An Attention Based Multi-gate Mixture-of-Experts Model for Quantitative Stock Selection

Keyao Li* and Jungang Xu

***Abstract*—As a kind of typical quantitative trading strategy, quantitative stock selection has attracted increasing attention of investors in recent years, and the application of many traditional economic models and machine learning models to the stock selection has yielded quite valuable results. However, most of the existing research are still limited to the learning of a single target, which does not serve the needs of multiple investment objectives well. To address these issues, we propose an Attention based Multi-gate Mixture-of-Experts (AMMOE), which is a multi-tasking model obtained by combining the MMOE and attention modules. We apply this model to extract information from stock characteristics using correlations among stock indicators to predict different stock indicators simultaneously, improve the predictive performance of each target, and provide a valuable reference for portfolio construction. The experimental results show that all portfolios with the AMMOE model achieve the highest returns and significant advantage in most backtesting metrics compared to other machine learning models.**

***Index Terms*—Multi-task learning, quantitative stock selection, portfolio construction, backtesting**

I. Introduction

With the investment market matures and the variety of financial instruments continues to increase, quantitative investment, as an emerging investment approach, is attracting more and more attention for its stability and efficiency compared to traditional investment approaches. The idea of quantitative investment is analyzing the performance of markets or financial assets using data analysis methods to achieve specific investment objectives, such as obtaining stable returns or reducing trading risks [1]. Currently, quantitative investment have been applied to almost aspects of the investment, including stock selection, market timing, asset scheduling and risk management.

The effectiveness of quantitative investment depends on the ability to analyze data, which is consistent with the strength of machine learning in data processing [2]. In fact, with the improvement of data-driven machine learning theory, the collection and analysis of massive market data including securities, industries, sectors, company statements, news policies, etc. has become possible [3]. Research in quantitative investment has gradually shifted from the study of economic theory to the study of computer algorithms and models [4]. However, compared to data analysis in other fields, there are some challenges for the processing and application of quantitative data, including:

1) The validity of the quantitative data characteristics and the correlation with stock indicators are highly volatile. On one hand, volatility within the market or the industry, macro-regulatory policies and the operating status of companies can affect the performance of stocks. On the other hand, stocks are a part of the market, thus the analysis of stock characteristics also needs to take into account the effects of time and space [5]. For general data, more features is beneficial for model training, but as the market environment change, the utility of different features may change. Some marginal factors may gradually dominate, and some factors that were effective in the past environment may lose their values, which make the analysis of quantitative data extraordinarily difficult.
2) Considering the investment behavior itself, investors often need to meet multiple investment objectives. On one hand, investment is subject to many different factors, thus investment behavior that relies on a single target is risky. On the other hand, there is a correlation between different investment objectives, and forecasts for a single target cannot take full advantage of this correlation. For example, excessive pursuit of return requires a correspondingly high level of risk. Ideal investment is high return and low risk, but the two ones are never independent.

Multi-Task Learning (MTL) is an important branch of machine learning. Many realistic problems cannot be solved by dividing them into separate sub-problems, thus the approach of decomposing complex problems and processing them with single-task models yields incomplete results [6]. Unlike most machine learning models for a single task, multi-task learning models take multiple tasks with correlation as learning targets at the same time and exploits the correlation between these tasks to better capture the complex properties of several events or several scenarios in the real world [7]. In addition, multi-task learning can satisfy the constraints of multiple tasks simultaneously by sharing the knowledge learned from different tasks, which is equivalent to a regularization method to avoid overfitting of individual tasks [8].

The key of multi-task learning is the sharing of information between different tasks. According to the way of information sharing, multi-task models can be divided into two categories, namely hard parameter sharing [9] and soft parameter sharing [10]. Early multi-task learning is dominated by hard parameter sharing models, i.e., all tasks share the same hidden layer, and the model structure is solidified with serious negative transfer. With the development of deep learning, especially neural networks, many difficulties in the design of parameter sharing structures have been solved, soft parameter sharing models have become the mainstream of current

Manuscript received December 15, 2022; revised March 28, 2023; accepted June 25, 2023.

K. Li and J. Xu are with the School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing, China.

*Correspondence: likeyao20@mails.ucas.ac.cn (K.L.)

multitask learning research. The Multi-gate Mixture-of-Experts (MMOE) is one of the most representative research results of soft parameter sharing model [11]. MMOE shares the substructure called expert among all tasks and automatically optimize the weights assigned to each Expert for each task through the gating network. Compared to the model where the underlying structure is completely shared, MMOE is able to flexibly assign features with little increase in the number of parameters, enabling feature sharing while preserving separate parts for each task. The model and its variants are now widely used in areas such as online video recommendation systems [12, 13].

