Financial Transaction Forecasting using Neural Network and Bayesian Optimization

Mark Wu*,
University of Wisconsin-Madison,
Madison , WI,
Hwu378@wisc.edu,

Abstract— The stock price forecasting has been a well-known topic attracting many researchers and investors. How to buy in a low price and sell it in a high price is an eternal focus. Until now, there are two schools about the stock market forecasting: fundamental analysis and technical analysis. This paper is mainly about the technical analysis which utilized the historical time-series of market data, such as trading price and volume to predict whether we do financial transaction or not. In this paper, we propose to use the stock trading action selection prediction model based on neural network and a hyper-parameter optimization method. Our method can efficiently analyze attributes of different dimensions to make predictions better. We evaluated our trading behavior on the Jane Street dataset provided by the kaggle competition. The results of the experiment show that our method achieves a good performance compared with other machine learning methods. On the dimension of unity score we proposed to evaluate the performance of model, our model's unity score is higher 2111 and 3179 higher than ResNet and XGBoost model.In addition, we also studied to get the better hyperparameters with the Bayesian optimization algorithm.

Index Terms—Transaction Predicting, technical analysis, Neural network, hyper-parameters optimization, Bayesian optimization

I. Introduction

In today's world, predicting stock price as a popular topic has attracted both investors and researches. And the issue about how to buy in a low price and sell it in a high price has raised. Owing to the short-term trading that could enables many transactions occur within one second. It is feasible for traders to find and take advantage of price differences in real time. In a perfect market, the price of stock will never be overpriced and undervalued. On the contrary, the financial markets will never completely effective in a real world. Therefore, how to develop a better trading strategies to identify the use of blind spots in the market is a popular direction and difficult question.

Throughout the algorithmic trading framework, stock forecasting components collect information from different sources, such as news and historical trades, because stock markets are characterized by high volatility, dynamism, and complexity[1][2][3]. Any little movements in stock markets are influenced by several factors. If we use the traditional method to forecast the stock price, it will be a challenging task. Until now, there are two schools about the stock market forecasting: fundamental analysis and technical analysis. The fundamental analysis needed the financial analyses of companies or industries.

Furthermore, thanks to the development of the technology and further study on machine learning, a plethora of methods has been applied in stock price forecasting. So in this paper we devote to using the neural network algorithm to solve the problem about stock price forecasting and with the Bayesian optimization algorithm, we get the optimal hyper-parameters of the network. The data is from Jane Street Market provided by Kaggle website for the prediction of transaction strategy. Python is the main tool for data processing, modeling and analysis.

A. Related Work

The profitability of stock market investment depends to a large extent on the predictability of stock movements. Investment risk and uncertainty can be minimized if the forecasting model or technology accurately predicts the direction of the market. This will increase investment in the stock market and help policymakers and regulators make appropriate decisions and take corrective action.

Based on the efficient market hypothesis[4], an event happens on a stock would affect the price of the stock. According to this, researches and traders have been mining the event information of the stock from many resources: news, social media and discussion board. [5]

The technical analysis predicts the stock trend based on the historical time-series of market data, such as trading price and volume.
Many machine learning algorithms have been applied in forecasting. [6]

A survey about usage of Machine Learning for stock market prediction is reported in Reference [7]. The authors provide a detailed benchmark about usage of Support Vector Machine (SVM), Random Forest, K-Nearest Neighbor (KNN), Naive Bayes, and Softmax in stock prediction. The experimental results confirmed that Random Forest algorithm performs the best for large datasets, and Naïve Bayesian Classifier is the best for small datasets.

What's more, because of the rise of deep learning, some studies attempted the neural network in the task of stock price forecasting. To further model the long-term dependency in time series, recurrent neural networks (RNN), especially Long Short-Term Memory (LSTM) network [8], had also been employed in financial perdition [9,10,11,12]. Besides, to improve the performance of technical analysis, some recent efforts [13,14] leveraged Graph Neural Networks [15] to capture the relationships between different stocks.

B. *Our Contribution*

- ✓ We propose a model based on Neural network, owing to the autoencoder that can effectively mine different feature contribution values to solve the task of whether the instructions issued by the trading strategy should be executed.
- ✓ During the feature engineering, we fill the missing feature value using the mean value. And we show the missing feature distribution image.
- ✓ We optimize our model and use the Bayesian Optimization to obtain the optimal hyper-parameters.
- ✓ We introduce experimental results and apply unity score as indicator, comparing our model with other model on the same dataset provided by Jane Street Market.

