DISCUSSION PAPER SERIES E

SHIGA UNIVERSITY

Discussion Paper No. E-8

Neural Network Application in Predicting Stock Returns: Evidence from Japan

Xin Zhong and Yiwen Wang

March 2021

The Institute for Economic and Business Research
Faculty of Economics
SHIGA UNIVERSITY

1-1-1 BANBA, HIKONE,
SHIGA 522-8522, JAPAN

Neural Network Application in Predicting Stock Returns: Evidence from Japan

Xin Zhong and Yiwen Wang

***Abstract*—The primary purpose of this study is to forecast the one-month forward Nikkei 225 stock return by employing neural networks. We first explore the predictive function of artificial neural networks by comparing the predictive power of models of different neurons and hidden layers. We find that the model with 100 neurons and two hidden layers has the best predictive ability. We also investigate the effects of different types of input variables on predictions. The results show that both technical and liquidity proxies contribute to the analysis. Finally, we combine neural networks with portfolio construction strategies and confirm that neural networks predictions can effectively distinguish good stocks and bad stocks. In summary, this study applies the neural network to the stock market and provides a new idea for using deep learning to investment decisions.**

I. Introduction

One of the main achievements of artificial intelligence techniques, the deep learning neural networks, have drawn considerable attention and become a popular research topic during recent decades. Neural networks are designed to mimic human brains with neural systems and learn past information in a higher intelligence than traditional mathematical methods. It is also able to evolve the model itself over time by continually obtaining available messages. Due to these advanced functions, neural networks are expected to have the capability of predicting stock prices and returns by many experts.

This research is conducted using the H2O platform[1], a developed neural network tool. The technical indicators and liquidity measures are collected as inputs from the year 2010 to 2017, ranging eight years after the 2008 financial crisis. The output is one-month-forward individual stock return. Our study also back tests the past and compares the actual returns with predicted ones for the same period. A unique design of this research is that moving window prediction system[2] is applied. This moving window prediction uses a 6-year learning period to forecast one data point, which is the one-month-forward return. By moving learning periods month by month, there are in total 24 predicted returns from January 2016 to December 2017. We are also interested in the characteristics of neural networks. With different settings in terms of model characteristics, we compare the goodness of various deep learning models and discover the optimal one with specific layers and neurons number. We also examine the goodness of deep learning models and the standard linear regression model. The goodness of models is evaluated by errors between predicted results and actual returns. As a result, we observe that neural network models always outperform the linear regression when it comes to predicting stock returns.

However, based upon the above comparisons, it is hard to draw further conclusions concerning the goodness of neural network models because the evaluation metrics such as Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) tell limited information about the goodness of models. Therefore, a portfolio construction strategy is employed to help understand the capability and application of predicted results. The results show that the return of high-return portfolios predicted by deep learning models precede the benchmark Nikkei 225 index and low-return portfolios with great statistical significance. Therefore, the goodness of the applying neural network models to the stock predicting is generally confirmed.

The remainder of the study is organized as follows. Section 2 briefly reviews the relevant studies concerning technical analysis, liquidity measures, and neural network applications. The data screening for neural networks and details of model settings are described and explained in Section 3. Section 4 summaries the empirical results and provides a robust test using portfolio construction strategy. Section 5 concludes our findings and gives suggestions for further work.

II. Literature Review

Technical analysis is a methodology of studying past market data, using many chart tools. Dow theory states that prices trend directionally, up, down, flat or some combinations. Its followers, or technical analysts, believe in the presence of these price trends and predictability of the future direction. They also suggest the market as well as investors are repeating themselves over time, hence the market is predictable. Friesen et al. (2009) present that there is bias in acquiring and interpreting market information for traders. They find the existence of autocorrelation of price movements, regarded as the subsequent effect of trader bias, and successfully predict price jumps for some stocks. Caporin et al. (2013) contribute to technical analysis by showing that the high and low prices of equity shares are largely predictable only on the basis of their past realizations. Sezer et al. (2017) transfer the most commonly preferred technical analysis indicators into a series of buy-sell-hold trigger signals and conclude that the neural network model can achieve comparable predictions of trigger signals against the buy and hold strategy in most of the cases. In fact, majority of past related papers take a method of direction prediction, which is not aimed to forecast expected return but only buy, sell or hold signals. It is quite reasonable because technical analysis itself is playing the role of finding market turnover in the changeable environment. Usually this method will also

[1] For detailed explanation, please refer to Part D of Section III.
[2] For detailed explanation, please refer to Part E of Section III.

lead to a relatively high prediction power.