In this paper, we propose an attention based MMOE (AMMOE) model by adding one MLP attention module into MMOE, and then apply the AMMOE to predict two stock indicators, i.e., excess return and moving average convergence / divergence (MACD). Based on daily stock data of the Chinese Shanghai and Shenzhen stock markets from 2010 to 2021, we verify the validity of the AMMOE for quantitative stock selection problems. We also test the performance of other baseline models such as support vector machine (SVM), feedforward neural networks (FNN) and convolutional neural networks (CNN) on the same dataset, and compare the test results and backtesting results of each model to demonstrate that the AMMOE has better stability in dealing with complex multi-objective problems.

The rest of this paper is organized as follows. Section II reviews the related work on machine learning and multi-task learning in the field of quantitative investment, Section III presents the model we proposed. Section IV gives the experiments, including the content of the dataset, experimental methods and experimental results. Finally, in Section V, we conclude the work.

II. Related Work

Most of the early research on quantitative investment relied on economic theories, such as the efficient market theory. Researchers used mathematical and statistical methods to analyze the relationships between market factors and asset returns to propose interpretable solutions for asset allocation or to develop specific investment strategies [14]. Some of the representative researches include the momentum Alpha strategy based on the classical CAPM pricing model [15] and the three-factor risk model (TFRM) based on the risk premium [16]. Due to the limitation of computing power, the research at that time could not achieve large-scale data analysis, and was more of a supplement and improvement to the existing investment theory.

With the development of artificial intelligence and machine learning, some researchers have started to try to apply machine learning methods to the field of quantitative investment. Long before the deep learning theory was proposed, Shahpazov analyzed the performance of different neural network models in financial time series forecasting and verified the effectiveness of artificial neural networks in predicting the market ability [17]. Li *et al*. used the clustering methods based on K-means algorithm to analyze the investment efficiency of smart investment, making the analysis results more close to the actual [18]. Rapach used regression model with lasso regularization to control the systemic risk and constructed a sector rotation portfolio that achieved significant advantages compared to traditional methods [19]. Some researchers have also used machine learning methods to explore the characteristics of different market environments. Leippold et al. built a comprehensive set of return prediction factors using various machine learning algorithms, summarizing the characteristics that distinguish the Chinese market from the U.S. market, and suggesting more valuable references for investors in the Chinese market [20].

The success of deep learning has greatly expanded the research of quantitative investment. The effectiveness of deep neural networks in handling such tasks was verified in [21] by using deep learning methods to extend traditional models and applying them to financial decision making and stock movement prediction. Multimodal neural networks [22], reinforcement learning [23] and generative adversarial networks [24] have also made some progress in exploiting temporal and spatial information of stocks. These results show that the application of machine learning and deep learning in the field of quantitative investment has been widely accepted. However, market is a very complex entity and there is still a lot to explore. Multi-task learning is one example. It has been increasingly noticed by researchers of quantitative investment because of its advantages in dealing with complex problems.

In recent years, some results have been achieved in the application of multi-task learning. Bitvai used multi-task learning as a modulating tool to extend linear models to capture predictable patterns in market movements [25]. Huang et al. used a multi-task learning model based on tensor fusion to achieve the fusion of data from multiple sources for stock prediction on more dimensions [26]. A multi-task learning model based on the GRU [27] s proposed to exploit stock features from different industries. For now, the application form of multi-task learning in the field of quantitative investment is more as a supplement or extension of other models, and still suffers from reliance on large amounts of dataset and complex network structures.

III. Model

In this section, we first give the description of problem. Then we specifically introduce network architecture and each part of our model.

A. Problem Description

Quantitative stock selection does not rely on human emotional perceptions, but does rely on quantifiable stock features and specific predictive indicators. The prediction results of the indicator can reflect the value of the stock, and thus give us a reference for selection.

Specifically, we need to train a model based on daily stock data and daily stock market index data to predict the excess return and MACD of a stock for the next trading day. These two prediction targets represent the price movement trend of the stock on the trading day and the price movement trend of the combined long-term and short-term performance of the stock. Both these two indicators have proven effective in practice. More specifically,

1) **Excess return**. The stock data source is all listed stocks in the Shanghai and Shenzhen markets in China, and accordingly, the benchmark stock index used to calculate excess returns is the CSI (China Securities Index) 300. The stock's excess return is different between the daily return of the stock and the daily return of the CSI 300 index. Using excess return instead of stock return can alleviate the imbalance in the number of training labels in some datasets, which is beneficial for model training.
2) **MACD**. A comprehensive technical indicator that uses the dispersion and convergence of a stock's fast and slow averages to determine the long short position and movement trend of a stock. Given the stock closing price $close_i$ on trading day i, we can calculate the stock's long-term average EMA_{26_i} and short-term average EMA_{12_i} (generally cover 26 and 12 days respectively) and the difference between them, DIF_i, then the MACD value is obtained by calculating the moving average (generally covers 10 days) of DIF_i, DEA_{10_i}.

