Guba Emotion

AI6130 Final Project Report

GitHub Repo: https://github.com/genehhz/AI6130-Large-Language-Model-Project

Lin Yue (LINY0145@e.ntu.edu.sg)
Ma Hong (HONG032@e.ntu.edu.sg)
Salinelat Teixayavong (salinela.001@e.ntu.edu.sg)
Jiang Kexun (kexun001@e.ntu.edu.sg)
Pan Feng (Feng006@e.ntu.edu.sg)
Eugene Ho (EHO010@e.ntu.edu.sg)

The Chinese A-share market presents a unique case for financial analysis. Unlike mature markets, it is characterized by a high concentration of retail investors whose trading is often sentiment-driven, resulting in high volatility and frequent deviations from fundamental value. Consequently, traditional forecasting models rooted in the Efficient Market Hypothesis (EMH) are ill-equipped to handle these behavioral dynamics. To address the limitation, we developed a multi-modal prediction model that integrates social media sentiment with traditional financial market indicators. Our framework analyzes textual data from the Eastmoney Stock Forum using a FinBERT-Tone-Chinese model, which we fine-tuned on a domain-specific financial corpus. This sentiment analysis is then combined with technical market data to forecast stock movements days ahead. Our empirical analysis confirms two key findings. First, the domain-specific FinBERT-Tone-Chinese model substantially surpasses the accuracy of general-purpose financial BERT models in sentiment classification. Secondly, after completing the full feature engineering pipeline, all features were fed into three models, the Vanilla Transformer, PatchTST, and LightGBM, to perform a 3 classes classification task ($c = 3$), predicting whether the next day's stock movement would go up, remain neutral, or go down. LightGBM achieved the strongest performance overall, outperforming both the Vanilla Transformer and PatchTST, especially in the low signal-to-noise environment. In summary, this research confirms that incorporating behavioral sentiment data with financial indicators is an effective strategy to improve predictive accuracy in emerging markets driven by sentiment.

1. Introduction

The integration of social media analytics into financial forecasting has gained considerable momentum in recent years, as researchers recognize the predictive value embedded in investor discussions and sentiment expressions.[1] In China, the rapid growth of online financial communities has created rich datasets of retail investor opinions, with platforms like Guba serving as primary venues for stock-related discussions. These platforms offer unique insights into the collective psychology of Chinese retail investors, who represent a substantial portion of market participants and whose sentiment can significantly influence price movements.[2]

China's A-share market plays a key role in global finance but differs notably from mature markets in structure and investor behavior. A defining trait is its retail investor dominance, where despite a rise in institutional participation, individuals still made up approximately 55% of trading volume in 2019.[2,3] This retail-dominated investor structure drives high volatility and turnover rates, as individual investors are more prone to emotions, rumors, and short-term speculation, frequently leading to price deviations from intrinsic value.[2,3] Given participants' strong sensitivity to information flows, sentiment shifts are quickly and sharply reflected in prices, creating complex nonlinear relationships between investor mood and market movements. Traditional forecasting models grounded in the Efficient Market Hypothesis (EMH) often fail to capture any fluctuations that could be driven by behavioral biases and emotional responses, thus underscoring the value of incorporating alternative data sources.[4] This is especially so for social media signals that could directly reveal investor sentiment and psychological states.[5]

Despite growing interest in financial sentiment analysis, research has predominantly concentrated on Western markets and English-language sources, creating a substantial gap in understanding Chinese market dynamics. Additionally, conventional statistical methods such as ARIMA and linear regression models face inherent limitations when modeling the behavior-driven price movements characteristic of retail-dominated markets. These traditional approaches assume market efficiency and struggle to capture the emotional volatility and nonlinear patterns prevalent in Chinese equity markets. The potential of combining granular sentiment measures with linguistic features and traditional technical indicators through advanced machine learning architectures has not been fully explored, particularly in the context of Chinese equity markets where behavioral factors play an outsized role.

This study investigates whether incorporating social media sentiment and linguistic information can improve stock price movement predictions for Chinese exchange-listed securities. We extract data from Guba discussion threads, compute sentiment valence scores from user posts, and develop a multivariate forecasting model that synthesizes three distinct information sources: historical price movements, quantified sentiment measures, and textual embeddings. Our approach builds upon methodologies demonstrated in recent financial NLP research,[6] adapting them to the unique characteristics of Chinese financial social media and the behavioral dynamics of retail-dominated markets.