Liquidity is generally described as the ability of financial securities to trade within a certain period without affecting market price. Nowadays, more and more researchers are emphasizing that liquidity plays an important role in the field of asset pricing, because as they propose, investors require higher compensation for bearing more liquidity risk. Therefore, more illiquid stocks usually have higher expected returns. Amihud (2002) estimates liquidity from the perspective of asset return and trading quantity. Liu (2006) emphasizes trading speed from the viewpoint of zero daily trading volume. Datar, Naik, and Radcliffe (1998) use the turnover ratio as a proxy for liquidity. Pastor and Stambaugh (2003) consider asset liquidity to be sensitive to innovations in aggregate liquidity from the perspective of price impact. Lesmond et al. (1999) focus on the marginal trading cost by estimating the incidence of zero returns. In brief, there are many empirical studies concerning the measure of liquidity of stocks and for this paper, the positive relationship between illiquidity and return is considered as great importance when it comes to predict stock returns.

Previous common measurements of market efficiency include the autocorrelation-based test and the variance ratio for a random walk, but since the advent of machine learning techniques, dozens of papers in recent years start to focus on return prediction with more sophisticated deep learning models. Cao et al. (2005) compare the predictive power of linear models and neural network models in Chinese stock market, showing that neural networks fit well with the emerging market and outperform the linear models. Mizuno et al. (1998) build a neural network model for stock index prediction, using some typical technical indices as inputs. It concludes that in predicting TOPIX, the neural networks outperform the single use of the technical indicator but underperform the buy-and-hold strategy. Abe and Nakayama (2018) apply deep learning models for individual stock from MSCI Japan Index. They prepare a list of fundamental factors as inputs and find the most accurate deep learning model with specific hidden layers and neurons. Shynkevich et al. (2017), who focus on S&P 500 components and use technical indicators as model inputs, observe the highest prediction accuracy when the input window length is roughly equal to the forecast window length. In brief, individual stock share is deemed to be the more reasonable research object of machine learning than stock indices. The input features and the evaluation of model predictive power are two keys for relevant research and the best design is still under discussion.

III. Data and Methodology

This section introduces the data formation, the contents of the deep learning model, and the main body of regression methodology. The dataset for back-testing and prediction modeling is described in the first part. The second part mainly presents the detail of the input factors, including 15 technical analysis indicators and five liquidity measures. The third part explains the principle of a typical kind of neural network, the multi-layer feedforward artificial neural network (ANN), followed by the fourth part that lists the related coding message of deep learning function in H2O platform. At last, the moving-window regression method is explained in the fifth part.

A. Data of Japanese Stocks

Referring to past studies that investigate different stock markets by individual stock, this paper also uses market index components and try to predict the one-month forward return of representative stocks. In terms of the Japanese market, TOPIX and Nikkei 225 are two of the most indicative and leading stock indices. This research selects Nikkei 225 for several reasons. At first, Nikkei 225 components are top 225 blue-chip companies in Japan and are likely to have a more meaningful number of historical prices and trading volumes. In other words, the effectiveness of data is one of our concerns. Moreover, this stocks universe has a more workable size. To cover more alternatives to inputs and maintain an acceptable running speed, the stock amount we would like to use could be sacrificed to some extent.

The sample period of this research is from 2010 to 2017, considering the 2008 financial crisis and the afterward effect in the Japanese market. Since the Nikkei 225 constituents are review annually, we select the stocks that have never been excluded in the Nikkei 225 over these eight years. The data sample reaches 217 stocks in the end. Involving liquidity measures differs in this paper from previous studies. As a new tentative type of variables for stock return prediction, we believe it can bring significant contributions since many empirical works have improved traditional asset pricing models by adding liquidity. Input data collection includes 15 technical indicators and five liquidity measures of each stock. The former indicators are directly downloaded from Bloomberg and the latter measures are calculated with historical trading volumes and prices. Finally, as the response or output variable of deep learning models, the monthly stock returns from 2010 to 2017 are prepared.

B. Technical Indicators and Liquidity Measures

Technical analysis is a way to predict the future through the study of the past. From our point of view, the way technical analysis works is like that of neural networks. Both tools aim to find current or future market patterns from the past, though some external drivers are not necessarily exclusive for neural networks. We want to combine and enhance the relationship between technical analysis and neural network, with the expectation that they can together provide a good forecast of the financial market.

