Explainable AI model for Stock trading

Seonghyun Ban[1], Dongyoung Choi[2], and Minseung Lee[3]

[1] SungKyunKwan University
2066, Seobu-ro, Jangan-gu, Suwon-si, Gyeonggi-do, Republic of Korea
bansh123@skku.edu
[2] ttiger0114.skku.edu
[3] mkms3580@skku.edu

Abstract. With the rapid development of artificial intelligence technology, attempts to incorporate it into the financial industry have also increased. Numerous approaches have been proposed to deal with the stock market price prediction problem. However, these approaches suffered from difficulties in generalizing the stock market and lack of explanation. Therefore, in order to solve these two problems, we would like to apply a so-called "divide and conquer" approach and methodologies that has contributed greatly to visualizing the decision-making of models such as Attention and Grad-CAM.

Keywords: Divide and Conquer · Explainable AI

1 Introduction

1.1 Importance of the suggested problem

Robo-Advisor is a word with a combination of "robot" and "advisor" and is a service in which algorithms analyze the stock market to manage investors' assets or seek advice instead of experts. As the domestic robo-advisor market grows, attempts to develop deep learning-based robo-advisors are increasing[1]. However, these attempts had two major problems.

First, most of the approaches were intended to be solved by generalizing the stock market. Developing a model that can sustain acceptable profit returns took a lot of resources and time due to rapid volatility and many variables in the stock market. As many researches on the model of visual and language perception were conducted in various ways, theories on the correlation between the capacity, accuracy of the model and the distribution of data were established. [2][3] The more complex data classes are, the larger capacity of the model required to solve the problem, and the more data used for training the model tends to be trained more precisely. Therefore, we plan to cluster data close to manifold space through heuristic ways or unsupervised learning and use it for model learning to effectively solve the problems with growing the required model capacity and reducing learning efficiency due to various variables in the stock market and complex data distribution.

Second, for deep learning-based services, the lack of explainability which means the interpretation of the process is not clear has been consistently pointed out as a problem. In order to officially launch a robo-advisor product in Korea, it is necessary to explain algorithms that investors can understand. Unlike conventional machine learning, deep learning does not explain accurate or intuitive decision-making processes. Due to these features, when numerous deep learning-based robo-advisor products are commercialized in the future, investors will be damaged by unclear differentiations and explanations in product algorithms. We argue that to solve this problem, the algorithm provided by robo-advisor needs explanation with minimal visualization tools, and we expect this to be established as the basis for providing transparent products to investors in the future.

1.2 Brief introduction on previous approaches

Before solving the problem through existing deep learning with zero-shot and end-to-end ways, the sequence of methods - clustering data, developing each artificial intelligence model specialized for each cluster, and then ensemble - was used. We do not use ensembles to construct models, but we would cluster data and create models specialized for each cluster.

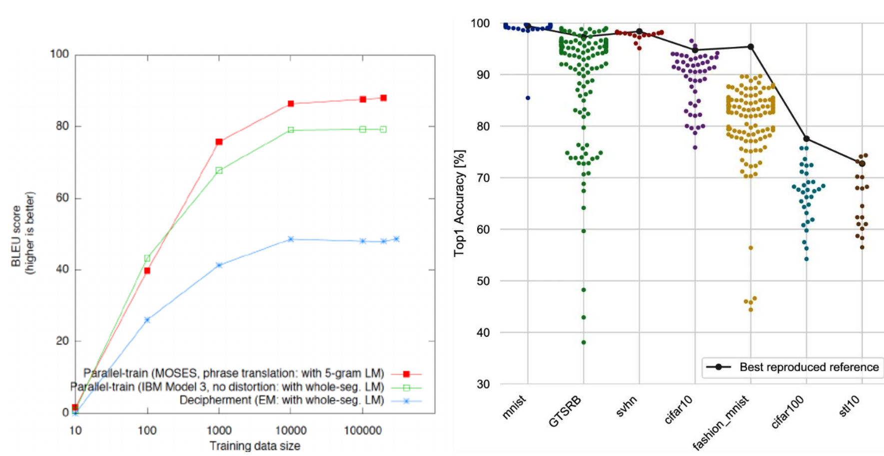


Fig. 1. Fig. 1 shows that the larger the data size used for NLP (natural language processing) model training, the greater the accuracy, **(Left)** and that the more complex the data distribution is, the less accurate it is, making it difficult to train the model. **(Right)**