$$EMA_{N_i} = \frac{2close_i + (N-1)EMA_{N_{i-1}}}{N+1} \tag{1}$$

$$DIF_i = EMA_{12_i} - EMA_{26_i} \tag{2}$$

$$DEA_{N_i} = \frac{2DIF_i + (N-2)DEA_{N_{i-1}}}{N} \tag{3}$$

$$MACD_{N_i} = 2\left(DIF_i - DEA_{10_i}\right) \tag{4}$$

This technical indicator can be used to determine the trend of a stock based on the increase or decrease of the DIF_i and the positive or negative movement of the MACD indicator. It is particularly effective in predicting divergence.

Backtesting is the best way to evaluate the effectiveness of a quantitative stock selection strategy, i.e., to actually simulate investment behavior with assumptions about the investment target, investment environment and investment conditions. The backtest results reflect the actual value of stock or portfolio. Backtesting indicators we used include

1) **Portfolio Revenue**. Net income earned on investments.
2) **Sharpe Ratio**. It is defined as the difference between the expected return on investment R and the risk-free return R_f divided by the standard deviation of the investment return, representing the additional return gained by the investor for each unit of risk increase.

$$Sharpe\ Ratio = \frac{R - R_f}{\sqrt{Var(R)}} \tag{5}$$

3) **Information Ratio**. It is defined as the difference between the expected return on investment R and the benchmark return R_b divided by the standard deviation of investment return. It is similar to the Sharpe Ratio, but it can reflect the relative value of stocks.

$$Information\ Ratio = \frac{R - R_b}{\sqrt{Var(R)}} \tag{6}$$

4) **Maximum Drawdown**. It is defined as the maximum value of the individual stock retracement rate included in the portfolio. It means the maximum decline that the selected portfolio has experienced during the specified period.

B. Structure of Model

The structure of AMMOE is shown in Fig. 1, which mainly contains the attention module, expert networks, gating networks and high-level task structure. The attention module uses a trainable fully connected layer structure to automatically adjust the weights of the original features and improve the learning efficiency of the model. Multiple expert networks at the bottom of the model extracts different dimensions of the original features and then assigns weights to different experts through the gating network for each specific task. Experts and gating networks enable the whole model to learn both the shared features and respective independent features. Feature extraction, assignment and learning are all completed automatically during the training process.

For AMMOE containing n experts E_i $(1 < i < n)$ with input $x \in R^d$, it contains a task t with a corresponding gating network $g_t \in R^n$ and a high-level independent substructure H_t. The implementation of an expert in the AMMOE is essentially the same as a multi-layer perceptron with a ReLU activation function, but here the activation function layer is followed by the addition of dropout layer with 30% dropout rate, which is added to prevent expert from over-relying on certain inputs during feature extraction.

The g_t of each task has the free parameter $W_{g_t} \in R^{n\times d}$ for linear transformation of the input, and then the weight value assigned to each expert is obtained by the Softmax function. Similar to expert, a dropout layer with 10% dropout rate is added after the linear transformation layer of gating network to prevent the gating network from over-relying on certain experts. Dropout can avoid the unbalanced distribution of weights in gating network and enhance its learning effect. However, the number of experts is generally smaller than the number of features, so the dropout rate is also low.

We add a MLP attention module to the input layer of the model to preprocess the original features of quantitative stock selection. The effectiveness of stock features is affected by various factors, and the correlation between stock selection factors and indicators varies in different market environments, which means some weakly correlated features will impair the performance of the expert, resulting in insignificant differences across different experts. The attention module is equivalent to adjusting the importance of the original features according to the relevance between features and targets, thus the expert and the whole model pay more attention to the features with higher relevance to the prediction target.

The structure of MLP attention module is shown in Fig. 2, the actual input to expert and gating network is x_{att} instead of x. The implementation of this attention module is quite simple, so it does not impose too much extra burden on the model. The original input x is processed by two unbiased fully connected layers, and then the weights obtained from the Sigmoid function are used to adjust the dimension of x_{att} so that it is equal to that of the original input. To make this attention module better fit the model structure, the number of neurons in the first fully connected layer is equal to the number of experts in the AMMOE.