II. Feature Engineering

In this part, we introduce data sets and feature engineering.

We get data that from Jane Street Market provided by Kaggle website. The dataset of market prediction contains training dataset showing historical data and returns and features dataset indicting metadata pertaining to the anonymized features .

There are 2,390,491 transaction sample points in the training dataset. Each row in the dataset represents a trading opportunity, for which you will be predicting an action value: 1 to make the trade and 0 to pass on it. Each trade has an associated weight and resp, which together represents a return on the trade. The date column is an integer which represents the day of the trade, while ts_id represents a time ordering.

It is worthy to note that there are 5 values of this return, which represent the return of different time frames. Because the transaction has a cycle of closing positions, different cycles may have different returns. This is a complete data set, so transactions with a weight of 0 are also included (17%), although such transactions is not helpful to the evaluation, and they were eventually deleted. In addition to anonymized feature values, the website also provide us with metadata about the features in features.csv.

III. Methodology

In this section, we will introduce our model in detail. We firstly show our whole model and then explained further about the AutoEncoder and the Bayesian Optimization.

The structure of our neural network is shown in figure 2. The network we designed own three dense layer and the data is inputted to both the traditional dense layer and the encoder section.

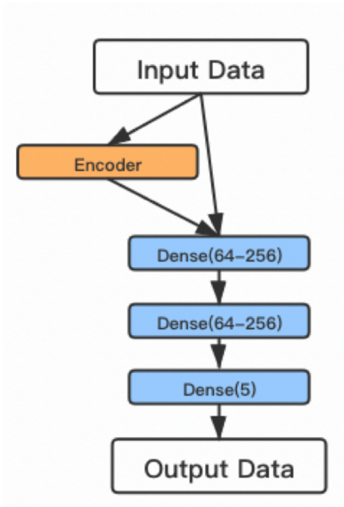

Figure 1: the structure of our model

- ➢ Auto Encoder

Autoencoder is a neural network designed to learn identity functions in an unsupervised manner to reconstruct the original input while compressing the data in the process to discover more efficient and compressed representations.

In our paper, the structure of AutoEncoder is shown in the following figure 2. With this network we can extract the main features and obtain the better performance. There are two networks in the AutoEncoder. They are encoder network and decoder network respective. The encoder network could translate the original high-dimension input into the latent low-dimensional code. While the decoder network could recover the data from the code, likely with larger and larger output layers.

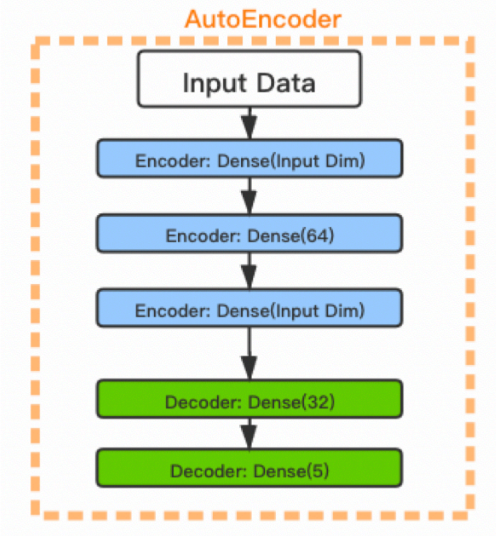

Figure 2: auto-encoder' structure

IV. EXPERIMENTS

In this section, we applied our model on the DataSet of Jane Street Market. We use a utility score as the evaluation function. The higher the utility score is, the better the performance will showed. And we use bayesian optimization to optimize our model, obtaining better the hyper parameters which could make our model perform surprisingly .
Each row in the test set represents a trading opportunity for which you will be predicting an action value, 1 to make the trade and 0 to pass on it. Each trade j has an associated weight and resp, which represents a return.

For each date i, we define:

$$p_i = \sum_j (\text{weight\_ij} * \text{resp\_ij} * \text{action\_ij})$$

$$t = \frac{\sum p_i}{\sqrt{\{\sum p_i^2\}}} * \sqrt{\left\{\frac{250}{|i|}\right\}}$$

Where |i| is the number of unique dates in the test set. The utility is then defined as:

$$u = \min(\max(t, 0), 6) \sum p_i$$

In table1 ,three models our model, ResNet and Xgboost with their Unity scores are shown.

Models	**Unity**
Proposed Model	**9331**
ResNet	7220
XGBoost	6182

table 1; different models with their unity score

It is obvious to see that our proposed model have a higher unity score. The higher the unity score is, the better performance we will obtain. Our proposed model's unity is 9331, the ResNet is 7220 and Xgboost is 6182. The unity of our model is 2111 and 3179 higher than other model.

