Stock Price Prediction System*

Chanyoung Lee[1], Yujin Seo[2], and Donghun Jung[3]

[1] Sungkyunkwan University, Computer Science and Engineering, Republic of Korea
[2] Sungkyunkwan University, Department of Mathematics, Republic of Korea
[3] Sungkyunkwan University, Department of Physcis, Republic of Korea

Abstract. This project aims to predict the stock price of some selected stocks in KODEX 200 and S&P 500 on a day-by-day basis. Leveraging data sourced from the Korea Exchange (KRX) and Nasdaq, we will undertake the training and evaluation of a diverse set of machine learning models, including Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Transformer. We focus on predicting the stock price of the next a few days. In a contrast to the general stock price prediction system of many stock firms, we do not analyse the fundamental of the company. We focus on the short-term changes of the stock price, and attempts to reflect the recent issues via VADER.

Keywords: Stock Price Prediction · Machine Learning

1 Introduction

In today's capitalist world, stocks present an attractive avenue for financial gain, offering significant profit potential. Investors, regardless of their depth of understanding of the stock market, are constantly seeking reliable methods to predict future stock prices, as their financial well-being often hinges on these predictions.

Predicting stock prices is fundamentally about discerning the real value of a company. Knowing a company's true value helps in determining whether its stock price will rise or fall from its current level. However, this is a complex task influenced by a myriad of factors, including global events and economic conditions.

Historically, professional fund managers have been pivotal in guiding investments in the stock market. However, their strategies and predictions are susceptible to personal biases. This limitation has spurred interest in leveraging machine learning for more objective and unbiased stock price forecasting.

While numerous services today use machine learning for this purpose, many are not open-source and are primarily geared towards professional investors. These systems, often complex and tailored for experienced users, focus more on asset allocation than on precise stock price prediction. In contrast, our project targets the general public, who rely more on simple chart analysis and intuition, without delving deep into a company's fundamentals or news events.

* Supported by LINC

Our proposal is to predict the short-term movement of blue-chip stocks, a strategy known to be effective over periods ranging from a few days to a few months. We aim to make short-term predictions, not in real-time, to assist general users and simplify the prediction process.

To achieve this, we have employed machine learning techniques, utilizing models such as Long Short-Term Memory(LSTM), Gated Recurrent Unit(GRU), and Transformer. These models have been instrumental in successfully predicting stock prices for the next few days by capturing short-term trends. In addition, we have integrated VADER, a sentiment analysis tool, to reflect recent issues and capture both short- and long-term stock price changes based on current events.

Our system has been deployed as a publicly accessible web service, offering users easy access to our predictive model. This deployment significantly contributes to investment decision-making, lowering the entry barrier for those interested in stock market investments. Empirical evaluations of our system have shown promising results in capturing short-term trend of stock price movements, thereby validating our approach and offering a valuable tool for investors.

2 Design for the Proposed System/Solution/Service

2.1 LSTM

Overview of LSTM LSTM is a sophisticated architecture within Recurrent Neural Networks (RNNs), designed to address the limitations of traditional RNNs in capturing long-range dependencies in sequential data, a challenge often compounded by the vanishing gradient problem. The hidden layer of an LSTM consists of memory cells interconnected through a series of gates: the input gate, forget gate, and output gate, which collectively facilitate short-term memory storage.

1. Forget gate: Marked by a blue box in the figure, the forget gate determines which information from the previous state should be discarded or retained. Mathematically, it is represented as:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{1}$$

   where f_t is the output of the forget gate layer at the time step t, W_f, U_f, b_f are the weight matrices and bias vectors for the forget gate computation, respectively.
2. Denoted by an orange box, the input gate decides what new information should be stored in the cell. It operates as follows:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{2}$$

$$g_t = tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{3}$$

   where i_t is the output of the input gate layer at the time step t, g_t is the candidate value to be added to the output layer at the time step t, W_i, W_c,

U_i, U_c, b_i, b_c are the weight matrices and bias vectors for the input gate and candiate value computation, respectively.

