JOURNAL OF INFORMATION SCIENCE AND ENGINEERING XX, XXXX-XXXX (2024)
DOI: XX.XXX.XXXX.XX

Generalized Stock Price Prediction for Multiple Stocks Combined with News Fusion

PEI-JUN LIAO[1,2], HUNG-SHIN LEE[5,+], YAO-FEI CHENG[4],
Li-Wei Chen[3,5], Hung-yi Lee[1], and Hsin-Min Wang[2,+]
[1]*Graduate Institute of Electrical Engineering, National Taiwan University, Taipei, Taiwan*
[2]*Institute of Information Science, Academia Sinica, Taipei, Taiwan*
[3]*United Link Co., Ltd., Taipei, Taiwan*
[4]*University of Washington, Seattle, WA, USA*
[5]*National Tsing Hua University, Hsinchu, Taiwan*

Predicting stock prices presents challenges in financial forecasting. While traditional approaches such as ARIMA and RNNs are prevalent, recent developments in Large Language Models (LLMs) offer alternative methodologies. This paper introduces an approach that integrates LLMs with daily financial news for stock price prediction. To address the challenge of processing news data and identifying relevant content, we utilize stock name embeddings within attention mechanisms. Specifically, we encode news articles using a pre-trained LLM and implement three attention-based pooling techniques—self-attentive, cross-attentive, and position-aware self-attentive pooling—to filter news based on stock relevance. The filtered news embeddings, combined with historical stock prices, serve as inputs to the prediction model. Unlike prior studies that focus on individual stocks, our method trains a single generalized model applicable across multiple stocks. Experimental results demonstrate a 7.11% reduction in Mean Absolute Error (MAE) compared to the baseline, indicating the utility of stock name embeddings for news filtering and price forecasting within a generalized framework.

Keywords: Stock Price Prediction, Language Model, Time Series Forecasting

1. Introduction

Predicting stock prices continues to be a challenge in financial forecasting, remaining an active area of research [1, 2, 3, 4, 5, 6]. Recent studies have increasingly integrated textual data with historical stock prices in their predictive models. Currently, benchmark datasets often use concise social media posts from platforms such as X (formerly Twitter) or Facebook [7, 8], utilizing information retrieval techniques to filter and extract relevant content for stock price forecasting.

However, articles obtained through information retrieval techniques often yield a limited quantity of data and may miss implicitly relevant information not easily captured by conventional keyword-based queries, such as those relying solely on stock names. Furthermore, separating the retrieval phase from the stock price prediction model introduces a dependency on the accuracy and reliability of the independent retrieval process.

Conversely, bypassing retrieval and aggregating all available daily articles introduces significant noise. This noise complicates processing and analysis, hindering the integra-

+ Corresponding authors: Hung-Shin Lee and Hsin-Min Wang.

tion of textual information with stock price data. Consequently, irrelevant textual information can obscure the prediction process and degrade model accuracy.

To address this problem, we employ an attentive pooling mechanism [9] to filter heterogeneous information. This mechanism encodes the extracted text into a fixed-length embedding using a pre-trained language model (PLM). Rather than filtering news for individual stocks a priori [4], we aggregate all financial news articles published daily, processing a significantly larger volume of text compared to short social media posts. Building upon models like BELT [10] and BigData22 baselines [8], which primarily rely on simple concatenation and cross-attention [11] mechanisms, we systematically evaluate various text and price fusion strategies. Specifically, we investigate fusion techniques that dynamically weight textual information and market price signals to capture market dynamics. This integration allows the model to utilize both structured financial signals and unstructured textual data for prediction.

In this work, we demonstrate that news, when combined with the pooling mechanism, can assist in stock price prediction. We selected six individual stocks from the Taiwan Stock Exchange (TWSE) for our experiments. Furthermore, we conduct additional validation on individual stocks from the BigData23 [4] dataset, which encompasses the U.S. stock market, to evaluate cross-market applicability. Unlike previous models trained on individual stocks, our objective is to develop a single generalized model across multiple stocks and evaluate its performance on unseen data. In a generalized model handling multiple stocks, news filtering requires dynamic, stock-specific context. To address this, we incorporate individual stock names into the attention mechanism as queries. We explore three variants of attention mechanisms to integrate stock name information: self-attentive pooling [12], cross-attentive pooling [13], and position-aware self-attentive pooling [14]. Our experimental results demonstrate that this approach achieves a 7.11% reduction in Mean Absolute Error (MAE) compared to the baseline. In the fusion stage, we build upon cross-attention and concatenation by incorporating Graph Convolutional Networks (GCNs) [15, 16]. These networks model the structural dependencies between stocks. Furthermore, we utilize bidirectional cross-attention to capture interactions between textual and price signals. This fused representation is subsequently utilized as input for Time-LLM [17], upon which our proposed model is grounded.

This work presents the following contributions:

1. **General Stock Price Prediction Model (Sec. 3)**: We introduce a "General" model, defined as a single unified model trained across multiple stocks. Unlike previous individual-asset models, this generalized framework captures collective market dynamics and demonstrates applicability across the Taiwan and U.S. markets.
2. **Semantic-Aware News Selection Mechanism (Sec. 3-2)**: We propose a pooling mechanism guided by *Stock Name Embeddings*. This allows the model to leverage the semantic knowledge of Large Language Models (LLMs) to automatically filter irrelevant noise from daily news feeds, addressing the information noise problem in financial text processing.
3. **Integration of News and Price (Sec. 3-3)**: Integration of News and Price (Sec. 3-3): By integrating GCNs with cross-attention, we fuse unstructured textual representations with structured numerical data. News is first encoded into contextual text representations by a PLM. The LLM then serves as the backbone to integrate and transform these

text representations together with price series. This enables the predictor to exploit information from unstructured text that is not explicitly present in price histories.

4. **Empirical Validation (Sec. 4 & 5)**: We conduct experiments on both the Taiwan Stock Exchange and the U.S. BigData23 dataset. The results demonstrate reductions in prediction errors compared to established baselines and provide an analysis of name-based versus ticker-based embeddings in different market contexts.

2. Related Work

2.1 Stock Price Prediction

We present a stock price prediction methodology that integrates fusion strategies and attention mechanisms to combine price signals with textual information.

Conventional approaches to stock price prediction often formulate the task using logistic regression [18] or moving average methodologies [19]. Among these, Autoregressive Integrated Moving Average (ARIMA) models [20, 21] are widely used for modeling time series through autoregressive and moving average components.

The development of deep learning has led to a range of models for stock price prediction. Recurrent Neural Networks (RNNs) [22, 23] and Long Short-Term Memory networks (LSTMs) [24, 25, 26, 1, 27, 28, 15, 29, 30, 31, 32] have been used to capture temporal dependencies. Convolutional Neural Networks (CNNs) have also been applied to model short-term market fluctuations by extracting localized patterns [33, 34, 29, 35].

More recently, Transformer-based models, initially developed for natural language processing (NLP), have been adopted for stock price prediction [3], alongside deep learning architectures based on multilayer perceptrons (MLPs) [36, 5].

Time series forecasting has also advanced with Transformer architectures, including PatchTST [37, 38] and Autoformer [39]. Motivated by the success of large language models (LLMs) in NLP, researchers have proposed models for time series forecasting, including TimesFM [40], FPT [41], AutoTimes [42], and Time-LLM [17]. These models leverage generative pre-trained transformer (GPT) [43, 44, 45, 46] and LLaMA [47, 48, 49] architectures.

2.2 Text Information

Prior work has shown that textual information can complement price-based signals for stock price prediction. For example, the BELT framework [10] uses BERT-based [50] encoding to extract information from X (formerly Twitter) and integrates it with LSTM-based stock price prediction models. Similarly, our model integrates textual information with price data to forecast future stock prices.