Our work makes the following contributions:

- We provide empirical analysis of sentiment patterns la-

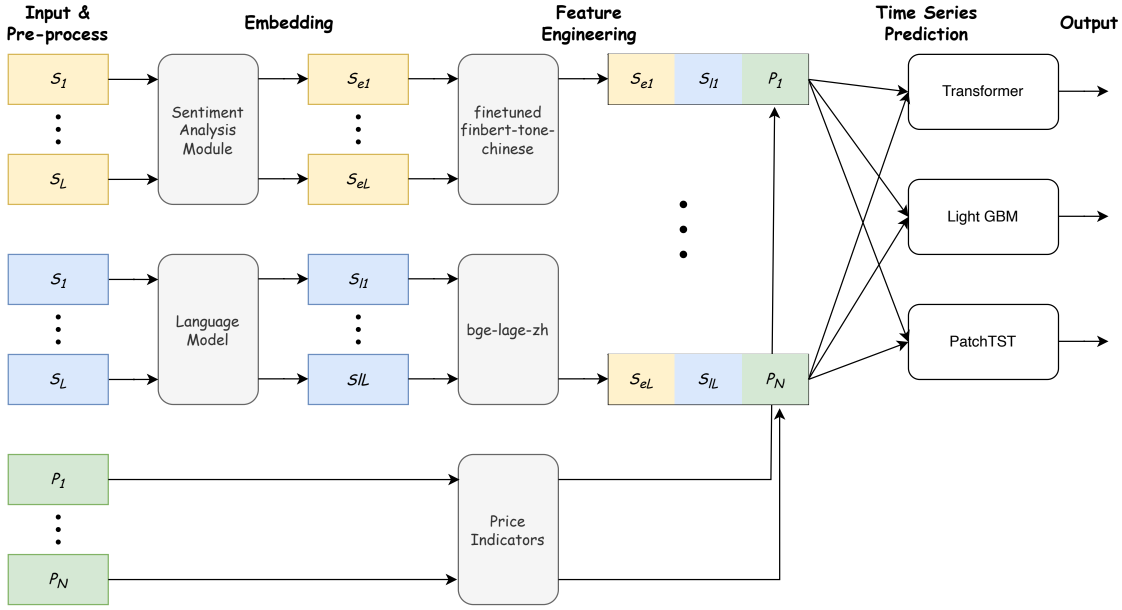

Fig. 1. LLM Data Processing and Modelling Pipeline

tent in posts and comments posted in the Guba platform, offering new perspectives on Chinese retail investor behavior and emotional responses to market events

- We develop a multi-modal, daily A-share price fluctuation prediction framework that integrates traditional market data with derived sentiment metrics and language representations for improved predictive accuracy
- We evaluate the marginal contribution of each feature type (price, sentiment, and text) to demonstrate their relative importance for forecasting stock price movements in the Chinese markets.

2. Related Works

Research related to this work can be grouped into three major areas: (1) financial sentiment analysis and domain-adapted language models, (2) sentiment-driven dynamics in the Chinese A-share market, and (3) multi-modal predictive models that integrate text, sentiment, and market data. This section reviews the most relevant studies and highlights the gaps that motivate our approach.

2.1. Financial Sentiment Analysis and Domain-Specific Language Models

Financial sentiment analysis has advanced substantially with the emergence of transformer-based models. While early work relied on lexicon-based methods or handcrafted rule systems, such approaches struggled with contextual nuances and domain-specific terminology. In contrast, models such as FinBERT, FinBERT-Tone, and Financial RoBERTa demonstrated that pretraining on large-scale financial corpora significantly improves performance on sentiment polarity detection, tone classification, and the identification of subtle linguistic cues in formal financial documents.

However, the majority of these models are trained on English news or analyst reports, leaving Chinese-language research comparatively underdeveloped. More importantly, Chinese financial discussion forums are characterized by informal, colloquial, and emotionally charged language that differs markedly from the formal style of financial reports. Existing Chinese financial BERT variants generally lack exposure to this specific communication style, consequently exhibiting poor generalization to retail-investor-generated content. This gap underscores the necessity for domain adaptation specifically targeted at stock forum discourse, which directly motivates our development of a fine-tuned FinBERT-Tone-Chinese sentiment model.

2.2. Sentiment and Behavioral Dynamics in the Chinese A-Share Market

The Chinese A-share market is distinguished by its high proportion of retail investors and demonstrably sentiment-driven trading patterns. Prior studies have shown that investor sentiment extracted from platforms like Eastmoney's Guba or social media channels correlates with key market dynamics, including price fluctuations, volatility clustering, and short-term reversals. These platforms therefore represent a valuable source for capturing the collective psychology of retail participants.

