



Deep Factor Model

Explaining Deep Learning Decisions for Forecasting Stock Returns with Layer-Wise Relevance Propagation

Kei Nakagawa^{1,3}, Takumi Uchida³, and Tomohisa Aoshima^{2,4}

¹ Nomura Asset Management Ltd., Chuo-ku, Japan
kei.nak.0315@gmail.com

² Fujitsu Cloud Technologies Limited, Shinjuku-ku, Japan

³ Graduate School of Business Sciences, University of Tsukuba, Tsukuba, Japan

⁴ Department of Risk Engineering, University of Tsukuba, Tsukuba, Japan

Abstract. We propose to represent a return model and risk model in a unified manner with deep learning, which is a representative model that can express a nonlinear relationship. Although deep learning performs quite well, it has significant disadvantages such as a lack of transparency and limitations to the interpretability of the prediction. This is prone to practical problems in terms of accountability. Thus, we construct a multifactor model by using interpretable deep learning. We implement deep learning as a return model to predict stock returns with various factors. Then, we present the application of layer-wise relevance propagation (LRP) to decompose attributes of the predicted return as a risk model. By applying LRP to an individual stock or a portfolio basis, we can determine which factor contributes to prediction. We call this model a deep factor model. We then perform an empirical analysis on the Japanese stock market and show that our deep factor model has better predictive capability than the traditional linear model or other machine learning methods. In addition, we illustrate which factor contributes to prediction.

Keywords: Deep factor model · Deep learning · Layer-wise relevance propagation

1 Introduction

An essential tool of quantitative portfolio management is the multifactor model. The model explains the stock returns through multiple factors. A general multifactor model in the academic finance field is sometimes used synonymously with the arbitrage pricing theory (APT) advocated by Ross [24]. The APT multifactor model includes a method of providing macroeconomic indicators a priori to explain stock returns and a method of extracting factors by factor analysis from past stock returns.

However, in practice, the Fama-French approach and the BARRA approach based on ICAPM [20] are widely used. The Fama-French or Barra multifactor

models correspond to a method of finding stock returns using the attributes of individual companies such as investment valuation ratios represented by PER and PBR.

The Fama-French approach was introduced for the first time by Fama and French [9]. The Barra approach was introduced by Rosenberg [23] and was extended by Grinold and Kahn [13]. It is calculated through cross-section regression analysis since it assumes that stock returns are explained by common factors.

In addition, there are two uses of the multifactor model. It can be employed both to enhance returns and to control risk. In the first case, if one is able to predict the likely future value of a factor, a higher return can be achieved by constructing a portfolio that tilts toward “good” factors and away from “bad” ones. In this situation, the multifactor model is called a return model or an alpha model.

On the other hand, by capturing the major sources of correlation among stock returns, one can construct a well-balanced portfolio that diversifies specific risk away. This is called a risk model. There are cases where these models are confused when being discussed in the academic finance field.

For both the return model and the risk model, the relationship between the stock returns and the factors is linear in the traditional multifactor model mentioned above. By contrast, linear multifactor models have proven to be very useful tools for portfolio analysis and investment management. The assumption of a linear relationship is quite restrictive. Considering the complexity of the financial markets, it is more appropriate to assume a nonlinear relationship between the stock returns and the factors.

Therefore, in this paper, we propose to represent a return model and risk model in a unified manner with deep learning, which is a representative model that can express a nonlinear relationship. Deep learning is a state-of-the-art method for solving various challenging machine learning problems [11], e.g., image classification, natural language processing, or human action recognition. Although deep learning performs quite well, it has a significant disadvantage: a lack of transparency and limitations to the interpretability of the solution. This is prone to practical problems in terms of accountability. Because institutional investors have fiduciary duty and accountability for their customers, it is difficult for them to use black-box type machine learning technique such as deep learning. Thus, we construct a multifactor model by using interpretable deep learning.

We implement deep learning to predict stock returns with various factors as a return model. Then, we present the application of layer-wise relevance propagation (LRP [3]) to decompose attributes of the predicted return as a risk model. LRP is an inverse method that calculates the contribution of inputs to the prediction made by deep learning. LRP was originally a method for computing scores for image pixels and image regions to denote the impact of a particular image region on the prediction of a classifier for a particular test image. By applying LRP to an individual stock or a quantile portfolio, we can determine which factor contributes to prediction. We call the model a deep factor model.