Textual data from social media and news platforms has been increasingly incorporated into stock price prediction [51, 52, 7, 8, 4, 53, 54, 55, 5, 19]. Advances in NLP, including large language models and pre-trained language models (PLMs), have improved the extraction and representation of information from financial text. The BERT family of models [50, 56] provides contextualized encodings that support more informative representations of text.

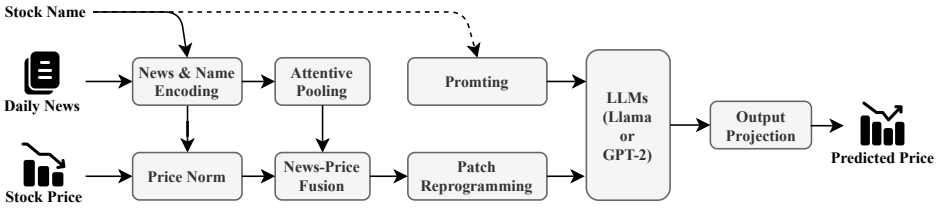


Fig. 1. Overview of the proposed stock price prediction framework. We employ Attentive Pooling to derive pooled representations of all news articles from each day. Subsequently, we integrate daily historical stock price data with corresponding news information via News-Price Fusion. Finally, these historical representations are input into an LLM to predict future stock prices. The dotted line represents a process that is optional.

Generative large language models have further expanded the use of text in stock price prediction by providing additional mechanisms to model latent relationships and sentiment linked to market dynamics. Specialized models such as FinLLMs [57] and FinGPT [58], trained on financial text, facilitate incorporating textual information into forecasting pipelines [59, 60, 61]. In this study, we primarily use BERT to encode Chinese-language text and, where applicable, DeBERTa [62] to encode the BigData23 [4] dataset to support fair comparisons. Building on prior research, we investigate the integration of price and text information to improve predictive accuracy. Various studies have explored methodologies to incorporate large language models into stock price prediction.

3. Proposed Method

We propose a stock price prediction model that extends the Time-LLM framework [17] by integrating relevant news data. Figure 1 illustrates the architecture of the proposed model. For each standard monthly trading window of 20 trading days, we apply separate normalization to (i) daily closing prices and (ii) the news articles associated with each trading day, and then encode the news using a PLM (i.e., News Encoding). Next, the encoded news is aggregated together with stock name embeddings via a pooling mechanism (i.e., Attentive Pooling) to obtain a daily news representation. Across the 20 trading days, the resulting sequence of pooled news vectors is fused with the corresponding price sequence using a fusion module (i.e., News-Price Fusion), thereby combining information from both modalities. The fused sequence representation is then provided to an LLM to predict future stock prices; in this study, we focus on one-day-ahead forecasting. The following subsections describe the design and functionality of each module.

3.1 News Encoding

Each news article is independently encoded using a pre-trained BERT[1] or DeBERTa[2] model. We use the [CLS] token representation from the final layer of BERT or DeBERTa

[1] https://huggingface.co/ckiplab/bert-base-chinese
[2] https://huggingface.co/microsoft/deberta-v3-base

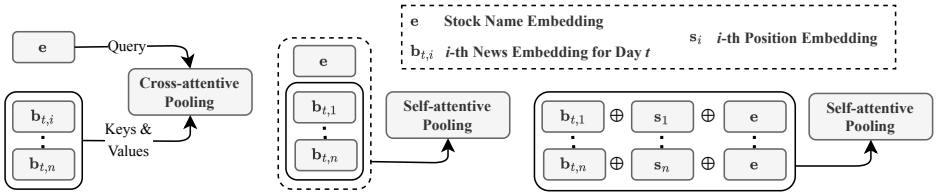


Fig. 2. Three proposed daily news pooling methods that integrate the stock name. (a) We use the sequence of news embeddings as Keys and Values and the stock name embedding as Query for cross-attention to obtain a pooled representation. (b) We append the stock name embedding in front of the sequence of news embeddings, and apply self-attention to obtain a pooled representation. (c) We add the stock name embedding and positional embeddings to the sequence of news embeddings via matrix addition $\oplus$, and apply self-attention to obtain a pooled representation.

as the document embedding, which provides a fixed-length semantic representation of the article.

3.2 Prompting in LLM

Our input contains 20 days of historical closing prices. Directly adopting the prompting style of the original Time-LLM model may lead to longer prompts, which can reduce the prominence of the numerical price inputs and affect the model's ability to capture temporal patterns.

To mitigate this issue, we use a concise prompt. Instead of providing detailed data descriptions or statistical summaries, the prompt contains only the target stock name. This design serves two purposes. First, it shortens the prompt, so the numerical inputs remain the primary information used for forecasting. Second, it conditions the LLM on the stock identity, allowing it to align the input sequence with the corresponding stock-specific context.

3.3 Attentive Pooling for Daily News Collection

Our dataset contains, on average, over 200 news articles per day. Encoding all article representations directly within the LLM input can be computationally expensive. Therefore, we apply an attentive pooling mechanism to summarize each day's news collection into a single vector representation. Given a sequence of BERT-encoded news embeddings $\{\mathbf{b}_{t,1}, \ldots, \mathbf{b}_{t,n}\}$, expressed as a matrix $\mathbf{B}_t \in \mathbb{R}^{n \times d}$, where $\mathbf{b}_{t,i} \in \mathbb{R}^d$ denotes the d-dimensional BERT embedding of the i-th news article on day t, we compute the daily news representation $\mathbf{c}_t \in \mathbb{R}^d$ as

$$\mathbf{c}_t = \text{Softmax}(\mathbf{w}\mathbf{B}_t^\intercal)\mathbf{B}_t, \tag{1}$$

where $\mathbf{w} \in \mathbb{R}^d$ is a trainable parameter. This formulation corresponds to dot-product attention, where the keys and values share the same representations, and the query is the trainable vector $\mathbf{w}$ [12]. The resulting $\mathbf{c}_t$ is then combined with the stock price embeddings and passed to a fusion module to integrate news and price information.

Moreover, the relevance of news articles can differ across stocks. While prompting can condition the LLM on the target stock, incorporating news embeddings together

with stock name embeddings provides an additional mechanism to filter news during inference. As illustrated in Figure 2, we integrate stock name embeddings with features extracted from news articles within the pooling module to produce a stock-conditioned daily news representation. In contrast to approaches that influence news utilization only at the LLM-based prediction stage, our pooling mechanism incorporates stock name embeddings during the news aggregation step, thereby injecting stock-specific information into the filtering process.

3.3.1 Cross-attentive Pooling (CAP)

As illustrated in Figure 2(a), we use the BERT-encoded stock name embedding, denoted as $\mathbf{e} \in \mathbb{R}^d$, as the query in a cross-attention mechanism. This embedding is obtained by averaging the BERT/DeBERTa token embeddings of the stock name at either the character level or the WordPiece level. For consistency with the attention formulation used in other pooling mechanisms, we revise eq. (1) accordingly. For day t, the news embeddings are denoted as $\mathbf{B}_t \in \mathbb{R}^{n\times d}$ and serve as both keys and values. The resulting daily representation $\mathbf{c}_t \in \mathbb{R}^d$ is computed as

$$\mathbf{c}_t = \text{Softmax}(\mathbf{e}\mathbf{W}_c\mathbf{B}_t^\intercal)\mathbf{B}_t, \tag{2}$$

where $\mathbf{W}_c \in \mathbb{R}^{d\times d}$ is a trainable parameter. This formulation uses the stock name embedding to modulate attention weights over the daily news embeddings $\mathbf{b}_{t,i}$ based on their similarity to the target stock, computed via inner products in the shared embedding space, thereby assigning lower weights to less relevant news articles.