TABLE I: The List of Technical Analysis Indicators

No.	Factor	No.	Factor
1	Percent Bandwidth (%B)	9	Parabolic Studies (PTPS)
2	Commodity Channel Index (CMCI)	10	Fear and Greed (FG)
3	Average Directional Movement Index (ADX)	11	Williams %R
4	Moving Average Convergence/Divergence (MACD)	12	Momentum
5	Relative Strength Index (RSI)	13	Rate of Change (ROC)
6	Stochastic oscillator %K (TAS_K)	14	Hurst Exponent

7	Stochastic oscillator %D (TAS_D)	15	MaxMin Retracement
8	Average True Range (ATR)		

Table I shows the list of the technical analysis indicators involved in this research. All of them are historical prices related and use the month as a period for all calculations when it is needed. The indicators are directly downloaded from Bloomberg.

However, all selected technical indicators above are not involved with stock trading volume. Considering this pitfall of the technical indicators, five liquidity proxies are computed and comprise the other input list. The role of liquidity becomes more and more important in recent years because it drags empirical research of stock returns closer to the real market, by considering real cases like transaction costs. There are many ways to calculate liquidity. This paper chooses five of them with the most known value of research. Like many advanced technical analysis indicators, these liquidity measures are expected to reveal some patterns after summarizing the past data. Following discusses the liquidity proxies used in this research.

1. Amihud (ILLIQ)

$$ILLIQ = \sum \frac{|r_t|}{Volume_t}. \quad (1)$$

This illiquidity measure is defined by Acharya and Pedersen in 2005, where r_t is the daily stock return on day *t* and $Volume_t$ is the dollar trading volume on day *t*. This indicator can capture the price response to one dollar of trading volume, also belonging to price impact measurement.

2. Liu (LMx)

It is a turnover-adjusted zero-return measure of liquidity.

$$LMx_{i,t} = \left[N_z + \frac{\frac{1}{TVx}}{DF} \right] \times \frac{21x}{Nx}. \quad (2)$$

N_z is the number of zero trading volume days, TV_x is the turnover rate in the previous *x* months, N_x is the number of trading days in the previous *x* months, and *DF* is a deflator to ensure that the second term in the square brackets falls in the range of zero to one (not inclusive) for all sample stocks. In brief, this measure captures the multidimensional nature of liquidity.

3. Datar, Naik, and Radcliffe (Turn)

$$Turn_t = \frac{\text{number of shares traded in day } t}{\text{number of shares issued}}. \quad (3)$$

Developed by Datar in 1998, this monthly turnover is computed using daily trading shares and issued shares one year before.

4. Pastor and Stambaugh (Gamma)

$$r^e_{t+1} = \theta + \phi r_t + \gamma sign(r^e_t)(Volume_t) + \varepsilon_t. \quad (4)$$

Gamma (γ) is a price impact measurement, calculated by the above regression defined by Pastor and Stambaugh in 2003. r^e_t is the security's excess return above the market return on day *t* and $Volume_t$ is the trading volume in dollars on day *t*. The coefficient on the signed trading volume, γ, is expected to be negative.

5. Lesmond, Ogden, and Trzcinka (LOT)

This estimator of effective transaction costs developed by Lesmond et al. in 1999. This measure assumes that the investors are rational and that they will trade only when the excess return of stock *j* above the market return exceeds transaction costs. The parameter a_{2j} represents the buyer's transaction costs and a_{1j} represents the seller's transaction costs. The measure of the total round-trip transaction costs, LOT, is computed as the difference in the buyers' and sellers' trading costs:

$$LOT = \alpha_{2j} - \alpha_{1j}. \quad (5)$$

The parameters a_{1j} and a_{2j} can be obtained by maximizing the logarithm of the likelihood function which captures the relationships among the unobserved stock return, the observed stock return, and the market return.

C. Multi-layer Feedforward Artificial Neural Network (ANN)

Deep learning is a crucial part of machine learning that is beneficial from the development of computer clusters recent in the last decade. One of the well-established technique systems of deep learning is neural networks, which are biologically inspired by the human brain and neuron structure. The simplest type of neural networks, ANN, is the basic model of this paper and is conducted with H2O platform and R environment.