We then perform an empirical analysis on the Japanese stock market and show that our deep factor model has better predictive power than the traditional linear model or other machine learning methods. In addition, we illustrate which factor contributes to prediction.

2 Related Works

Stock return predictability is one of the most important issues for investors. Hundreds of papers and factors have attempted to explain the cross section of expected returns [14, 19, 25]. Academic research has uncovered a large number of such factors, 314 according to Harvey et al. [14], with the majority being identified during the last 15 years.

The most popular factors of today (Value, Size, and Momentum) have been studied for decades as part of the academic asset pricing literature and practitioner risk factor modeling research. One of the best-known efforts in this field came from Fama and French in the early 1990s. Fama and French [9] put forward a model explaining US equity market returns with three factors: the market (based on the traditional CAPM model), the size factor (large vs. small capitalization stocks), and the value factor (low vs. high book to market). The Fama-French three-factor model, which today includes Carhart’s momentum factor [6], has become a canon within the finance literature. More recently, the low risk [4] and quality factors [21] have become increasingly well accepted in the academic literature. In total, five factors are studied the most widely [15].

Conversely, the investors themselves must decide how to process and predict returns, including the selection and weighting of such factors. One way to make investment decisions is to rely upon the use of machine learning. This is a supervised learning approach that uses multiple factors to explain stock returns as input values and future stock returns as output values. Many studies on stock return predictability using machine learning have been reported. Cavalcante et al. [7] presented a review of the application of several machine learning methods in financial applications. In their survey, most of these were forecasts of stock market returns; however, forecasts of individual stock returns using the neural networks dealt with in this paper were also conducted.

In addition, Levin [18] discussed the use of multilayer feed forward neural networks for predicting a stock return with the framework of the multifactor model. To demonstrate the effectiveness of the approach, a hedged portfolio consisting of equally capitalized long and short positions was constructed, and its historical returns were benchmarked against T-bill returns and the S&P500 index. Levin achieved persistent returns with very favorable risk characteristics.

Abe and Nakayama [2] extended this model to deep learning and investigated the performance of the method in the Japanese stock market. They showed that deep neural networks generally outperform shallow neural networks, and the best networks also outperform representative machine learning models. These results indicate that deep learning has promise as a skillful machine learning method to predict stock returns in the cross section.

However, these related works are only for use as a return model, and the problem is that the viewpoint as a risk model is lacking.

3 Methodology – Deep Factor Model

3.1 Deep Learning

The fundamental machine learning problem is to find a predictor $f(x)$ of an output Y given an input X . As a form of machine learning, deep learning trains a model on data to make predictions, but it is distinguished by passing learned features of data through different layers of abstraction. Raw data is entered at the bottom level, and the desired output is produced at the top level, which is the result of learning through many levels of transformed data. Deep learning is hierarchical in the sense that in every layer, the algorithm extracts features into factors, and a deeper level’s factors become the next level’s features.

A deep learning architecture can be described as follows (1). We use $l \in 1, \dots, L$ to index the layers from 1 to L , which are called hidden layers. The number of layers L represents the depth of our architecture. We let $z^{(l)}$ denote the l -th layer, and so $X = z^{(0)}$. The final output is the response Y , which can be numeric or categorical.

The explicit structure of a deep prediction rule is then

$$\begin{aligned} z^{(1)} &= f^{(1)}(W^{(0)}X + b^{(0)}) \\ z^{(2)} &= f^{(2)}(W^{(1)}z^{(1)} + b^{(1)}) \\ &\vdots \\ z^{(L-1)} &= f^{(L-1)}(W^{(L-2)}z^{(L-2)} + b^{(L-2)}) \\ Y &= f^{(L)}(W^{(L-1)}z^{(L-1)} + b^{(L-1)}) \end{aligned} \tag{1}$$

Here, $W^{(l)}$ are weight matrices, and $b^{(l)}$ are the threshold or activation levels. $z^{(l)}$ are hidden features that the algorithm extracts. Designing a good predictor depends crucially on the choice of univariate activation functions $f^{(l)}$. Commonly used activation functions are sigmoidal (e.g., $\frac{1}{(1+\exp(-x))}$), $\cosh(x)$, or $\tanh(x)$) or rectified linear units (ReLU) $\max\{x, 0\}$.