3.3.2 Self-attentive Pooling (SAP)

As shown in Figure 2(b), we concatenate the stock name embedding with the news embeddings along the sequence dimension. We then apply self-attentive pooling to obtain a daily aggregate news representation:

$$\mathbf{c}_t = \text{Softmax}(\mathbf{w}_s\tilde{\mathbf{B}}_t^\intercal)\tilde{\mathbf{B}}_t, \tag{3}$$

where $\tilde{\mathbf{B}}_t = [\mathbf{e};\mathbf{B}_t] \in \mathbb{R}^{(n+1)\times d}$ treats the stock name as an additional item in the sequence, and $\mathbf{w}_s \in \mathbb{R}^d$ is a trainable parameter. Under this formulation, the attention weights are determined jointly by the stock name embedding and the news embeddings, reflecting both stock–news relations and interactions among the news articles.

3.3.3 Position-aware Self-attentive Pooling (PA-SAP)

As depicted in Figure 2(c), we incorporate positional embeddings to account for temporal order within each day's news sequence. Let $\mathbf{S} \in \mathbb{R}^{n\times d}$ denote sinusoidal positional embeddings as in the Transformer architecture. We replicate the stock name embedding $\mathbf{e}$ to match the length of $\mathbf{B}_t$, yielding $\mathbf{E} \in \mathbb{R}^{n\times d}$. We then add positional embeddings to both the replicated stock embedding and the news embeddings, and form the modified embeddings $\bar{\mathbf{B}}_t$. The resulting sequence is summarized using a self-attentive pooling mechanism, referred to as position-aware self-attentive pooling:

$$\mathbf{c}_t = \text{Softmax}(\mathbf{w}_p\bar{\mathbf{B}}_t^\intercal)\bar{\mathbf{B}}_t, \tag{4}$$

where $\bar{\mathbf{B}}_t = \mathbf{B}_t + \mathbf{E} + \mathbf{S}$, and $\mathbf{w}_p \in \mathbb{R}^d$ is a trainable parameter.

3.4 News-Price Fusion

To integrate textual and price information, we use three fusion methods to combine data from news articles and stock prices, generating Stock-aware Features, as illustrated in Fig. 3.

3.4.1 Cross-attention Fusion

We use cross-attention to integrate daily news embeddings and stock price embeddings. Specifically, we apply both price-to-news and news-to-price cross-attention [11]. Cross-attention is defined as

$$\text{CrossAtt}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\intercal}}{\sqrt{d}}\right)\mathbf{V}, \tag{5}$$

where $\mathbf{Q} \in \mathbb{R}^{T_Q \times d}$ and $\mathbf{K},\mathbf{V} \in \mathbb{R}^{T_K \times d}$. Here, T_Q denotes the length of the query sequence and T_K denotes the length of the key (and value) sequence; the key and value sequences share the same length T_K. Before applying attention, $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ are each projected by separate linear layers.

Accordingly, the price-to-news ($\mathbf{S_{P2N}}$) and news-to-price ($\mathbf{S_{N2P}}$) representations are computed as

$$\mathbf{S_{P2N}} = \text{CrossAtt}(\mathbf{P},\mathbf{N},\mathbf{N}),\ \mathbf{S_{N2P}} = \text{CrossAtt}(\mathbf{N},\mathbf{P},\mathbf{P}), \tag{6}$$

where $\mathbf{N} = (\mathbf{c}_t \in \mathbb{R}^d \mid t = 1,\ldots,T)$ and $\mathbf{P} = (\mathbf{p}_t \in \mathbb{R}^d \mid t = 1,\ldots,T)$ denote the sequences of news embeddings and price embeddings over $T = 20$ days (a monthly trading period), respectively.

$\mathbf{S_{P2N}}$ and $\mathbf{S_{N2P}}$ are the outputs of cross-attention under two query–key–value configurations. Specifically, $\mathbf{S_{P2N}}$ is obtained by using $\mathbf{P}$ as the query and $\mathbf{N}$ as both the key and the value. Conversely, $\mathbf{S_{N2P}}$ is obtained by using $\mathbf{N}$ as the query and $\mathbf{P}$ as both the key and the value.

Both $\mathbf{S_{P2N}}$ and $\mathbf{S_{N2P}}$ are matrices in $\mathbb{R}^{T \times d}$. Before applying attention, $\mathbf{N}$ and $\mathbf{P}$ are each passed through separate linear transformation layers to produce their corresponding query, key, and value representations. These two directions of cross-attention capture dependencies between news and stock prices and allow information to be exchanged between the two modalities.

3.4.2 GCN Fusion

In addition to cross-attention, we incorporate a two-layer GCN [15, 16] to integrate news and price information. The first layer fuses the daily news embeddings and price embeddings to model within-day relations. The second layer applies a CausalCNN (Causal Convolutional Neural Network) to aggregate the resulting news-aware price features over a 5-day window. We use a 5-day window to match the horizon of a five-day short-term moving average, which provides a standard time scale for summarizing short-term price movements. The formulation is

$$\mathbf{H} = \text{GCNConv}(\mathbf{N},\mathbf{P}),\ \mathbf{S_{GCN}} = \text{CausalCNN}\,(\mathbf{H})\,. \tag{7}$$

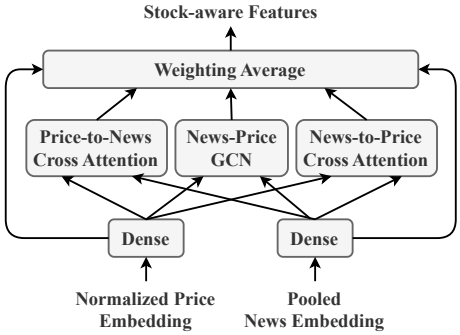


Fig. 3. The News-Price Fusion Module. Initially, we feed the 20-day price embeddings and pooled news embeddings into distinct dense layers. Subsequently, we apply bidirectional cross-attention, alternating between prices and news as Query/Key and Value. Additionally, a two-layer GCN architecture is employed to model the interaction between prices and news. Finally, we integrate the three fused representations with the original outputs from the dense layers (both price and news embeddings) using a weighted average, yielding the final fused result.

The input to the first layer consists of the news collection embeddings $\mathbf{N}$ and normalized price embeddings $\mathbf{P}$ over T days, as introduced in section 3.4.1. The GCNConv layer aggregates information within this T-day window by connecting the news and price features. In the second layer, we apply a CausalCNN with kernel size 5 to the output of the first layer, $\mathbf{H}$. The causal convolution restricts the receptive field so that, at each time step, the model depends only on past and current inputs, thereby preserving temporal causality. This convolution aggregates information from five consecutive days to represent short-term temporal dynamics. The output of this layer is denoted as $\mathbf{S_{GCN}}$.

Finally, we compute a learnable weighted sum of the representations from the fusion methods, together with the original news collection and price embeddings, to obtain a unified representation (Stock-aware Features), as illustrated at the top of Fig. 3. This representation is then fed into the patch-reprogramming module, where it is concatenated with the prompt within the input sequence for future price prediction.

3.5 Patch Reprogramming and LLM

LLMs are pre-trained on large-scale text corpora and can model patterns and relationships in natural language. When applying LLMs to quantitative domains such as finance, a key difficulty is the modality gap between numerical time-series inputs and the discrete, token-based representations used by language models. To mitigate this gap, we adopt the patch reprogramming technique of Time-LLM [17], which enables a frozen LLM to process stock-related features without modifying its core architecture.

The main idea is to map stock-aware features into a representation that is compatible with the LLM embedding space. We first construct a compressed representation of the LLM vocabulary embedding space. Specifically, a linear layer projects the vocabulary embeddings from the original size V to a lower dimension U (with $V \gg U$), producing a set of *prototypes* that summarize the vocabulary embedding space.