Despite this potential, existing sentiment-based studies typically rely on coarse aggregated signals—such as positive–negative post ratios, posting volume, or keyword frequency. These signals are often insufficient for capturing finer-grained emotional dynamics or complex linguistic subtleties. Moreover, few works incorporate temporally aligned sentiment signals at the stock-day level, and fewer still explore multiple sentiment modalities (e.g., emotion polarity, activity volume, sentiment momentum) simultaneously. This limitation in existing research—namely, the lack of granular, stock-specific sentiment modeling—motivates our construction of a multi-modal dataset that integrates multiple dimensions of investor sentiment aligned with daily market outcomes.

2.3. Text Embeddings and Multi-modal Models for Stock Prediction

Beyond sentiment labels, high-dimensional text embeddings have emerged as a critical tool for modeling semantic and contextual information in financial texts. Sentence-level embedding models, including Sentence-BERT, SimCSE, and the more recent Chinese embedding family (e.g., BAAI's bge-large-zh),[7] offer expressive representations that capture deep semantic structure. However, their application in Chinese equity markets remains limited, particularly in the context of retail investor discussions, where domain mismatch and noisy language impede their effectiveness.

In parallel, multi-modal learning approaches—integrating price indicators, sentiment scores, and textual embeddings—have garnered significant interest in financial forecasting. While traditional machine learning models (e.g., gradient boosting machines) remain competitive, particularly with robust feature engineering, deep learning approaches such as LSTMs and Transformers aim to learn temporal dependencies end-to-end. Although time-series transformers like PatchTST have recently achieved state-of-the-art performance on benchmark forecasting tasks, their application in equity prediction and multi-modal fusion scenarios remains nascent.

3. Methodology and Experimental Design

Figure 1 shows the overall pipeline of our proposed framework. After getting stock and text data (Section **3.1.**), feature engineering (Section **3.3.**) is applied to extract deeper information of given data. Then we modelling based on existing data (Section **3.4.**) and predict the rise/fall of the stock.

3.1. Data Acquisition and Preprocessing

Our data sources consist of both text and market data. For text data, using the existing "Guba" dataset that only has post titles, we massively crawled all related comments as supplementary data on the Eastmoney Stock Forum "Guba" regarding the constituent stocks of the CSI 300 indices. We chose "Guba" as the data source because it is China's largest and most influential stock forum, providing the best window for observing retail investor sentiment.[8] For market data, we retrieved daily opening price, highest price, lowest price, closing price, volume (OHLCV) and turnover data for the same period and stock range as the text data by AKShare library.

The preprocessing workflow involved text cleaning, merging, and time alignment. A systematic cleaning and filtering process was performed to remove common noise from stock forum texts, such as URLs, stock codes, meaningless special characters, and emoticons. To maximize contextual information, the title of each post and all its comments were merged into a single document; this approach helps to more accurately determine the overall sentiment. To ensure the timeliness of information, all text data are grouped into trading-time text and closing-time text. All posts and comments published for a specific stock within a single trading day's time window (e.g., from the close of trading on day T-1 to the open of trading on day T) are aggregated to predict stock price movements on day T.

3.2. Sentiment Fine-tuning

The objective of this module is to create a high-precision sentiment classifier tailored to the unique Chinese financial social media context of "Stock Forum". The selected base model is `yiyanghkust/finbert-tone-chinese`, which is based on `bert-base-chinese`, fine-tuned on ˜8000 analyst reports, and specifically designed for Chinese financial sentiment analysis with a strong domain foundation. This base model was then fine-tuned on 3000 manually annotated Stock Forum posts to adapt to the colloquial and informal language paradigm. The fine-tuning process tuned all parameters and optimized with cross-entropy loss, resulting in a customized model capable of accurately judging the sentiment (positive, negative, neutral) of Stock Forum posts.

3.3. Feature Engineering

3.3.1. Price Indicators

We selected six key technical indicators derived from market data to serve as features. These indicators are chosen to capture different aspects of market dynamics, including core price movement, momentum, trend, volatility, and volume.