The multi-layer feedforward ANN is also known as a deep neural network or multi-layer perceptron. It consists of an input layer, multiple hidden layers, and an output layer. The data digestion goes through all those layers to get results. Within each layer, there is a list of nodes or neurons executing designate calculations. Between layers, various algorithms can be applied to adjust the weights of layer inputs and optimize the results for different aims. A general method for nonlinear optimization called gradient descent is often implemented and internalized with H2O. The powerfulness of ANN is its capability of storing experiential knowledge in the learning process and implementing multiple nonlinear regressions.

Another essential characteristic of neural networks is the back-propagation learning algorithm, which employs a backward phrase to minimize estimate errors after the training procedure. In this paper, we would like to take advantages of H2O programming where back-propagation is natively added into its multi-layer feedforward ANN. In this paper, all mentioned deep learning models or neural networks are conducted under multi-layer feedforward ANN of H2O with consistent characteristics, introduced comprehensively in the next part.

D. Application Program H2O

H2O is a well-known and easy-to-use open resource to conduct machine learning analytics. It is chosen other than Karas and Tensorflow because H2O has a deep learning function based on a multi-layer feedforward artificial neural network that is trained with stochastic gradient descent using back-propagation. Therefore, all deep learning models of this paper are built within the H2O deep learning function.

Though the H2O deep learning provides lots of conveniences process for beginning learners, there are some important arguments in need of manual settings. Firstly, the activation function is fixed for all models at "TanhWithDropout". Tanh activation is usually preferred in pursuit of higher accuracy to the default activation, Rectifier. At the same time, Rectifier activation has vanished gradient for negative inputs while Tanh activation has both positive and negative scales, which is preferred in return prediction of this research. Tanh activation with dropout rate is chosen in order into reduce overfitting problems, according to Kochura et al. (2017), and the number is set at 0.1. Secondly, having a similar role with dropout, L1 and L2 penalty are applied instead of using the default. Secondly, having a similar role with dropout, L1 and L2 penalty are applied instead of using the default. Thirdly, the validation set is specified to help tune the deep learning model. The validation frame can be used to stop the model earlier when overwrite_with_best_model = T and keep the optimized model without running too many rounds. An additional set of seed is needed to generate robust results. The number of seeds must be the same for splitting validation set at the beginning and processing deep learning later. At last, the reproducible argument is needed to be true to make sure there is only one unique series of output. Other unmentioned arguments are not specified and using the default.

The number of hidden layers and layer size are crucial for deep learning models. According to different situations, there could be some optimization of models after trials. Therefore, six types of hidden layers and two types of layer size are designed in this paper to compare deep learning models. The number of layers follows a sequence from 1 to 6, while in each layer the number of neurons is set as either 50 or 100, corresponding to recent prevalent study on the Internet. The neuron number is relatively large because using the available inputs to predict stock returns seems rather complicated and may require a more complex model with more hidden layers and neurons. Big size and many layers might result in the overfitting problem. Therefore, arguments like dropout are necessary. Since we have two types of inputs, there are three alternatives for input-dependent models, which are 15 technical indicators, five liquidity proxies, and 20 variables together.

By running lay-dependent deep learning models, two optimal ones with either 50 neurons or 100 neurons are generated in the first part of Section 4. After generating the optimal models in both cases, the input-dependent models are conducted and to find out if there is a significant difference among different inputs.

E. Dynamic Training Window

A common way to run deep learning model is to split the sample data to training, validation and test sets. When the sample data is fixed, the prediction numbers of test period will come out at one time after training and validation. Abe and Nakayama (2018) suggest using a moving window for training and testing to predict the one-month forward return for each stock. In their paper, the latest N month value of input variables comprise a set of training data to predict the stock return, while most of input variables are fundamental factors. In addition, Skabar and Cloete (2002) apply the similar methodology while using technical indicators as input variables. Both papers claim that the prediction model can achieve better optimization with dynamic training and testing data because it can change from past to future. Therefore, this paper also uses the moving window for training and validation. In the 6-year training and validation procedure, neural networks analyze the relation between 20 input variables and one-month forward returns for an individual stock. After optimizing the model, the return of the first month after that 6-year period is predicted based upon testing-period input dataset, which is across only one month. For instance, the first deep learning model of stock *i* will use its technical indicators and liquidity measures from January 2010 through November 2015, as the set of inputs. The corresponding output is the return series of that stock from February 2010 to December 2015. After accomplishing the first deep learning model, its return of January 2016 is predicted using input data of December 2015. The next model is reconstructed in the same way by moving the training period and the testing period one month forward. In the end, the stock *i* will have 24 values of predicted returns from January 2016 to December 2017.