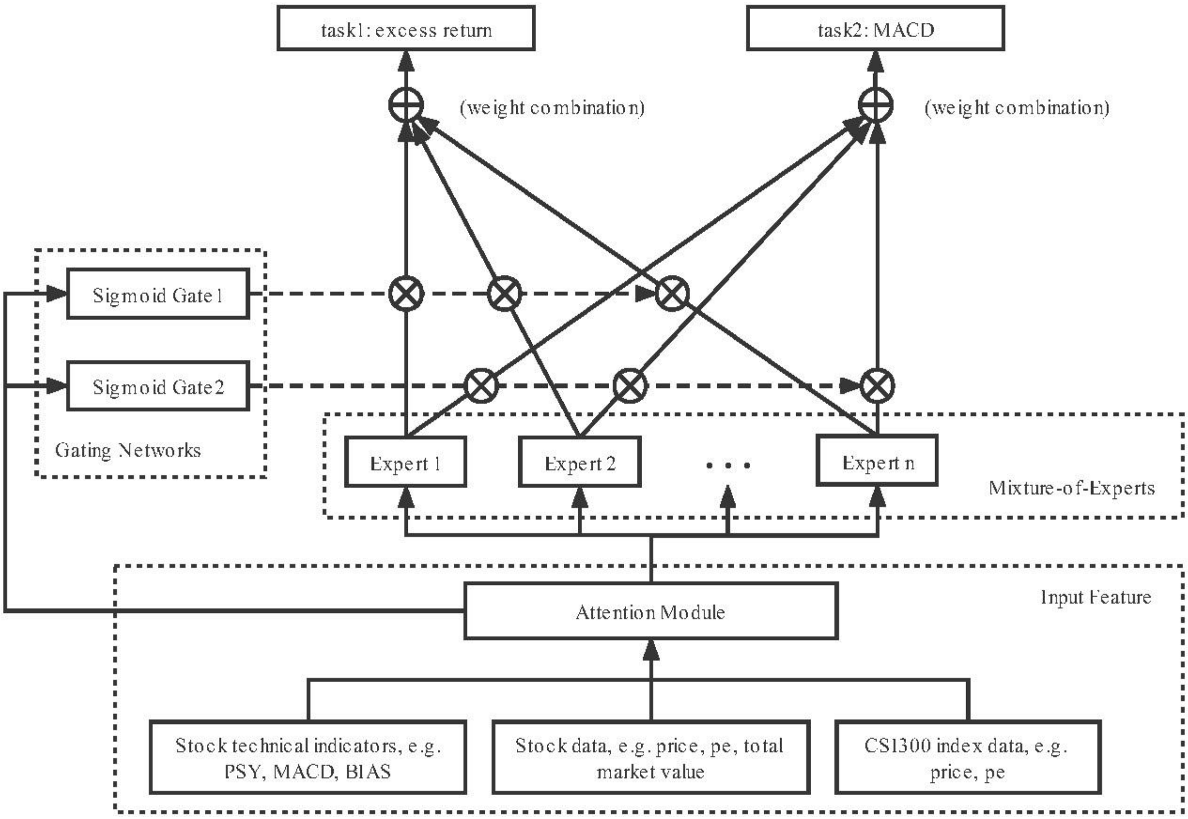

Fig. 1. The structure of AMMOE.

Summarize the above, the output of task t, y_t, is defined as follows:

$$y_t = H_t(i_t) \tag{7}$$

$$i_t = \sum_{i=0}^{n} g_t^{(i)}(x_{att}) E_i(x_{att}) \tag{8}$$

$$g_t = Softmax\left(W_{g_t} x_{att}\right) \tag{9}$$

In addition, the number of experts is considered as a hyperparameter in this model. However, in the case of fewer feature inputs, a smaller number of experts encourages the gating network to fully utilize the learning results of each expert [13].

IV. Experiments

A. Dataset

We select all listed stocks on the Shanghai Stock Exchange and Shenzhen Stock Exchange as our dataset during the period 2010/09 to 2021/09. Each stock sample in the dataset consists of three components: stock daily trading data, CSI 300 daily trading data and stock technical indicators. Appendix provides the details.

The label of each sample represents the excess return and MACD of the stock for the next trading day. The positive and negative excess return correspond to whether the sample is positive or negative. However, for MACD, we cannot directly get the label based on its original value. We need to decide the trend of the stock for the next trading day based on the MACD rules, which include the movement, positive / negative values of the DIF and MACD. If the trend is rising, then it is a positive sample, otherwise it is a negative sample. So we can indirectly get a prediction of the rising or fall of the stock price by predicting the positive or negative of these two targets. After transforming the labels, the similarity coefficient of these two targets is about 70%, which has a strong correlation.

In addition, in dataset, we remove some samples with small stock price changes to prevent them from interfering with the judgment of the model. Therefore, the sample of stocks with excess return between -0.01 and 0.01 will be excluded.

B. Experimental Design

To verify the effectiveness of AMMOE-based stock selection models, we constructed some baseline models for comparison, including feedforward neural networks (FNN), convolutional neural network (CNN), support vector machine (SVM) and random forest. For each of these models, we fully trained them under a variety of different hyperparameter configuration.