The core indicators include **Log Return** (log(PctChange)), which measures daily revenue and possesses good statistical properties, and **Amplitude**, which measures the range of intraday price fluctuations. For momentum, the **RSI (Relative Strength Index)** is used to identify overbought or oversold conditions and capture mean reversion signals. It is calculated as:

$$\text{RSI} = 100 - \left(\frac{100}{1 + RS}\right)$$

The **MACD (Moving Average Convergence Divergence)** is a trend indicator that helps identify the strength, direction, and momentum of a trend, defined by the difference between two Exponential Moving Averages (EMAs):

$$\text{MACD} = EMA_{12}(\text{Close}) - EMA_{26}(\text{Close})$$

To measure volatility, the **Bollinger Band Width** quantifies market contraction or expansion status by normalizing the distance between the bands:

$$\text{BBW} = \frac{\text{UpperBand} - \text{LowerBand}}{\text{MiddleBand}}$$

Finally, **OBV (On-Balance Volume)** is a volume indicator that uses changes in trading volume to measure buying and selling pressure and confirm price trends. Its cumulative value is calculated as:

$$OBV_t = OBV_{t-1} \pm \text{Volume}_t$$

3.3.2. Sentiment Indicators

To capture sentiment dynamics relative to market activity, all text-based indicators are computed separately for two periods: trading hours and closing hours. The primary metric is the **Emotion Index**, a normalized measure quantifying the overall sentiment polarity we gained insight,[9] defined as:

$$\text{Emotion Index} = \frac{\text{Total\_Positive} - \text{Total\_Negative}}{\text{Total\_Positive} + \text{Total\_Negative}}$$

In addition to this index, we include **Activity Volume** metrics, specifically the Total_Posts_Count and Total_Click_Count, to measure user engagement. Finally, **Emotion Momentum** is calculated as the 3-day or 5-day moving average of the Emotion Index, designed to capture evolving trend changes in investor sentiment.

3.3.3. Text Embeddings

In addition to the derived sentiment indicators, we generated high-dimensional semantic features directly from the text. We utilized the bge-large-zh-v1.5 model[7] from BAAI (Beijing Academy of Artificial Intelligence) to create sentence embeddings. This model was applied to each preprocessed document (the merged title and comments), converting the raw text into a dense vector representation that captures its semantic meaning as part of our feature engineering.

3.4. Modelling

3.4.1. Vanilla Transformer with Encoder

The Vanilla Transformer implemented in this study adopts a three-modality architecture designed to integrate heterogeneous but complementary sources of information. The model jointly learns from stock price features, emotion-based features derived from sentiment signals, and high-dimensional text features. Each modality contributes distinct temporal cues relating to market movements, and the integrated framework allows the model to capture interactions spanning technical signals, market sentiment, and semantic information embedded in textual data.

The model receives three parallel input sequences of identical temporal length: a 6-dimensional stock feature sequence, a 10-dimensional emotion sequence, and a 2048-dimensional text feature sequence. Each sequence spans a fixed sliding window of historical days. Because these modalities differ greatly in scale and statistical characteristics, each sequence is first projected into a common latent space of dimension 64 through separate projection networks. The stock sequence is processed through a linear transformation followed by ReLU and LayerNorm;[10] the emotion features undergo LayerNorm before projection to stabilise input variance; and the text features are compressed through a two-stage reduction from 2048 to 256 and finally to the shared projection dimension, with ReLU and LayerNorm applied to preserve representational quality. This projection step ensures that all modalities are expressed within a comparable embedding space, enabling effective multi-modal fusion.

After projection, the three embeddings are concatenated at each time step to form a 192-dimensional fused vector, which is subsequently passed through a fusion module that maps the combined signal to a 256-dimensional hidden representation using a linear transformation with ReLU, dropout, and LayerNorm. To retain temporal ordering within the sliding window, the model adds a learnable positional embedding rather than the fixed sinusoidal encoding used in the original Transformer.[11] The use of learnable positional parameters is consistent with subsequent Transformer-based models such as BERT,[12] allowing the network to adaptively distinguish different positions within the sequence.

The fused and position-enhanced sequence is then processed by a stack of two Transformer encoder layers,[11] each consisting of multi-head self-attention with four attention heads and a feed-forward sublayer with expanded dimensionality (set to four times the hidden dimension). The encoder adopts a pre-norm configuration,[13] which generally provides more stable optimisation behaviour. Through the self-attention mechanism, the model learns dynamic temporal dependencies across the entire sequence, capturing both short-range fluctuations and longer-term contextual patterns essential for modelling market dynamics.

