How Do We Predict Stock Returns in the Cross-Section with Machine Learning?

Masaya Abe
Nomura Asset Management Co, Ltd.2-2-1, Toyosu,
Koto-ku, Tokyo,135-0061, Japan

Kei Nakagawa
Nomura Asset Management Co, Ltd.2-2-1, Toyosu,
Koto-ku, Tokyo,135-0061, Japan

ABSTRACT

Stock return prediction is one of the most important themes for investors. Until now, there are many studies for the application of machine learning methods to predict stock returns in the cross-section. However, those studies focus only on differences in machine learning methods. This study investigates how the difference in problem settings such as problem definition and data preprocessing affects the performance of stock return prediction. Our results show that the performance varies depending on problem settings regardless of the prediction models. These findings indicate that not only the prediction models but also the problem settings are important for stock return prediction.

CCS CONCEPTS

• **Applied computing**; • **Law, social and behavioral sciences**; • **Economics**;

KEYWORDS

Deep Learning, Stock Return Prediction, Cross-Section

ACM Reference Format:
Masaya Abe and Kei Nakagawa. 2020. How Do We Predict Stock Returns in the Cross-Section with Machine Learning?. In *2020 3rd Artificial Intelligence and Cloud Computing Conference (AICCC 2020), December 18–20, 2020, Kyoto, Japan.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3442536.3442547

1 INTRODUCTION

Stock return prediction is one of the most important themes for both researchers and investors. Until now, there are many studies for the application of machine learning methods to predict stock returns [1, 2].

However, most part of those research has been basically focused on providing better algorithms. For example, the stock return prediction models centering on deep learning have been proposed as supervised learning and those are reported to outperform representative machine learning models. We also have to focus on problem settings: problem definition and data preprocessing despite both are recognized as critical steps for successful machine learning processes [3].

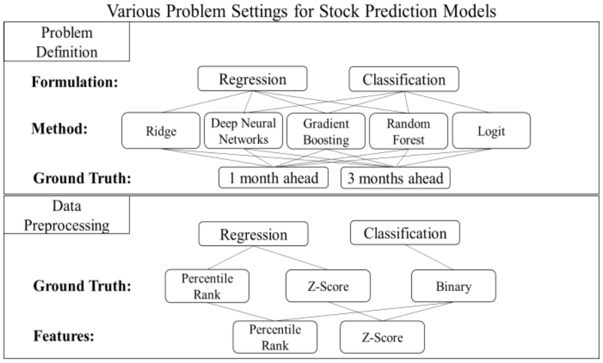

Figure 1: The patterns of the problem definition and the data preprocessing.

In this study, we investigate how the difference in problem settings affects the performance of stock return prediction in the cross-section. As far as we know, there are no previous works that address the problem settings of stock return prediction in the cross-section [4-9]. Fig. 1 shows the problem definition and the data preprocessing in our study.

For the problem definition, our research questions are

[**How to formulate the problem**] To define the stock return prediction problem, there are two types of formulating problems: regression or classification. A regression problem predicts continuous outputs i.e. stock returns. A classification problem predicts a discrete output i.e. buy or sell.

[**Which prediction method to use**] Most part of research on the stock return prediction has been basically focused on which prediction method to use.

[**What should be ground truth**] Consider What should be ground truth to use for learning various prediction models. Here, we use the one-month-ahead return and the three-months-ahead returns.

For the data preprocessing, our research questions are

[**How to preprocess the features**] we preprocess the features by percentile rank or Z-score.

[**How to preprocess the ground truth**] For the regression problem, we preprocess the ground truth by percentile rank or Z-score.

We investigate the impact of the problem definition and data preprocessing on the performance of stock return prediction in the Japanese stock market.

AICCC 2020, December 18–20, 2020, Kyoto, Japan
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-8883-2/20/12…$15.00
https://doi.org/10.1145/3442536.3442547