Since Attention and Grad-CAM derived from natural language processing and image processing can be applied to 1D data with the same mechanism, There are cases in which those methods for expressing the weight of data for model decision are applied to 1D data. [5]

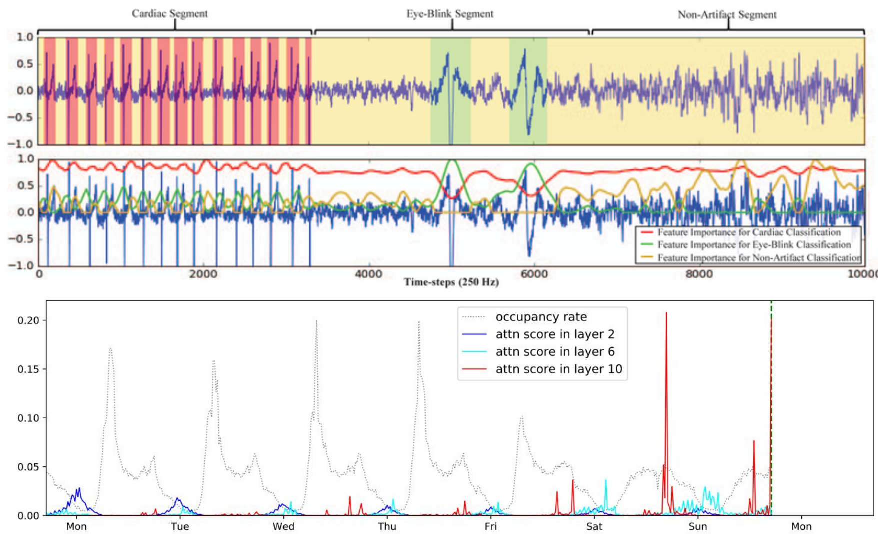


Fig. 2. Fig. 2 shows an example of applying visualization of model decision-making with one-dimensional data, using grad-CAM **(Up)** and Attention **(Bottom)**

2 Motivation and Objective

Deep learning models tend to learn better with simple tasks, so we argue that model learning can be made easier if data is subdivided into simpler environments through heuristic algorithms or unsupervised learning such as PCA, t-SNE, and autoencoder. The robo-advisor market requires explanation on the algorithm of the model in order to commercialize the robo-advisor product. On the other hand, since deep learning-based algorithms are specialized in problem-solving and focus on producing consequential outputs, users cannot determine the basis for decision of deep learning models or which data the model focuses on. Due to the characteristics of deep learning models being non-deterministic, despite the model that produces good returns, if the service provider or user does not grasp the model's algorithm or basis for decision, the service provider may not provide minimal safeguards for model being exposed to unrecognized external variables. For this reason, we predict that in the robo-advisor market, where the upcoming deep learning model is dominant, various confusion will occur in the future if the market is filled with unexplainable robo-advisor products.

Proper regulations are needed for these "black box" products, and this study will be a good opportunity to provide the basis for standardization to establish these regulations. In addition, as the market share of deep learning-based models increases, derivatives such as portfolio optimization can be expected, identifying the characteristics of the model and providing customized products to users. Therefore, the goal of this project is to develop a model that makes reasonable profits through segmentation of stock charts and to create an explainable and transparent robo-advisor by combining a methodology that visualizes the model's decision-making process.

3 Background

Various algorithms have been proposed to establish a stock trading strategy. Neural network models such as RNN [6] have also been introduced, following genetic algorithms [7]. In addition, reinforcement learning, in which agents recognize the current state and learn how to maximize cumulative rewards within a given environment, is regarded as an appropriate way for learning the stock market with rapid changes and many interference factors, and is also actively used to establish stock trading strategies [8].

4 Problem Statement and Proposed Solution

4.1 It is ideal to train a well-generalized model, but it consumes a lot of time and resources.

Attempts to train the stock market from the past aimed at producing a model with generalized performance for a given indicator, regardless of methods such as reinforcement learning, machine learning, and supervised learning. But we point out some problems about previous approaches.

The first is that as mentioned above, time and resource consumption are large. It is definitely ideal to train a single model using data under all trading conditions. However, as can be seen in BERT and GPT-3 in the field of natural language processing, training a very large, precise and well-designed model required to train complex and diverse data distribution, takes a lot of time, and hardware resources such as high-performance GPU and memory.

