Explainable Stock Price Movement Prediction using Contrastive Learning

Kelvin Du
Nanyang Technological University
Singapore, Singapore
zidong001@e.ntu.edu.sg

Rui Mao
Nanyang Technological University
Singapore, Singapore
rui.mao@ntu.edu.sg

Frank Xing
National University of Singapore
Singapore, Singapore
xing@nus.edu.sg

Erik Cambria
Nanyang Technological University
Singapore, Singapore
cambria@ntu.edu.sg

ABSTRACT

Predicting stock price movements is a high-stakes task that demands explainability for human decision-makers. A key shortcoming in current methods is treating sub-predictions independently, without learning from accumulated experiences. We propose a novel triplet network for contrastive learning to enhance the explainability of stock movement prediction by considering instances of "integrated textual information and quantitative indicators". We refer to the target past-l-day tweet-price time series as the "anchor instance". Each anchor instance is paired with a "positive instance" characterized by highly correlated return trends yet significant differences across the entire feature space, and a "negative instance" that exhibits similar return trends along with high proximity in the feature space. The model is designed with the objective of **(1)** minimizing the cross entropy loss between input logits and target, **(2)** minimizing the distance between the anchor instances and positive instances, and **(3)** maximizing the distance between the anchor instances and negative instances. Our framework's effectiveness is demonstrated through extensive testing, showing superior performance on stock prediction benchmarks.

CCS CONCEPTS

• **Human-centered computing**; • **Applied computing** → **Economics**;

KEYWORDS

AI, Explainability, Contrastive learning, NLP, Stock price

ACM Reference Format:
Kelvin Du, Rui Mao, Frank Xing, and Erik Cambria. 2024. Explainable Stock Price Movement Prediction using Contrastive Learning. In *Proceedings of ACM International Conference on Information and Knowledge Management (CIKM '24)*. ACM, New York, NY, USA, 9 pages. https://doi.org/XXXXXXX.XXXXXXX

CIKM '24, October 21–25, 2024, Boise, ID

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Stock price movement prediction is a challenging task due to the dynamic and complex nature of the financial market. Early studies [23] believed that achieving a prediction accuracy rate over 50% is seen as a substantial benefit to the daily trading activities for practical traders. Conventionally, there are two dominant methods used in stock market analysis, i.e., fundamental analysis and technical analysis [23]. Fundamental analysis examines stocks from their inherent value perspective, considering factors such as from the macro-economy and industry circumstances to the individual financial robustness of companies. Aspects such as earnings, expenditures, assets, and liabilities are integrative parts of this analysis. Conversely, technical analysis seeks to identify potential opportunities by examining statistical trends in stock price movements. Well-known technical indicators include the Simple Moving Average (SMA), Exponential Moving Average (EMA), and Moving Average Convergence/Divergence (MACD) etc. Many previous studies on market forecasting have predominantly used historical stock trading data, technical indicators, and macroeconomic factors as the basis for their analysis [23]. However, there is a novel and rapidly expanding field of research that focuses on the integration of financial textual data with fundamental and technical indicators in financial forecasting, using natural language processing (NLP) [7, 17–19, 22, 31, 32], because textual information reflects the cognitive patterns of market participants [20]. Among financial textual sources, social media and self-media data offer subjective judgment and analysis from investors and analysts [8] and are crucial data resources for financial market prediction [33].

A notable limitation of existing research in this domain is the insufficient exploration of the intrinsic connections inherent to stock price dynamics, which encompass two distinct categories: firstly, the inherent patterns of price movements intrinsic to a specific stock, discernible across historical price data; secondly, the analogous price movement patterns exhibited by other stocks in the market. One prospective approach for uncovering price connections is through contrastive learning. Adopted in computer vision and NLP research [10, 27], contrastive learning is a potent technique that facilitates the enhancement of feature representation by accentuating the differentiation between positive (similar) and negative (dissimilar) samples. Previous works [21, 34, 37] highlighted that hard negative triplets, which are the most similar negative instances that have opposite labels, are the most informative.

To the best of our knowledge, [29] was the first study that introduced contrastive learning to financial time series forecasting. Addressing the problem that time series datasets are usually sample-insufficient for predictive learning, it was proposed to first learn compact time series representations, then classify new samples based on their conditional mutual information to the representations [29]. In the same vein, [11] proposed the representations to include high-frequency data, and the multi-granularity heterogeneity problem handled with adaptive fusion. [15] used similar approaches not on time series, but on stocks to generate "stock embeddings" using internal and relational attributes. The efficacy of stock embeddings is subsequently evaluated using covariance matrix insertion in portfolio optimization. However, these studies did not incorporate textual features and explainable mechanisms, resulting in a limitation in their ability to enhance accuracy, explainability, and trustworthiness [1, 2]. As a result, the previous methods cannot help financial analysts and investors investigate a critical question: **What factors induce alterations in the stock price's trajectory on the subsequent trading day, assuming a comparable historical price trend?**

To address the aforementioned gaps, we propose a novel stock price movement prediction framework that integrates quantitative indicators and textual information, leveraging contrastive learning techniques to enhance its predictive capability and explainability. Each l-day lag time series is designated as an anchor instance, accompanied by its most correlated instance in the price trend time series but with the largest distance in other feature spaces displaying the same price movement, identified as the positive instance, and the most closely correlated instance in price trend and smallest distance in other feature space exhibiting an opposite movement, considered as the negative instance. The essence of this approach is to scrutinize the feature disparities among the anchor, positive, and negative instances, allowing the model to strategically position the anchor and positive instances in closer proximity through a thorough comprehension of their underlying representations and characteristics. This strategic alignment captures subtle features instrumental in predicting price movement deviations on the subsequent trading day. Remarkably, it proves effective in scenarios where the anchor and positive instances, as well as the anchor and negative instances, demonstrate congruent trends over the preceding l days. In practice, the positive and negative samples can serve as references, allowing traders to assess the inference logic and outcomes generated by machines. This approach enhances the accuracy of trading by incorporating human influence within the decision-making loop, which is a significant advantage, compared to other black-box models.

The efficacy of our proposed framework is validated through extensive experimentation on three benchmark datasets. On average, our model exceeds the strongest baseline by 1.5% in accuracy across datasets, showcasing its superior performance compared to existing approaches. We also demonstrate the explainability of our method. Our contributions can be summarized from three perspectives: (1) A novel and explainable triplet network architecture for contrastive learning is proposed. The framework revolves around predicting stock price movements by comparing textual and quantitative features of the current time interval against those of a prior time span characterized by the most analogous price movement trend to the present period. (2) Building upon the aforementioned explainable framework, we conduct a comprehensive analysis of the factors leading to discernible fluctuations in the subsequent trading day, despite the presence of highly comparable price trends in historical data. (3) The proposed model demonstrates competitive performance on publicly available datasets, underscoring its effectiveness in stock price movement prediction.