Table 1: List of Factors

No.	Factor	Description
1	B/P	Net Asset/Market Value
2	E/P	Net Profit/Market Value
3	D/P	Dividend/Market Value
4	S/P	Sales/Market Value
5	CF/P	Operating cash flow/Market Value
6	ROE	Net Profit/Net Asset
7	ROA	Net Operating Profit/Total Asset
8	ROIC	Net Operating Profit After Taxes/(Liabilities with interest + Net Asset)
9	Accrual	-(Changes in Current Asset and Liability-Depreciation)/Total Asset
10	Total Asset Growth Rate	Change Rate of Total Assets from the previous period
11	Current Ratio	Current Asset/Current Liability
12	Equity Ratio	Net Asset/Total Asset
13	Total Asset Turnover Rate	Sales/Total Asset
14	CAPEX Growth Rate	Change Rate of CAPEX from the previous period
15	EPS Revision (1 month)	1 month EPS Revision
16	EPS Revision (3 month)	3 month EPS Revision
17	Momentum (1 month)	Stock Returns in last month
18	Momentum (12-1 month)	Stock Returns in the past 12 months except for last month
19	Volatility	Standard Deviation of Stock Returns in the past 60 months
20	Skewness	Skewness of Stock Returns in the past 60 months
21-32	Dummy Variables (Month)	Month from January to December (one-hot encoding)

2 RELATED WORK

There are two major strategies in stock return prediction: Namely, the one based on time-series analysis [1, 2] and the one based on cross-sectional analysis [10].

The methods of the former strategy analyze past stock prices as time-series data and are applied to a practical trading strategy that focuses on a particular stock. Indeed, financial time-series forecasting can be considered one of the significant challenges in time series and machine learning literature [11]. [12, 13] showed that the shape of stock price fluctuation is an important feature in the prediction of future prices. They proposed a method to predict future stock prices with the past fluctuations similar to the current with indexing dynamic time warping method [14].

The methods of the latter strategy, which include this paper, stock return prediction with cross-sectional data of corporate attributes. One of the most significant interests in a cross-sectional analysis lies in finding factors that have strong predictive powers to the expected return in the cross-section. The Fama-French three-factor model [15, 16] is one of the nominal works in this field. They argued that the cross-sectional structure of the stock price can be explained by three factors: Namely, the beta (market portfolio), the size (market capitalization), and the value (Book-value to price ratio; BPR). This model inspires many subsequent research papers that propose more sophisticated versions of factors [17]. [18] surveyed the history of the proposed factors and argued that the number of reported factors shows a rapid increase in the last two decades.

In particular, machine learning approaches, which can capture the nonlinear relationship among multiple factors, are recently applied to a cross-sectional analysis. [19] applied the LASSO [20] in the U.S. stock market, and [4-9] applied deep learning in the Japanese stock market. However, these studies focus only on the proposed stock return prediction model being superior to other models.

3 DATASET AND METHODOLOGY

3.1 Dataset

We prepare the dataset for MSCI Japan Index constituents. The MSCI Japan Index comprises the large and mid-cap segments of the Japanese market. The index is also often used as a benchmark for overseas institutional investors investing in Japanese stocks.

We use the 32 factors listed in Table 1. The last 12 are used as dummy variables that are one to flag the months from January to December. These are used relatively often in practice and calculated monthly (at the end of the month). To calculate these factors, we acquire necessary data from WorldScope, Thomson Reuters, I/B/E/S, and EXSHARE. The actual financial data is acquired from WorldScope and Reuters Fundamentals (WorldScope priority). Considering the time when investors are actually available, we have a lag of four months. Forecast data (Nos. 15 and 16) is obtained from Thomson Reuters Estimates and I/B/E/S Estimates (Thomson Reuters priority).

3.2 Problem Definition

We define the stock return prediction problem as follows.

For stock i in MSCI Japan Index constituents at month $t-l$ (end of month) represented as U_{t-l}, 32 factors listed in Table 1 are defined by $\boldsymbol{x}_{i,t-l} \in \mathbb{R}^{32}$as features. The ground truth, $r_{i,t(l)} \in \mathbb{R}$, is defined by the next month stock return ($l = 1$) or the next three-months stock return ($l = 3$), respectively. Note that $r_{i,t(l)}$ is defined as

Ground Truth ($l = 1$)

Training: 1 set	
December 1999	
Features (November 1999)	Ground Truth (December 1999)

Ground Truth (l= 3)

Training: 1 set	
December 1999	
Features (September 1999)	Ground Truth (October- December 1999)

Figure 2: One set of training data with each ground truth for stock *i* in December 1999.

$p_{i,t}/p_{i,t-l} - 1$. Here, $p_{i,t}, p_{i,t-l}$ denotes the closing price at month t and $t - l$. As a more specific example, Fig. 2 shows the relationship between the features and the ground truth for stock i from one set of training data at December 1999 as $t = T$ for each ground truth, $l = 1$ and $l = 3$. Each set consists of all stocks in MSCI Japan Index constituents in November 1999 and September 1999 as time point $T - l$, respectively. The features are the factors as of November 1999 and September 1999.