The second problem is that the stock market is a very complex and 'fluctuating' environment heavily affected by external variables. In fact, in order to train a model that generally works well in all situations, not only the numerical data of the stock market, but also information such as national policy, corporate news, and public response must be considered. Training an artificial intelligence model with data that does not include these important external variables can lead to unexplained decision-bias.

To complement the problems pointed out above, we plan to cluster data through heuristic algorithms and unsupervised learning and focus on data distribution with similar latent features to conduct model learning. This method is expected to solve the first problem to some extent by limiting the environment in which we want to train. In order to select stocks in real time using the current open API, data must be selectively provided through a condition search formula that returns only stocks that meet specific conditions to select stocks of interest first. In addition, the shorter the stock data interval, the closer it is to short-term investment, which is less affected by external variables. For example, if tick unit data is used within a very short period of time, it is expected that the impact of external information such as national policy or corporate news, which is not expressed as daily data, can be reflected in the data before and after that. However, since training using tick-unit data requires excessive time and hardware resources, we would like to compromise by using minute-unit data to complete the project within a semester.

4.2 Lack of explainability of artificial intelligence can adversely affect the robo-advisor market.

The deep learning model has recently been in the spotlight in various fields because the amount of information that can be expressed nonlinearly is very large and easy to train. However, at the same time, the problem of lack of explainability was also raised because it was difficult to determine which part of the given input data had an effect on the output of the determined model. And unlike the market focusing on user convenience, these issues are more sensitive to financial industries where evidence and variables are important, consequently robo-advisor products are obligated to explain the algorithm. We would like to solve the above problem by using various methodologies that help visualize the decision-making process of deep learning models such as grad-CAM and Attention. We expect that significant alleviation of lack of explanation will help to introduce standard policies for robo-advisor products by securing clarity and specificity of robo-advisor algorithms and warning incompleteness of unexplained artificial intelligence models.

5 Planning in detail

The overall project is carried out through the following process.

1. **Data acquisition and preprocessing**
2. **Development of artificial intelligence model**
3. **Construction of backend system**
4. **AI model evaluation**
5. **Construction of visualization system**

5.1 Data acquisition and preprocessing.

– **Who is in charge of this** : Seonghyun Ban, Minseung Lee

- Using ‘Daeshin Securities’ Open API collects stock market data for two years.
- Remove outliers in the data or handle with missing values
- Generate label data. Label data is used as the ground truth for model training in process 2. Label depends on the structure and definition of the model. For example, if the model is defined as a binary classification model that makes outputs only between 'Buy' and 'Hold on', the following example defines the Label value.

  UP: When the closing price rose 3% compared to the last time-series data within a set time period.

  HOLD: Even after a certain amount of time, the stock price fluctuated between 3 percent and -3 percent, or the closing price fell by 3 percent compared to the last time series data.

  Based on the last time series data to be considered, there are various hyperparameters such as how long to observe, time, stock price, how much stock price rises to predict a rise, and the key is to find the appropriate label definition for features of the data and model through trial and error.
- Analyze data distribution and cluster data using heuristic methods or unsupervised learning

5.2 Development of artificial intelligence model

- **Who is in charge of this** : Seonghyun Ban, Minseung Lee
- The model is trained using the data collected in process 1. The model has a structure suitable for handling time series, such as 1D-CNN and Transformer.
- Evaluate the trained model by the validation set of offline data and select the model with the best accuracy.
- Basically, a model is a binary classifier that can take two actions, 'Buy' and 'Hold on', and the detailed definition of this model could be changed flexibly in the process of improving the model's performance and problems.

5.3 Construction of backend system

- **Who is in charge of this** : Dongyoung Choi
- Based on 'Kiwoom Securities' open API, real-time data processing programs, so called backend systems, will be implemented, and socket programming will help to transmit and receive packet-based data between AI models and backend systems.
- In order to limit only clustered data to test targets, the backend system prepares a search formula for conditional transactions that reflects the characteristics of each cluster well, and then automatically subscribes to only the resulting stocks of the search method through the Kiwoom Securities Open API to receive data in real time.

- If certain trading conditions are met within the selected stock, it is appropriately preprocessed and packetized to the AI model.
- When the output result of the model receiving the preprocessed stock information is ‘Buy’, the stock code and result are retransmitted in packet form to the backend system.
- The backend system that receives the stock code and ‘Buy’ purchases the stock at the market price through the API, and the quantity is autonomously adjusted according to the current portfolio.
- Among the purchased stocks, stocks that exceed the threshold of losses (-3%) or profits (+3%) are automatically sold.