2 RELATED WORK

Research in the field of stock market prediction encompasses various aspects, such as market index, stock price, stock price movement, return rate, and volatility etc. To achieve prediction goals, time series models, machine learning techniques, deep learning approaches, and reinforcement learning methods have been explored. Specifically, [5] proposed a novel neural tensor network combined with a deep convolutional neural network (CNN) to predict event-driven stock price movements in the S&P 500 index and individual stocks. [36] proposed event-driven trading strategies that detect corporate events, considered as driving forces of market movements, from news articles. A bi-level event detection model is trained using the masked-language model (MLM) loss. Two trading strategies were tested on the EDT dataset, with trade-at-end strategy yielding a return outperforming sentiment-based models. The trade-at-best strategy, which completed transactions within a specified time frame, resulted in a return also surpassing all sentiment-based models.

Another study by [16] focused on extracting features from news titles through a CNN and event tuples (agent, predicate, and object) via knowledge graph embedding. The features were combined with daily trading and technical analysis data, and support vector machine (SVM) and long short-term memory (LSTM) models were utilized for stock price movement prediction. Joint learning of event tuples and text was found to be the most effective approach, addressing the text sparsity problem in feature extraction. A deep generative model called StockNet was proposed by [33] for stock market prediction based on binary movement, denoting a rise in stock price as one and a fall as zero. The model consisted of three components: Market Information Encoder, Variational Movement Decoder, and Attentive Temporal Auxiliary. [9] proposed to implement adversarial training as a means to enhance the predictive model's generalization capacity within the neural network framework and achieved a significant performance improvement as compare to [33].

Multiple studies have also investigated the integration of company relationships into the prediction of stock market movements. Notably, [4] incorporated company relationships using Graph Convolutional Neural Networks, while [26] proposed a deep attentive learning approach for predicting stock movements based on information from social media texts and company correlations. [35] introduces DTML (Data-axis Transformer with Multi-Level contexts), a novel approach for accurate stock movement prediction by efficiently correlating multiple stocks. DTML leverages temporal and global market context to learn dynamic correlations and outperforms existing methods, yielding a significant annualized return in investment simulations.

Another effort to enhance stock market movement prediction is through self-supervised learning from sparse noisy tweets [28], which overcomes biases towards popular stocks and filters out noisy data. IIt employs self-supervised learning to create shared embeddings for stocks and tweets, enabling accurate predictions for less popular stocks and enhancing robustness by leveraging multi-level relationships from tweets. In the study by [17] which also strategically incorporates company relationships, two important hypotheses were proposed: (1) market sentiment differs from semantic sentiment, and (2) the stock price of a target company is influenced by its related companies. To address these hypotheses, a multi-source aggregated classifier is developed for stock price movement prediction. [6] proposed a dual-graph neural network, which dynamically generates and integrates price relationships and semantic relationships between companies for stock price movement prediction. While machine learning has significantly contributed to the enhancement of stock price prediction, there exist two primary constraints within earlier research endeavors: (1) The transparency of the stock decision-making mechanism is deficient, leaving users uninformed about the underlying factors responsible for the predicted increases or declines in stock prices. (2) It is unclear what factors induce alterations in the stock price's trajectory on the subsequent trading day, assuming a comparable historical trend.

3 METHODOLOGY

The proposed framework is illustrated in Fig. 1. In a specific delineation, the framework comprises six principal components: Tweet Embedding Layer, Price Normalization Layer, Triplet Selector, Textual Information Encoder-Decoder (TIE) and Quantitative Indicator Encoder-Decoder (QIE) with triplet network, and Stock Price Movement Classifier. We employ contrastive learning-based loss regularization, namely a triplet loss for the textual feature comparative learning, and a triplet loss for the quantitative indicator feature comparative learning.

3.1 Tweets Embedding Layer

The Tweets Embedding Layer encodes tweets and generates embeddings. Specifically, each tweet e can be represented by an embedding vector $\mathbf{v} \in \mathbb{R}^d$ using sentence-BERT [25], which represents a modification of the pre-trained BERT network, using Siamese network architecture. This adaptation is specifically engineered to derive semantically significant embeddings for sentences, and the dimension of the embedding d is 384. Given n number of tweets on day i associated with stock A, it can be represented as $[e_1, e_2, \dots e_n]$. The Tweet Embedding layer provides a representation for each tweet, resulting in vectors $[v_1, v_2, \dots v_n]$ where $v_j \in \mathbb{R}^d$, $j \in [1, 2, ..., n]$ and n denotes the total number of tweets for stock A on day i.

3.2 Textual Information Encoder-Decoder

As the internet era advances, social networking platforms such as X (formerly known as Twitter) and StockTwits have increasingly shaped investor perspectives and market reactions [12]. In our study, we adopt the Twitter messages provided by [33], [30] and [28]. The Textual Information Encoder-Decoder primarily comprises a CNN and a dual-stage Attentive LSTM (ALSTM) network, collectively designed to effectively encode the embeddings of tweets.

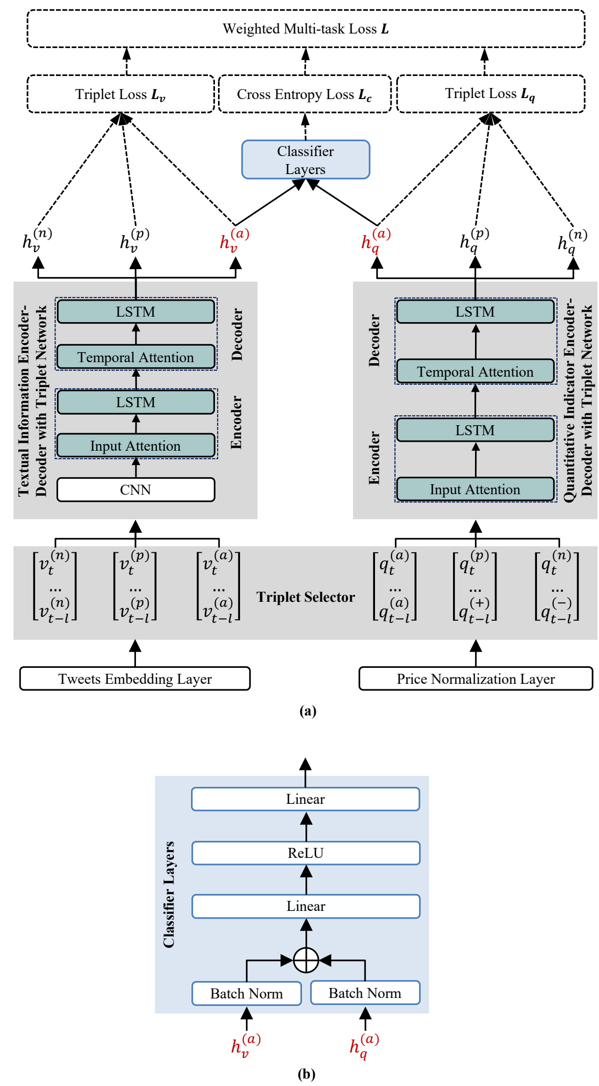

Figure 1: Proposed Framework. (a) The overall architecture. (b) The classifier layers in the overall architecture. ⊕ denotes concatenation. a, p, n denotes anchor, positive and negative stock samples, respectively. v denotes textual features; q denotes quantitative indicator features.