The ground truth ($l = 1$) is the actual stock return in December 1999 for one-month return and the ground truth ($l = 3$) is the actual cumulative stock return from October 1999 to December 1999.

We formulate following regression/classification problem with the dataset defined above.

[Regression] For a regression problem, we use the mean squared error (MSE) as the loss function and define MSE_T when training the model at T as follows:

$$MSE_T = \frac{1}{K} \sum_{t=T-N+1}^{T} \sum_{i \in U_{t-l}} \left(r_{i,t(l)} - f\left(\boldsymbol{x}_{i,t-l} | \boldsymbol{\theta}_T\right) \right)^2 \qquad (1)$$

where $K = \sum_{t=T-N+1}^{T} |U_t|$ is the number of all training examples. $\boldsymbol{\theta}_T$ is the parameter by solving Eq. 1).

[Classification] For a classification problem, we deal with binary classification. The loss function is cross-entropy (CE) defined as follows:

$$CE_T = -\sum_{t=T-N+1}^{T} \sum_{i \in U_{t-l}} \{ r_{i,t(l)} \ln f\left(\boldsymbol{x}_{i,t-l} | \boldsymbol{\theta}_T\right) + \left(1 - r_{i,t(l)}\right) \ln f\left(\boldsymbol{x}_{i,t-l} | \boldsymbol{\theta}_T\right) \} \qquad (2)$$

where $\boldsymbol{\theta}_T$ is the parameter calculated by solving Eq. 2).

3.3 Data Preprocessing

Data preprocessing is performed to convert cross-sectional data for regression and classification problems.

[Regression] We rescale ground truth $r_{i,t(l)}$ and each feature $\boldsymbol{x}_{i,t-l}$ within MSCI Japan Index constituents at each time point. Regression have two types of preprocessing, a percentile rank (P-rank) and a Z-score. A P-rank is performed so that each value is from 0 to 1 by ranking each value in ascending order and then dividing by the maximum rank value. A Z-score is the number of standard deviations from the mean a data point is. The value is calculated by subtracting the mean and dividing by the standard deviation within MSCI Japan Index constituents at each time point. Winsorization is performed three times so that all data below -3 set to -3, and data above 3 set to 3.

[Classification] For classification, similar rescaling is done for each feature $\boldsymbol{x}_{i,t-l}$, but ground truth $r_{i,t(l)}$ is convert to binary. We replace the values larger than the median stock return within MSCI Japan Index constituents at each time point to class 1, and otherwise class 0.

Note that $r_{i,t(l)}$ and $\boldsymbol{x}_{i,t-l}$ after preprocessing are set to $\hat{r}_{i,t(l)}$ and $\boldsymbol{v}_{i,t-l}$.

3.4 Training and Prediction

We train the model by using the latest 120 sets of training from the past 10 years. To calculate the prediction, we substitute the latest features into the model after training has occurred.

For classification problems, the probability to outperform the cross-sectional median return is calculated for each stock. The cross-sectional predictive stock return (score) of stock i is calculated from time T by Eq. 3) substituting $\boldsymbol{v}_{i,T}$ into the function $f(\cdot)$ in Eq. 3) with the parameter $\boldsymbol{\theta}^*_T$, where $\boldsymbol{\theta}^*_T$ is calculated from Eq. 1) or (2) with $N = 120$:

$$Score_{i,T(L)} = f\left(\boldsymbol{v}_{i,T} | \boldsymbol{\theta}^*_T\right) \qquad (3)$$

For example, to calculate the prediction score for next one-month ($l = 1$) in January 2000 ($T + 1$) from December 1999 (T), the features are the factors as of December 1999 (T). The prediction model is updated by sliding one-month-ahead and carrying out a monthly forecast. In the case of the next three-month prediction ($l = 3$), the features are used as of December 1999 (T) as well, but the prediction model is updated by sliding three-months-ahead and carrying out three-month forecast.

The prediction period is from January 2000 to December 2018. The illustrations of the flow of the processing are shown in Fig. 3, which shows the relationship between prediction and training data for one-month prediction and three-month prediction at each time point.

3.5 Performance Measure

We need to analyze performance related to actual return more directly than accuracy measures such as MSE and CE. In this paper, we consider two investment strategies that are widely used in the literature of finance [15-18]. Namely, (i) the long portfolio strategy, and (ii) the long-short portfolio strategy. We consider an equally-weighted portfolio, which is simple yet sometimes outperforms more sophisticated alternatives [21].

We define the return of long portfolio R^L_t and long-short portfolio R^{LS}_t.