3.2 Layer-Wise Relevance Propagation

LRP is an inverse method that calculates the contribution of the prediction made by the network. The overall idea of decomposition is explained in [3]. Here, we briefly reiterate some basic concepts of LRP with a toy example (Fig. 1). Given input data x , a predicted value $f(x)$ is returned by the model denoted as function f . Suppose the network has L layers, each of which is treated as a vector with dimensionality $V(l)$, where l represents the index of layers. Then, according to

the conservation principle, LRP aims to find a relevance score R_d for each vector element in layer l such that the following equation holds:

$$f(x) = \sum_{d \in V(L)} R_d^{(L)} = \dots = \sum_{d \in V(l)} R_d^{(l)} = \dots = \sum_{d \in V(1)} R_d^{(1)} \quad (2)$$

As we can see in the above formula (2), LRP uses the prediction score as the sum of relevance scores for the last layer of the network, and maintains this sum throughout all layers.

Figure 1 shows a simple network with six neurons. w_{ij} are weights, z_i are outputs from activation, and $R_i^{(l)}$ are relevance scores to be calculated. Then, we have the following equation:

$$\begin{aligned} f(x) &= R_6^{(3)} \\ &= R_5^{(2)} + R_4^{(2)} \\ &= R_3^{(1)} + R_2^{(1)} + R_1^{(1)} \end{aligned} \quad (3)$$

Furthermore, the conservation principle also guarantees that the inflow of relevance scores to one neuron equals the outflow of relevance scores from the same neuron. $z_{ij}^{(l,l+1)}$ is the message sent from neuron j at layer $l+1$ to neuron i at layer l . In addition, $R_d^{(l)}$ is computed using network weights according to the equation below:

$$R_i^{(l)} = \sum_j \frac{z_{ij}^{(l,l+1)}}{\sum_k z_{kj}^{(l,l+1)}} R_j^{(l+1)}, \quad z_{ij}^{(l,l+1)} = w_{ij} z_i^{(l)} \quad (4)$$

Therefore, LRP is a technique for determining which features in a particular input vector contribute most strongly to a neural network's output.

3.3 Deep Factor Model

In this paper, we propose to represent a return model and risk model in a unified manner with deep learning, which is a representative model that can express a nonlinear relationship. We call the model a deep factor model. First, we formulate a nonlinear multifactor model with deep learning as a return model.

The traditional fundamental multifactor model assumes that the stock return r_i can be described by a linear model:

$$r_i = \alpha_i + X_{i1}F_1 + \dots + X_{iN}F_N + \varepsilon_i \quad (5)$$

where F_i are a set of factor values for stock i , X_{in} denotes the exposure to factor n , α_i is an intercept term that is assumed to be equal to a risk-free rate of return under the APT framework, and ε_i is a random term with mean zero and is assumed to be uncorrelated across other stock returns. Usually, the factor exposure X_{in} is defined by the linearity of several descriptors.

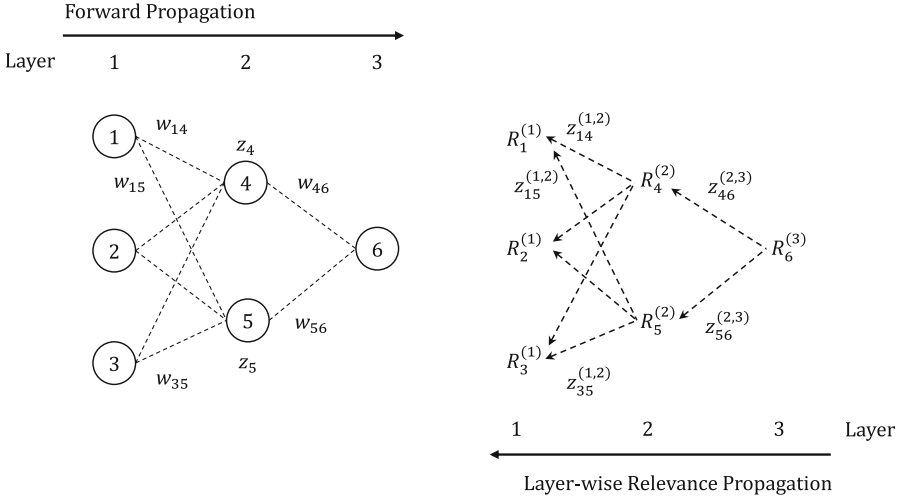


Fig. 1. LRP with toy example

While linear multifactor factor models have proven to be very effective tools for portfolio analysis and investment management, the assumption of a linear relationship is quite restrictive. Specifically, the use of linear models assumes that each factor affects the return independently. Hence, they ignore the possible interaction between different factors. Furthermore, with a linear model, the expected return of a security can grow without bound as its exposure to a factor increases.