3.2.1 *Convolutional Neural Network.*

As emphasized by [12], tweets possess diverse impacts within the market landscape. Consequently, the Tweets Embedding Layer encodes intraday tweets and yields output, a three-dimensional vector denoted as $\mathbf{V} \in \mathbb{R}^{l \times m \times d}$. Here, l signifies the sequence length, m denotes the maximum number of daily tweets, and d represents the dimensionality of the sentence embedding, which serves as input to a 2D-CNN. The 2D-CNN's primary role is to discern and internalize intricate feature representations, denoted by the equation:

$$\mathbf{C} = \text{Conv2d}(\mathbf{V}) \tag{1}$$

The convolved features $\mathbf{C}$ undergo batch normalization, followed by activation through the Rectified Linear Unit (ReLU) function, and are subsequently subjected to adaptive max pooling, resulting in an output size of $(1, 128)$. The pooled feature maps are then concatenated to form $\mathbf{D} \in \mathbb{R}^{l \times 3 \times 128}$. The same process, using a 2D-CNN with a kernel size of $(3, 1)$ and l filters, is applied to generate $\mathbf{X} \in \mathbb{R}^{l \times 64}$.

3.2.2 Attentive LSTM. The dual-stage ALSTM is a widely recognized attentive RNN architecture that was originally introduced by [24] for time series prediction. We extended it to multivariate time series for financial forecasting. In our implementation, the ALSTM is composed of several layers, namely: Encoder (Input Attention and LSTM layers) and Decoder (Temporal Attention and LSTM layers), as illustrated in Fig. 1. The input attention module serves to assess the importance of input features at time t, and the temporal attention mechanism is applied in the decoder to adaptively select relevant encoder hidden states across all time steps. In the context of a given multivariate time series, which is represented as $\mathbf{x}^k = (x_1^k, x_2^k, \ldots, x_l^k)^\top \in \mathbb{R}^l$, an input attention mechanism can be constructed using a deterministic attention model, specifically a Multilayer Perceptron (MLP). This construction involves referencing the previous hidden state $\mathbf{h}_{t-1}$ and the cell states $\mathbf{s}_{t-1}$ within the encoder LSTM unit. This is expressed as:

$$e_t^k = \mathbf{v}_e^\top \cdot \tanh\left(\mathbf{W}_e\,[\mathbf{h}_{t-1}; \mathbf{s}_{t-1}] + \mathbf{U}_e \mathbf{x}^k\right) \quad , \tag{2}$$

where $\mathbf{v} \in \mathbb{R}^l$, $\mathbf{W}_e \in \mathbb{R}^{l \times 2d_1}$, and $\mathbf{U}_e \in \mathbb{R}^{l \times l}$ are parameters that are learned with d_1 being the size of hidden states. The attention weight α_t^k is given by Eq. 3, indicating the importance of the k-th input feature at time t. n is the number of features. A softmax function is applied to e_t^k to ensure that all attention weights sum up to 1.

$$\alpha_t^k = \frac{\exp(e_t^k)}{\sum_{i=1}^n \exp(e_t^i)} \tag{3}$$

This input attention mechanism is implemented as a forward network that can be jointly trained with other components of the Attentive LSTM. Utilizing these attention weights, the hidden state at time t can be updated as:

$$\mathbf{h}_t = f_1(\mathbf{h}_{t-1}, \widetilde{\mathbf{x}}_t) \tag{4}$$

$$\widetilde{\mathbf{x}}_t = \left(\alpha_t^1 x_t^1, \alpha_t^2 x_t^2, \ldots, \alpha_t^n x_t^n\right)^\top \tag{5}$$

Here, f represents a LSTM unit with $\mathbf{x}_t$ replaced by the newly computed $\widetilde{\mathbf{x}}_t$. Through this proposed input attention mechanism, the encoder can selectively focus on specific feature series rather than treating all input feature series uniformly. After the encoder stage, the decoder with temporal attention is adopted to predict the output using LSTM for decoding the encoded information. To adaptively select relevant encoder hidden states, a temporal attention mechanism is employed in the decoder. Specifically, the attention weight of each encoder hidden state at time t is calculated based upon the previous decoder hidden state $\mathbf{h'}_{t-1} \in R^{d_2}$ and the cell state of the LSTM unit $\mathbf{s'}_{t-1} \in \mathbb{R}^{d_2}$, with d_2 being the size of the decoder hidden states, can be represented as:

$$d_t^j = \mathbf{v}_d^\top \cdot \tanh\left(\mathbf{W}_d\left[\mathbf{h}'_{t-1}; \mathbf{s}'_{t-1}\right] + \mathbf{U}_d \mathbf{h}_j\right), \quad 1 \le j \le l \quad , \tag{6}$$

where $[\mathbf{h'}_{t-1}; \mathbf{s'}_{t-1}] \in \mathbb{R}^{2d_2}$ is a concatenation of the previous hidden state and cell state of the LSTM unit. $\mathbf{v}_d \in R^{d_1}$, $\mathbf{W}_d \in R^{d_1 \times 2z_2}$, and $\mathbf{U}_d \in \mathbb{R}^{d_1 \times d_1}$ are parameters to learn with bias terms being omitted for clarity. The attention weight β_t^j of each encoder hidden state at time t, which represents the importance of the i-th encoder hidden state for the prediction, is calculated as:

$$\beta_t^j = \frac{\exp(d_t^j)}{\sum_1^l \exp(d_t^i)} \tag{7}$$

The context vector $\mathbf{c}_t$ is computed as a weighted sum of encoder hidden states:

$$\mathbf{c}_t = \sum_{i=1}^{l} \beta_t^i \mathbf{h}_i \tag{8}$$

Once we obtain the weighted summed context vectors, we can combine them with the given target series $(y_1, y_2, \ldots, y_{T-1})$ as follows:

$$\widetilde{y}_{t-1} = \widetilde{\mathbf{w}}^\top [y_{t-1}; \mathbf{c}_{t-1}] + \widetilde{b} \tag{9}$$

where $[y_{t-1}; \mathbf{c}_{t-1}] \in \mathbb{R}^{d_1+1}$ is a concatenation of the decoder input y_{t-1} and the computed context vector $\mathbf{c}_{t-1}$. Parameters $\widetilde{\mathbf{w}} \in \mathbb{R}^{d_1+1}$ and $\widetilde{b} \in \mathbb{R}$ map the concatenation to the size of the decoder input. The newly computed $\tilde{y}_{t-1}$ is used to update the decoder hidden state at time t:

$$\mathbf{h}'_t = f_2(\mathbf{h}'_{t-1}, \tilde{y}_{t-1}) \tag{10}$$

This established attention strategy recognizes the variability in the information quality of tweets and their differential impacts across market phases.