Given these prototypes, we use cross-attention to map stock-aware features into the

Table 1. Maximum and average number of characters in news articles and maximum and average number of news articles per day. For excessively long articles, methods such as sliding window or truncation are applied.

Max # Chars	Avg # Chars	Max # News	Avg # News
7538	509.96	654	219.50

LLM embedding space. In this mechanism, the stock-aware features serve as the *query*, while the vocabulary prototypes serve as both the *key* and the *value*. This mapping yields a prototype-weighted representation of the input features in the LLM embedding space. With patch reprogramming, stock-related signals (e.g., price dynamics and news-derived features) are represented in a token-compatible embedding form for subsequent processing by the frozen LLM. To preserve the pre-trained parameters, we keep the LLM frozen and fine-tune only the reprogramming layers and the final prediction layer. This design limits the number of trainable parameters during adaptation to the forecasting task.

4. Experimental Setup

4.1 Datasets

4.1.1 Taiwan Stock Market Data Collection

We collected Taiwan stock market data (TW21) from July 1, 2021 to April 30, 2024 using the twstock API[3], which contains 1,018 trading days. We conducted experiments on six stocks: TSMC, MediaTek, Foxconn, Realtek, Novatek, and Delta. Following the time series partitioning strategy used in the Time-Series Library[4] and Time-LLM, we split the dataset into training (70%), validation (10%), and test (20%) sets, corresponding to 712, 103, and 203 trading days, respectively. We also collected Taiwanese financial news from the same period and filtered articles related to finance, consumer markets, and technology. On average, the dataset contains 219.50 articles per day, and each article has 569.96 Chinese characters (the PLM encodes at the character level), as summarized in Table 1. The prediction task uses 20 days of historical prices and the corresponding news as input (approximately one trading month) and predicts the closing price of the next trading day.

4.1.2 BigData23 U.S. Stock Market Data

As an additional analysis across markets, we conducted experiments on 42 U.S. stocks using the BigData23 [4] dataset, which includes companies from multiple sectors such as technology, finance, and consumer goods. This composition allows us to examine whether the model relies on broad market signals rather than sector-specific patterns. Following the same time series partitioning strategy, we split the data into training (70%), validation (10%), and test (20%) sets. For each stock, the Twitter texts within a single day

[3] https://twstock.readthedocs.io/zh-tw/latest/

[4] https://github.com/thuml/Time-Series-Library

were aggregated, and their chronological order was removed. The resulting daily texts were then aligned with each stock's time sequence, while texts from different stocks were sorted alphabetically and merged into a single day's news input.

4.2 Model and Training Configurations

To develop a unified model that generalizes across multiple stocks, we aggregate data from all six stocks in the Taiwan dataset for training and validation, and evaluate the model on the individual test set of each stock. Similarly, for the BigData23 U.S. market dataset, we combine data from all 42 stocks and align Twitter comments in chronological order to preserve the temporal correspondence between stock prices and social media discussions. The model was trained to minimize the Mean Squared Error (MSE) loss function.

In our experiments on the Taiwan dataset, the backbone LLM is the LLaMA3-8b model from the TAIDE project (Llama3-TAIDE-LX-8B-Chat-Alpha1, 8B parameters)[5]. To examine the dependence on the backbone, we also run experiments with CKIP's GPT-2 (gpt2-base-chinese, 102M parameters)[6] as an alternative LLM. For the BigData23 dataset, we follow the LLaMA and GPT-2 configurations used in Time-LLM, where the backbone LLM is kept frozen during training.

We use early stopping with a patience of 5 and set the maximum number of epochs to 15. The batch size is 4, and the learning rate is 0.01 in all experiments. Other settings follow Time-LLM. All experiments are conducted on an NVIDIA RTX-3090 GPU with 24GB memory.

4.3 Evaluation Metrics

We use Mean Absolute Error (MAE) and Mean Squared Error (MSE) as evaluation metrics by comparing model predictions with the corresponding ground-truth closing prices. In all experiments, stock prices are normalized using Standard Scaling[7] during training and inference, and MAE and MSE are computed on the normalized prices.

5. Experimental Results

5.1 Prediction on TW21

The first experiment evaluates a future-1 day prediction scenario, where a model trained on a mixed-stock dataset is tested across six individual stocks. Two Time-LLM baselines are considered: (1) without news (TimeLLM) and (2) with news but without stock name embeddings (+News). Additionally, we compare against two classical models: a pure LSTM and an FPT model. As demonstrated in Table 2, cross-attentive pooling (+CAP) and self-attentive pooling (+SAP) yield lower MAE and MSE compared to the baselines across the majority of individual stocks.

[5] https://huggingface.co/taide/Llama3-TAIDE-LX-8B-Chat-Alpha1

[6] https://huggingface.co/ckiplab/gpt2-base-chinese

[7] https://scikit-learn.org/dev/modules/generated/sklearn.preprocessing.StandardScaler.html

Table 2. Performance of individual stocks in future-1 day prediction. TimeLLM represents the Time-LLM baseline without news, +News represents including news but excluding the stock name, +CAP, +SAP, and +PA-SAP are the proposed news pooling methods integrating the stock name: cross-attentive pooling, self-attentive pooling, and position-aware self-attentive pooling, respectively. Bold text indicates that it surpasses both TimeLLM and +News, and underline indicates the best in each column.

Company	Metric	Methods						
		LSTM	FPT	TimeLLM	+News	+CAP	+SAP	+PA-SAP
TSMC	MAE	0.2677	0.1477	0.1496	0.1530	**0.1398**	**0.1433**	**0.1419**
	MSE	0.2000	0.0457	0.0456	0.0446	**0.0399**	**0.0429**	**0.0432**
MediaTek	MAE	0.2931	0.1403	0.1365	0.1347	**0.1276**	**0.1294**	**0.1200**
	MSE	0.2270	0.0356	0.0340	0.0346	**0.0292**	**0.0301**	**0.0261**
Foxconn	MAE	1.0633	0.5378	0.5016	0.5035	**0.4763**	**0.4953**	0.5125
	MSE	0.8225	0.7734	0.6935	0.7776	**0.6479**	0.7216	0.7237
Realtek	MAE	0.2449	0.0930	0.0892	0.0961	0.0921	**0.0880**	0.0898
	MSE	0.1752	0.0175	0.0164	0.0192	0.0167	**0.0161**	**0.0147**
Novatek	MAE	0.2727	0.0890	0.0938	0.0949	**0.0929**	**0.0921**	**0.0910**
	MSE	0.2152	0.0167	0.0171	0.0189	**0.0169**	**0.0159**	**0.0157**
Delta	MAE	0.3536	0.1938	0.1887	0.1781	**0.1740**	**0.1740**	**0.1772**
	MSE	0.2614	0.0763	0.0673	0.0613	**0.0612**	**0.0564**	**0.0581**

In the TW21 dataset, we observe that among all pooling methods, +CAP yields the lowest MSE for TSMC (0.0399) and Foxconn (0.6479), and the lowest MAE for both TSMC (0.1398) and Foxconn (0.4763).

+SAP outperforms the baselines across most cases, though it yields the lowest overall error on fewer individual stocks than +CAP. These results indicate that, within TW21, +CAP minimizes errors for large-cap stocks such as TSMC and Foxconn, whereas +SAP exhibits lower error variance across the different individual stocks.

Position-aware self-attentive pooling (+PA-SAP) yields the lowest errors for specific stocks, particularly MediaTek, where it attains the lowest MAE (0.1200) and MSE (0.0261), and Novatek, where it achieves the lowest MSE (0.0157). However, its error metrics show higher variance across the dataset compared to +SAP.