➢ Bayesian Optimization

On account of many optimization problems in machine learning are black box optimization problems that we do not have an analytical expression for the mapping function nor do we know its derivatives. Evaluation of the function is restricted to sampling at a point x and getting a possibly noisy response.

Hyper-parameters are parameters that the model cannot learn directly from the data during training. For example, the learning rate in random gradient drop algorithm, for the sake of computational complexity and algorithm efficiency, we can not directly learn from the data a relatively good learning speed. However, the learning rate is very important, the larger learning rate is not easy to converge the model to a more suitable smaller value solution, while the smaller learning rate often makes the model training speed greatly reduced. For hyper-parameters such as learning rates, we usually need to set them before we train the model. As a result, fine-tuning hyper-parameters can be painful for complex models with many hyper-parameters. Bayesian optimization is an approximate approximation method based on post-test probability.

In our experiment, we use the Bayesian optimization algorithm to optimize our model. We get the epoch, batchsize and the number of neurons in each layers as well as the number of middle layers. It is shown in the table 2.

Hyper-parameter	**quantity**
epoch	3
neurons in each layers	64-256
middle layers	3
batchsize	4096

table 2: hyper-parameters in our model

V. CONCLUSIONS

In this paper, a new method based on neural network had been proposed. We used the auto encoder to extract key features and combined the Bayesian optimization to obtained the better hyper parameters. The data is from the Jane Street market provided by the kaggle platform. We do experiments which shows that our proposed model effectively comprehensively processes data of different dimensions, and performs better than ResNet, even better than XGBoost with default parameters. We know the profit ratio of the parameters and features, and got some interesting and constructive guidance. In the future, we can improve the model and data based on this information to improve the forecasting effect and apply the model to other market forecasting problems.

References

[1]Almudhaf F (2018) Predictability, Price bubbles, and efficiency in the Indonesian stock-market. Bull Indones Econ Stud 54(1):113 – 124

[2]Al-Shaib M (2006) The predictability of the Amman stock exchange using Univariate autoregressive integrated moving average (ARIMA) model. Journal of Economic and Administrative Sciences 22(2):17 – 35

[3]Asadi S, Tavakoli A, Hejazi SR (2010) A new hybrid for improvement of auto-regressive integrated moving average models applying particle swarm optimization. Expert Syst Appl 39:5332 – 5337

[4]Awajan AM, Ismail MT, Wadi SA (2018) Improving forecasting accuracy for stock market data using EMD-HW bagging. PLoS One 13(7):1 – 20

[5]Diebold FX, Marino RS (1995) Comparing predictive accuracy. J Bus Econ Stat 13(3):134–144

[6]Harrison B, Moore M (2012) Stock market efficiency, non-linearity, thin trading and asymmetric information in MENA stock markets. Economic Issues 17(1):77–93

[7]Humala A (2013) Some stylized facts of return in the foreign exchange and stock markets in Peru. Stud Econ Financ 30(2):139–158

[8]Hung SL, Adeli H (1993) Parallel backpropagation algorithms on CRAY Y-MP8/864 supercomputer. Neurocomputing 5(6):287 – 302

[9]Makridakis S, Wheelwright SC, Hyndman RJ (2015) Forecasting: methods and applications. Wiley India, New Delhi

[10]Mondal P, Shit L, Goswami S (2014) Study of effectiveness of time series Modelling (ARIMA) in forecasting stock prices. International Journal of Computer Science, Engineering and Applications 4(2):13 – 29

[11]Mostafa MM (2010) Forecasting stock exchange movements using neural networks: empirical evidence from Kuwait. Expert Syst Appl 37:6302 – 6309

[12]Robert D Edwards, WHC Bassetti, and John Magee. 2007. Technical analysis of stock trends. CRC press.

[13]Navid Rekabsaz, Mihai Lupu, Artem Baklanov, Alexander Dur, Linda Andersson, and Allan Han- ¨ bury. 2017. Volatility prediction using financial disclosures sentiments with word embedding-based ir models. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Vancouver, Canada, volume 1, pages 1712 – 1721.

[14]Wang JZ, Wang JJ, Zhang ZG, Guo SP (2011) Forecasting stock indices with backpropagation neural network. Expert Syst Appl 38:14346 – 14355

[15]Winkler RL, Makridakis S (1983) The combination of forecasts. J R Stat Soc 146(2):150–157