After passing through the encoder, the sequence of hidden states is aggregated using an attention-based temporal pooling mechanism. Instead of relying on a simple average or selecting the final time step, the attention pooling assigns learnable importance weights to each time step, enabling the model to emphasise the most informative periods in the historical window. Attention pooling follows the general formulation of self-attentive pooling.[14] The weighted sum of hidden states produces a fixed-dimensional representation of size 256, which is subsequently fed into a two-layer classification head that outputs the three target classes defined by the prediction task.

To promote stable training and mitigate overfitting, several regularisation strategies were incorporated. These include dropout within the fusion and feed-forward layers, weight decay applied through the AdamW optimiser,[15] and a training schedule of 100 epochs with a small learning rate. Class weighting was also introduced in the loss function to correct for imbalanced class distributions in the training data.

It is important to note that the implemented baseline is an encoder-only architecture. The model does not include a decoder component, as the task is a classification problem rather than a sequence-to-sequence generation task. Encoder-only transformers are well suited for representation learning in settings where the objective is to map an input sequence to a fixed-dimensional output, which aligns with the design of this study.

To train the model, we employ class-weighted cross-entropy loss, AdamW optimisation,[15] and early stopping to prevent overfitting. The key training hyperparameters used in the experiments are summarised in Table 1.

Table 1. Transformer Hyperparameters

Hyperparameter	**Value**
Window size	30
Batch size	64
Projection dimension	64
Fused dimension	512
Learning rate	3×10^{-5}
Weight decay	1×10^{-4}
Dropout (global)	0.10
Gradient clipping	1.0
Epochs	100
Early stopping	Enabled
Label smoothing	0.10
Class weights	[0.839, 1.2833, 0.9719]

3.4.2. PatchTST

The PatchTST model implemented in this project uses a pretrained granite PatchTST backbone,[16] originally trained on the EETh1 dataset, to perform multi-modal time-series classification using 2 complementary input streams: technical stock indicators and emotion-based sentiment features. Similar to the Vanilla Transformer setup, each daily record is indexed by stock and date and contains 4 emotion variables (closing and trading emotion indices and their 3-day momentum) and 2 stock features (log return and amplitude), together with a 3-class next-day label. The model leverages the contrast between realised price movements and behavioural market sentiment, while delegating temporal pattern extraction to the pretrained PatchTST encoder.

Temporal samples are generated using a 90-day sliding window applied independently to each stock. Each window produces an aligned pair of pricing and emotion sequences, which are fed directly into the model.

Both modalities are projected into a 128-dimensional latent space using lightweight projection networks. The stock branch applies a linear projection followed by ReLU and LayerNorm, while the emotion branch applies LayerNorm before and after projection to stabilise its higher-variance statistics. The 2 projected embeddings are concatenated at each time step and passed through a fusion module that maps the resulting 256-dimensional vector to a 512-dimensional fused representation with ReLU and LayerNorm. A channel-alignment layer then reduces this 512-dimensional fused vector to the 7 input channels expected by the pretrained PatchTST backbone.

The pretrained `PatchTSTModel` performs the core temporal modelling. All backbone parameters are initially frozen to stabilise early training while allowing the new layers to adapt. A lightweight Houlsby-Adapter module is applied to the PatchTST hidden states,[17] consisting of LayerNorm, a bottleneck projection, ReLU, dropout, and an up-projection added via a scaled residual connection. Temporal aggregation is performed using mean pooling applied across the encoder's output sequence, producing a fixed-length vector that is fed into a 2-layer classification head with dropout and ReLU before projecting to the 3 output classes. The hyperparameters selected for fine-tuning the PatchTST model are summarised in Table 2.

Table 2. PatchTST Hyperparameters

Parameter	**Value**
Window size	90
Batch size	64
Projection dimension	128
Fused dimension	512
Transformer backbone	https://huggingface.co/ibm-granite/granite-timeseries-patchtst
Learning rate (head)	1e-3
Learning rate (backbone)	5e-6
Weight decay	1e-3
Dropout (global)	0.10
Gradient clipping	1.0
Adapter	Houlsby Adapter
Epochs	60
Regularisers	**Value**
Label smoothing	0.10
MixUp α	0.2
MixUp probability	0.5
Class weighting	Yes

Training follows a 2-stage regimen: (1) a head-warmup phase in which only the projection, fusion, adapter, and classification layers are trained, and (2) a partial-unfreezing phase in which the last 2 backbone blocks are selectively unfrozen and optimised with a very small learning rate. Parameter groups are constructed to maintain separate learning rates and decay policies for backbone versus non-backbone layers. Optimisation uses AdamW with a strong head learning rate, a minimal backbone learning rate, weight decay, gradient clipping, and a warmup–cosine learning rate schedule.