Overall, these findings indicate that within the TW21 dataset, the proposed pooling mechanisms reduce prediction errors across the evaluated stocks. In addition, multi-seed evaluation (Table 3) shows that while +CAP yields the lowest mean errors in large-cap stocks, +SAP yields comparable mean errors with lower or equal standard deviations across most individual stocks. The relative ranking between +CAP and +SAP remains unchanged across seeds, indicating that the observed error reductions are not driven by random initialization. These results suggest that +SAP exhibits lower sensitivity to initialization variance.

Table 3. Mean performance over three independent random seeds with sample standard deviation in parentheses. The upper block reports MAE and the lower block reports MSE. Lower is better.

	TSMC	MediaTeK	Foxconn	Realtek	Novatek	Delta
			MAE Mean (Std)			
TimeLLM	0.1502 (0.0099)	0.1335 (0.0104)	0.5092 (0.0474)	0.0909 (0.0047)	0.0931 (0.0011)	0.1876 (0.0080)
+News	0.1444 (0.0077)	0.1295 (0.0121)	0.5153 (0.0409)	0.0924 (0.0061)	0.0923 (0.0023)	0.1815 (0.0044)
+SAP	0.1414 (0.0043)	0.1227 (0.0078)	0.4927 (0.0138)	0.0853 (0.0023)	0.0894 (0.0024)	0.1744 (0.0099)
+CAP	0.1390 (0.0007)	0.1252 (0.0090)	0.5019 (0.0408)	0.0897 (0.0040)	0.0933 (0.0014)	0.1807 (0.0074)
+PA-SAP	0.1489 (0.0145)	0.1333 (0.0116)	0.5510 (0.0345)	0.0967 (0.0068)	0.0936 (0.0049)	0.1889 (0.0139)
			MSE Mean (Std)			
TimeLLM	0.0463 (0.0048)	0.0327 (0.0044)	0.7285 (0.1439)	0.0166 (0.0018)	0.0174 (0.0004)	0.0670 (0.0070)
+News	0.0441 (0.0009)	0.0305 (0.0061)	0.7659 (0.1430)	0.0174 (0.0022)	0.0177 (0.0014)	0.0600 (0.0013)
+SAP	0.0432 (0.0024)	0.0271 (0.0034)	0.6902 (0.0571)	0.0150 (0.0010)	0.0164 (0.0004)	0.0562 (0.0045)
+CAP	0.0418 (0.0017)	0.0282 (0.0037)	0.7264 (0.1399)	0.0163 (0.0010)	0.0173 (0.0007)	0.0613 (0.0010)
+PA-SAP	0.0482 (0.0067)	0.0321 (0.0053)	0.9120 (0.1888)	0.0185 (0.0039)	0.0174 (0.0026)	0.0650 (0.0108)

Table 4. Performance on TW21 for future-1 day prediction. TimeLLM represents the baseline without news, while +News includes news but excludes the stock name. +CAP, +SAP, and +PA-SAP are proposed news pooling methods integrating the stock name. Bold text denotes surpassing both TimeLLM and +News, and underline highlights the best in each column.

Method	LLaMA Backbone		GPT-2 Backbone	
	MAE	MSE	MAE	MSE
without SNP				
TimeLLM	0.1913	0.1497	0.1889	0.1380
+News	0.1874	0.1432	0.1849	0.1391
+CAP	**0.1830**	**<u>0.1252</u>**	**0.1775**	**0.1200**
+SAP	**<u>0.1777</u>**	**0.1311**	**0.1756**	**0.1191**
+PA-SAP	0.1936	0.1535	**0.1788**	**0.1321**
with SNP				
TimeLLM	0.1932	0.1457	0.2000	0.1574
+News	0.1934	0.1594	0.1725	0.1170
+CAP	**0.1838**	**0.1353**	**<u>0.1704</u>**	**<u>0.1106</u>**
+SAP	**0.1870**	0.1472	0.1765	0.1276
+PA-SAP	**0.1887**	0.1469	0.1752	0.1256

5.2 Pooling Methods and Stock Name Prompts Across LLMs

To evaluate the various pooling mechanisms, we first analyze the LLaMA backbone by averaging the test MAE and MSE across all stocks. As shown in Table 4, +SAP reduces prediction errors. Without the Stock Name Prompt (SNP), +SAP reduces MAE from 0.1913 to 0.1777 (7.11%) and MSE from 0.1497 to 0.1311 (12.42%), yielding lower errors than both the TimeLLM and +News baselines.

With SNP, the error reductions from +SAP are less pronounced. MAE decreases

from 0.1932 to 0.1870 (3.21%), while MSE slightly increases from 0.1457 to 0.1472 (1.03%). Compared to +News, +SAP reduces MAE and MSE by 3.31% and 7.65%, respectively. In this setting, +CAP yields the lowest MSE (0.1353) with an MAE of 0.1838. This indicates that the interactions captured by +CAP between stock-related features and news contribute to lower prediction errors.

We further evaluate position-aware self-attentive pooling (+PA-SAP). Without SNP, +PA-SAP achieves MAE and MSE values of 0.1788 and 0.1321, respectively; while these are lower than the +News baseline, they do not consistently drop below the errors of +SAP or +CAP. Specifically, although +PA-SAP achieves a lower MAE compared to +CAP (0.1788 vs. 0.1830), it results in a higher MSE (0.1321 vs. 0.1252). With SNP, +PA-SAP yields an MAE of 0.1752 and an MSE of 0.1256, which remains higher than the MSE of +CAP.

A potential reason for the performance of +PA-SAP is that causal relationships among different news articles may not be distinct enough to benefit from positional encoding. Unlike structured financial reports or explicitly linked event sequences, financial news articles often exhibit weak or implicit causal dependencies. Consequently, incorporating positional relationships does not further reduce prediction errors. One direction for future work is to pre-filter news articles based on causal relationships before applying pooling mechanisms. Identifying causally linked news items and prioritizing them in the pooling process could assist in capturing market signals.

To verify that these trends are not model-specific, we extended our analysis to CKIP-GPT2. The results indicate that +SAP reduces MAE/MSE by 7.04%/13.70% without SNP. When SNP is included, +SAP yields MAE/MSE reductions of 11.75%/18.93%. However, under CKIP-GPT2, +SAP yields higher errors than +CAP, which achieves the lowest overall errors. This suggests that +CAP captures interactions between stock and news information, particularly when using a backbone with lower prompt-following capabilities.

+SAP exhibits lower error variance across settings, particularly without SNP. It yields consistent error metrics, whereas other methods exhibit higher variance. Furthermore, under the LLaMA backbone, the absence of SNP leads to larger error reductions, likely because stock name embeddings already provide sufficient contextual information, reducing the requirement for additional disambiguation through prompt-based mechanisms.

5.3 Ablation Study

In the ablation studies, we analyze +SAP under the without SNP setting on the TAIDE-LLaMA3 backbone, as detailed in Table 5. The results indicate that, under this scenario, the integration of all fusion mechanisms yields the lowest prediction errors. Each component contributes to error reduction; removing the GCN layer alone results in a larger increase in error compared to the simultaneous removal of P2N and GCP.

In contrast, under the SNP setting, the N2P-only configuration yields the lowest errors. This indicates that when prompts are present, the additional fusion mechanisms do not further reduce prediction errors.

Table 5. Ablation study results for the +SAP method. The "-" symbol indicates the removal of specific components from the News-Price Fusion module in Figure 1. Underlined values represent the best performance in each column.