Meanwhile, the same sets of data are regressed linearly for return prediction, as the counterpart of neural networks. Similarly, a set of 6-year input variables are explanatory factors, which are defined as x_i at time *t* in the formula below and the return in one month is the explained factor.

$$R_{t+1} = a_t + \beta_{jt} X_{jt} + \varepsilon t. \qquad (6)$$

The linear regression will move month by month to predict return series. The result of deep learning regressions and linear regressions can be comparable due to the same regression dataset.

IV. Empirical Results

This section compares deep learning models and the linear model in many situations. The first part compares the predictive neural network models with different settings of layer and layer size. The results show that the complex model is not necessarily better than simple models with few hidden layers. In the second part, to investigate how the embedded information varies for different types of inputs, the input is adjusted for the multi-layer models with the best performances from the first comparison. A portfolio construction strategy is applied in the last part to evaluate the prediction power of neural networks.

A. Predicting Results of Multi-layer Models

We build different types of neural network model for one-month forward returns. From January 2016 to December 2017, the return of each stock in each month will be predicted by 12 different deep learning models and one linear model, respectively. In total, we have 5208 predicted stock returns for each model. Since the deep learning regression models share the same data sample with the linear model, the goodness of the model is comparable.

The common evaluation metrics available for regression models include Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE). There is no perfect evaluation method currently for deep learning models. Therefore, by defining as follows, this paper chooses MAE and RMSE as evaluation metrics for neural networks. Specifically, y_i is the actual return of the stock *i*, $\hat{y_i}$ is the predicted return of the stock *i*, and *n* is the number of total predicted returns, which equal to 5208.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{7}$$

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \tag{8}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right|. \tag{9}$$

The result of MAE and RMSE of models with different neuron numbers and layer numbers is summarized in Table II. In general, models with 100 neurons have smaller MAE and RMSE than those having 50 neurons with the same hidden layer. It indicates that more neurons lift the accuracy level. Another important observation is that a good model is not necessary to have as many layers as possible. For example, given all models with 50 neurons in each hidden layer, the best model with lowest MAE and RMSE has four hidden layers, and given all models with 100 neurons in each hidden layer, the best model with lowest MAE and RMSE has two hidden layers. Therefore, it also implies that there is no general optimal solution for neural networks. By involving in the results of the linear regression model, a conclusion can be drawn that all deep learning models outperform the simple linear regression.

Fig. 1 displays MAE and RMSE for models with 50 neurons, while Fig. 2 is made for 100-neuron models. In general, there is no significant difference between the trend of MAE and RMSE, and both curves are U-sharp with optimal solutions.

TABLE II: MAE AND RMSE OF DEEP LEARNING MODELS AND THE LINEAR REGRESSION MODEL

	50 neurons		100 neurons	
Hidden layer	MAE （%）	RMSE（%）	MAE （%）	RMSE （%）
1	5.0005	6.5451	4.4308	5.8721
2	4.9184	6.5078	**4.2180**	**5.6266**
3	4.8271	6.4628	4.2917	5.7330
4	**4.7369**	**6.3715**	4.4176	5.9559
5	5.1255	6.8640	4.7850	6.4469
6	5.3215	7.1462	5.0918	6.8311
Linear Regression	5.3799	7.2284	5.3799	7.2285

