ATTENTION-BASED MULTIMODAL STOCK MOVEMENTS PREDICTION WITH HIGH-LEVEL FEATURES

PROJECT REPORT

Luo Li
Department of Computer Science
University College London
zceeidx@ucl.ac.uk

April 02, 2023

ABSTRACT

Natural Language Processing (NLP) and Computer Vision (CV) techniques demonstrate great potential in predicting stock price movements. Researchers were able to extract useful information from finanical text data and image representations of historical stock price data to predict trends of stocks. In this work, we proposed a multimodal-method that combines NLP and CV for binary stock price movements prediction, by leveraging pre-trained models for feature extraction and the attention-mechanism for feature fusion. We first transformed the daily historical stock prices into candlestick charts and used a pre-traiend ResNet50 model for extracting high level image features. Then, we used a financially fine-tuned FinBERT model to extract embeddings from stock-related tweets. Finally, we used a Transformer-based attetion fusion method for multimodal feature fusion. The proposed model was trained and evaluated on the stocknet dataset and achieved an accuracy of 53.0% and a MCC score of 0.048, beating a number of baseline models.

1 Introduction

Stock prediction has been an active field of research for a long time. However, it remains to be a very challenging task due to the stochasticity of the stock market. The stock market is a complex and dynamic system that is influenced by a wide range of factors, including macroeconomics indicators, company news, political events and investors' sentiment.

There are three main categories of methods used for stock price prediction. Traditional methods include fundamental and technical analysis, which extract relevant indicators from basic information about the target company and the historical prices of the stock. With the emergence of machine learning and text mining techniques, researchers start extracting textual features from stock-related text data such as financial news and social media, under the assumption that market sentiment is a strong indicator of stock price movements. The last category involves using extracting image features extracted from candlestick charts using convolutional neural networks.

For text-based methods, traditional features such as term frequency-inverse document frequency (TF-IDF) and bag-of-words (BOW) frequency features are widely used in combination of support vector machine (SVM). On the other hand, high-level features such as word embeddings are extracted for deep learning prediction models. For example, [1] used a fine-tuned BERT model to extract contextualized embeddings and used recurrent neural network (RNN) for stock movements prediction. [2] used a financially improved BERT model, FinBERT, for sentiment features extraction from financial news and used a long-short term memory (LSTM) model to predict the stock price on the 11th day given the previous 10 days' data. [3] evaluated and compared the effectiveness of combinations between different types of embeddings (context-free and contextualized) and different learning algorithms (SVM, CNN and LSTM), where BERT+CNN and Word2Vec+CNN gave the best results.

For image-based methods, many studies demonstrated the effectiveness of the image features extracted from candlestick charts generated using historical stock prices [4-6]. [4] proposed a convolutional neural network (CNN) architecture that uses candlestick charts as image inputs and was able to predict stock movements with an accuracy of above 90% on the Taiwan stock market dataset collected. However, [5] claimed that the results shown in [4] was suspiciously high and was unable to be reproduced even iwth the same model configuration and dataset. [6] proposed a noval method based on the graph structure embedding method that can represent candlestick chart features more accurately compare to models including SVM, CNN, and LSTM.

More recently, researchers start exploring multimodal approaches for stock price prediciton. For example, [7] used a feature fusion LSTM-CNN model to predict stock prices based on the different representations of the stock historical prices: time-series data and image data (candlestick charts, line charts, filled line charts, and bar charts). It demonstrated the superior prediction power of leverage multimodal data compare to using only the time-series or image data as input. [8] proposed a method based on graph neural network to model the multimodal signals (time-series data and text data) for accurate stock forecasting. [9] used a 1D-CNN to extract sentiment features from stock related tweets and a 2D-CNN to extract image features from candlestick charts. Finally, it fused the multimodal feature sand make predictions on stock price movements.

Despite these advances in multimodal methods, there is a gap in the literature regarding the fusion of high-level text (social media) and image (candlestick charts) features for stock price movements prediction. This work aims to explore the use of pre-trained models (FinBERT and ResNet50) for extracting high-level features from stock related tweets and candlestick charts. We also proposed three attention-based feature fusion models (self-attention, cross-attention, Transformer-based) for fusing the extracted multimodal features and make predicitons on stock price movements. We compared the proposed models against several baselines and demonstrated the effectiveness of the proposed feature extraction and fusion methods.