Method	with SNP		without SNP	
	MAE	MSE	MAE	MSE
+SAP	0.1870	0.1472	<u>0.1777</u>	<u>0.1311</u>
- GCN	0.1835	0.1420	0.1940	0.1583
- P2N	0.1882	0.1538	0.1934	0.1611
- N2P	0.2166	0.2125	0.2112	0.1889
- P2N - N2P	0.2149	0.2191	0.2145	0.2147
- N2P - GCN	0.1902	0.1605	0.1808	0.1351
- P2N - GCN	<u>0.1776</u>	<u>0.1313</u>	0.1808	0.1341
- P2N - N2P - GCN	0.1865	0.1421	0.1871	0.1444

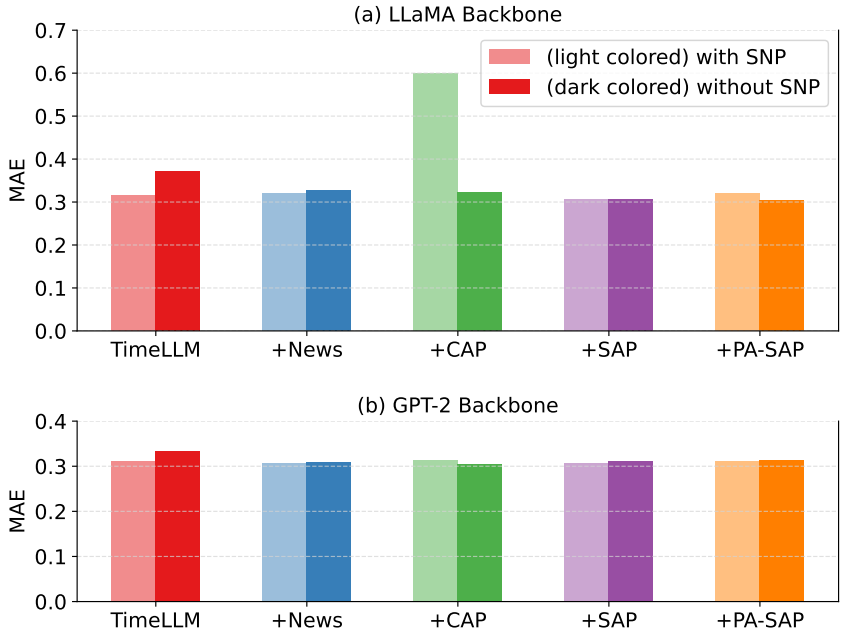


Fig. 4. (a) and (b) compare the MAE of future 5-day price prediction methods with different backbones: LLaMA (a) and GPT-2 (b). Darker bars indicate results without SNP, while lighter bars show results with SNP. TimeLLM serves as the baseline, +News includes news without stock name embeddings, and +CAP, +SAP, and +PA-SAP are three attentive pooling methods.

5.4 Future-5 Day Prediction

We analyze the average MAE for future-5 day predictions across two backbone models and three attentive pooling mechanisms. As shown in Fig. 4, +SAP yields the lowest MAE for short-term forecasting, while +CAP exhibits higher error variance across different stocks. Incorporating the SNP reduces prediction errors, indicating its utility in leveraging LLMs for sequence modeling over extended horizons. Exhibiting lower mean

Table 6. Performance on BigData23 for future-1 day prediction (without SNP). TimeLLM is the baseline without news, while +News includes news but excludes the stock name. +SAP, +CAP, and +PA-SAP are news pooling methods integrating the stock name. Bold surpasses both TimeLLM and +News, and underline highlights the best in each column.

Methods	LLaMA Backbone		GPT-2 Backbone	
	MAE	MSE	MAE	MSE
TimeLLM	0.1030	0.0221	0.1037	0.0222
+News	0.0975	0.0199	0.1034	0.0223
+CAP	0.2653	0.1291	0.2443	0.1102
+SAP	**0.0974**	**0.0198**	**0.1024**	**0.0218**
+PA-SAP	0.1019	0.0215	0.1097	0.0245

errors and lower variance, +SAP provides more consistent forecasting results across the dataset. These findings suggest that the current 20-day input window aligns with short-term forecasting objectives, whereas extending it to 60 days could be evaluated for long-term trend predictions in future work.

5.5 BigData23 Results

The linguistic characteristics of the BigData23 dataset differ from TW21, presenting a challenge for semantic-based models. While TW21 employs full company names that serve as semantic anchors, BigData23 relies on abbreviated ticker symbols (e.g., "AAPL", "F"). In the pre-trained embedding space of a PLM, single-letter tickers like "F" (Ford) or "T" (AT&T) exhibit polysemy and lack the specific entity associations found in full corporate names. This semantic ambiguity reduces the precision of name-based entity recognition.

Following the methodology of Time-LLM [17], we evaluated the model using LLaMA2 [48] and OpenAI's GPT-2 [44], focusing on the *without SNP* setting for consistency with TW21. As shown in Table 6, +SAP yields the lowest prediction errors under the LLaMA backbone. Conversely, +CAP exhibits an increase in error metrics.

This divergence provides insight into the attention mechanisms: +CAP relies on the stock name embedding acting as a "query" to filter news. When the ticker embedding is semantically ambiguous, the error reductions from this filtering mechanism diminish. In contrast, +SAP, which fuses information via self-attention across the concatenated sequence, reduces dependency on the explicit semantic representation of the stock identifier. These findings suggest that while our framework is applicable across markets, minimizing prediction errors on ticker-based datasets may require mapping abbreviations to full entity names to provide contextual precision.

5.6 Computational Cost vs. Performance

Using an LLM backbone increases computational cost compared to lightweight models such as LSTM. On a single NVIDIA RTX-3090, in our experiments, LSTM training

completes (with early stopping) in approximately 30 minutes, whereas the LLM-based variants (TimeLLM and +SAP) require approximately two days under the same training protocol (early stopping with patience $= 5$ and a maximum of 15 epochs). Averaged over the six TW21 stocks in Table 2, +SAP reduces MAE by 56.14% and MSE by 73.23% relative to LSTM. Compared with the LLM backbone alone (TimeLLM), +SAP reduces MAE by 3.60% and MSE by 6.40% on average, with additional computational overhead relative to the base LLM. For daily forecasting, the inference latency aligns with the prediction frequency; therefore, the cost–performance trade-off is determined by the application's accuracy requirements and available compute budget.

6. Conclusion and Future Directions

In this study, we introduced a multi-stock training framework designed to integrate news sentiment for stock price prediction while mitigating the impact of irrelevant information. We implemented and evaluated three attention-based pooling mechanisms: self-attentive (+SAP), cross-attentive (+CAP), and position-aware self-attentive pooling. The approach utilizes stock names to guide these mechanisms, enabling the model to filter information from real-world news feeds in the Taiwan stock market. The empirical results on the TW21 dataset indicate that the framework reduces prediction errors, and its evaluation on the BigData23 dataset demonstrates its applicability across different data environments.

Our comparative analysis across 1-day and 5-day prediction horizons, using both LLaMA and GPT-2 as backbone models, yielded the following observations. The +SAP mechanism exhibited lower error variance and consistently reduced prediction errors across stocks. In contrast, while the +CAP mechanism yielded low errors for specific large-cap stocks, its metrics exhibited higher variance, indicating a sensitivity to hyperparameter tuning that warrants further investigation. Furthermore, incorporating the "Name Prompt" did not yield a further reduction in error metrics, suggesting that the attention mechanisms alone are sufficient for identifying relevant news content within our framework.

Building on these findings, we outline directions for future research. First, regarding data scope and generalization limits, we intend to expand the TW21 dataset to include a broader range of securities, specifically targeting small-cap and low-visibility stocks. This will allow us to evaluate the model's applicability beyond large-cap technology companies and analyze the utility of semantic embeddings in less information-rich environments. We will also explore the integration of longer-term historical data, such as 60- or 120-day input windows, to forecast medium-term price trends (e.g., the 20-day moving average). This will assess the model's capability to capture both short-term volatility and long-term momentum.