3.3 Price Normalization Layer

The historical price data encompass open, high, low, close, and adjusted close prices for every trading session. The price movement indicators over intervals of 5, 10, 15, 20, 25, and 30 days are also computed. To capture the market fluctuations, we normalize both the price and its associated movement indicators using current day's close price for open, high and low price (e.g., $n\_open = \frac{open_t}{close_t} - 1$), the previous day's corresponding price for close and adjusted close price (e.g., $n\_close = \frac{close_t}{close_{t-1}} - 1$), and current day's adjusted close price for 5, 10, 15, 20, 25, and 30-day movement (e.g., $n\_5-day = \frac{\sum_{i=0}^4 \text{adj\_close}_{t-i}/5}{\text{adj\_close}_t} - 1$). Consequently, a total of 11 quantitative indicators have been derived in this study.

3.4 Quantitative Indicator Encoder-Decoder

Technical analysis indicates that historical price offers significant insights into prospective market movements [13]. The Quantitative Indicator Encoder-Decoder is designed to encapsulate the temporal sequence representation of quantitative indicators spanning a lookback period of l days, which is 7 days in our study.

To represent the sequential interdependence of trading days, the same dual-stage Attentive LSTM is adopted. Given multivariate time series for quantitative indicators $\mathbf{q}^k = (q_1^k, q_2^k, \ldots, q_t^k) \in \mathbb{R}^l$. The encoder output on the t^{th} day defined as:

$$\mathbf{p}_t = f_1(\mathbf{p}_{t-1}, \widetilde{\mathbf{q}}_t) \tag{11}$$

Here, $q_i \in \mathbb{R}^{11}$ signifies the price vector on day i for each stock s in the window of time steps.

Existing literature suggests that each trading day's trend influences stock trend prediction differently [9]. In alignment with this understanding, we incorporate the same temporal attention mechanism in the decoder that ascertains the significance of particular days, synthesizing a holistic feature representation from the LSTM's entire hidden states [24]. This mechanism, similar to the implementation in Textual Information Encoder-Decoder, employs the newly computed $\tilde{y}_{t-1}$ to update the decoder's hidden state at time t:

$$\mathbf{p}'_t = f_2(\mathbf{p}'_{t-1}, \tilde{y}_{t-1}) \tag{12}$$

3.5 Triplet Selector

Our triplet selection process unfolds during training and is adept at distinguishing between positive and negative instances within a batch. Specifically, we begin by calculating the pairwise Pearson correlation coefficient using l-day return trends derived from adjusted closing prices, and the Euclidean distances between both quantitative and textual features to determine the relative closeness of each instance to others during the forward pass. Based on the input labels, we identify which pairs of instances are similar (positive) and dissimilar (negative). For each anchor instance in the batch, we select the top k instances that exhibit the highest correlation in terms of price return, from instances with the same and opposite label respectively, using the Pearson correlation coefficient. We then create a distance matrix by calculating pairwise Euclidean distance using quantitative and textual features. From the top k instances identified through the highest Pearson correlation coefficient, we isolate the instance that exhibits the largest distance in the entire feature space while maintaining the same polarity as positive, and the instance with the smallest distance and opposite polarity as negative. By leveraging these positive and negative instances, the model computes the triplet loss. The training objective is to ensure that the distance between the anchor and the positive instance is smaller than the distance between the anchor and the negative instance, by at least the defined margin.

3.6 Stock Price Movement Classifier

The stock price movement classifier comprises a series of operations including batch normalization, concatenation of textual and quantitative features, and the incorporation of linear and activation layers. More precisely, the hidden states associated with anchor instances for both textual and quantitative features undergo a sequence of operations: batch normalization, concatenation, a linear layer with ReLU activation, followed by dropout, and finally, a linear layer for the computation of cross-entropy loss.

3.7 Weighted Multi-task Learning Objective

The model is designed with the objectives of minimizing entropy loss between input logits and target, and the triplet loss between the hidden states of anchor, positive and negative instances. Specifically, it aims to minimize the distance between the anchor instance and positive instances that share a common identity, while simultaneously maximizing the distance between the anchor and negative instances with contrasting identities. The triplet loss was introduced by [27] in the field of face recognition. The anchor instance (a), a positive instance (p) of the same class as the anchor, and a negative instance (n) from a different class than the anchor, form a triplet (a, p, n). The learning objective of the triplet loss is to minimize the distance $D(a, p)$ between the anchor and positive instances and maximize the distance $D(a, n)$ between the anchor and negative instance:

$$L_t = \sum_{i=1}^{N} \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right]_+ \tag{13}$$

$$\forall (f(x_i^a), f(x_i^p), f(x_i^n)) \in \mathcal{T}, \tag{14}$$

where $\mathcal{T}$ is the set of all possible triplets in the training set. The distance is measured by the Euclidean distance. The + indicates that when the value inside $D(a, p) - D(a, n) + \alpha$ is greater than zero, it is taken as that value, and when it is less than zero, it is zero. This can also be expressed as:

$$L_t = \max(D(a, p) - D(a, n) + \alpha, 0), \tag{15}$$

where α enforces a margin between positive pairs (a, p) and negative pairs (a, n). Regarding triplets, they can be categorized as:

- **Easy triplets**: In the case of $L = 0$, where $D(a, p) + \alpha < D(a, n)$, the intra-class distance is small, and the inter-class distance is large; there's no need for optimization.
- **Hard triplets**: For $D(a, n) < D(a, p)$, the inter-class distance is smaller than the intra-class distance and requires special attention.
- **Semi-hard triplets**: For $D(a, p) < D(a, n) < D(a, p) + \alpha$, the intra-class and inter-class distances are close, but there is a margin α, which is easier to optimize.