2 Methodology

In this section, we present our attention-based methods for multi-modal stock movements prediction using self-supervised embeddings. We start by introducing the image representation (candlestick charts) of the historical stock prices data. We then describe the feature extraction process using pre-trained self-supervised models. Next, we discuss the self- and cross-attention mechanisms used for capturing inner- and inter-modality information. Finally, we explain the feature fusion method based on the Hadamrd product.

2.1 Candlestick Charts

Candlestick chart are a type of image representation of the historical stock prices data. It is a powerful tool commonly used by traders in technical analysis of stocks to identify patterns and trends. As shwon in Figure 1 a sample candlestick chart used in this work, each candlestick consists of stock information (opening, closing, high and low prices) on a specific day. Moreover, the colors of the candlesticks indicate whether the stock had a higher (green, upward trend) or lower (red, downward trend) closing price than the opening price at that day.

Considering the rich information a single candlestick chart can provide, we transformed all the historical prices of the stocks from time-series data into their corresponding candlestick charts. To align with previous work [10], we used a 5-day observation window where each candlestick chart consists of 5 candlesticks. We chose black as the background color to make the candlesticks more prominent. We performed the transformation using the *mplfinance* library which is built on top of the *Matplotlib* library and provides tools for easy visualization financial data. To remove uninformative part of the transformed images, we performed center-cropping to discard purely black space at the edge of the images.

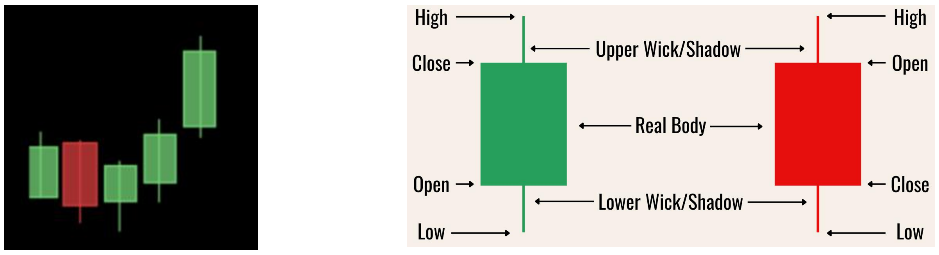

Figure 1. Left: A 5-day candlestick chart used in this work. Right: Detail annotations of the candlesticks.

2.2 Feature Extraction with Pre-trained Models

To extract high-level features from the text and image data, we used two pre-trained models: FinBERT and ResNet50. We downloaded the checkpoints of these models through the Hugging Face's *Transformers* library and the *torchvision* library, which both provide an easy-to-use interface for extracting features using pre-trained models. Due to the limited computational resources at hand, no fine-tuning was carried out.

2.21 Image Feature Extractor: ResNet50

ResNet, which stands for Residual Network, is a convolutional neural network (CNN) architecture proposed by Kaiming He et al. in 2015. It uses residual connections, which allow information to bypass one or more layers, to address the problem of vanishing gradients in deep neural networks. Through residual connections, ResNet can not only learn the desired mapping between the inputs and outputs, but also learn the difference between the inputs and outputs. ResNet has achieved state-of-the-art performance in various computer vision tasks, such as image classificaiton, object detection, and semantic segmentation.

ResNet50 is a specific varaint of the ResNet architecture that consists of 50 layers, including convolutional layers, pooling layers, fully connected layers and a global average pooling layer. ResNet50 can be devided into several stages as shown in Figure 2. Stage 1 performs initial feature extraction using a single convolution layer. Stage 2-5 each contains 3, 4, 6, and 3 residual blocks respectively where a single residual block consists of two 3×3 convolutional layers, followed by batch normalization and ReLU activation. Finally, it ends with a global average pooling layer and a fully connected layer with softmax for the default task of image classification.

We used a ResNet50 model that is pre-trained on the famous ImageNet dataset with a classifcation task as our feature extractor. After pre-training on a large amout of images, the ResNet50 model learnt the high-level representation of images. Specifically, we input the candlestick charts to the pre-trained ResNet50 model and used the output before the fully connected layer as our encoded features with a size of 1024. The input images were preprocessed (resized and normalized) the same way as that during the pre-training of the ResNet50 model.