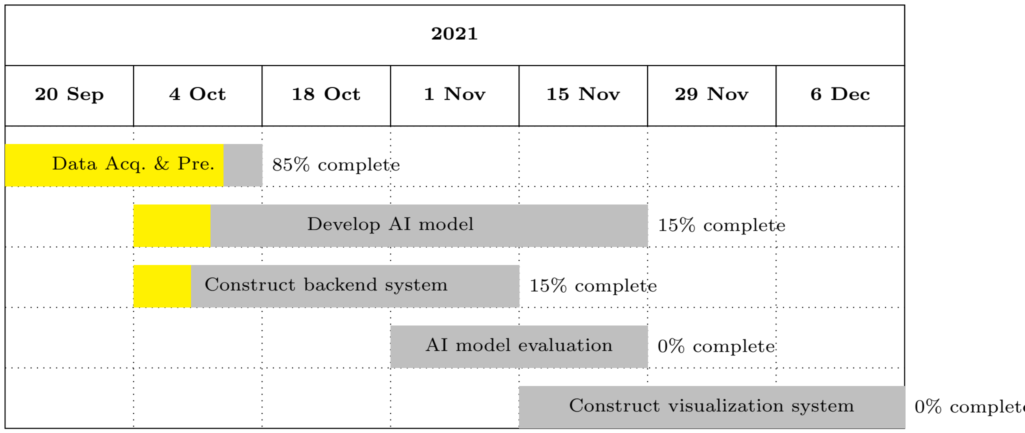
5.4 AI model evaluation

- **Who is in charge of this** : Dongyoung Choi
- Real-time data is transmitted to the model through the backend system, and mock investment is conducted by reflecting the model’s decision on a given data.
- The simulated investment runs from a day to a week, and the yield will be counted. It is desirable to verify the model by conducting mock investment for at least a month, but it seems to be too much time compared to the project period and we are planning to proceed with mock investment for a minimum period of time.
- In Process 2, models with an error range of 2-3% between the validation yield observed in offline data and the mock investment yield are considered appropriate and used in Process 5.

5.5 Construction of visualization system

- **Who is in charge of this** : Seonghyun Ban, Dongyoung Choi and Minseung Lee
- A visualization system is constructed using a model verified in Process 4. If 1D-CNN is used as a mode network, visualization is performed using grad-CAM, and if Transformer is used as a model network, Attention will be used.
- Using the backend system of Process3, the original data filtered by the condition search formula, the metadata and the weight values of the data for a model decision are transmitted to the frontend.
- The front end is implemented as python, but if possible, it extends to the web form using Java scripts. The front end that received the data weight aims to identify the physical location of the indicator data used on the actual stock trading GUI and paint the color according to the weight in real time using pyautogui.

5.6 Brief schedule for the progress of ongoing tasks



References

1. Koscom. (2021). Trends in RA market size. https://www.ratestbed.kr:7443/portal/bbs/B0000006/list.do?menuNo=
2. Sujith Ravi and Kevin Knight : Deciphering Foreign Language. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (2011), pages 12–21
3. Scheidegger, F., Istrate, R., Mariani, G. et al. Efficient image dataset classification difficulty estimation for predicting deep-learning accuracy. Vis Comput 37, 1593–1610 (2021).
4. Garg, P., Davenport, E., Murugesan, G., Wagner, B., Whitlow, C., & Maldjian, J., Montillo, A. . Automatic 1D Convolutional Neural Network-based Detection of Artifacts in MEG acquired without Electrooculography or Electrocardiography. International Workshop on Pattern Recognition in NeuroImaging. International Workshop on Pattern Recognition in Neuroimaging, (2017).
5. LI, S., Jin, X., Xuan, Y., Zhou, X., Chen, W., Wang, Y., & Yan, X. Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting. Advances in Neural Information Processing Systems 32 (2019).
6. iJADE stock advisor: an intelligent agent based stock prediction system using hybrid RBF recurrent network IEEE Transactions on Systems, Man, Cybernetics, Part A: Systems and Humans, 34 (3) (2004), pp. 421-428
7. A hybrid genetic-neural architecture for stock indexes forecasting Information Sciences, 170 (1) (2005), pp. 3-33
8. Jangmin O, Jongwoo Lee, Jae Won Lee, Byoung-Tak Zhang, Adaptive stock trading with dynamic asset allocation using reinforcement learning, Information Sciences, Volume 176, Issue 15, (2006), pp 2121-2147