2001. An empirical study of the naïve bayes classifier. *IJCAI 2001 Work Empir Methods Artif Intell*, 3.

Shai Shalev-shwartz, Koby Crammer, Ofer Dekel, and Yoram Singer. 2003. Online passive-aggressive algorithms. In *Advances in Neural Information Processing Systems*, volume 16. MIT Press.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need. *Preprint*, arXiv:1706.03762.