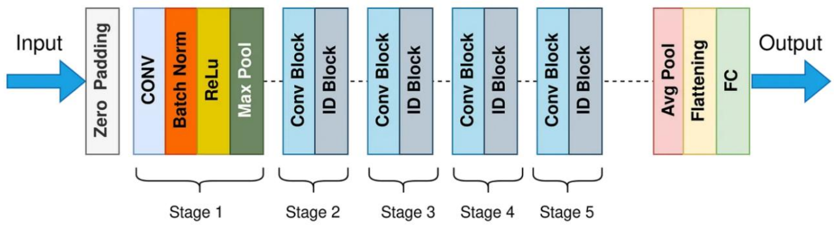

Figure 2. Architecture of the ResNet50 model in terms of blocks and stages.

2.22 Text Feature Extractor: FinBERT

Bidirectional Encoder Representations from Transformer (BERT) is a state-of-the-art pre-trained language model based on the Transfromer architecture. BERT consists of 12 layers of the Transformer's encoder block with the purpose to generate contextualized word representations for downstream tasks. BERT uses two pre-training objectives: masked language modelling (MLM) and next sentence prediction (NSP), to learn high-level representations of words using a large corpus of text. As shown in Figure 3, BERT learns the semantic meanings of the sequential text input through the NSP task by predicting the next sentences and learns the contextualized information of words through the MLM task by predicting the masked words.

FinBERT is built by further training the pre-trained BERT on a large financial corpus (46143 documents, around 29M words) with the same pre-training tasks (MLM & NSP). Moreover, FinBERT is fine-tuned on 4845 labelled sentences from financial news on the downstream task of sentiment analysis. In such way, FinBERT is more capable in capturing financial related information as well as sentiment clues within specialized financial text data compare to BERT. The movements of stock prices is often referred as an reflection of the market's sentiment. Therefore, we consider FinBERT as a suitable feature extractor that can be used to obtain infroamtive representations of the stock related financial news we used in this project.

To preprocess the raw financial new data, we used FinBERT's default tokenizer form the Transformers library. The tokenizer takes text sentences as input and output sequences of token IDs that can be fed directly to the FinBERT model. As mentioned above, we used a 5-day window to generate training samples. Therefore, we concatenated all the news within the 5-day window and truncated the sequences to a maximum length of 256 in order to reduce the computational expense. We then used the tokenizer to preprocess the concatenated and truncated sequences and finally use the FinBERT mode to extract features with an embedding size of 768.

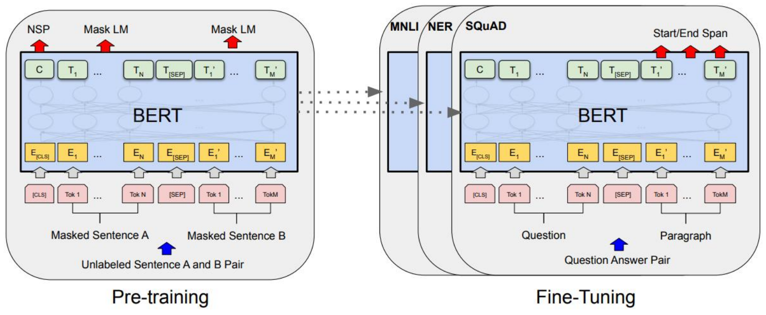

Figure 3. Architecture of the BERT model.

2.3 Feature Fusion Methods

We used self- and cross-attention for exploiting the inner-modality and inter-modality information between the text and image features. Moreover, we leveraged the Transformer's encoder block for more complex modeling. Before the final prediction layer, we experimented with two feature fusion methods: Hadamard product and statistical pooling.

2.31 Inner-Modality Modelling: Self-Attention

We used self-attention to selectively attend to parts of the input text sequences that provide most valuable information that might be related to the stock price movements. For example, sentiment clues like "increased sales" and "decline in sles" may be given higher attention weights within a text sequence and therefore guiding the model to focus on these informative parts. Self-attention is computed as shown in Equation 1 where all three inputs Query (Q), Key(K) and Value (V) come from the same input text sequence.

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d}})V \qquad (1)$$

To take step further, we used multi-head self-attention using the *nn.MultiheadAttention* module, allowing the model to capture the complex information within the input text data in different subspaces.