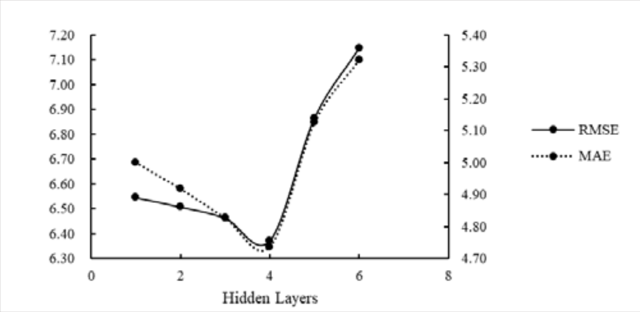

Fig. 1. Deep learning models with 50 neurons in each hidden layer

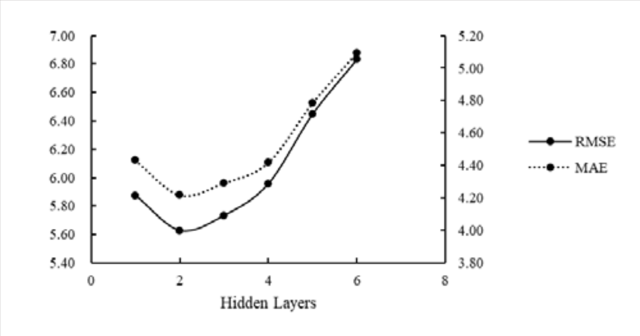

Fig. 2. Deep learning models with 100 neurons in each hidden layer

B. *Predicting Results of Input Models*

Using the best models with 50 and 100 neurons, we further compare the predicting abilities of different types of inputs. The result is shown in Table III.

The models with 50 neurons in every four hidden layers and with 100 neurons in every two hidden layers are selected from the previous work. At the same time, the linear model using the same sets of data is conducted, regarding the result in Table 2. A new evaluation method is used to compare models with different types of input, which called Volatility of Forecasted Errors (VFE). It is the standard deviation of forecast errors. The VFE and RMSE have almost the same values but the VFE is more direct and easier to understand intuitively. When it comes to models with different size of samples, especially for the comparison in this part, the VFE is preferred for more precise estimation of models.

TABLE III: VFE OF NEURAL NETWORKS AND LINEAR REGRESSION

	4*50	2*100	Linear Regression
All Inputs	6.3695	5.6265	7.1976
Tech. Only	6.8165	5.9534	6.7223
Liquid. Only	8.1689	8.0394	8.9180

Table III shows that both technical indicators and liquidity measures contribute to the deep learning prediction because models with all inputs have the lowest VFE compared with technical-indicator-only models and liquidity-measure-only models. Also, technical indicators might be better input for return predictions than liquidity measures because technical-indicator-only models have much lower VFE than liquidity-measure-only models.

C. *Portfolio Construction*

Based on the above comparisons, it is hard to draw further conclusions concerning the goodness of neuron network models because the evaluation metrics such as MAE and RMSE tell limited information about the goodness of models.

Therefore, a portfolio construction strategy is employed to help understand the capability and application of predicted results.

The best model assessed previously with the lowest errors is the one with two hidden layers and 100 neurons in each. Therefore, we implement this model as a representative of deep learning forecasting. Before moving to further statistic test, we check the distribution of the returns predicted from the deep learning model the linear regression model.

Fig. 3 and Fig. 4 show a cross-sectional distribution of the predicted returns. In these figures, the horizontal axis represents the equal interval of monthly returns from -14.4% to 21.6% and the vertical axis counts the number of firms in each interval. Thus, we generally confirm that the cross-sectional returns predicted from two models subject to a normal distribution.