(i) The long portfolio strategy considered here buys the top quintile (i.e., one-fifth) scores of the stocks with equal weight aiming to outperform the average return of all the stocks. Let $L_t \subset U_t$: $1/5|U_t|$ be the long portfolio. The return from the long portfolio is defined as the average return of L_t : $R^L_t = 1/|L_t| \sum_{i \in L_t} r_{i,t(1)}$. (ii) The long-short portfolio strategy not only buys the top quintile scores of the stocks but also sells the bottom quintile scores of the stocks. Let $S_t \subset U_t$: $1/5|S_t|$ be the short portfolio. The return from the short portfolio is $R^S_t = 1/|S_t| \sum_{i \in S_t} r_{i,t(1)}$. The average return in the long-short portfolio strategy is defined as $R^{LS}_t = R^L_t - R^S_t$.

Regarding the long portfolio strategy, the annualized return is the excess return (Alpha) against the average return of all stocks

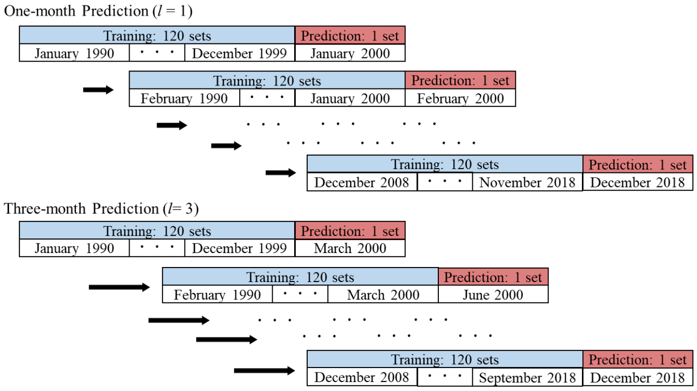

Figure 3: Training-prediction set for one-month prediction and three-month prediction.

in the universe, the risk (tracking error; TE) is calculated as the standard deviation of Alpha and risk-normalized return is measured by information ratio (IR).

$$\text{Alpha} = \prod_{t=1}^{T} (1+\alpha_t)^{\frac{12}{T}} -1\,, \text{TE} = \sqrt{\frac{12}{T-1}(\alpha_t-\mu_\alpha)^2}, \text{IR} = \text{Alpha}/\text{TE}$$

Here, $\alpha_t = R_t^L - 1/|U_t| \sum_{i\,\in U_t} r_{i,t(1)}$, $\mu_\alpha = 1/T \sum_{t=1}^{T} \alpha_t$.

Likewise, we evaluate the long-short portfolio strategy by its annualized return (AR), risk as the standard deviation of return (RISK), risk/return (R/R) as return divided by risk.

$$\text{AR} = \prod_{t=1}^{T} \left(1+R_t^{LS}\right)^{\frac{12}{T}} - 1\,, \text{RISK} = \sqrt{\frac{12}{T-1}\left(R_t^{LS}-\mu_{LS}\right)^2}, \text{R/R} = \text{AR}/\text{RISK} \tag{4}$$

Here, $\mu_{LS} = 1/T \sum_{t=1}^{T} R_t^{LS}$. For evaluating the rebalance amount, we calculate the annualized one-way portfolio turnover (TN), which define as the average percentage of stocks traded in each period.

TN from the long portfolio defines as

$$TN^L = \frac{12}{2l\,(T/l-1)} \sum_{t=1}^{T/l-1} \sum_{i \in L_t \cup L_{t+1}} \left\| w_{i,t+l}^L - w_{i,t}^{L+} \right\|_1$$

where $w_{i,t+l}^L$ is the portfolio weight at $t+l$ and $w_{i,t}^{L+}$ is the long portfolio weight after considering stock price fluctuation between t and $t+l$. Likewise, for the long-short portfolio, we define TN^{LS} as

$$TN^{LS} = TN^L + TN^S$$

$$TN^S = \frac{12}{2l\,(T/l-1)} \sum_{t=1}^{T/l-1} \sum_{i \in S_t \cup S_{t+1}} \left\| w_{i,t+l}^S - w_{i,t}^{S+} \right\|_1$$

where $w_{i,t+l}^S$ is the portfolio weight at $t+l$ and $w_{i,t}^{S+}$ is the short portfolio weight after considering stock price fluctuation between t and $t+l$. Portfolio rebalancing interval is every l month. The performance of these strategies is calculated monthly during the prediction period 19 years from January 2000 to December 2018.