In the experiments, the width and depth of the FNN are the same as the width and depth of the single task substructure in AMMOE, with a maximum width of 16 and a maximum depth of 8, to ensure that they have the same model complexity. The training samples of CNN are stitched together from the same stock samples of 3 consecutive trading days, which is equivalent to manually constructing a "picture" of the stock.

This CNN dataset has only a very small difference in the proportion of labels compared to the original dataset. We use a convolution kernel of width 3 for the convolution operation. Except for the underlying convolutional layer, the other structures of CNN remain the same as FNN. For SVM and Random Forest, we will adjust their important hyperparameters to optimize the models.

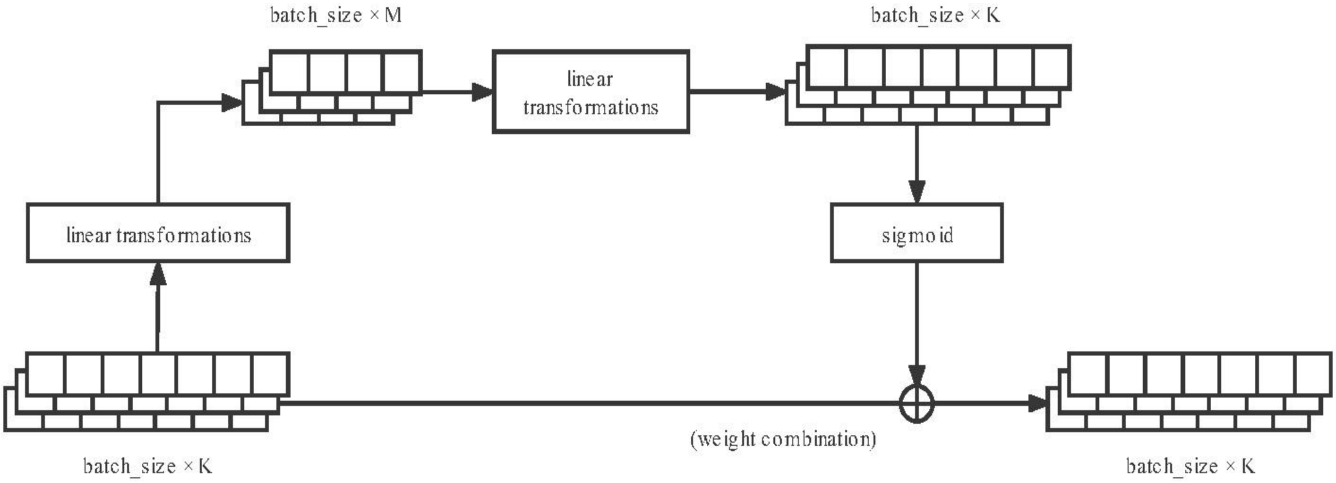

Fig. 2. The structure of MLP attention module.

We use MinMaxScaler to normalize the features of the training, validation and test sets so that the features are distributed between [0,1]. Because the size of single dataset is not large, the learning rate is set to 0.005, the batch size to 128, the number of training rounds to 600, and the model is trained with the Adam optimizer to prevent overfitting.

The model is trained using a rolling training approach. The number of samples is not enough if the training is done with a dataset of only one trading day. However, if samples from different trading days are mixed together to expand the dataset, the simultaneous presence of samples from several different datasets within the same epoch will lead to confusion in the training direction of the model, even affect the final training results. The market does not have large fluctuations in the short term in the absence of significant external influences [28], thus we use a rolling training approach with strict division of each original dataset. Specifically, the last two datasets in a month are used as the validation datasets, and the remaining datasets are used as the training datasets. After the model is trained on the previous training dataset, it continues to be trained with the next training datasets until the final model is obtained. Then, we use the datasets of top 5 trading days of the next month as the test datasets, apply the trained model to rank the stocks in the test datasets, select top 10 performing stocks.

Backtesting based on these five test datasets, we buy at the closing price of the day and sell at the closing price of the next day, then calculate backtesting indicators based on the returns from buying and selling.

We use three approaches to construct the portfolio: based on excess returns only, based on MACD only, and based on both excess returns and MACD. The last approach is obtained by weighting the list of stock rankings based on the two labels with the same weights. Within the date range of the test datasets, we take a full position in all selected stocks with the same principal amount on each trading day and liquidate the position on the next trading day. Then we can calculate backtesting indicators according to the profits obtained from these buying and selling.

C. *Experiment Results*

We select the models with the highest portfolio revenue in each category based on the three ways of constructing portfolios and calculate their backtesting indicators on all test datasets. For the portfolio revenue, we calculate the sum of them. For the sharpe ratio, information ratio and maximum drawdown of them, we calculate the average. These results are shown in Table I to Table III.