Considering the complexity of the financial markets, it is more appropriate to assume a nonlinear relationship between the stock returns and the factors. Generalizing (5), maintaining the basic premise that the state of the world can be described by a vector of factor values and that the expected stock return is determined through its coordinates in this factor world leads to the nonlinear model:

$$r_i = \tilde{f}(X_{i1}, \dots, X_{iN}, F_1, \dots, F_N) + \varepsilon_i \tag{6}$$

where \tilde{f} is a nonlinear function.

The prediction task for the nonlinear model (6) is substantially more complex than that in the linear case since it requires both the estimation of future factor values as well as a determination of the unknown function \tilde{f} . As in a previous study [18], the task can be somewhat simplified if factor estimates are replaced with their historical means \bar{F}_n . Since the factor values are no longer variables, they are constants. For the nonlinear model (6), the expression can be transformed as follows:

$$\begin{aligned} r_i &= \tilde{f}(X_{i1}, \dots, X_{iN}, \bar{F}_1, \dots, \bar{F}_N) + \varepsilon_i \\ &= f(X_{i1}, \dots, X_{iN}) + \varepsilon_i \end{aligned} \tag{7}$$

where X_{in} is now the security’s factor exposure at the beginning of the period over which we wish to predict. To estimate the unknown function f , a family of models needs to be selected, from which a model is to be identified. In the following, we propose modeling the relationship between factor exposures and future stock returns using a class of deep learning.

However, deep learning has significant disadvantages such as a lack of transparency and limitations to the interpretability of the solution. This is prone to practical problems in terms of accountability. Then, we present the application of LRP to decompose attributes of the predicted return as a risk model. By applying LRP to an individual stock or a quantile portfolio, we can determine which factor contributes to prediction. If you want to show the basis of the prediction for a stock return, you can calculate LRP using the inputs and outputs of the stock. In addition, in order to obtain the basis of prediction for a portfolio, calculate LRPs of the stocks included in that portfolio and take their average. Then, by aggregating the factors, you can see which factor contributed to the prediction. Figure 2 shows an overall diagram of the deep factor model.

4 Experiment on Japanese Stock Markets

4.1 Data

We prepare a dataset for TOPIX index constituents. TOPIX is a well-accepted stock market index for the Tokyo Stock Exchange (TSE) in Japan, tracking all domestic companies of the exchange’s First Section. It is calculated and published by the TSE. As of March 2016, the index is composed of 1,948 constituents. The index is also often used as a benchmark for overseas institutional investors who are investing in Japanese stocks.

We use the 5 factors and 16 factor exposures listed in Table 1. These are used relatively often in practice and are studied the most widely in academia [15].

In calculating these factors, we acquire necessary data from the Nikkei Portfolio Master and Bloomberg. Factor exposures are calculated on a monthly basis (at the end of month) from December 1990 to March 2016 as input data. Stock returns with dividends are acquired on a monthly basis (at the end of month) as output data.

4.2 Model

Our problem is to find a predictor $f(x)$ of an output Y , next month’s stock returns given an input X , various factors. One set of training data is shown in Table 3. In addition to the proposed deep factor model, we use a linear regression model as a baseline, and support vector regression (SVR [8]) and random forest [5] as comparison methods. The deep factor model is implemented with TensorFlow [1], and the comparison methods are implemented with scikit-learn [22]. Table 2 lists the details of each model.