3.6 Prediction Methods

[Deep Neural Network (DNN)] is implemented with TensorFlow [22]. We examine 3 patterns of the neural networks. The hidden layer size is set to be (150-150-100-100-50-50), (200-200-100-100-50-50), and (300-300-150-150-50-50). The dropout rate for each layer is set to (30%-30%-20%-20%-10%-10%), (50%-50%-30%-30%-10%-10%), and (50%-50%-50%-50%-50%-50%). The number of units in each layer is designed to decrease as the layer becomes closer to the output layer. We use the ReLU function as the activation function, and Adam [23] for the optimization algorithm. Batch normalization [24] is applied to activation. The mini-batch size is set to 1000. For training the model, we follow [8]. The network weights are updated until the average of the rank correlation coefficient between the predicted returns calculated by each training data set and the realized returns (ground truth) reaches 0.20. And those reached 0.16 are used for the initial network weights at the next point as sequential analysis. We initialize to generate the network weights from TensorFlow's function "truncated_normal" set to mean "0" and standard deviation "$\sqrt{2/M}$" (M is the size of the previous layer).

[Gradient Boosting Tree (GB)] is implemented with xgboost [25] with the class "XGBRegressor" for regression and the class "XGBClassifier" for classification. For the hyper parameters, we set the max number of features("max_features") to 32, the number of trees ("n_estimators") to 500, and 3 patterns of the depth of tree ("max_depth") to {3, 5, 7}.

[Random Forest (RF)] is implemented with scikit-learn [26] with the class "ensemble.RandomForestRegressor" for regression and the class "ensemble.RandomForestClassifier" for classification. For the hyper parameters, we set the max number of features("max_features") to 32, the number of trees ("n_estimators") to 500, and 3 patterns of the depth of tree ("max_depth") to {3, 5, 7}.

[Ridge Regression (RR)] is implemented with scikit-learn [26] with the class "linear_model.Ridge". For the hyper parameters, we set 3 patterns of the regularization strength("alpha") to {0.01, 1, 100}.

[Logistic Regression (LR)] is implemented with scikit-learn with the class "linear_model.LogisticRegression". For the hyper parameters, we set 3 patterns of the inverse of regularization strength("C") to {0.01, 1, 100}.

Table 2: Experimental Results of Long portfolio.

Alpha [%]		One-month return				Three-month return			
		DNN	GB	RF	RR	DNN	GB	RF	RR
Regression	P-rank	**9.66**	6.74	8.18	7.64	**5.32**	2.56	3.09	3.67
	Z-score	**8.58**	6.70	7.41	7.43	2.95	1.38	2.99	**3.36**
Classification	P-rank	**8.35**	5.44	7.12	6.76	**3.45**	1.88	1.92	2.90
	Z-score	**7.76**	6.40	6.87	6.75	2.95	1.67	1.81	**3.38**
IR		One-month return				Three-month return			
		DNN	GB	RF	RR	DNN	GB	RF	RR
Regression	P-rank	1.46	1.28	**1.47**	1.34	**0.80**	0.49	0.57	0.69
	Z-score	**1.39**	1.25	1.24	1.20	0.52	0.28	0.56	**0.61**
Classification	P-rank	1.28	1.06	**1.35**	1.20	0.47	0.41	0.36	**0.55**
	Z-score	**1.37**	1.30	1.30	1.15	0.48	0.37	0.35	**0.63**
TN^{L} [%]		One-month return				Three-month return			
		DNN	GB	RF	RR	DNN	GB	RF	RR
Regression	P-rank	889	823	784	**743**	251	243	**166**	167
	Z-score	831	810	785	**706**	219	239	**142**	150
Classification	P-rank	867	778	729	**684**	266	244	179	**166**
	Z-score	817	793	727	**645**	234	237	172	**159**

Table 3: Average performance of Alpha, IR, and TN^{L} in each category.

Avg.	Formulation		Prerprocess		Ground Truth		Method			
	Regression	Classification	P-rank	Z-score	1M	3M	DNN	GB	RF	RR
Alpha [%]	**5.48**	4.71	**5.29**	4.90	**7.36**	2.83	**6.13**	4.10	4.92	5.24
IR	**0.95**	0.85	**0.92**	0.88	**1.29**	0.51	**0.97**	0.80	0.90	0.92
TN^{L} [%]	**497**	481	499	**479**	776	**202**	547	521	461	**428**

4 EXPERIMENTAL RESULTS

4.1 Results of Long Portfolio

Table 2 shows the results of Long portfolio strategies; the values of Alpha, IR, and TN^{L}, respectively. Each value is selected from the pattern with the best IR. Bold characters indicate the best ones among each category, and the best value in all categories of each ground truth is also underlined. First, we focus on the Alpha. DNN in each category of formulation and data preprocessing of one-month return outperforms all other methods. In terms of three-month return, RR outperforms in Z-score categories while DNN outperforms in percentile rank (P-rank) categories. The best value in each ground truth comes from DNN. As for the values of IR, DNN in each Z-score category of one-month return outperforms all other methods, and RF outperforms in each P-rank category. The best value in three-month return category comes from DNN, but the patterns of RR are best in other categories.