As depicted in Fig. 1, each of the identified triplets from the triplet selector undergoes processing by both Textual Information Encoder-Decoder and Quantitative Indicator Encoder-Decoder. The Textual Information Encoder-Decoder generates distinct hidden states denoted as $h_v^{(a)}$, $h_v^{(p)}$, and $h_v^{(n)}$ for the anchor, positive, and negative instances, while the Quantitative Indicator Encoder-Decoder produces corresponding hidden states represented as $h_q^{(a)}$, $h_q^{(p)}$, and $h_q^{(n)}$. Consequently, the triplet loss is computed for both the textual information, denoted as L_v, and the quantitative indicators, denoted as L_q. Likewise, the identical triplet loss computation is applied to L_q. This procedural step is designed to exam the disparities present in the triplets, enabling the model to strategically position the anchor and positive instances in higher proximity, while simultaneously ensuring that the negative instance is distanced further away from the anchor instance. Simultaneously, the hidden states corresponding to anchor instances undergo classifier layers. The process culminates in the computation of the cross-entropy loss L_c for binary classification:

$$L_c = -w_{y_n} \cdot \left(y_n \log\left(\frac{1}{1+\exp(-x_n)}\right) + (1 - y_n) \log\left(1 - \frac{1}{1+\exp(-x_n)}\right)\right)$$

The learning objective is finally defined as follows where the weighting of multiple loss functions is determined through an assessment of the homoscedastic uncertainty associated with each respective task [14] with λ_i as the learnable parameters:

$$L = \sum \frac{1}{2\lambda_i^2} L_i + \log(1 + \lambda_i^2) \tag{16}$$

Dataset	No. of Stocks	No. of Tweets	Period
ACL18 [33]	87	106,271	2014-01-02 to 2015-12-30
CIKM18 [30]	38	955,788	2017-01-01 to 2017-12-28
BIGDATA22 [28]	50	272,762	2019-07-05 to 2020-06-30

Table 1: Description of "Price Movement - Text" Datasets.

4 EXPERIMENTAL SETUP

4.1 Datasets

StockNet dataset[1], released by [33], is one of the most representative datasets for stock price movement prediction, with 87 stocks selected from various industries from January 2014 to December 2015. We also consider another social text-driven stock prediction dataset built by [30], by aggregating stock prices from Yahoo Finance alongside relevant social media discourse, primarily from Twitter. The dataset covers the time frame of January 2017 through to December 2017 with 38 stocks selected from the broader Standard & Poor's 500 list, ensuring each had a substantial representation on Twitter. Lastly, we include newly released data for stock market forecasting, created by [28]. This dataset encompasses a selection of 50 stocks, spanning from July 2019 to June 2020. We transform the price movements into a binary classification problem as in recent works for stock movement prediction [28, 33]. Specifically, we label movements equal to or less than -0.5% as 0 and movements greater than 0.55% as 1. We partitioned each dataset chronologically into training, validation, and test subsets, aligning with established methodologies applied in recent studies on stock movement prediction [28]. Our reported results are averaged over 10 runs on the testing sets with random seeds.

4.2 Evaluation Metrics

To make a fair comparison with previous studies on stock forecasting [9, 33], our chosen evaluation metrics are bidirectional accuracy and the Matthews Correlation Coefficient (MCC). Accuracy is extensively adopted across various classification problems. The MCC becomes especially pertinent when the dataset presents notable disparities in class distribution. The calculation of accuracy and MCC necessitates the construction of a confusion matrix, which enumerates true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN), and is defined as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{17}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP{+}FP)\times(TP{+}FN)\times(TN{+}FP)\times(TN{+}FN)}} \tag{18}$$

4.3 Baseline Models

We compare our model with strong baselines for stock price movement prediction, as follows: **Momentum (MOM)** serves as a technical metric that forecasts the likelihood of positive or negative movement for each instance based on the prevailing trend over the preceding 10 days. **Mean Reversion (MR)** projects the movement of each instance as a deviation from the recent price, aligning towards the 30-day moving average [3]. **StockNet** is presented by [33], in which the stock input undergoes encoding through a Variational Autoencoder (VAE) to encapsulate its inherent stochasticity. **Attentive LSTM (ALSTM)** synergizes the attention strategy with diverse LSTM cell states, as highlighted by [24]. In the same thread, two other variations, **ALSTM-W** and **ALSTM-D** are introduced by [28], with Word2Vec and Doc2Vec for tweet embeddings, respectively. **Attentive LSTM using adversarial training (Adv-ALSTM)** is introduced by [9], which incorporates adversarial training with ALSTM to augment generalization capabilities. **DTML** (Data-axis Transformer with Multi-Level contexts) is an innovative approach for precise stock movement prediction that effectively correlates multiple stocks introduced by [35]. **SLOT** improves predictions of stock market movements using self-supervised learning from sparsely available and noisy tweets [28].

[1]https://github.com/yumoxu/stocknet-dataset

4.4 Experimental Details

The models in our study are trained on an NVIDIA Tesla T4 processor. The training process encompasses 100 epochs, during which the validation dataset is employed to select the optimal model, while the test dataset serves as the basis for reporting performance metrics. A learning rate of 1e-6 is used in conjunction with the Adam optimizer, with a batch size of [64, 128], a hidden size of [256, 512], the top k of [10, 20] for triplet selector and l of [5, 7, 10] days for hyperparameter tuning.

5 RESULT AND ANALYSIS

Our investigation presents accuracy and MCC in Table 2. Across all designated categories and evaluation metrics, our novel model consistently exhibits superior performance relative to conventional technical analysis methodologies [3], as well as machine learning models [9, 24, 28, 33], with notably substantial margins of improvement. Specifically, we attained the highest performance on the CIKM'18 and BIGDATA'22 datasets, achieving the accuracy of 0.5790, and 0.5728, accompanied by improvements of 3.7% and 4.5%, respectively. Furthermore, the model demonstrated its highest performance with an MCC score of 0.1326 on the BIGDATA'22 dataset. It also achieved the second-highest MCC score on the CIKM'18 dataset, and the third-highest MCC scores on the ACL'18 dataset, yielding values of 0.0579 and 0.1481, respectively. On average, our model exceeds the strongest baseline by 1.5% in accuracy across the three datasets. It's important to emphasize that, unlike models functioning as black-box systems, our proposed approach empowers users to perform a comparative analysis between the anchor stock and instances of both positive and negative stocks. The objective of this comparison is to identify the most influential features that contribute to prediction accuracy, as demonstrated in Section 7.

6 ABLATION STUDY

An ablation study is conducted to ascertain the efficacy of contrastive learning. The outcomes derived from the training of models utilizing 10 distinct random seeds have been detailed in Table 3, which serves to underscore the favorable influence of contrastive learning on both the performance and stability of the models. Evidently, the incorporation of contrastive learning has led to a noteworthy augmentation in model performance, particularly when

Model	ACL18		CIKM18		BIGDATA22	
	Accuracy	MCC	Accuracy	MCC	Accuracy	MCC
MOM [3]	0.4701	-0.0640	-	-	-	-
MR [3]	0.4621	-0.0782	-	-	-	-
ALSTM [24]	0.5182	0.0429	0.5254	-0.0077	0.4869	-0.0254
ALSTM-W [28]	0.5332	0.0754	0.5364	0.0315	0.4828	-0.0116
ALSTM-D [28]	0.5298	0.0681	0.5040	-0.0449	0.4916	-0.0090
Adv-ALSTM [9]	0.5311	0.0685	0.5369	0.0217	0.5036	0.0120
StockNet [33]	0.5360	-0.0248	0.5235	-0.0161	<u>0.5299</u>	-0.0163
DTML [35]	<u>0.5812</u>	<u>0.1806</u>	<u>0.5386</u>	<u>0.0049</u>	0.5165	<u>0.0651</u>
SLOT [28]	**0.5872**	**0.2065**	<u>0.5586</u>	**0.0899**	<u>0.5481</u>	<u>0.0952</u>
Ours	<u>0.5670</u>	<u>0.1481</u>	**0.5790**	<u>0.0589</u>	**0.5728**	**0.1326**

Table 2: Performance comparison on the benchmark dataset. The boldface indicates the highest scores.