3.4.3. LightGBM

Light Gradient Boosting Machine (LightGBM) is a high-performance, open-source gradient boosting decision tree (GBDT) framework designed for scalability and efficiency. Unlike traditional GBDT algorithms that grow trees level-wise, LightGBM utilizes a leaf-wise growth strategy, which selects the leaf with the maximum delta loss to split. This approach often leads to faster convergence and higher accuracy, though it requires careful tuning of parameters like `num_leaves` to prevent overfitting. To optimize performance, LightGBM incorporates two novel techniques: Gradient-based One-Side Sampling (GOSS) and Feature Bundling (EFB). GOSS focuses the training process by retaining all instances with large gradients (i.e., "hard" examples) and performing random sampling on instances with small gradients (i.e., "easy" examples). EFB efficiently reduces the feature space by bundling sparse, mutually exclusive features together. Combined with its use of a highly optimized histogram-based algorithm for finding optimal split points, LightGBM significantly reduces memory consumption and accelerates training speed, making it exceptionally well-suited for large-scale tabular datasets.

In our methodology for the financial prediction task, we employed this LightGBM framework as a representative traditional machine learning model. The model was trained on a set of engineered features to output the final classification categories. Given that LightGBM is inherently struc-

Table 3. Details of our multi-modal Guba-Emotion dataset

Component	Details
Stock Data	110k daily stock data
Stock Label	Up(34.5%), Down(39.3%), Flat(26.2%)
Text Data	7.3M posts with 8M crawled comments
Text Sentiment	Positive(32.8%), Negative(35.6%), Neutral(31.6%)
Time Span	January 1, 2023 to December 31, 2024.

Table 4. Fine-tuning Results

Epoch	Training Loss	Validation Loss	Accuracy
1	0.766	0.788	0.673
2	0.681	0.776	0.673
3	0.625	0.817	0.690

tured for tabular data, we adapted our time-series problem by setting the `window_size` to 1, effectively using only the most recent day's data for each prediction. The model's objective was configured for multiclass classification by setting the `objective` parameter to `'multiclass'` and specifying `num_class=3` to correspond to the target categories. To identify the most effective model configuration, the complete set of specific hyperparameters was rigorously optimized using the Optuna hyperparameter-tuning framework.

4. Experimental Results

4.1. Dataset Settings

Table 3 shows the key statistics of our dataset. Our multi-modal dataset contains 2 years' daily data from 230 companies in CSI300 index, resulting in 110k daily stock data and 7.3M posts. For the stock label's category, we choose a threshold of 0.5% change to classify the up, down and flat stocks. This threshold setting ensures a balanced classification in each class, making it easier for the model to generate a stable prediction. The text sentiment is also balanced, showing a promising result of our fine-tuned Guba-text sentiment analysis.

4.2. Sentiment Fine-tuning Results

We fine-tuned our Guba-text sentiment classification model based on finbert-tone-chinese,[18] and trained on 3k manually labeled samples using Huggingface's Trainer API. We set batch size to 16, trained with 3 epochs and a 0.01 weight decay. Table 4 shows the fine-tuning results per epoch, and we could see the decrease of loss and increase of accuracy after fine-tuning. The main issue of the base model is it focuses on financial reports, cannot recognize the casual words and slang in stock forums, so it simply classifies every sentence to neutral. As shown in Table 5, after fine-tuning, we could see the model is able to classify the sentence correctly, including correctly understand "awesome" as sarcasm, and "PingAn" is a company name other than a positive word (safety).

4.3. Evaluating Results on Modelling

To evaluate model performance comprehensively, we combined two types of metrics: traditional machine learning classification indicators and empirical financial indicators.

Table 5. Sentiment Classification Comparison before/after Fine-tuning

Content	Before	After
PingAn Bank is my new favorite.	Neutral	Neutral
Selling and leaving, securing profits.	Neutral	Negative
Last place, awesome.	Neutral	Negative
How about a limit-up?	Neutral	Postive
Trash PingAn.	Neutral	Negative

Table 6. General-Model Comparison Based on Machine Learning Metrics

Metric	Transformer	PatchTST	LightGBM
Up precision	0.35	0.37	0.36
Up recall	0.09	0.49	0.17
Up f1	0.15	0.42	0.23
Down precision	0.42	0.40	0.39
Down recall	0.69	0.26	0.74
Down f1	0.52	0.32	0.51
Flat precision	0.32	0.26	0.33
Flat recall	0.34	0.29	0.13
Flat f1	0.33	0.27	0.19
Accuracy	**0.39**	**0.35**	**0.38**

4.3.1. Machine Learning Metrics

We used standard classification measures: F1-Score, Accuracy, Recall, and Precision to assess model performance. For all three models: vanilla Transformer, PatchTST and LightGBM, all general models' results are similar on the combined 230 stocks dataset, as shown in Table 6.