We train all models by using the latest 60 sets of training data from the past 5 years. The models are updated by sliding one month ahead and carrying

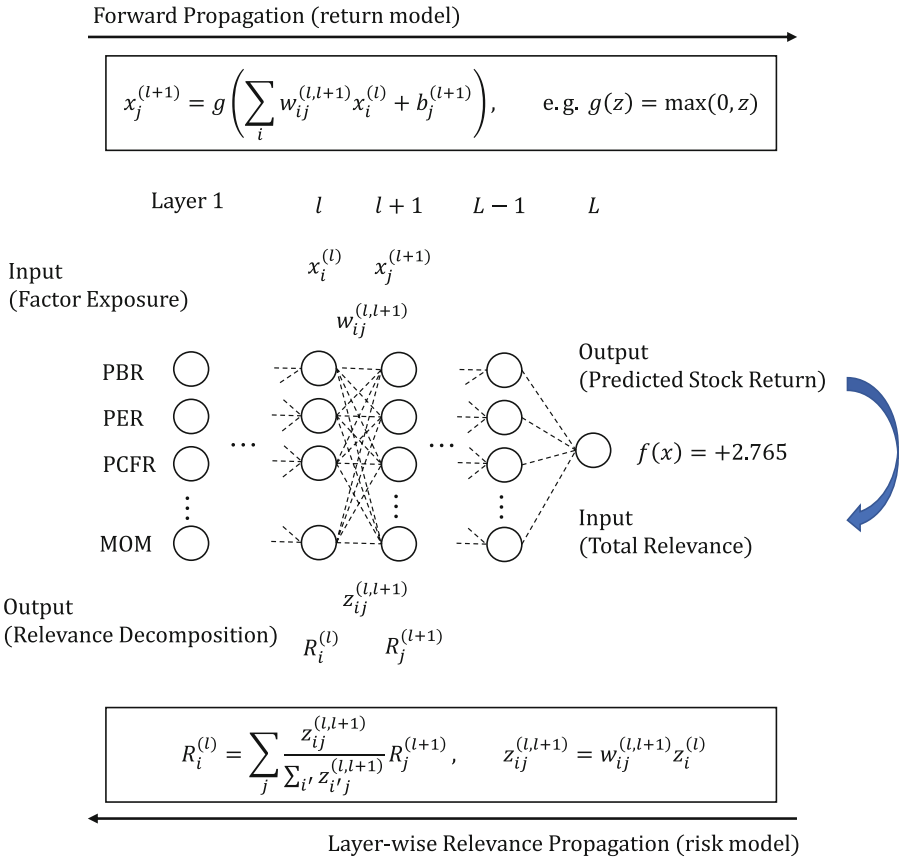


Fig. 2. Deep factor model

out a monthly forecast. The prediction period is 10 years, from April 2006 to March 2016 (120 months). This is because we wanted to hold a test period over 10 years including the date of Lehman shock. But, we have to check the impact of reference period choice on performance for further study. Figure 3 shows the image of our prediction framework. In order to verify the effectiveness of each method, we compare the prediction accuracy of these models and the profitability of the quintile portfolio. We construct a long/short portfolio strategy for a net-zero investment to buy top stocks and to sell bottom stocks with equal weighting in quintile portfolios. For the quintile portfolio performance, we calculate the annualized average return, risk, and Sharpe ratio. In addition, we calculate the average mean absolute error (MAE) and root mean squared error (RMSE) for the prediction period as the prediction accuracy.

Table 1. Factors and factor descriptors

Factor	Descriptor	Formula
Risk	60VOL	Standard deviation of stock returns in the past 60 months
	BETA	Regression coefficient of stock returns and market risk premium
	SKEW	Skewness of stock returns in the past 60 months
Quality	ROE	Net income/Net Assets
	ROA	Operating Profit/Total Assets
	ACCRUALS	Operating Cashflow – Operating Profit
	LEVERAGE	Total Liabilities/Total Assets
Momentum	12-1MOM	Stock returns in the past 12 months except for past month
	1MOM	Stock returns in the past month
	60MOM	Stock returns in the past 60 months
Value	PSR	Sales/Market Value
	PER	Net Income/Market Value
	PBR	Net Assets/Market Value
	PCFR	Operating Cashflow/Market Value
Size	CAP	$\log(\text{Market Value})$
	ILLIQ	$\text{average}(\text{Stock Returns}/\text{Trading Volume})$

Table 2. Details of each method

Model	Description
Deep factor model	Model 1 The hidden layers are {80-50-10}. We use the ReLU as the activation function and Adam [16] for the optimization algorithm
	Model 2 The hidden layers are {80-80-50-50-10-10}. We use the ReLU as the activation function and Adam [16] for the optimization algorithm
Linear model	Linear models is implemented with scikit-learn with the class “sklearn.linear_model.LinearRegression”. All parameters are default values in this class
SVR	Support vector regression (SVR) is implemented with scikit-learn with the class “sklearn.svm.SVR”. All parameters are default values in this class
Random forest	Random Forest is implemented with scikit-learn with the class “sklearn.ensemble.RandomForestRegressor”. All parameters are default values in this class

Table 3. One set of training data for March 2016.