2.32 Inter-Modality Modelling: Cross-Attention

Cross-attention serves the purpose of sharing and fusing information across the two modalities. The computation is the same as that shown in Equation 1 except that the Query (Q) comes from the image modality and the Key (K) and Value (V) come from the text modality. Therefore, the output of the cross-attention module is a weighted sum of the text sequential data with attention weights representing the similarity between an element in the text sequence to the image feature. Through cross-attention, the model can learn to focus on the relevant parts between the input text and image data and understand the association between the trends observed in the images and the sentiment clues in the text.

Similar to self-attention, we used the *nn.MultiheadAttention* (MHA) module to perform multi-head cross-attention. Since MHA requires the inputs to have the same embedding size, linear layers were applied before MHA to reduce the dimensions of both the image and text data to a fixed and common value.

2.32 Transformer-based Modelling

To improve upon the simple cross-attention method, we leverged the Transformer's encoder block to perform more complex inter-modality modelling, as shown in Figure 4. Besides multi-head cross-attention, there exist a position-wise feed-forward network (FFN), layer normalization (LN) layers and residual connections. The FFN layer performs computation as depicted in Equation 2 where each element of the input is linearly transformed twice with a ReLU activation function in between for non-linearity. The LN layers standardize the input across the feature dimension, ensuring a consistent distribution of the activation values during training, and therefore prevent gradient vanishing and slow convergence. Finally, the residual connections bypass the cross-MHA layer and the FFN layer to create a better flow of information through the network.

$$\mathrm{PWFF}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \qquad (2)$$

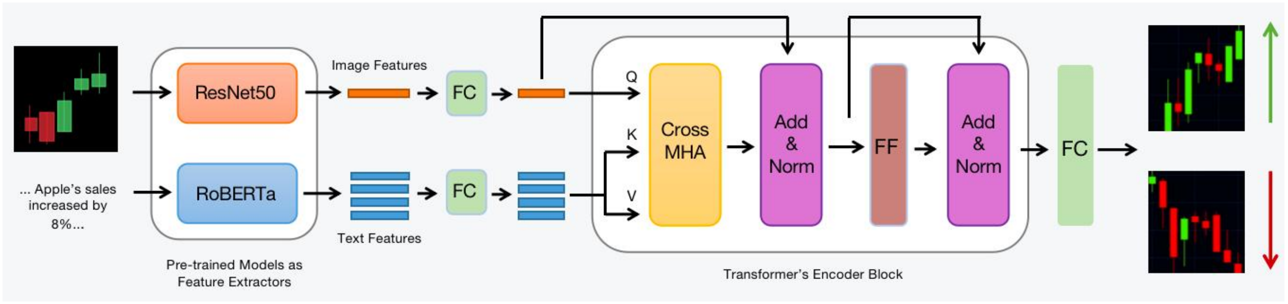
... Apple's sales increased by 8%...
ResNet50
RoBERTa
Pre-trained Models as Feature Extractors
Image Features
Text Features
FC
FC
Q
K
V
Cross MHA
Add & Norm
FF
Add & Norm
Transformer's Encoder Block
FC

Figure 4. Architecture of the proposed Transformer-based multi-modal fusion model.

2.34 Fusion Methods: Hadamard Product

Unlike the model with cross-MHA and the Transformer's encoder block, the model using self-MHA does not have a fusion mechanism and therefore ends up with 2 output vectors corresponding to image and text. We need to fuse the 2 output vectors prior to the final prediciton layer. Before performing any fusion, we first applied temporal average pooling over the sequential text features to obtain a single embedding. With the two vectors (text embedding and audio embedding), Hadamard product computes the element-wise product as shown in Equation 3. Intuitively, the

Hadamard product can extract the mutual information among the text features and image features. Privous work also demonstrated the effectiveness of Hadamard product in similar applications.

$$\mathrm{HP}(\mathrm{A}, \mathrm{B})_{\mathrm{ij}} = \mathrm{A}_{\mathrm{ij}} \times \mathrm{B}_{\mathrm{ij}} \qquad (3)$$

2.4 Binary Classification

We are aiming to predict the stock price movements as either going up or down, i.e. binary classification. Therefore, the prediction layer is a fully connected layer with output dimension of 1. The predicted logit is then passed through a sigmoid function to obtain the probability that the stock price will go up.