Model	ACL18		CIKM18		BIGDATA22	
	Accuracy	MCC	Accuracy	MCC	Accuracy	MCC
TIE + QIE	0.5423	0.0886	0.5551	0.0197	0.5609	0.0898
CL + TIE	0.5177	0.0398	0.5426	0.0312	0.5184	0.0114
CL + QIE	0.5530	0.1029	0.5693	0.0369	0.5702	0.1188
CL + TIE + QIE	**0.5670**	**0.1481**	**0.5790**	**0.0589**	**0.5728**	**0.1326**

Table 3: Ablation Analysis for Contrastive Learning (CL). The boldface indicated the best result.

applied to both the Quantitative Indicator Encoder-Decoder and Textual Information Encoder-Decoder. Notably, employing contrastive learning solely on quantitative indicators has also yielded superior results compared to scenarios where it is not applied. However, it is noteworthy that the model's performance experiences a significant decline when only textual information is considered. This phenomenon can be attributed to several factors. Primarily, the inherent relationships within quantitative indicators tend to be more discernible in contrast to textual information. This distinction arises primarily from the inherent sparsity of content within tweets. Additionally, quantitative indicators exert a more direct influence on price fluctuations. To elaborate, the model enriched with contrastive learning, specifically for the Quantitative Indicator Encoder-Decoder, has demonstrated a substantial increase of 1 to 2 percentage points in accuracy and 3 to 4 percentage points in MCC. This underscores the pivotal role played by quantitative information in the prediction process, with textual information assuming a supplementary role.

7 EXPLAINABILITY DEMONSTRATION

We have conducted a series of case studies to show how does the explainability of our model work. Generally, we observe the variation in the attention weights of quantitative features to be greater than that of textual features, aligning with the observation that quantitative features play a crucial role in predictions, with textual information serving a supportive role. Although the distance between the anchor and positive instance in the quantitative and textual feature space is larger than that between the anchor and negative instance, the visualization of temporal attention weights reveals that the attention weights of the anchor and positive instances become more correlated after training than the negative instances. This suggests that the model recognizes the common features from positive samples and distinct features from negative samples. Meanwhile, it shows that the more recent information often has a more significant impact on subsequent market movement.

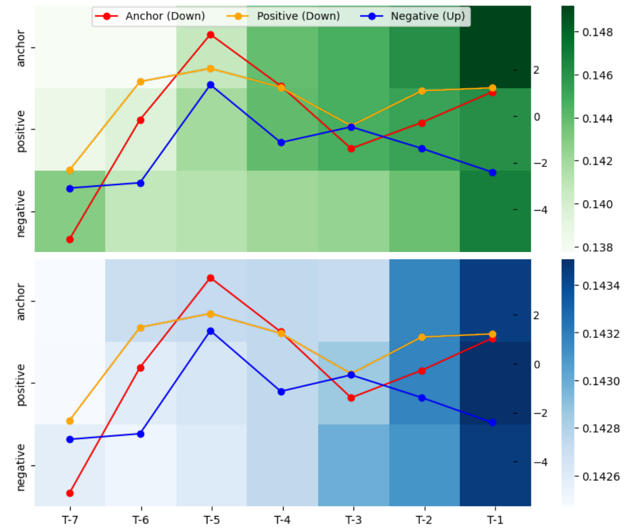

Figure 2a: Visualization of Temporal Attention Weights. The green heatmap denotes the significance of *quantitative* features by different lagging days from T-1 to T-7; The blue heatmap represents the significance of *textual* features. The line charts represent normalized adjusted closing prices in percentage. The anchor instance C and positive instance MO are in a downtrend, while the negative instance REX is in an uptrend on day T.

In Fig. 2a, the anchor, positive, and negative instances have similar historical trends from T-7 to T-1. The anchor pertains to Citigroup (C) exhibiting a downtrend on trading day T, with Altria Group (MO) serving as the positive reference with the same downward movement direction, and Rex International Holding (REX) serving as the negative instance with an upward trend on trading day T. The Pearson correlation of attention weights of quantitative features is 0.9754 for anchor and positive instances, and 0.7753 for anchor and negative instances, and that of textual features is 0.9751 for anchor and positive instances and 0.9039 for anchor and negative instances. In the green heatmap, quantitative indicators at T-4 (4 lagging days) still impact the current prediction for the anchor and positive stocks, with a general downward trend as time progresses, especially noticeable after the peak at T-5. For the quantitative indicators of the negative instance REX on T-1, there was a continuous downward trend in the price return from T-3 to T-1, which may have an impact on its upward trend on day T. On the other hand, by viewing the textual feature importance heatmap (blue), the most significant tweets for the prediction of the anchor and positive stocks appear at T-1 and T-2, while the tweets on T-3 and others are comparatively less significant. For example, we found that C's tweets mentioned that "top decliners". For REX, we also found significant tweets that likely resulted in the rise in its stock price, e.g., "investment research upgrades".

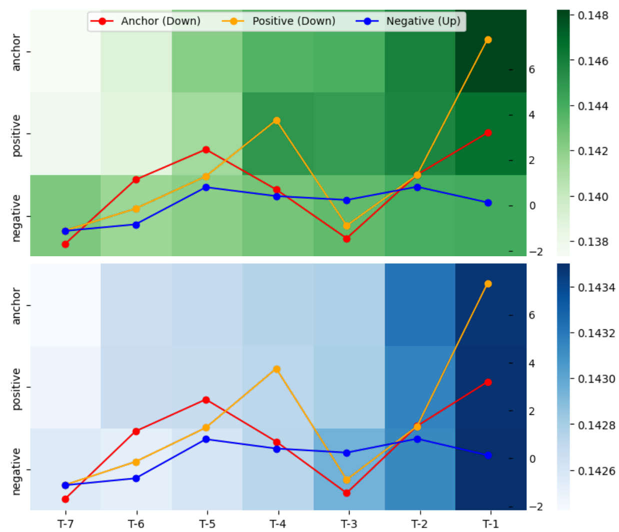

Figure 2b: Visualization of Temporal Attention Weights. The green heatmap denotes the significance of *quantitative* features by different lagging days from T-1 to T-7; The blue heatmap represents the significance of *textual* features. The line charts represent normalized adjusted closing prices in percentage. The anchor instance C and positive instance DIS are in a downtrend, while the negative instance DUK is in an uptrend on day T.