3. Output gate: Shown as a gray box, the output gate determines which information from the cell should be used to generate the output. It is defined by:

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{4}$$
$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{5}$$
$$h_t = o_t \odot tanh(c_t) \tag{6}$$

where o_t is the output of the output gate layer at the time step t, c_t is the status of the cell at the time step t, h_t is the output of the LSTM unit at the time step t, W_o, U_o, b_o are the weight matrices and bias vectors for the output gate computation, respectively.

Compared to the traditional RNN, LSTM performs various mathematical operations, including including element-wise multiplication and addition, to control the flow of information and perform updates to the memory cell and hidden state.

Compared to traditional RNNs, LSTMs perform a variety of mathematical operations, including element-wise multiplication and addition, to control the flow of information and update the memory cell and hidden state. This architecture enables LSTMs to effectively handle long sequential data, making them capable of capturing time-dependent patterns in the data.

Background of employing LSTM Stock prices are typically regarded as time series data, characterized by their sequential nature and the presence of temporal dependencies. LSTM is particularly well-suited for modeling such data due to their ability to capture both short-term and long-term dependencies and patterns. This capability stems from their unique architecture, which allows them to remember information over extended periods and forget irrelevant data, a critical feature for analyzing the often volatile and non-linear patterns observed in stock market data.

In the context of stock price prediction, LSTMs can learn from historical price data, identifying underlying trends and patterns that are indicative of future movements. This makes them a powerful tool for financial analysts and investors seeking to make informed decisions based on predictive analytics. The use of LSTM in this domain is driven by the hypothesis that past stock performance, along with other relevant financial indicators, can provide valuable insights into future price trajectories.

2.2 GRU

Overview of GRU GRU is a type of RNN architecture, offering a simpler alternative to LSTM. Despite their simpler structure, GRUs have proven highly

effective in various applications. They are designed to address the vanishing gradient problem, enabling RNNs to better capture long-range dependencies in sequential data. A distinctive feature of GRUs, as compared to LSTMs, is their unified approach to managing cell state and output, which simplifies the architecture.

GRUs utilize two main gates for their operation:

1. Update gate: The update gate determines how much of the past information needs to be passed along to the future.

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{7}$$

2. Reset gate: The reset gate decides how much of the past information to forget.

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{8}$$

These gates help the GRU to make decisions about what information is relevant to keep from past time steps and what can be discarded, enabling it to capture dependencies over different time scales effectively.

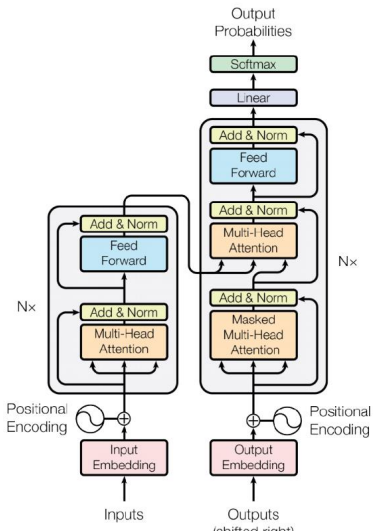
Background of employing GRU Similar to LSTMs, GRUs are employed with the expectation that they can capture both short-term and long-term dependencies and patterns in sequential data. GRU, with their simplified structure, offers an efficient way to model these complexities.

The choice of GRU for stock price prediction is motivated by its efficiency in training and its effectiveness in handling sequences where the gap between the relevant information and the point where it is needed is unknown or variable. This makes GRUs particularly suitable for analyzing financial time series data, where they can learn from historical price movements to predict future trends and patterns.

2.3 Transformer

The Transformer is a neural network architecture that has revolutionized the field of natural language processing (NLP) and has shown promising potential in time series prediction tasks. Unlike traditional RNN-based models like LSTM and GRU, the Transformer relies entirely on an attention mechanism.

Overview of Transformer Transformer operates as an Encoder-Decoder model, leveraging an attention mechanism. In the Encoder-Decoder architecture, the Encoder takes an input sequence and encodes the information into a single context vector. Conversely, the



Decoder utilizes this context vector to generate an output sequence. Within the model structure, a crucial component is the "embedding" process. This process involves the conversion of input values into a unified vector representation.