Second, to address the limitations observed in ticker-based datasets (such as BigData23), we plan to investigate Entity Linking and knowledge retrieval techniques. By mapping ambiguous tickers (e.g., "F") to their corresponding semantic entities (e.g., "Ford Motor Company") before feeding them into the LLM, we aim to provide the contextual representation required for news filtering. Additionally, we will experiment with incorporating features from traditional econometric models, such as ARIMA, to augment

the model's input representation.

Finally, future work will involve backtesting these predictive models as components of trading strategies to evaluate their performance against established financial benchmarks.

ACKNOWLEDGMENT

We thank Ming-Chi Yen for maintaining the GPU infrastructure. Gratitude is extended to TAIDE (LLM) and twstock (stock data) for their resources. Gemini 3.1 Pro and GPT-5.2 Thinking were used for language refinement.

REFERENCES

1. K. Chen, Y. Zhou, and F. Dai, "A LSTM-based method for stock returns prediction: A case study of China stock market," in *Proceedings of IEEE International Conference on Big Data*, 2015, pp. 2823–2824.
2. A. Nayak, M. M. Pai, and R. M. Pai, "Prediction models for Indian stock market," *Procedia Computer Science*, Vol. 89, 2016, pp. 441–449.
3. T. Li, Z. Liu, Y. Shen, X. Wang, H. Chen, and S. Huang, "MASTER: Market-guided stock transformer for stock price forecasting," in *Proceedings of AAAI Conference on Artificial Intelligence*, 2024, pp. 162–170.
4. S. Wang, Y. Bai, T. Ji, K. Fu, L. Wang, and C.-T. Lu, "Stock movement and volatility prediction from tweets, macroeconomic factors and historical prices," in *Proceedings of IEEE International Conference on Big Data*, 2023, pp. 1863–1872.
5. S. BL and S. BR, "Combined deep learning classifiers for stock market prediction: integrating stock price and news sentiments," *Kybernetes*, Vol. 52, no. 3, 2023, pp. 748–773.
6. K. J. Koa, Y. Ma, R. Ng, and T.-S. Chua, "Learning to generate explainable stock predictions using self-reflective large language models," in *Proceedings of ACM on Web Conference*, 2024, pp. 4304–4315.
7. Y. Xu and S. B. Cohen, "Stock movement prediction from tweets and historical prices," in *Proceedings of Annual Meeting of the Association for Computational Linguistics*, 2018, pp. 1970–1979.
8. H. Wu, W. Zhang, W. Shen, and J. Wang, "Hybrid deep sequential modeling for social text-driven stock prediction," in *Proceedings of ACM International Conference on Information and Knowledge Management*, 2018, pp. 1627–1630.
9. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, Vol. 30, 2017, pp. 5998–6008.
10. Y. Dong, D. Yan, A. I. Almudaifer, S. Yan, Z. Jiang, and Y. Zhou, "BELT: A pipeline for stock price prediction using news," in *Proceedings of IEEE International Conference on Big Data*, 2020, pp. 1137–1146.
11. M. Seo, "Bidirectional attention flow for machine comprehension," *arXiv preprint arXiv:1611.01603*, 2016.

12. P. Safari, M. India, and J. Hernando, "Self-attention encoding and pooling for speaker recognition," in *Proceedings of Annual Conference of the International Speech Communication Association*, 2020, pp. 941–945.
13. S. M. Kye, Y. Kwon, and J. S. Chung, "Cross attentive pooling for speaker verification," in *Proceedings of IEEE Spoken Language Technology Workshop*, 2021, pp. 294–300.
14. C. Zhang, S. Gao, H. Wang, and W. Zhang, "Position-aware joint entity and relation extraction with attention mechanism." in *Proceedings of International Joint Conference on Artificial Intelligence*, 2022, pp. 4496–4502.
15. Y. Chen, Z. Wei, and X. Huang, "Incorporating corporation relationship via graph convolutional neural networks for stock price prediction," in *Proceedings of ACM International Conference on Information and Knowledge Management*, 2018, pp. 1655–1658.
16. S. Zhang, H. Tong, J. Xu, and R. Maciejewski, "Graph convolutional networks: a comprehensive review," *Computational Social Networks*, Vol. 6, no. 1, 2019, pp. 1–23.
17. M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan, and Q. Wen, "Time-LLM: Time series forecasting by reprogramming large language models," in *Proceedings of International Conference on Learning Representations*, 2024.
18. J. A. Ou and S. H. Penman, "Financial statement analysis and the prediction of stock returns," *Journal of Accounting and Economics*, Vol. 11, no. 4, 1989, pp. 295–329.
19. S. Lauren and S. D. Harlili, "Stock trend prediction using simple moving average supported by news classification," in *International Conference on Advanced Informatics: Concept, Theory and Application*, 2014, pp. 135–139.
20. A. A. Ariyo, A. O. Adewumi, and C. K. Ayo, "Stock price prediction using the ARIMA model," in *Proceedings of International Conference on Computer Modelling and Simulation*, 2014.
21. G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.
22. K.-i. Kamijo and T. Tanigawa, "Stock price pattern recognition-a recurrent neural network approach," in *Proceedings of International Joint Conference on Neural Networks*, 1990, pp. 215–221.
23. S. Wang, Y. Bai, K. Fu, L. Wang, C.-T. Lu, and T. Ji, "ALERTA-Net: A temporal distance-aware recurrent networks for stock movement and volatility prediction," in *Proceedings of IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 2023, pp. 538–542.
24. W. Lu, J. Li, J. Wang, and L. Qin, "A CNN-BiLSTM-AM method for stock price prediction," *Neural Computing and Applications*, Vol. 33, no. 10, 2021, pp. 4741–4753.
25. J. Agrawal, V. Chourasia, and A. Mittra, "State-of-the-art in stock prediction techniques," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, Vol. 2, no. 4, 2013, pp. 1360–1366.
26. K. Khare, O. Darekar, P. Gupta, and V. Attar, "Short term stock price prediction using deep learning," in *Proceedings of IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology*, 2017, pp. 482–486.

27. Z. Hu, Y. Zhao, and M. Khushi, "A survey of forex and stock price prediction using deep learning," *Applied System Innovation*, Vol. 4, no. 1, 2021, p. 9.
28. B. S. Pramod and M. P. M. Shastry, "Stock price prediction using LSTM," *Test Engineering and Management*, Vol. 83, 2020, pp. 5246–5251.
29. S. Mehtab and J. Sen, "Stock price prediction using CNN and LSTM-based deep learning models," in *Proceedings of International Conference on Decision Aid Sciences and Application*, 2020, pp. 447–452.
30. Y. Wang, H. Wu, J. Dong, Y. Liu, M. Long, and J. Wang, "Deep time series models: A comprehensive survey and benchmark," *arXiv preprint arXiv:2407.13278*, 2024.
31. S. Wang, T. Ji, J. He, M. Almutairi, D. Wang, L. Wang, M. Zhang, and C.-T. Lu, "AMA-LSTM: Pioneering robust and fair financial audio analysis for stock volatility prediction," in *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, 2024, pp. 379–386.
32. M. M. Billah, A. Sultana, F. Bhuiyan, and M. G. Kaosar, "Stock price prediction: comparison of different moving average techniques using deep learning model," *Neural Computing and Applications*, Vol. 36, no. 11, 2024, pp. 5861–5871.
33. A. A. Adebiyi, A. O. Adewumi, and C. K. Ayo, "Comparison of ARIMA and artificial neural networks models for stock price prediction," *Journal of Applied Mathematics*, Vol. 2014, no. 1, 2014, p. 614342.
34. D. Daiya, M.-S. Wu, and C. Lin, "Stock movement prediction that integrates heterogeneous data sources using dilated causal convolution networks with attention," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 8359–8363.
35. E. Hoseinzade and S. Haratizadeh, "CNNpred: CNN-based stock market prediction using a diverse set of variables," *Expert Systems with Applications*, Vol. 129, 2019, pp. 273–285.
36. J. Fan and Y. Shen, "StockMixer: A simple yet strong MLP-based architecture for stock price forecasting," in *Proceedings of AAAI Conference on Artificial Intelligence*, 2024, pp. 8389–8397.
37. Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," in *Proceedings of International Conference on Learning Representations*, 2023.
38. S. Wang, T. Ji, L. Wang, Y. Sun, S.-C. Liu, A. Kumar, and C.-T. Lu, "StockTime: A time series specialized large language model architecture for stock price prediction," *arXiv preprint arXiv:2409.08281*, 2024.
39. H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," *Advances in Neural Information Processing Systems*, Vol. 34, 2021, pp. 22 419–22 430.
40. A. Das, W. Kong, R. Sen, and Y. Zhou, "A decoder-only foundation model for time-series forecasting," in *Proceedings of International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, 2024, pp. 10 148–10 167.
41. T. Zhou, P. Niu, L. Sun, and R. Jin, "One fits all:power general time series analysis by pretrained LM," *Advances in Neural Information Processing Systems*, Vol. 36, 2023, pp. 43 322–43 355.