The results of TN^{L} are shown that RR in each category of one-month return is lower turnover than all other methods. In terms of three-month return, RF is lower than other methods in each regression category, and RR is lower each classification category.

Table 3 shows the average performance of Alpha, IR, and TN^{L} in each category. Bold characters indicate the best ones among each category. We can easily confirm that the better results of Alpha in each category are regression in formulation, P-rank in data preprocessing, one-month return in ground truth, and DNN in method. In terms of IR, the results are the same as Alpha. Turnover is lower in three-month return because of low rebalance frequency. For fairness, we compared with Alpha after deducting the one-way transaction cost of 10bps (0.10%), but the result in each category is almost the same. As for problem setting, problem formulation and data preprocessing have the same impact on the Alpha and IR as selecting the medium and best performing methods. In particular, ground truth has the same impact on the Alpha and IR as selecting the best and worst methods.

4.2 Results of Long-Short Portfolio

Table 4 show the experimental results of Long-Short portfolio strategies; the values of AR, R/R, and TN^{LS}, respectively. Each value is selected from the pattern with the best R/R. Bold characters indicate the best ones among each category, and the best value in all categories of each ground truth is also underlined. The results of AR also show that DNN in each category of formulation and data preprocessing of one-month return outperforms all other methods. Besides, the best value in each category of three-month return comes from DNN. The best value in all category, therefore, comes from DNN. As for the values of R/R, DNN outperforms all other methods in one-month return each category. The best value in three-month return category comes from DNN, but there is no clear trend in each category. The results of TN^{LS} are similar to those of TN^{L} in Table 2. RR and RF are lower turnover than DNN and GB, and RR is mostly lower when comparing RF.

Table 4: Experimental Results of Long-Short portfolio.

AR [%]		One-month return				Three-month return			
		DNN	GB	RF	RR	DNN	GB	RF	RR
Regression	P-rank	**19.56**	11.25	15.73	14.38	**9.75**	5.43	6.38	7.44
	Z-score	**17.06**	11.86	13.05	14.65	**7.98**	3.58	6.60	6.67
Classification	P-rank	**16.18**	9.27	13.61	13.09	**7.14**	3.29	5.65	6.23
	Z-score	**14.68**	11.72	12.37	13.09	**6.89**	3.73	6.01	6.48
R/R		One-month return				Three-month return			
		DNN	GB	RF	RR	DNN	GB	RF	RR
Regression	P-rank	**1.52**	1.27	1.50	1.24	**0.71**	0.57	0.67	0.70
	Z-score	**1.52**	1.26	1.29	1.30	0.65	0.41	**0.68**	0.64
Classification	P-rank	**1.37**	0.99	1.36	1.15	0.48	0.40	0.57	**0.58**
	Z-score	**1.46**	1.31	1.23	1.15	0.58	0.41	**0.60**	0.59
TN^{LS} [%]		One-month return				Three-month return			
		DNN	GB	RF	RR	DNN	GB	RF	RR
Regression	P-rank	1,692	1,700	1,484	**1,411**	496	539	348	**316**
	Z-score	1,596	1,572	1,437	**1,397**	434	466	**261**	302
Classification	P-rank	1,701	1,532	1,379	**1,299**	522	574	353	**318**
	Z-score	1,592	1,546	1,374	**1,255**	468	474	325	**316**

Table 5: Average performance of AR, R/R, and TN^{LS} in each category.