3 Experiments

3.1 Dataset and Problem Formualtion

To produce comparable results, we used the the publically available stocknet [10] dataset which has been extensively used by previous work on the task of stock price movements prediction with financial text data. It contains 2-year daily price movements from 2014/01/01 to 2016/01/01 of 88 stocks, as well as the tweets related to these stocks within the 2-year period. This dataset consists of two main components: historical stock prices and related tweets. The historical prices were collected from Yahoo Finance and the tweets were retrieved through Twitter's API under the official license. The 88 stocks were selected by taking the top 10 stocks in captical size in each of the 8 industries (Basic Materials, Consumer Goods, Healthcare, Services, Utilities, Financial, Industrial Goods and Technology) and all the 8 stocks in the Conglomerates industrial.

We define the problem as a binary classification task with a 5-day window. That is, to predict whether the closing price of a stock of the 6th day is higher (positive) or lower (negative) than the that on the 5th day, given the historical prices and the tweets in the privious 5 days. To align with previous studies, we label the samples based on the percentage of movements where $\geq 0.55\%$ and $\leq -0.5\%$ were the two thresholds for positive and negative labels. Samples with movement percentages within the variation thresholds were simply removed. The entire dataset originally contains 43430 samples in total. After applying the 5-day sliding window and labelling, we ended up with 18543 samples with a fairly even distribution between the two classes: 50.2% (positive) and 49.8% (negative). Time-series data cannot be shuffled and randomly split. Therefore, we temporally split the dataset into training (2014/01/01-2015/08/01, 14040 samples), validation (2015/08/01-2015/10/01, 1910 samples) and test (2015/10/01-2016/01/01, 2593 samples) sets.

3.2 Implementation Details

The entire framework were implmenet in PyTorch. The first step was to preprocess the raw data and extract features using pre-trained models. The model checkpoints were downloaded through Hugging Face's *Transformer* library. The embedding sizes of the text and image features were set to be the default values of the corresponding models: 768 and 1024. The maximum sequence length of the text data was truncated to 256 for computational efficiency. Then, we split the extracted features into training, validation and test sets and create the corresponding data loaders.

We used three commonly used evaluation metrics for binary classification: accuracy, f1-score, Matthews Correlation Coefficient (MCC) score. MCC is a balanced metric that is more informative as it takes into account the imbalance between the positive and negative classes, as well as the misclassification rates of both classes. We trained the models on the training set over a maximum of 30 epochs and save the best checkpoint with the best MCC score obtained on the validation set. Finally, we recorded the performance of this best model on the test set. This training process was repeated 5 times for each model and we report the average values of the metrics.

Over an extensive hyperparameter tuning process, we finalized our choice of hyperparameters as shown in Table 1. To avoid being trapped by local optimal during training, we adopted a training strategy called learning rate scheduling where we reduce the learning rate by a factor of 0.5 when there is no decrease seen in validation loss for 3 consecutive epochs. Furthermore, we leveraged early stopping to prevent overfitting when the validation loss hasn't been improved over 8 consecutive epochs.

Number of Attention Heads	6
Dropout-out Rate	0.1
Batch Size	32
Learning Rate	0.0001

Table 1. Hyperparameters used in training.

3.3 Compare Against Baseline Models

To examine the effectiveness of the proposed models, we compared the performance agaisnt the following baseliens:

- Random Guessor: a random model that randomly guess the outputs as either 0 or 1 with equal probabilities
- ARIMA: Autoregressive Intergrated Moving Average, a time-series forcasting model widely used in finance, uses only the time-series historical prices data as input
- Image-CNN: a convolutional neural network model, takes only the candlestick charts as input
- Text-CLF: a single modality model, uses only the text features to make predictions directly
- Image-CLF: a single modality model, uses only the image features to make predictions directly

The three proposed attention-based multi-modal models are:

- Text-Image-SA: the bi-modal feature fusion model using self-attention
- Text-Image-CA: the bi-modal feature fusion model using cross-attention
- Text-Image-Transformer: the bi-modal feature fusion model using the Transformer's encoder block