The experimental results show that among all portfolios constructed based on three different methods, the portfolio constructed by the AMMOE prediction results has the highest portfolio revenue and the highest information ratio. In Table. I, the sharpe ratio of AMMOE is slightly lower than the other baselines, but the information ratio of the other baselines perform worse than AMMOE, which indicates that some baselines perform worse than the benchmark on some test datasets. The backtesting results shown in Table I are based on excess return, so it is normal that the information ratio and the sharpe ratio do not perform the same. But in Table II and Table III, the sharpe ratio and information ratio of the portfolios constructed by AMMOE are consistent. These results indicate that AMMOE has better stability than other baselines. However, with the limited investment approach, AMMOE takes more risk while achieving higher revenue, so the maximum drawdown is silghtly worse than some baselines. But in general, the AMMOE-based portfolio still has a clear advantage over other baselines in terms of predicting investment indicators.

Without restricting the investment approach, we select the portfolio with the highest revenue, calculate their backtesting indicators on all test datasets according to the calculations in Table I to Table III. These results are shown in Table IV. AMMOE has the best performance on all backtesting indicators. Moreover, unlike the results in Table I to Table III, AMMOE also has the lowest maximum drawdown among all models. In practice, there are generally no restrictions on investment approaches, thus AMMOE has a greater advantage than other models because it balances revenue and risk well.

Fig. 3 shows the average and maximum portfolio revenue for each category of models after selecting the portfolio with the highest revenue on each test datasets. Despite the differences in the performance of models with different hyperparameter configuration, the average and optimal results of AMMOE are still the best. Because AMMOE is constrained by multiple tasks, it always has good robustness in the presence of changes in data or model structure. It has more significant advantages over other single-task learning models when dealing with complex data.

TABLE I: BACKTESTING RESULTS OF THE HIGHEST-REVENUE PORTFOLIOS CONSTRUCTED BY EACH TYPE OF MODEL BASED ON EXCESS RETURN

Model	AMMOE	FNN	CNN	SVM	Random Forest
Portfolio Revenue	**3113.34**	2876.21	2120.92	2094.44	2629.63
Sharpe Ratio	0.21735	**0.29059**	0.23839	0.17778	0.11136
Information Ratio	**0.02960**	0.02956	-0.04220	-0.04888	-0.11920
Maximum Drawdown	0.02922	0.02808	0.02599	0.02684	**0.02297**

TABLE II: BACKTESTING RESULTS OF THE HIGHEST-REVENUE PORTFOLIOS CONSTRUCTED BY EACH TYPE OF MODEL BASED ON MACD

Model	AMMOE	FNN	CNN	SVM	Random Forest
Portfolio Revenue	**4882.25**	3502.41	1597.89	2095.53	3121.64
Sharpe Ratio	**0.24589**	0.18705	0.09983	0.23012	0.23477
Information Ratio	**0.10638**	0.02908	-0.14044	-0.04575	-0.00169
Maximum Drawdown	0.03183	0.03126	**0.02430**	0.02683	0.03093

TABLE III: BACKTESTING RESULTS OF THE HIGHEST-REVENUE PORTFOLIOS CONSTRUCTED BY EACH TYPE OF MODEL BASED ON BOTH EXCESS RETURN AND MACD

Model	AMMOE	FNN	CNN	SVM	Random Forest
Portfolio Revenue	**4293.96**	2916.11	2207.78	2287.50	4079.60
Sharpe Ratio	**0.27069**	0.25151	0.20899	0.24827	0.27068
Information Ratio	**0.15438**	0.07095	-0.00992	-0.02367	-0.04927
Maximum Drawdown	0.02653	0.03060	**0.02532**	0.02695	0.02600

TABLE IV: BACKTESTING RESULTS OF THE HIGHEST-REVENUE PORTFOLIOS CONSTRUCTED BY EACH TYPE OF MODEL WITHOUT RESTRICTING THE INVESTMENT APPROACH

Model	AMMOE	FNN	CNN	SVM	Random Forest
Portfolio Revenue	**5802.73**	4110.94	3066.34	2369.54	4471.72
Sharpe Ratio	**0.34455**	0.32841	0.28926	0.24773	0.30121
Information Ratio	**0.18208**	0.16331	0.08279	-0.02325	0.10328
Maximum Drawdown	**0.02500**	0.02519	0.02623	0.02638	0.02838

AMMOE models have the characteristics of sharing the underlying structure and training multiple targets simultaneously. Therefore, its training efficiency is significantly higher than other neural network-like models without apparently increasing the model complexity. This will greatly reduce the resource consumption of model training and deployment in practical applications. Therefore, AMMOE can also be extended to other quantitative investment fields, such as high-frequency trading, which requires higher computational speed.