Avg.	Formulation		Prerprocess		Ground Truth		Method			
	Regression	Classification	P-rank	Z-score	1M	3M	DNN	GB	RF	RR
AR [%]	**10.71**	9.34	**10.27**	9.78	**13.85**	6.20	**12.40**	7.52	9.92	10.25
R/R	**1.00**	0.89	0.94	**0.94**	**1.31**	0.58	**1.04**	0.83	0.99	0.92
TN^{LS} [%]	966	**939**	979	**926**	1,498	**407**	1,063	1,050	870	**827**

Table 5 shows the average performance of AR, R/R, and TN^{LS} in each category. Bold characters indicate the best ones among each category. The results are generally consistent with the results of Long portfolio in Table 3. The better results of AR in each category are regression in formulation, P-rank in data preprocessing, one-month return in ground truth, and DNN in method. In terms of R/R, the results are nearly the same as AR. We confirmed that even after deducting transaction costs, the results are mostly unchanged. As for problem setting, problem formulation and data preprocessing have the same impact on the AR and R/R as selecting the medium and best performing methods, and ground truth have the same impact on the AR and R/R as selecting the best and worst methods as in the case of Long portfolio.

5 CONCLUSION

In this study, we investigate how the difference in problem settings affects the performance of stock return prediction in the cross-section. Our conclusions are as follows:

- The better results of Alpha (resp. AR) in each category are regression in formulation, P-rank in data preprocessing, one-month return in ground truth, and DNN in method. The results for IR (resp. R/R) are the same. This is an answer to our research question.
- In comparison to machine learning methods, DNN outperforms in each AR category, and the best value of each Alpha and R/R category comes from DNN. As for turnover, RF and RR have lower turnover.
- As for problem setting, problem formulation and data preprocessing have the same impact on the performance as selecting the medium and best performing methods. In particular, ground truth has the same impact as selecting the best and worst methods.
- These results do not mostly change after deducting the transaction costs. Directions of promising future work includes the followings.
- More sophisticated portfolio construction: In this study, we use a simple equally-weighted portfolio. On the other hand, the portfolio theory [27] states that explicit consideration of the risk in portfolio selection is important. Regarding this direction, combining our method with more sophisticated portfolio strategies, such as complex valued risk diversification [28], TPLVM based Portfolio [29] or RM-CVaR Portfolio [30] will be an interesting direction for the future work.
- Stateful DNN methods: This paper considered a rolling-horizon learning of a neural network, whereas there are several other approaches for portfolio selection. In particular, the recurrent neural networks [31] and its variants are stateful neural networks that can capture the time evolution of the stock universe.

REFERENCES

[1] Atsalakis, G. S., and Valavanis, K. P. 2009. Surveying stock market forecasting techniques–Part II: Soft computing methods. *Expert Systems with applications*, 36(3), 5932-5941. DOI= https://doi.org/10.1016/j.eswa.2008.07.006

[2] Cavalcante, R. C., Brasileiro, R. C., Souza, V. L., Nobrega, J. P., and Oliveira, A. L. 2016. Computational intelligence and financial markets: A survey and future directions. *Expert Systems with Applications*, 55, 194-211. DOI=https://doi.org/10.1016/j.eswa.2016.02.006

[3] García, S., Luengo, J., and Herrera, F. 2015. *Data preprocessing in data mining* (pp. 195-243). Cham, Switzerland: Springer International Publishing. DOI=https://doi.org/10.1007/978-3-319-10247-4

[4] Abe, M., and Nakayama, H. 2018. Deep learning for forecasting stock returns in the cross-section. *In Pacific-Asia Conference on Knowledge Discovery and Data Mining* (pp. 273-284). Springer, Cham.DOI=https://doi.org/10.1007/978-3-319-93034-3_22

[5] Nakagawa, K., Uchida, T., and Aoshima, T. 2018. Deep factor model. *In ECML PKDD 2018 Workshops* (pp. 37-50). Springer, Cham. DOI=https://doi.org/10.1007/978-3-030-13463-1_3

[6] Sugitomo, S., and Minami, S. 2018. Fundamental Factor Models Using Machine Learning. *Journal of Mathematical Finance*, 8, 111-118.DOI=https://doi.org/10.4236/jmf.2018.81009

[7] Nakagawa, K., Ito, T., Abe, M., and Izumi, K. 2019. Deep Recurrent Factor Model: Interpretable Non-Linear and Time-Varying Multi-Factor Model. *In AAAI-19 Workshop on Network Interpretability for Deep Learning.* arXiv preprint arXiv:1901.11493.

[8] Nakagawa, K., Abe, M., and Komiyama, J. 2019. A Robust Transferable Deep Learning Framework for Cross-sectional Investment Strategy. *In AAAI-20 Workshop on Knowledge Discovery from Unstructured Data in Financial Services* arXiv preprint arXiv:1910.01491.