Input: 80 dim	Output: 1 dim
Factor descriptors: 16 × 5 dim	Return: 1 dim
February 2016	March 2016
November 2015	
August 2015	
May 2015	
February 2015	

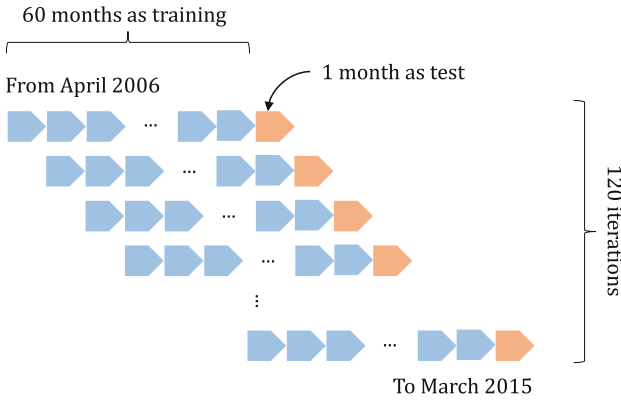


Fig. 3. Stock prediction framework.

4.3 Results

Table 4 lists the average MAE and RMSE of all years and the annualized return, volatility, and Sharpe ratio for each method. In the rows of the table, the best number appears in bold. Deep factor model 1 (shallow) has the best prediction accuracy in terms of MAE and RMSE as in the previous study [2, 18]. On the other hand, deep factor model 2 (deep) is the most profitable in terms of the Sharpe Ratio. The shallow model is superior in accuracy, while the deep one is more profitable. In any case, we find that both models 1 and 2 exceed the baseline linear model, SVR, and random forest in terms of accuracy and profitability. These facts imply that the relationship between the stock returns in the financial market and the factor is nonlinear, rather than linear. In addition, a model that can capture such a nonlinear relationship is thought to be superior.

Table 4. Average MAE and RMSE of all years and annualized return, volatility, and Sharpe ratio for each method.

	Deep factor model		Linear model	SVR	Random forest
	Model 1	Model 2			
Return [%]	10.81	10.31	8.17	1.46	0.12
Volatility [%]	7.65	6.86	8.20	9.66	5.43
Sharpe ratio	1.41	1.50	1.00	0.15	0.02
MAE	0.0663	0.0669	0.0679	0.1713	0.0728
RMSE	0.0951	0.0953	0.0965	0.1962	0.1024

4.4 Interpretation

Here, we try to interpret the stock of the highest predicted stock return and the top quintile portfolio based on the factor using deep factor model 2 as of the last time point of February 2016. In general, the momentum factor is not very effective, but the value and size factors are effective in the Japanese stock markets. Nowadays, there is a significant trend in Japan to evaluate companies that will increase ROE over the long term because of the appearance of the Corporate Governance Code. In response to this trend, the quality factor including ROE is gaining attention. But, [17] found that both the RMW and the CMA related to our quality factor are weakly associated with the cross-sectional variations of stock returns in long term, which is significantly different from the US evidence.

Figure 4 shows which factor contributed to the prediction in percentages using LRP. The contributions of each descriptor calculated by LRP are summed for each factor and are displayed as a percentile.

We observe that the quality and value factors account for more than half of the contribution to both the stock return and quintile portfolio. The quality factor and the momentum factor are not effective in the linear multifactor model, whereas their contribution is remarkably large in the Deep Factor Model. Moreover, the contribution of the size factor is small, and it turns out that there is a widely profitable opportunity regardless of whether the stock is large or small. Figure 5 shows that these trends do not change in time series. Therefore, the Deep Factor Model is stable in terms of interpretability.