However, when training individually on each stock, the fine-tuned LightGBM model shows the most promising results. As shown in Table 7, LightGBM achieves about 60% accuracies in its top 10 stocks, which is significantly higher than PatchTST's accuracies. We didn't fine-tune the vanilla Transformer model, since PatchTST is based on Transformer and has similar results.

4.3.2. Financial Metrics

For the model's financial value, we evaluate it via a long-short portfolio backtest. We choose the fine-tuned LightGBM model and backtest on the unseen December 2024's data. We use a very simple strategy: buy if the model predicts a rise, sell if the model predicts a fall. Figure 2 shows the cumulative returns, the highest stock return reaches about 10%, and the portfolio (same weight) reaches about 5%. The above results demonstrate the model's capability to mine alpha and generate return.

Table 7. PatchTST and LightGBM fine-tuned model Top 10 Stock Accuracies

PatchTST		LightGBM	
Stock Code	Accuracy	Stock Code	Accuracy
000301	0.52	688303	0.71
002460	0.52	601360	0.67
603833	0.52	600176	0.62
600023	0.50	688506	0.62
000338	0.50	002050	0.57
300433	0.48	002252	0.57
601799	0.48	002475	0.57
002252	0.47	002493	0.57
300059	0.47	300308	0.57
002179	0.47	301236	0.57

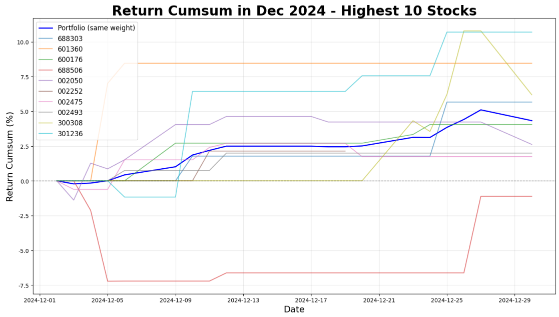

Fig. 2. Backtested Cumulative Returns in Dec 2024

5. Analysis and Discussion

Here we discuss the performance's difference between Transformer-based models and traditional machine learning models such as LightGBM. Surprisingly, LightGBM outperforms Transformers, and here is the possible reasons.

5.1. Extreme Signal-to-Noise Ratio

Financial sequences are dominated by random, unpredictable "noise." A Transformer, with its massive capacity, is designed to find complex patterns. When given a raw, noisy sequence, it dramatically overfits the noise, learning non-repeatable, false patterns. The LightGBM approach (using features) first reduces this noise via feature engineering (e.g., using moving averages which smooth data), making the problem easier.

5.2. Implicit vs. Explicit Signal Representation

The LightGBM path relies on Feature Engineering (e.g., Momentum, RSI, Volatility). These features are explicit signals derived from financial domain knowledge. LightGBM is state-of-the-art at selecting which of these strong, pre-processed signals are most predictive. The Transformer path gives the model a raw sequence and asks it to implicitly learn the concepts of "momentum" or "volatility" from scratch. This is a much harder task, and the model almost always gets lost in the noise (see **5.1.**) before it can discover these complex, subtle signals.

6. Conclusion

In conclusion, by fine-tuning the FinBERT-Tone-Chinese model on the Eastmoney Stock Forum corpus, we are able to achieve more accurate sentiment extraction specifically adapted to the Chinese financial discourse. The results demonstrate that domain-specific fine-tuning substantially enhances sentiment classification performance compared to general-purpose models, underscoring the value of contextual adaptation in financial natural language processing.

Three distinct predictive models were examined: LightGBM, a gradient boost model optimized for tabular data; the Vanilla Transformer, which captures long-range dependencies via self-attention; and PatchTST, a Transformer variant designed for local and long-term time-series modeling. Two modeling strategies were used: The first trained a general model on an aggregated dataset of 230 stocks to capture market-wide dynamics. The second, a general-to-specific approach, pre-trained models on the full dataset and fine-tuned them on individual stocks, allowing adaptation to unique, stock-specific behaviors.