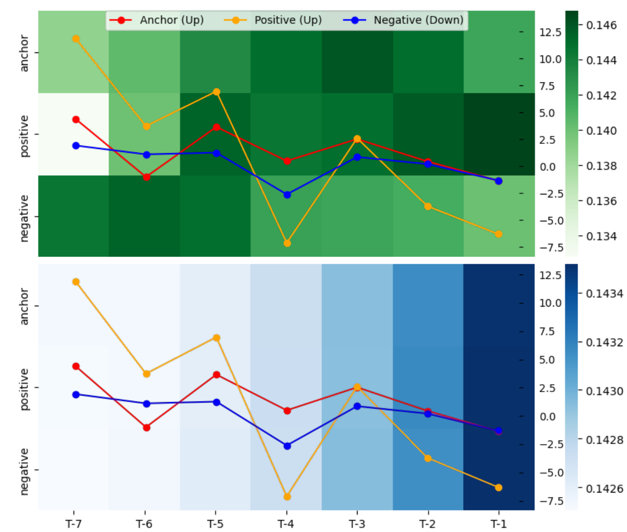

Figure 2c: Visualization of Temporal Attention Weights. The green heatmap denotes the significance of *quantitative* features by different lagging days from T-1 to T-7; The blue heatmap represents the significance of *textual* features. The line charts represent normalized adjusted closing prices in percentage. The anchor instance VZ and positive instance JPM are in a uptrend, while the negative instance NEE is in an downtrend on day T.

In Fig. 2b, the Pearson correlation of attention weights of quantitative features is 0.9654 for anchor and positive instances and 0.8368 for anchor and negative instances, and that of textual features is 0.9957 for anchor and positive instances and 0.9480 for anchor and negative instances. The anchor Citigroup (C) and the positive instance the Walt Disney Company (DIS), both exhibiting a downtrend, demonstrate a heightened emphasis on time periods T-1 and T-2 in relation to quantitative indicators, particularly a sharp increase in significance from T-3 to T-1 after a downward trend. Conversely, the negative instance Duke Energy Corporation (DUK) does not emphasize particular days significantly, underscoring a more distinctive contrast in quantitative information prioritization. Notably, the distribution of attention weights to textual information is nearly commensurate for the anchor and positive instance, with a shared emphasis on T-1 and T-2, spreading gradually to T-6. In this context, the model discerns the distinctions in quantitative indicators and comprehends the nuances between positive and negative instances. Upon analyzing the quantitative indicators of the negative instance DUK, a notable plateau period is observed in the price before subsequently increasing. A meticulous analysis of the textual corpus pertaining to C unveils a shift from an optimistic view to a pessimistic view from tweets such as "stocks with greater movement" and "don't miss the bounce and squeeze" to "technical alert: nasdaq crosses below 8100" and "make money even when the price is declining". This pattern finds resonance in the discourse surrounding DIS, evident in expressions from tweets like "nice move so far as it heads out of the consolidation box on strong volume URL" and "closed up over 7% as the company's new streaming service" to "call destruction" and "new alert".

In Fig. 2c, the correlation of attention weights of quantitative features is 0.7855 for anchor and positive instances and -0.5198 for anchor and negative instances, and that of textual features is 0.9998 for anchor and positive instances and 0.9994 for anchor and negative instances. This shows a notable consistency in the treatment of textual features with respect to temporal attention. However, a distinctive pattern emerges in the realm of quantitative information. Specifically, Verizon Communications (VZ), serving as the anchor, along with the positive instance JPMorgan Chase & Co (JPM) exhibiting an uptrend, and the negative instance NextEra Energy (NEE) with a downtrend, demonstrate a heightened focus on time periods T-1, T-2, and T-3 in textual features. Nevertheless, both the anchor and positive lines exhibit closely aligned downward trends from T-7 to T-1, with notable rises on T-5 and T-3. In contrast, the negative instance shows a stable price trend before T-4, a sharp decrease on T-4, and then follows a similar pattern as before T-4, evidenced by the heightened attention from T-4 to T-7. Sentiments regarding T-1 exhibit positivity for anchor instance, as evidenced by expressions such as "here is the list of stocks. beating s&p500 nicely." and the positive instance encompasses phrases like "buying opportunity keeps knocking" and "outperform". In the negative instance highlighting a downtrend, NextEra Energy's tweets on T-2 and T-3 revealed that "receives average recommendation". This collective decrease in stake by multiple brokerages exerted a detrimental influence on NEE's price movement.

8 CONCLUSION

An explainable framework for stock price movement prediction that strategically encodes textual information and quantitative indicators is proposed in this study. The "anchor instances" explain how predictions are made by leveraging positive and negative instances. Unique to our methodology is the adoption of Attentive LSTM combined with contrastive learning to discern the intrinsic correlations within stock prices. This is achieved by reducing entropy loss and to minimize the distance between anchor instances and positive instances with shared identities. Concurrently, our approach endeavors to increase the distance between the anchor instance and negative instances, which represent disparate identities. Lastly, extensive experiments on a benchmark dataset for stock movement prediction corroborate the improved accuracy and MCC metrics by using our framework in comparison to several established methodologies.

REFERENCES

[1] Erik Cambria, Rui Mao, Melvin Chen, Zhaoxia Wang, and Seng-Beng Ho. 2023. Seven Pillars for the Future of Artificial Intelligence. *IEEE Intelligent Systems* 38, 6 (2023), 62–69.

[2] Erik Cambria, Xulang Zhang, Rui Mao, Melvin Chen, and Kenneth Kwok. 2024. SenticNet 8: Fusing emotion AI and commonsense AI for interpretable, trustworthy, and explainable affective computing. In *Proceedings of International Conference on Human-Computer Interaction (HCII)*.

[3] Ernest Chan. 2013. *Algorithmic Trading: Winning Strategies and Their Rationale*. Wiley.

[4] Yingmei Chen, Zhongyu Wei, and Xuanjing Huang. 2018. Incorporating corporation relationship via graph convolutional neural networks for stock price prediction. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 1655–1658.

[5] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2015. Deep learning for event-driven stock prediction. In *Twenty-fourth International Joint Conference on Artificial Intelligence (IJCAI)*. 2327–2333.

[6] Kelvin Du, Rui Mao, Frank Xing, and Erik Cambria. 2024. A dynamic dual-graph neural network for stock price movement prediction. In *2024 International Joint Conference on Neural Networks (IJCNN), Yokohama, Japan*.

[7] Kelvin Du, Frank Xing, and Erik Cambria. 2023. Incorporating Multiple Knowledge Sources for Targeted Aspect-based Financial Sentiment Analysis. *ACM Transactions on Management Information Systems* 14, 3 (2023), 23.