Next, we quantitatively verify the risk model by LRP. Table 5 shows the correlation coefficients between each factor and the predicted return in the top quintile portfolio. The correlation coefficients are calculated by averaging the correlation coefficients between each descriptor and the predicted return by each factor. The influence of the value and size factor differs when looking at LRP and correlation. The value factor has a large contribution to LRP and a small contribution to the correlation coefficients. The size factor has the opposite contributions. Therefore, without LRP, we could misinterpret the return factors.

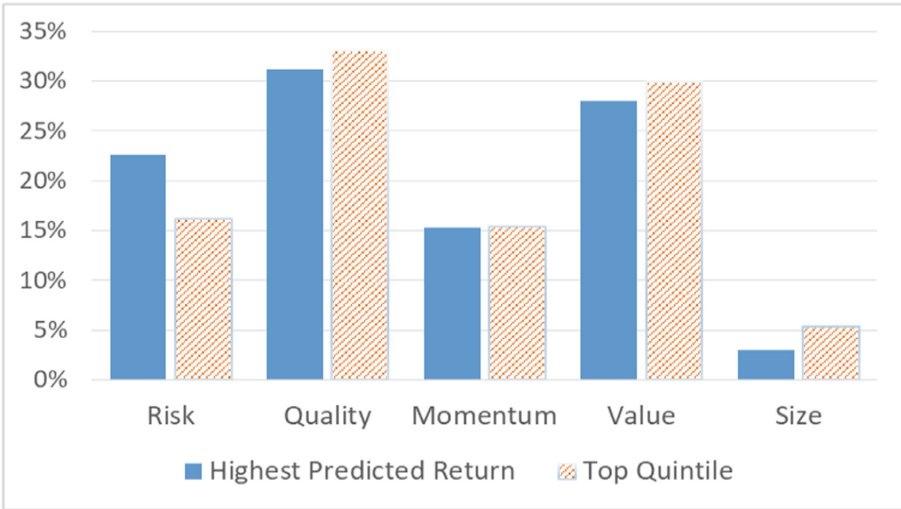


Fig. 4. Interpreting highest predicted return and top quintile portfolio based on factor using network as of last time point of February 2016

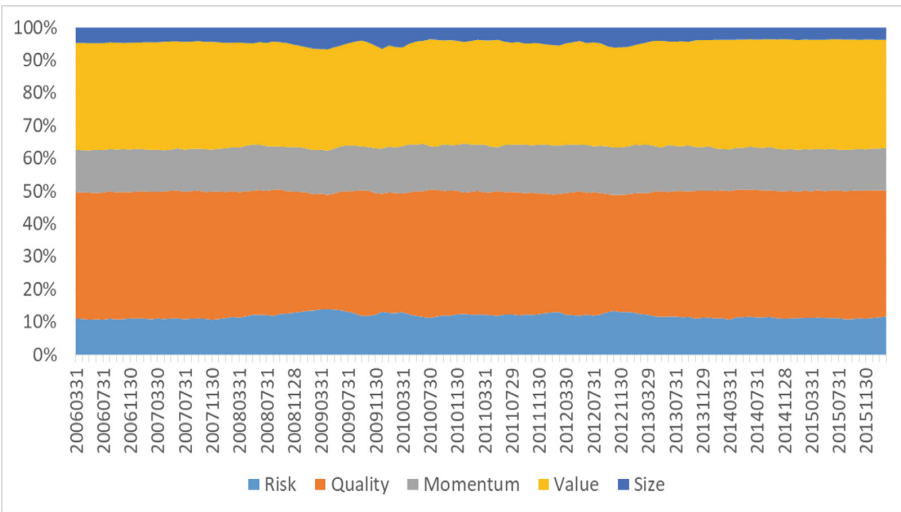


Fig. 5. Interpreting top quintile portfolio based on factor using network from April 2006 to February 2016

Table 5. Correlation coefficients between each factor and predicted return in top quintile portfolio.

	Risk	Quality	Momentum	Value	Size
Spearman	0.14	0.22	0.24	0.08	0.14
Kendall	0.10	0.15	0.17	0.06	0.10

5 Conclusion

We presented a method by which deep-learning-based models can be used for stock selection and risk decomposition. In terms of fiduciary duty and accountability for institutional investors, risk decomposition is important in practice.