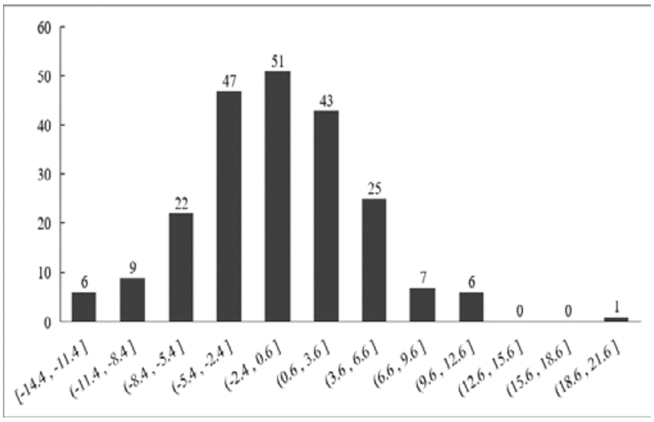

Fig. 3. Histogram of Oct. 2017 predicted returns by deep learning

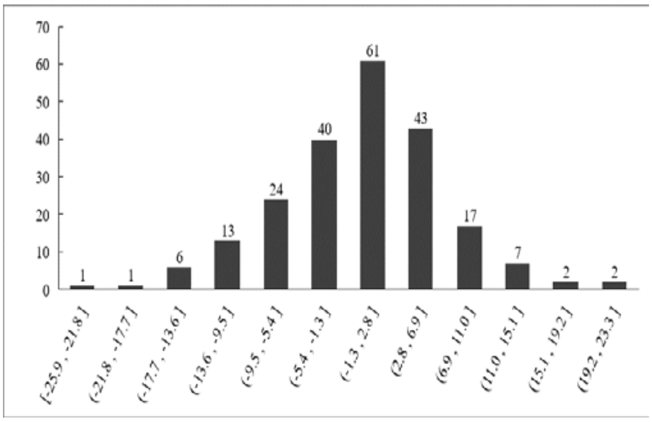

Fig. 4. Histogram of Oct. 2017 predicted returns by linear regression

Then, we build equally weighted high-return portfolios and low-return portfolios. The return is estimated from neural networks and linear regression, respectively. At first, 217 of return predictions in each predicting month are sorted in descending order. The top-50 return group and bottom-50 return group are formed 24 times during the two-year period. The next step is to equally weight each stock in the top and the bottom categories, generating High-return Portfolio (HRP) and Low-return Portfolio (LRP) in each month. The stock composition could be different over time, indicating both portfolios need to be monthly rebalanced. At last, the return of portfolios is calculated based on the actual return of each composition and the series of return spreads between high and low is tested with student t-test. In the following paragraph, HRP-DP and LRP-DP refer to portfolios based on the best deep learning model and HRP-LR and LRP-LR indicates portfolios constructed with linear regression. All predictions are generated with all types of inputs.

HRP-DP shows an average 7.50% monthly return and LPR-DP shows an average monthly return of -4.61%, seen in Table IV. Meanwhile, the average monthly return of Nikkei from 2016 to 2017 is 1.24%, as the benchmark. Return of HRP-LR is 6.67% and return of LRP-LR is -4.15%. The difference between the top portfolio and the bottom portfolio is statistically significant since p-values are smaller than 5%. Portfolios built under deep learning regression have bigger t statistics, indicating that deep learning method distinguishes the well-performed and poor-performed stocks better than simple linear regression. If the good and bad stock shares are apart, it is easier for investors to apply further strategies, like long-short strategy, and generate excess return.

As a result, the return of high-return portfolios predicted by deep learning models precede the benchmark Nikkei 225 index and low-return portfolios both with great statistical significance. The larger number of t statistic in terms of the difference between HRP-DP and Nikkei 225 than that in terms of the difference between HRP-LR and Nikkei 225 shows that the deep learning model has greater forecasting power than linear regression.

We draw return series of HRP-DP, LRP-DP and Nikkei 225 from 2016 to 2017 in Fig.5. Return curve of HRP-DP is always above that of LRP-DP while monthly returns of Nikkei 225 are roughly between the two curves. The trend of all curves is similar and shows co-movement in general.

TABLE IV: RESULT OF WELCH'S TWO-SAMPLE T-TEST

Mean (%): HRP-DP	7.5023	Mean (%): HRP-LP	6.6689
Mean (%): LRP-DP	-4.6064	Mean (%): LRP-LP	-4.1526
Diff. (%):	12.1087	Diff. (%)	10.8215
t statistic:	7.6980	t statistic	7.0039
Mean (%): HRP-DP	7.5023	Mean (%): HRP-LP	6.6689
Mean (%): Nikkei 225	1.2434	Mean (%): Nikkei 225	1.2434
Diff. (%):	6.2588	Diff. (%):	5.4255
t statistic:	4.7684	t statistic:	4.1845