Experimental results demonstrated that LightGBM consistently outperformed both Transformer-based models, particularly under extremely low signal-to-noise ratios on the market. Leveraging explicit engineered features such as momentum, volatility, and moving averages, LightGBM effectively reduced noise and achieved approximately 67% accuracy among the top-performing stocks. In contrast, Transformer models—when given raw, noisy sequences—tended to overfit non-repeatable patterns, highlighting their sensitivity to random market fluctuations.

From a practical point of view, the proposed framework may provide valuable insight for quantitative investors seeking new sources of alpha and for regulatory authorities monitoring sentiment-driven market fluctuations. Nevertheless, the study is limited by its use of daily-frequency data, which may miss finer intra-day sentiment variations, and by a limited exploration of model architectures. Future research will extend this work by incorporating high-frequency data, exploring graph-based neural networks (GNNs) to model cross-stock sentiment propagation, and conducting causal inference analyzes to elucidate the underlying mechanisms linking sentiment and asset price movements.

References

[1] Li, J., Bu, H., and Wu, J.: Sentiment-aware Stock Market Prediction: a Deep Learning Method, 2017 International Conference on Service Systems and Service Management, International Conference on Services Systems and Services Management, ICSSSM, None, 2017.

[2] Luo, C., Li, Z., and Liu, L.: Does Investor Sentiment Affect Stock Pricing? Evidence from Seasoned Equity Offerings in China, *National Accounting Review*, **3**, 1 (2021), pp. 115–136.

[3] Hilliard, J. and Zhang, H.: Size and Price-to-book Effects: Evidence from the Chinese Stock Markets, *Pacific-Basin Finance Journal*, **32**, C (2015), pp. 40–55.

[4] Pham, Q., Pham, H., Pham, T., and Tiwari, A. K.: Revisiting the Role of Investor Sentiment in the Stock Market, *International Review of Economics Finance*, **100** (2025), 104089.

[5] Xu, Y., Wang, J., Chen, Z., and Liang, C.: Sentiment Indices and Stock Returns: Evidence from China, *International Journal of Finance & Economics*, **28**, 1 (2023), pp. 1063–1080.

[6] Lee, J., Youn, H. L., Poon, J., and Han, S. C.: StockEmotions: Discover Investor Emotions for Financial Sentiment Analysis and Multivariate Time Series, AAAI-23 Bridge Program (AI for Financial Services),

[7] Xiao, S., Liu, Z., Zhang, P., and Muennighoff, N.: C-Pack: Packaged Resources to Advance General Chinese Embedding,

[8] Wang, G., Yu, G., and Shen, X.: The Effect of Online Investor Sentiment on Stock Movements: an LSTM Approach, *Complexity*, **2020**, 1 (2020), 4754025.

[9] Yong, S., Wen, L. et al.: A TEXT MINING BASED STUDY of INVESTOR SENTIMENT and ITS INFLUENCE on STOCK RETURNS., *Economic Computation & Economic Cybernetics Studies & Research*, **52**, 1 (2018).

[10] Ba, J., Kiros, J., and Hinton, G.: Layer Normalization, *arXiv preprint arXiv:1607.06450* (2016).

[11] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L. et al.: Attention Is All You Need, *Advances in neural information processing systems*, **30** (2017).

[12] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, NAACL, 2019.

[13] Xiong, R., Yang, Y., He, D., Zheng, K., Zheng, S. et al.: On Layer

Normalization in the Transformer Architecture, International Conference on Machine Learning, 2020.

[14] Lin, Z., Feng, M., Santos, C. N. d., Yu, M., Xiang, B. et al.: A Structured Self-attentive Sentence Embedding, ICLR, 2017.

[15] Loshchilov, I. and Hutter, F.: Decoupled Weight Decay Regularization, International Conference on Learning Representations, 2019.

[16] Nie, Y., Nguyen, N. H., Sinthong, P., and Kalagnanam, J.: A Time Series Is Worth 64 Words: Long-term Forecasting with Transformers,

[17] Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., de Laroussilhe, Q. et al.: Parameter-Efficient Transfer Learning for NNLP, Proceedings of the 36th International Conference on Machine Learning (ICML), 2019.

[18] Yang, Y., UY, M. C. S., and Huang, A.: FinBERT: a Pretrained Language Model for Financial Communications,