The attention mechanism employed by the Transformer is a key feature. It assigns varying weights to elements within the input sequence, placing greater emphasis on pertinent information. This emphasis is then reflected in the model's output. The Transformer employs this mechanism to comprehensively evaluate the significance of the entire input sequence when generating the output.

The utilization of the Transformer model holds the promise of delivering superior performance. It has the capacity to incorporate diverse sources of information such as news, stock indices, and corporate disclosures, distinguishing it from other models.

Background of employing Transformer The decision to employ the Transformer model in stock price prediction stems from its advanced capabilities in handling sequential data. While LSTM and GRU have made significant strides in addressing the vanishing gradient problem, they still have limitations, particularly in their sequential processing nature and in fully capturing long-range dependencies. The expectation is that the Transformer, with its advanced architecture, will outperform traditional models like LSTM and GRU in capturing the intricate patterns and dependencies inherent in stock price data, leading to more accurate and reliable predictions.

2.4 Loss function

Overview of Loss function In our model, the loss function is defined as:

$$\mathcal{L} = \mathcal{L}_H + \lambda \sqrt{\sum \left(\frac{\hat{y}_t - y_t}{y_{t-1} - y_t} \right)^2} \tag{9}$$

where $\mathcal{L}_H$ is Huber loss function, λ is a regularization parameter, y_t is the actual value at time step t, and $\hat{y}_t$ is the predicted value at time step t. The Huber loss function is given by:

$$\mathcal{L}_H = \sum_{t=1}^{n} \begin{cases} \frac{1}{2}(y_t - \hat{y}_t)^2 & \text{tf } |y_t - \hat{y}_t| \leq \delta \\ \delta |y_t - \hat{y}_t| - \frac{1}{2}\delta^2 & \text{otherwise} \end{cases} \tag{10}$$

The Huber loss function is chosen for its balanced approach, combining the robustness of the L1 norm with the smoothness of the L2 norm. This makes it differentiable and less sensitive to outliers in the data. Visually, the Huber loss resembles the L2 norm for smaller errors but transitions to an L1 norm-like behavior for larger errors, effectively handling outliers.

Additionally, we incorporated a regularization term into the loss function. This term is designed to provide some leniency for rapid changes in stock prices. It implies that the model is less penalized for missing rapid price changes, acknowledging the inherent volatility and unpredictability in stock price movements.

Background of employing Loss function Initially, our model employed the Huber loss function without a regularization term. However, we observed that the model tended to predict the next day's stock price as being equal to the current day's price, a pattern inconsistent with real-world stock market behavior. This phenomenon aligns with the martingale theory in probability, where the expected next value of a sequence of random variables is simply the current value.

Recognizing that stock prices are not merely a sequence of random variables but are influenced by trends and patterns, we introduced the regularization term. This addition encourages the model to recognize and respond to short-term trends in stock prices, rather than strictly adhering to the immediate past value. It allows the model to be more responsive to rapid changes, aligning its predictions more closely with the dynamic nature of the stock market.

2.5 VADER

Overview of VADER VADER (Valence Aware Dictionary and sEntiment Reasoner) is a rule-based model specifically designed for sentiment analysis of social media text. It excels in analyzing the sentiment of individual words and phrases, providing an overall sentiment score for the text. VADER employs a combination of lexical elements and grammatical heuristics to assess sentiment, effectively capturing both the polarity (positive or negative) and intensity of emotions expressed in the text.

One of VADER's strengths lies in its ability to adeptly handle the unique characteristics of social media text, such as slang, emoticons, and abbreviations, which are often challenging for traditional sentiment analysis models. Its performance in sentiment analysis tasks, especially in the context of social media, is notably high.

Background of employing VADER In our stock price prediction model, we integrated VADER to incorporate additional factors that could influence stock prices. While our model effectively captures short-term trends in stock prices—predicting rises or falls based on recent price movements—it sometimes struggles to anticipate when these trends might not hold.