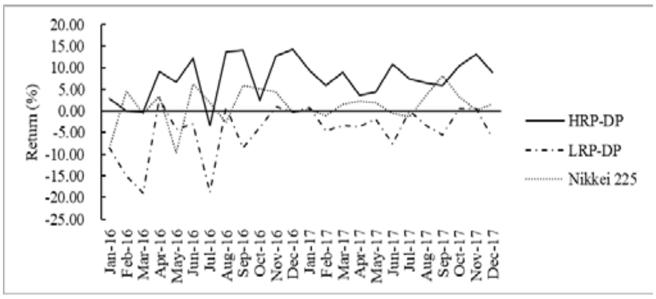

Fig. 5. Histogram of Oct. 2017 predicted returns by deep learning

V. CONCLUSIONS AND FUTURE WORK

In this paper, we explore the prediction function and relative performance of artificial neural networks with the H2O platform. The hidden layers and layer size of neural networks attach great importance for prediction power. For problems like return prediction in this paper, specific scales of hidden layers and neurons are pre-set to find the optimal combination. As for the input, this paper chooses technical indicators and liquidity measures to predict monthly stock returns with moving windows, which could improve the robustness of the model. The main conclusion is that neural network model with all the inputs and 100 neurons in two hidden layers has the best forecast results than others, depending on the evaluation metrics MAE and RMSE, and both technical and liquidity proxies contribute to a better prediction. We also investigate the effects of different types of input variables on predictions. The results show that both technical and liquidity proxies contribute to the analysis. The further application of neural network predictions is to build ranked portfolios. We find that relatively high return stocks predicted by neural networks have significantly higher actual return than that of relatively low return stocks grouped in the same way. Hence the prediction lead by neural networks can provide a certain of accuracy in predicting the Japanese stock market. In summary, this study applies the neural network to the stock market and provides a new idea for using deep learning to investment decisions.

This study has a few important limitations. In this research, we only compare neural network models with linear regression. The reason is that the linear regression is generally used in the current field of asset pricing, such as the capital asset pricing model (CAPM) and the Fama-French three-factor model. In terms of tools, there is a more advanced function of H2O called H2O grid search that can optimize the hyperparameter. If this function is conducted, there could be a more precise model of neural networks with not only 50 or 100 neurons. However, running this function is quite time-consuming. The trade-off between running time and prediction accuracy has been a popular research topic in the deep learning field. Considering that the further model exploration has gone beyond the primary purpose of this research, we only program using standard H2O functions. We expect a more efficient and accurate deep learning model which could be applied in the financial area in the future.

REFERENCES

[1] Abe, M. and Nakayama, H. (2018), "Deep learning for forecasting stock returns in the cross-section," In PAKDD 2018, Advances in Knowledge Discovery and Data Mining, 273-284.

[2] Amihud, Y. (2002), "Illiquidity and stock returns: Cross-section and time-series effects," Journal of Financial Markets, 5 (1), 31-56.

[3] Cao, Q., Leggio, K.B. and Schniederjans, M.J. (2005), "A comparison between Fama and French's model and artificial neural networks in predicting the Chinese stock market," Computers and Operations Research, 32 (10), 2499-2512.

[4] Caporin, M., Ranaldo, A. and Santucci de Magistris, P. (2013), "On the predictability of stock prices: A case for high and low prices", Journal of Banking and Finance, 37 (12), 5132-5146.

[5] Datar, V.T., N. Y. Naik and R. Radcliffe (1998), "Liquidity and stock returns: An alternative test," Journal of Financial Markets, 1 (2), 203-209.

[6] Friesen, G.C., Weller, P.A. and Dunham, L.M. (2009), "Price trends and patterns in technical analysis: A theoretical and empirical examination", Journal of Banking and Finance, 33 (6), 1089-1100.

[7] Kochura, Y., Stirenko, S. and Gordienko, Y. (2017), "Comparative performance analysis of neural networks architectures on H2O platform for various activation functions," In 2017 IEEE International, Young Scientists Forum on Applied Physics and Engineering (YSF), 70-73.

[8] Lesmond, D.A., J. P. Ogden and C. A. Trzcinka (1999), "A new estimate of transaction costs," Review of Financial Studies, 12 (5), 1113-1141.

[9] Liu, W. (2006), "A liquidity-augmented capital asset pricing model," Journal of Financial Economics, 82 (3), 631-671.

[10] Mizuno, H., Kosaka, M., Yajima, H. and Komoda, N. (1998), "Application of neural network to technical analysis of stock market prediction," Studies in Information and Control, 7 (3), 111-120.

[11] Pastor, L. and R. F. Stambaugh (2003), "Liquidity risk and expected stock returns," Journal of Political Economy, 111 (3), 642-685.

[12] Sezer, O., Ozbayoglu, A. and Dogdu, E. (2017), "A Deep Neural-Network Based Stock Trading System Based on Evolutionary Optimized Technical Analysis Parameters," Procedia Computer Science, 114, 473–480.

[13] Shynkevich, Y., T. M. McGinnity, S. A. Coleman, A. Belatreche and Y. Li (2017), "Forecasting price movements using technical indicators: Investigating the impact of varying input window length," Neurocomputing, 264, 71–88.

[14] Skabar, A. and Cloete, I. (2002), "Neural networks, financial trading and the efficient markets hypothesis," Australian Computer Science Communications, 24 (1), 241-249.