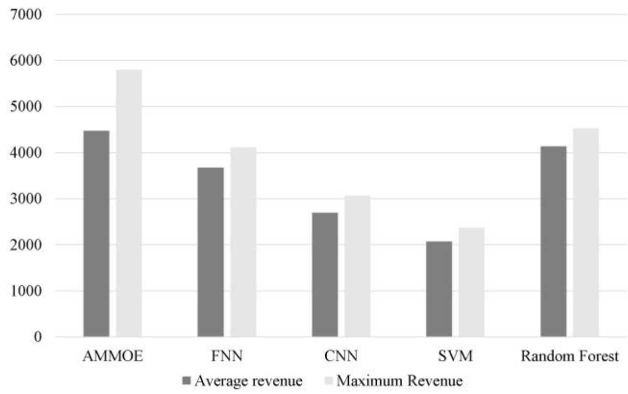

Fig. 3. The comparison of different model's maximum revenue and average revenue.

V. CONCLUSION

In this paper, we propose a multi-task learning model called AMMOE, in which an attention module is added to MMOE for preprocessing original features. We apply it to the quantitative stock selection and analyze its performance. The experimental results show that AMMOE can handle complex stock data well and get better prediction results for multiple targets. In addition, the model exhibits better stability and robustness, and higher training efficiency than other single-task stock selection models. The effectiveness of the method is proved in backtesting of actual trading data, which can provide valuable suggestions for actual investment behavior. Exploring AMMOE's application in other quantitative investment fields or improving the performance of it in more complex application scenarios will be the focus of future work.

APPENDIX

Table V shows all stock features used in the experiment.

TABLE V: STOCK FEATURES USED IN THE EXPERIMENT

Feature	Type of Feature	Description
ln close	stock daily trading data	logarithm of stock closing price
pch chg	stock daily trading data	stock daily return
turnover rate	stock daily trading data	stock turnover rate
volumn rate	stock daily trading data	stock volumn rate
pe	stock daily trading data	P/E ratio
pb	stock daily trading data	P/B ratio
ps	stock daily trading data	P/S ratio
dv ratio	stock daily trading data	dividend rate
ln total mv	stock daily trading data	logarithm of total market value
volatility 6	stock daily trading data	short-term (within 6 days) stock price volatility
volatility 12	stock daily trading data	medium-term (within 12 days) stock price volatility
volatility 24	stock daily trading data	long-term (within 24 days) stock price volatility
ln bench close	CSI 300 daily trading data	logarithm of CSI 300 closing price
bench return rate	CSI 300 daily trading data	CSI 300 daily return
bench turnover rate	CSI 300 daily trading data	CSI 300 turnover rate
bench pe	CSI 300 daily trading data	CSI 300 P/E ratio

bench pb	CSI 300 daily trading data	CSI 300 P/B ratio
ln bench total mv	CSI 300 daily trading data	logarithm of CSI 300 total market value
DEA	stock technical indicators	moving average (within 10 days) of DIF
DIF	stock technical indicators	difference between long-term average and short-term average
MACD	stock technical indicators	moving average convergence / divergence
PSY	stock technical indicators	psychological line
RSI6	stock technical indicators	relative strength index within 6 days
RSI12	stock technical indicators	relative strength index within 12 days
RSI24	stock technical indicators	relative strength index within 24 days
BIAS6	stock technical indicators	bias within 6 days
BIAS12	stock technical indicators	bias within 12 days
BIAS24	stock technical indicators	bias within 24 days

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Keyao Li: Conceptualization, Methodology, Formal analysis, Writing-original draft. Jungang Xu: Writing-review & editing, Supervision.

References

[1] S. Emerson, R. Kennedy, L. O'Shea, and J. O'Brien, "Trends and applications of machine learning in quantitative finance," in *Proc. 8th International Conference on Economics and Finance Research*, 2019.

[2] H. Ghoddusi, G. G. Creamer, and N. Rafizadeh, "Machine learning in energy economics and finance: A review," *Energy Economics*, vol. 81, pp. 709–727, 2019.

[3] R. N. Kahn, *The Future of Investment Management*, CFA Institute Research Foundation, 2018.

[4] Z. Zhang, S. Zohren, and S. Roberts, "Deep learning for portfolio optimization," *The Journal of Financial Data Science*, vol. 2, no. 4, pp. 8–20, 2020.

[5] M. Sharma, S. Sharma, and G. Singh, "Performance analysis of statistical and supervised learning techniques in stock data mining," *Data*, vol. 3, no. 4, p. 54, 2018.