[8] Kelvin Du, Frank Xing, Rui Mao, and Erik Cambria. 2024. Financial Sentiment Analysis: Techniques and Applications. *Comput. Surveys* 56, 9 (2024).

[9] Fuli Feng, Huimin Chen, Xiangnan He, Ji Ding, Maosong Sun, and Tat-Seng Chua. 2019. Enhancing Stock Movement Prediction with Adversarial Training. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI)*. 5843–5849.

[10] Kai He, Rui Mao, Yucheng Huang, Tieliang Gong, Chen Li, and Erik Cambria. 2023. Template-Free Prompting for Few-Shot Named Entity Recognition via Semantic-Enhanced Contrastive Learning. *IEEE Transactions on Neural Networks and Learning Systems* (2023).

[11] Min Hou, Chang Xu, Yang Liu, Weiqing Liu, Jiang Bian, Le Wu, Zhi Li, Enhong Chen, and Tie-Yan Liu. 2021. Stock Trend Prediction with Multi-granularity Data. In *Proceedings of the 30th International Conference on Information and Knowledge Management (CIKM)*. https://doi.org/10.1145/3459637.3482483

[12] Ziniu Hu, Weiqing Liu, Jiang Bian, Xuanzhe Liu, and Tie-Yan Liu. 2018. Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining (WSDM)*. 261–269.

[13] Monique Jeanblanc, Marc Yor, and Marc Chesney. 2009. *Mathematical methods for financial markets*. Springer Science & Business Media.

[14] Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7482–7491.

[15] Shicheng Li, Wei Li, Zhiyuan Zhang, Ruihan Bao, Keiko Harimoto, and Xu Sun. 2022. No Stock is an Island: Learning Internal and Relational Attributes of Stocks with Contrastive Learning. In *Proceedings of the Fourth Workshop on Financial Technology and Natural Language Processing (FinNLP)*. 147–153. https://doi.org/10.18653/v1/2022.finnlp-1.20

[16] Yang Liu, Qingguo Zeng, Huanrui Yang, and Adrian Carrio. 2018. Stock price movement prediction from financial news with deep learning and knowledge graph embedding. In *Pacific rim Knowledge Acquisition Workshop*. Springer, 102–113.

[17] Yu Ma, Rui Mao, Qika Lin, Peng Wu, and Erik Cambria. 2023. Multi-source aggregated classification for stock price movement prediction. *Information Fusion* 91 (2023), 515–528.

[18] Yu Ma, Rui Mao, Qika Lin, Peng Wu, and Erik Cambria. 2024. Quantitative Stock Portfolio Optimization by Multi-task Learning Risk and Return. *Information Fusion* 104 (2024), 102165.

[19] Lorenzo Malandri, Frank Xing, Carlotta Orsenigo, Carlo Vercellis, and Erik Cambria. 2018. Public Mood–Driven Asset Allocation: the Importance of Financial Sentiment in Portfolio Management. *Cognitive Computation* 10, 6 (2018), 1167–1176.

[20] Rui Mao, Kelvin Du, Yu Ma, Luyao Zhu, and Erik Cambria. 2023. Discovering the Cognition behind Language: Financial Metaphor Analysis with MetaPro. In *2023 IEEE International Conference on Data Mining*.

[21] Rui Mao, Kai He, Claudia Beth Ong, Qian Liu, and Erik Cambria. 2024. MetaPro 2.0: Computational Metaphor Processing on the Effectiveness of Anomalous Language Modeling. In *Findings of the Association for Computational Linguistics: ACL*. Association for Computational Linguistics, Bangkok, Thailand.

[22] Andrius Mudinas, Dell Zhang, and Mark Levene. 2019. Market trend prediction using sentiment analysis: lessons learned and paths forward. *arXiv preprint arXiv:1903.05440* (2019).

[23] Arman Khadjeh Nassirtoussi, Saeed Aghabozorgi, Teh Ying Wah, and David Chek Ling Ngo. 2015. Text mining of news-headlines for FOREX market prediction: A Multi-layer Dimension Reduction Algorithm with semantics and sentiment. *Expert Systems with Applications* 42, 1 (2015), 306–324.

[24] Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison Cottrell. 2017. A dual-stage attention-based recurrent neural network for time series prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*. 2627–2633.

[25] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Conference on Empirical Methods in Natural Language Processing*. https://api.semanticscholar.org/CorpusID:201646309

[26] Ramit Sawhney, Shivam Agarwal, Arnav Wadhwa, and Rajiv Shah. 2020. Deep attentive learning for stock movement prediction from social media text and company correlations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 8415–8426.

[27] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 815–823.

[28] Yejun Soun, Jaemin Yoo, Minyong Cho, Jihyeong Jeon, and U Kang. 2022. Accurate Stock Movement Prediction with Self-supervised Learning from Sparse Noisy Tweets. In *IEEE International Conference on Big Data*. 1691–1700.

[29] Hanwei Wu, Ather Gattami, and Markus Flierl. 2020. Conditional mutual information-based contrastive loss for financial time series forecasting. In *Proceedings of the First ACM International Conference on AI in Finance*. 1–7. https://doi.org/10.1145/3383455.3422550

[30] Huizhe Wu, Wei Zhang, Weiwei Shen, and Jun Wang. 2018. Hybrid deep sequential modeling for social text-driven stock prediction. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM)*. 1627–1630.

[31] Frank Xing, Erik Cambria, and Roy Welsch. 2018. Natural language based financial forecasting: a survey. *Artificial Intelligence Review* 50, 1 (2018), 49–73.

[32] Frank Xing, Lorenzo Malandri, Yue Zhang, and Erik Cambria. 2020. Financial Sentiment Analysis: An Investigation into Common Mistakes and Silver Bullets. In *COLING*. 978–987.

[33] Yumo Xu and Shay B Cohen. 2018. Stock movement prediction from tweets and historical prices. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*. 1970–1979.

[34] Hong Xuan, Abby Stylianou, Xiaotong Liu, and Robert Pless. 2020. Hard negative examples are hard, but useful. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*. Springer, 126–142.

[35] Jaemin Yoo, Yejun Soun, Yong-chan Park, and U Kang. 2021. Accurate multivariate stock movement prediction via data-axis transformer with multi-level contexts. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2037–2045.

[36] Zhihan Zhou, Liqian Ma, and Han Liu. 2021. Trade the Event: Corporate Events Detection for News-Based Event-Driven Trading. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. 2114–2124. https://doi.org/10.18653/v1/2021.findings-acl.186

[37] Luyao Zhu, Wei Li, Rui Mao, Vlad Pandelea, and Erik Cambria. 2023. PAED: Zero-Shot Persona Attribute Extraction in Dialogues. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, Vol. 1. Association for Computational Linguistics, Toronto, Canada, 9771–9787. https://doi.org/10.18653/v1/2023.acl-long.544