[9] Abe, M., and Nakagawa, K. 2020. Cross-sectional stock price prediction using deep learning for actual investment management. In Proceedings of the 2020 Asia Service Sciences and Software Engineering Conference, ASSE '20,. Association for Computing Machinery. DOI= https://doi.org/10.1145/3399871.3399889

[10] Subrahmanyam, A. 2010. The cross-section of expected stock returns: what have we learnt from the past twenty-five years of research?. *European Financial Management*, 16(1), 27-42. DOI= https://doi.org/10.1111/j.1468-036X.2009.00520.x

[11] Tay, F. E., and Cao, L. 2001. Application of support vector machines in financial time series forecasting. *omega*, 29(4), 309-317. DOI=https://doi.org/10.1016/S0305-0483(01)00026-3

[12] Nakagawa, K., Imamura, M., and Yoshida, K. 2017. Stock Price Prediction with Fluctuation Patterns Using Indexing Dynamic Time Warping and k^* -Nearest Neighbors. In *JSAI International Symposium on Artificial Intelligence* (pp. 97-111). Springer, Cham. DOI=https://doi.org/10.1007/978-3-319-93794-6_7

[13] Nakagawa, K., Imamura, M., and Yoshida, K. 2019. Stock price prediction using k -medoids clustering with indexing dynamic time warping. *Electronics and Communications in Japan*, 102(2), 3-8. DOI=https://doi.org/10.1541/ieejeiss.138.986

[14] Itakura, F. 1975. Minimum prediction residual principle applied to speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 23(1), 67-72. DOI= https://doi.org/10.1109/TASSP.1975.1162641

[15] Fama, E. F., and French, K. R. 1992. The cross-section of expected stock returns. *the Journal of Finance*, 47(2), 427-465.DOI=https://doi.org/10.1111/j.1540-6261.1992.tb04398.x

[16] Fama, E. F., and French, K. R. 1993. Common risk factors in the returns on stocks and bonds. *Journal of financial economics*, 33(1), 3-56. DOI=https://doi.org/10.1016/0304-405X(93)90023-5

[17] McLean, R. D., and Pontiff, J. 2016. Does academic research destroy stock return predictability?. *Journal of Finance*, 71(1), 5-32. DOI= https://doi.org/10.1111/jofi.12365

[18] Harvey, C. R., Liu, Y., and Zhu, H. 2016. . . . and the cross-section of expected returns. *The Review of Financial Studies*, 29(1), 5-68.DOI=https://doi.org/10.1093/rfs/hhv059

[19] Chinco, A., Clark-Joseph, A. D., and Ye, M. 2019. Sparse signals in the cross-section of returns. *Journal of Finance*, 74(1), 449-492. DOI= https://doi.org/10.1111/jofi.12733

[20] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1), 267-288. DOI= https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

[21] DeMiguel, V., Garlappi, L., and Uppal, R. 2009. Optimal versus naive diversification: How inefficient is the 1/N portfolio strategy?. *The review of Financial studies*, 22(5), 1915-1953.DOI=https://doi.org/10.1093/rfs/hhm075

[22] Abadi, M Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. TensorFlow: a system for large-scale machine learning. *In Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation (OSDI'16).* pp.265-283.

[23] Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[24] Ioffe, S., and Szegedy, C. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *In International Conference on Machine Learning* (pp. 448-456).

[25] Chen, T., and Guestrin, C. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794).

[26] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... and Vanderplas, J. 2011. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, 12, 2825-2830.

[27] Markowitz, H. 1952. Portfolio selection. *Journal of finance*, 7(1):77–91. DOI= https://doi.org/10.1111/j.1540-6261.1952.tb01525.x

[28] Uchiyama, Y., Kadoya, T., and Nakagawa, K. 2019. Complex valued risk diversification. *Entropy*, 21(2), 119. DOI=https://doi.org/10.3390/e21020119

[29] Uchiyama, Y., and Nakagawa, K. 2020. TPLVM: Portfolio Construction by Student's t-Process Latent Variable Model. *Mathematics*, 8(3), 449. DOI=https://doi.org/10.3390/math8030449

[30] Nakagawa, K., Noma, S., and Abe, M. 2020. RM-CVaR: Regularized multiple beta-cvar portfolio. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*.DOI= https://doi.org/10.24963/ijcai.2020/629

[31] Selvin, S., Vinayakumar, R., Gopalakrishnan, E. A., Menon, V. K., and Soman, K. P. 2017. Stock price prediction using LSTM, RNN and CNN-sliding window model. In 2017 international conference on advances in computing, communications and informatics (icacci) (pp. 1643-1647). IEEE. DOI=https://doi.org/10.1109/ICACCI.2017.8126078