To address this, we turned to external factors like news, which can significantly impact stock prices both in the short and long term. News sentiment, whether positive or negative, along with the intensity of the sentiment, can be pivotal in shaping stock market trends. VADER's ability to evaluate these sentiments provides a nuanced understanding of how news might affect stock prices.

We use VADER to generate sentiment scores for news articles, incorporating these scores as new factors in our model. This approach aims to enhance the model's predictive accuracy by accounting for the influence of news sentiment on stock price movements, capturing new trends that might not be evident from historical price data alone.

3 Implementation

3.1 Data Set

Stock Selection We selected about 15 stocks from KODEX 200 and S&P 500 for predicttion. Also, we categorized the stocks into two groups: stable stocks and unstable stocks based on the commercial EFT funds holdings. The following table shows the stocks we selected.

Korea Stable	**Korea unstable**	**US Nasdaq Stable**	**US Nasdaq unstable**
Item 1	Item 2	Item 3	Item 4
Item 1	Item 2	Item 3	Item 4

Normalization

Training Set

Test Set

3.2 UI/UX Design

The user experience (UX) and user interface (UI) aspects of the system were prioritized to ensure a seamless and intuitive user interaction. The frontend was designed with a user-centric approach, focusing on usability and visual appeal. The UI elements were designed to be responsive and accessible across different devices and screen sizes, enhancing the overall user experience.

3.3 Frontend

3.4 Backend

Django, a Python web framework, was employed for server-side development.

4 Evaluation

4.1 Evaluation Metric

4.2 Result

Model	MPA MSE
LSTM	
GRU	
Transformer	
LSTM + VADER	

5 Limitation and Discussions

5.1 Prediction Quality

6 Related Work

Recent research trends in stock price prediction include advancements in deep learning-based regression models. [1] These models often utilize Long Short-Term Memory (LSTM) networks and innovative validation techniques, such as walk-forward validation, to enhance their predictive capabilities.

In addition, some researchers have explored Particle Filter Recurrent Neural Networks (PF-RNNs), a new RNN family explicitly designed to model uncertainty within their internal structure. [2] Unlike traditional RNNs that rely on a deterministic latent state vector, PF-RNNs maintain a latent state distribution approximated as a set of particles. To enable effective learning, researchers have introduced a fully differentiable particle filter algorithm that updates the PF-RNN latent state distribution based on Bayes' rule. Experimental results have shown that PF-RNNs can outperform conventional gated RNNs across various domains, including synthetic robot localization datasets and real-world sequence prediction tasks, which is stock price prediction.

Furthermore, recent studies have proposed novel approaches, such as the development of a sentiment-ARMA model, which combines the autoregressive moving average model (ARMA) with sentiment analysis of financial news articles. [3] This model is integrated into an LSTM-based deep neural network

consisting of three components: LSTM, VADER model, and a differential privacy (DP) mechanism. The proposed DP-LSTM scheme has demonstrated the potential to reduce prediction errors and enhance model robustness. Extensive experiments conducted on S&P 500 stocks have indicated promising results, including a 0.32% improvement in mean Mean Percentage Absolute Error (MPA) and a significant up to 65.79% reduction in Mean Squared Error (MSE) for the prediction of the market index S&P 500.

7 Conclusion

We capture the short-term trend of stock price movements, thereby validating our approach and offering a valuable tool for investors.

References

1. Li, X., Li, Y., Yang, H., Yang, L., Liu, X.Y.: Dp-lstm: Differential privacy-inspired lstm for stock prediction using financial news. arXiv preprint arXiv:1912.10806 (2019)
2. Ma, X., Karkus, P., Hsu, D., Lee, W.S.: Particle filter recurrent neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 5101–5108 (2020)
3. Mehtab, S., Sen, J., Dutta, A.: Stock price prediction using machine learning and lstm-based deep learning models. In: Machine Learning and Metaheuristics Algorithms, and Applications: Second Symposium, SoMMA 2020, Chennai, India, October 14–17, 2020, Revised Selected Papers 2. pp. 88–106. Springer (2021)