Our conclusions are as follows:

- The deep factor model outperforms the linear model. This implies that the relationship between the stock returns in the financial market and the factors is nonlinear, rather than linear. The deep factor model also outperforms other machine learning methods including SVR and random forest.
- The shallow model is superior in accuracy, while the deep model is more profitable.
- Using LRP, it is possible to intuitively determine which factor contributed to prediction.

This study reports the main idea of deep factor model and initial results using Japanese stock market. We should check the stability of our model by using various stock market such as country-specific or global market [10, 12].

For further study, we would like to expand our deep factor model to a model that exhibits dynamic temporal behavior for a time sequence such as RNN.

References

1. Abadi, M., et al.: TensorFlow: a system for large-scale machine learning. In: OSDI, vol. 16, pp. 265–283 (2016)
2. Abe, M., Nakayama, H.: Deep learning for forecasting stock returns in the cross-section. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining (2018)
3. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS ONE* **10**(7), e0130140 (2015)
4. Blitz, D.C., Van Vliet, P.: The volatility effect: lower risk without lower return. *J. Portfolio Manag* (2007). Citeseer
5. Breiman, L.: Random forests. *Mach. Learn.* **45**(1), 5–32 (2001)
6. Carhart, M.M.: On persistence in mutual fund performance. *J. Financ.* **52**(1), 57–82 (1997)
7. Cavalcante, R.C., Brasileiro, R.C., Souza, V.L., Nobrega, J.P., Oliveira, A.L.: Computational intelligence and financial markets: a survey and future directions. *Expert Syst. Appl.* **55**, 194–211 (2016)
8. Drucker, H., Burges, C.J., Kaufman, L., Smola, A.J., Vapnik, V.: Support vector regression machines. In: *Advances in Neural Information Processing Systems*, pp. 155–161 (1997)
9. Fama, E.F., French, K.R.: The cross-section of expected stock returns. *J. Financ.* **47**(2), 427–465 (1992)
10. Fama, E.F., French, K.R.: Size, value, and momentum in international stock returns. *J. Financ. Econ.* **105**(3), 457–472 (2012)

11. Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y.: *Deep Learning*, vol. 1. MIT Press, Cambridge (2016)
12. Griffin, J.M.: Are the Fama and French factors global or country specific? *Rev. Financ. Stud.* **15**(3), 783–803 (2002)
13. Grinold, R.C., Kahn, R.N.: *Active portfolio management* (2000)
14. Harvey, C.R., Liu, Y., Zhu, H.: ... and the cross-section of expected returns. *Rev. Financ. Stud.* **29**(1), 5–68 (2016)
15. Jurczenko, E.: *Risk-Based and Factor Investing*. Elsevier, Amsterdam (2015)
16. Kinga, D., Adam, J.B.: A method for stochastic optimization. In: *International Conference on Learning Representations (ICLR)*, vol. 5 (2015)
17. Kubota, K., Takehara, H.: Does the Fama and French five-factor model work well in Japan? *Int. Rev. Financ.* **18**(1), 137–146 (2018)
18. Levin, A.E.: Stock selection via nonlinear multi-factor models. In: *Advances in Neural Information Processing Systems*, pp. 966–972 (1996)
19. McLean, R.D., Pontiff, J.: Does academic research destroy stock return predictability? *J. Financ.* **71**(1), 5–32 (2016)
20. Merton, R.C.: An intertemporal capital asset pricing model. *Econometrica: J. Econometric Soc.* 867–887 (1973)
21. Novy-Marx, R.: The other side of value: the gross profitability premium. *J. Financ. Econ.* **108**(1), 1–28 (2013)
22. Pedregosa, F., et al.: Scikit-Learn: machine learning in Python. *J. Mach. Learn. Res.* **12**(Oct), 2825–2830 (2011)
23. Rosenberg, B., McKibben, W.: The prediction of systematic and specific risk in common stocks. *J. Financ. Quant. Anal.* **8**(2), 317–333 (1973)
24. Ross, S.A., et al.: The arbitrage theory of capital asset pricing. *J. Econ. Theory* **13**(3), 341–360 (1976)
25. Subrahmanyam, A.: The cross-section of expected stock returns: what have we learnt from the past twenty-five years of research? *Eur. Financ. Manag.* **16**(1), 27–42 (2010)