[6] S. Vandenhende, S. Georgoulis, W. Van Gansbeke, M. Proesmans, D. Dai, and L. Van Gool, "Multi-task learning for dense prediction tasks: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[7] S. VANDENHENDE, S. Georgoulis, W. Van Gansbeke, M. Proesmans, D. Dai, and L. Van Gool, "Multi-task learning for dense prediction tasks: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[8] S. Ruder, "An overview of multi-task learning in deep neural networks," arXiv preprint arXiv:1706.05098, 2017.

[9] M. Guo, A. Haque, D.-A. Huang, S. Yeung, and L. Fei-Fei, "Dynamic task prioritization for multitask learning," in *Proc. the European Conference on Computer Vision (ECCV)*, 2018, pp. 270–287.

[10] X. Sun, R. Panda, R. Feris, and K. Saenko, "Adashare: Learning what to share for efficient deep multi-task learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 8728–8740, 2020.

[11] J. Ma, Z. Zhao, X. Yi, J. Chen, L. Hong, and E. H. Chi, "Modeling task relationships in multi-task learning with multi-gate mixture-of-experts," in *Proc. the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 1930–1939.

[12] Z. Zhao, L. Hong, L. Wei, J. Chen, A. Nath, S. Andrews, A. Kumthekar, M. Sathiamoorthy, X. Yi, and E. Chi, "Recommending what video to watch next: a multitask ranking system," in *Proc. the 13th ACM Conference on Recommender Systems*, 2019, pp. 43–51.

[13] H. Tang, J. Liu, M. Zhao, and X. Gong, "Progressive layered extraction (PLE): A novel multi-task learning (MTL) model for personalized recommendations," in *Proc. Fourteenth ACM Conference on Recommender Systems*, 2020, pp. 269–278.

[14] J. S. Abarbanell and B. J. Bushee, "Fundamental analysis, future earnings, and stock prices," *Journal of Accounting Research*, vol. 35, no. 1, pp. 1–24, 1997.

[15] J. Conrad and G. Kaul, "An anatomy of trading strategies," The Review of Financial Studies, vol. 11, no. 3, pp. 489–519, 1998.

[16] J. L. Davis, E. F. Fama, and K. R. French, "Characteristics, covariances, and average returns: 1929 to 1997," *The Journal of Finance*, vol. 55, no. 1, pp. 389–406, 2000.

[17] V. L. Shahpazov, V. B. Velev, and L. A. Doukovska, "Design and application of artificial neural networks for predicting the values of indexes on the bulgarian stock market," in *Proc. 2013 Signal Processing Symposium (SPS)*, IEEE, 2013, pp. 1–6.

[18] L. Li, J. Wang, and X. Li, "Efficiency analysis of machine learning intelligent investment based on k-means algorithm," *IEEE Access*, vol. 8, pp. 147 463–147 470, 2020.

[19] D. E. Rapach, J. K. Strauss, J. Tu, and G. Zhou, "Industry return predictability: A machine learning approach," The Journal of Financial Data Science, vol. 1, no. 3, pp. 9–28, 2019.

[20] M. Leippold, Q. Wang, and W. Zhou, "Machine learning in the Chinese stock market," *Journal of Financial Economics*, vol. 145, no. 2, pp. 64–82, 2022.

[21] S. Feuerriegel and R. Fehrer, "Improving decision analytics with deep learning: the case of financial disclosures," arXiv preprint arXiv:1508.01993, 2015.

[22] S. I. Lee and S. J. Yoo, "Multimodal deep learning for finance: integrating and forecasting international stock markets," *The Journal of Supercomputing*, vol. 76, no. 10, pp. 8294–8312, 2020.

[23] X.-Y. Liu, H. Yang, J. Gao, and C. D. Wang, "Finrl: Deep reinforcement learning framework to automate trading in quantitative finance," in *Proc. the Second ACM International Conference on AI in Finance*, 2021, pp. 1–9.

[24] P. Sonkiya, V. Bajpai, and A. Bansal, "Stock price prediction using bert and gan," arXiv preprint arXiv:2107.09055, 2021.

[25] Z. Bitvai and T. Cohn, "Day trading profit maximization with multi-task learning and technical analysis," *Machine Learning*, vol. 101, no. 1, pp. 187–209, 2015.

[26] J. Huang, Y. Zhang, J. Zhang, and X. Zhang, "A tensor-based sub-mode coordinate algorithm for stock prediction," in *Proc. 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, IEEE, 2018, pp. 716–721.

[27] D. Prabhu, A. Chhabra, S. Goyal, and B. Das, "Multi task learning for financial forecasting," *Emerging Technologies in Data Mining and Information Security*, Springer, 2021, pp. 397–411.

[28] D. Shah, H. Isah, and F. Zulkernine, "Stock market analysis: A review and taxonomy of prediction techniques," *International Journal of Financial Studies*, vol. 7, no. 2, p. 26, 2019.