4 Results and Analysis

As shown in Table 2, Text-Image-SA achieved the best accuracy and MCC among all the models., following by Text-Image-Transformer and Text-Image-CA. It can be observed that all the multi-modal models outperformed the single-modal models, proving the effectiveness of the proposed attention-based methods in capturing useful information from multi-modal data. From the performance of the single-modal models, we see that both text and image features can provide valuable information on their own for predicting stock price movements, beating the Random Guessor and ARIMA baselines. Among the single-modal models, Image-CNN was the only one that did not manage to beat the ARIMA baseline with all the metric values much lower that that of Image-CLF. This suggests that the high-level features extracted using pre-trained models are more informative than raw images as input, therefore empirically validated the effectiveness of our proposed feature extraction method. It is also worth noticing that the MCC scores of all the proposed models are positive, suggesting their real capabilities in capturing the underlying pattern in the data.

Model	**Accuracy**	**F1**	**MCC**
Random	0.499 ± 0.003	0.504 ± 0.002	-0.002 ± 0.007
ARIMA	0.513 ± 0.003	0.514 ± 0.003	-0.021 ± 0.003
Image-CNN	0.509 ± 0.011	0.515 ± 0.091	0.017 ± 0.011
Image-CLF	0.529 ± 0.004	0.619 ± 0.050	0.044 ± 0.007
Text-CLF	0.523 ± 0.005	0.627 ± 0.033	0.021 ± 0.008
Text-SA	0.523 ± 0.003	0.565 ± 0.016	0.015 ± 0.010
Text-Image-SA	$\mathbf{0.530 \pm 0.009}$	0.633 ± 0.058	$\mathbf{0.048 \pm 0.009}$
Text-Image-CA	0.524 ± 0.008	$\mathbf{0.646 \pm 0.059}$	0.033 ± 0.015
Text-Image-Transformer	0.529 ± 0.002	0.638 ± 0.048	0.043 ± 0.007

Table 2. Experimental results of the models' performance (averaged over 5 repeated trials).

5 Discussion

In this work, we demonstrated the effectiveness of the proposed attention-based models in leveraging multi-modal (text and image) data for stock price movements prediction. We evaluated the proposed models and some baselines models on the stocknet dataset and shown that the proposed model outperformed all the baselines. However, it is neccesary to further validate the proposed models on multiple dataset (ideally with different stocks and time periods) to check for generalization. This was not implemented in this work due to the limited time and computational rescources, but strongly recommended to be done in the future. For future work, it is also encouraged to perform trading tests with strategies based on the predictions of the models to further validate the potential investment values that the models might have. One key limitation of this work is that candlestick charts generated using daily stock price data might be insufficient for image-based models to learn informative features, due to the limited number fo candles presented in each chart. However, we were unable to retrieve minute-level stock prices correponding to the stocks and time-periods used in the stocknet for this project. Truncation of the text data also limits the informative being passed to the model. Longer sequence of text input could be used in future work where computational rescourse is sufficent.

References

[1] Q. Chen. Stock Movement Prediction with Financial News using Contextualized Embedding from BERT, 2021

[2] S. Halder. FinBERT-LSTM: Deep Learning based stock price prediction using News Sentiment Analysis, 2022

[3] W. C. Lin, C. F. Tsai, and H. Chen. Factors affecting text mining based stock prediction: Text feature representations, machine learning models, and news platforms, 2022

[4] R. M. I. Kusuma, T.-T. Ho, W.-C. Kao, Y.-Y. Ou, and K.-L. Hua. Using Deep Learning Neural Networks and Candlestick Chart Representation to Predict Stock Market, 2019.

[5] Y. Wang. Financial Time Series Classification using Convolutional Neural Network and Candlestick Charts, 2022

[6] J. Wang, X. Li, H. Jia, T. Peng and J. Tan. Predicting Stock Market Volatility from Candlestick Charts: A Multiple Attention Mechanism Graph Neural Network Approach, 2022

[7] T. Kim, and H. Y. Kim. Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data, 2019

[8] R. Sawhney, S. Agarwal, A. Wadhwa, and R. R. Shah. Deep Attentive Learning for Stock Movement Prediction From Social Media Text and Company Correlations, 2020

[9] T. -T. Ho, and Y. Huang. Stock Price Movement Prediction Using Sentiment Analysis and Candlestick Chart Representation, 2021

[10] Y. Xu and S. B. Cohen. Stock Movement Prediction from Tweets and Historical Prices, 2018