42. Y. Liu, G. Qin, X. Huang, J. Wang, and M. Long, "AutoTimes: Autoregressive time series forecasters via large language models," *Advances in Neural Information Processing Systems*, Vol. 37, 2024, pp. 122 154–122 184.
43. A. Radford, "Improving language understanding by generative pre-training," *OpenAI Blog.*, 2018.
44. A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI blog*, Vol. 1, no. 8, 2019, p. 9.
45. T. B. Brown, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.
46. P. P. Ray, "ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope," *Internet of Things and Cyber-Physical Systems*, Vol. 3, 2023, pp. 121–154.
47. H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, and F. Azhar, "LLaMA: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.
48. H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, and S. Bhosale, "LLaMA 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.
49. A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, and A. Fan, "The LLaMA 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.
50. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019, pp. 4171–4186.
51. G. P. C. Fung, J. X. Yu, and W. Lam, "Stock prediction: Integrating text mining approach using real-time news," in *Proceedings of IEEE International Conference on Computational Intelligence for Financial Engineering*, 2003, pp. 395–402.
52. S. Mohan, S. Mullapudi, S. Sammeta, P. Vijayvergia, and D. C. Anastasiu, "Stock price prediction using news sentiment analysis," in *Proceedings of IEEE International Conference on Big Data Computing Service and Applications*, 2019.
53. Y. Soun, J. Yoo, M. Cho, J. Jeon, and U. Kang, "Accurate stock movement prediction with self-supervised learning from sparse noisy tweets," in *Proceedings of IEEE International Conference on Big Data*, 2022, pp. 1691–1700.
54. R. Gupta and M. Chen, "Sentiment analysis for stock price prediction," in *Proceedings of IEEE Conference on Multimedia Information Processing and Retrieval*, 2020, pp. 213–218.
55. S. Deng, T. Mitsubuchi, K. Shioda, T. Shimada, and A. Sakurai, "Combining technical analysis with sentiment analysis for stock price prediction," in *IEEE International Conference on Dependable, Autonomic and Secure Computing*, 2011, pp. 800–807.
56. Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.
57. J. Lee, N. Stevens, S. C. Han, and M. Song, "A survey of large language models in finance (FinLLMs)," *arXiv preprint arXiv:2402.02315*, 2024.

58. H. Yang, X.-Y. Liu, and C. D. Wang, "FinGPT: Open-source financial large language models," *arXiv preprint arXiv:2306.06031*, 2023.
59. Q. Xie, W. Han, X. Zhang, Y. Lai, M. Peng, A. Lopez-Lira, and J. Huang, "Pixiu: A comprehensive benchmark, instruction dataset and large language model for finance," *Advances in Neural Information Processing Systems*, Vol. 36, 2023, pp. 33 469–33 484, datasets and Benchmarks Track.
60. H. Tong, J. Li, N. Wu, M. Gong, D. Zhang, and Q. Zhang, "Ploutos: Towards interpretable stock movement prediction with financial large language model," *arXiv preprint arXiv:2403.00782*, 2024.
61. S. Wu, O. Irsoy, S. Lu, V. Dabravolski, M. Dredze, S. Gehrmann, P. Kambadur, D. Rosenberg, and G. Mann, "BloombergGPT: A large language model for finance," *arXiv preprint arXiv:2303.17564*, 2023.
62. P. He, X. Liu, J. Gao, and W. Chen, "DeBERTa: Decoding-enhanced BERT with disentangled attention," in *Proceedings of International Conference on Learning Representations*, 2021.



Pei-Jun Liao received the B.S. degree from the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, and the M.S. degree from the Graduate Institute of Communication Engineering, National Taiwan University. He is currently pursuing a Ph.D. degree at the Graduate Institute of Electrical Engineering, National Taiwan University. His research interests include speech processing and self-supervised learning.



Hung-Shin Lee received the Ph.D. degree in electrical engineering from National Taiwan University (NTU), Taipei, Taiwan, in 2021. He is currently an AI Engineer with United Link Co., Ltd., Taiwan. Since February 2026, he has also been an Adjunct Assistant Professor with Graduate Institute of AI Interdisciplinary Applied Technology, National Taiwan Normal University, Taipei, Taiwan. His research interests include speech recognition and synthesis, as well as speech and video enhancement.



Yao-Fei Cheng received the B.S. degree from the Department of Computer Science and Information Engineering, National Taipei University of Technology, Taipei, Taiwan, and the M.S. degree in Computational Linguistics from the University of Washington, Seattle. His research interests include speech processing, natural language processing, and self-supervised learning.



Li-Wei Chen received the B.S. degree from the Department of Computer Science and Information Engineering, National Taipei University of Technology, Taipei, Taiwan. He is currently pursuing a M.S. degree at the Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan. His research interests include speech enhancement and speech synthesis.



Hung-yi Lee is a professor of EE and CS at National Taiwan University (NTU). His recent research focuses on developing technology that can reduce the requirement of annotated data for speech processing and natural language processing. He is a co-organizer of the workshop "Self-Supervised Learning for Speech and Audio Processing" at NeurIPS (2020) and AAAI (2022), special issue "New Trends in self-supervised speech processing" at INTERSPEECH (2020), workshop "Meta Learning and Its Applications to Natural Language Processing" at ACL (2021). He was the technical program co-chair of ISCSLP 2022, technical program co-chair of INLG 2023 and program chair of Machine Learning Summer School (MLSS) 2021 Taipei. He was a lead guest editor of the Journal of Selected Topics in Signal Processing (JSTSP), editing a special issue, "Self-Supervised Learning for Speech and Audio Processing".



Hsin-Min Wang received the B.S. and Ph.D. degrees in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1989 and 1995, respectively. In October 1995, he joined the Institute of Information Science, Academia Sinica, Taipei, Taiwan, where he is currently a Research Fellow. His research interests include spoken language processing, natural language processing, multimedia information retrieval, and machine learning. He was an Associate Editor of IEEE/ACM Transactions on Audio, Speech and Language Processing from 2016 to 2020. He is a Senior Editor of APSIPA Transactions on Signal and Information Processing. He was a General Co-Chair of ISCSLP2016, ISCSLP2018, ASRU2023, and O-COCOSDA2024 and a Technical Co-Chair of ISCSLP2010, O-COCOSDA2011, APSIPAASC2013, ISMIR2014, ASRU2019, APSIPAASC2023, and SLT2026. He was the recipient of the Chinese Institute of Engineers Technical Paper Award in 1995 and the ACM Multimedia Grand Challenge First Prize in 2012. He was an APSIPA distinguished lecturer for 2014–2015.