

# Is it possible to forecast KOSPI direction using deep learning methods?

Songa Choi<sup>a</sup>, Jongwoo Song<sup>1, a</sup>

<sup>a</sup>Department of Statistics, Ewha Womans University, Korea

---

## Abstract

Deep learning methods have been developed, used in various fields, and they have shown outstanding performances in many cases. Many studies predicted a daily stock return, a classic example of time-series data, using deep learning methods. We also tried to apply deep learning methods to Korea's stock market data. We used Korea's stock market index (KOSPI) and several individual stocks to forecast daily returns and directions. We compared several deep learning models with other machine learning methods, including random forest and XGBoost. In regression, long short term memory (LSTM) and gated recurrent unit (GRU) models are better than other prediction models. For the classification applications, there is no clear winner. However, even the best deep learning models cannot predict significantly better than the simple base model. We believe that it is challenging to predict daily stock return data even if we use the latest deep learning methods.

**Keywords:** deep learning, machine learning, time-series data, 1D-CNN, LSTM

---

## 1. Introduction

Deep learning is one of the most popular machine learning methods. It can be applied to a wide range of data, including images, texts, and audio. Various deep learning methods have been successful in many fields, such as computer vision (Voulodimos *et al.*, 2018) machine translation (Bahdanau *et al.*, 2016), speech recognition (Hannun *et al.*, 2014), image recognition (He *et al.*, 2016), and recommendation systems (Covington *et al.*, 2016). Convolutional neural networks (CNN) is one of the deep learning methods. CNN generally refers to 2D CNN used for image classification (Krizhevsky *et al.*, 2012; Yu *et al.*, 2017; Simard *et al.*, 2013). It shows the best performance in image classification by keeping spatial information of images using the convolution and efficiently processing the image size. 1D CNN is mainly used for time-series data and signal data (Lee *et al.*, 2017). It is useful in identifying patterns in one-dimensional data. Recurrent neural network (RNN) is a neural network specialized in processing sequential data. Long short-term memory (LSTM) is one of the most popular methods in RNN which solve long-term dependency problem in RNN. Mostly, LSTM is used mainly for speech recognition (Graves *et al.*, 2013a), natural language processing (Young *et al.*, 2018), and acoustic modeling (Graves *et al.*, 2013b), and so forth. Gated recurrent unit (GRU) is developed by Cho *et al.* (2014). It is a simplified version of LSTM with fewer parameters. Therefore, the learning time is much shorter than that of the LSTM in most cases.

Recently forecasting time series data using deep learning methods are very popular. Guresen *et al.* (2011) used a new artificial neural network (ANN) model which is DAN2 developed by Ghiassi

---

<sup>1</sup> Corresponding author: Department of Statistics, Ewha Womans University, 52, Ewhayeodae-gil, Seodaemun-gu, Seoul 03760, Republic of Korea. E-mail: josong@ewha.ac.kr

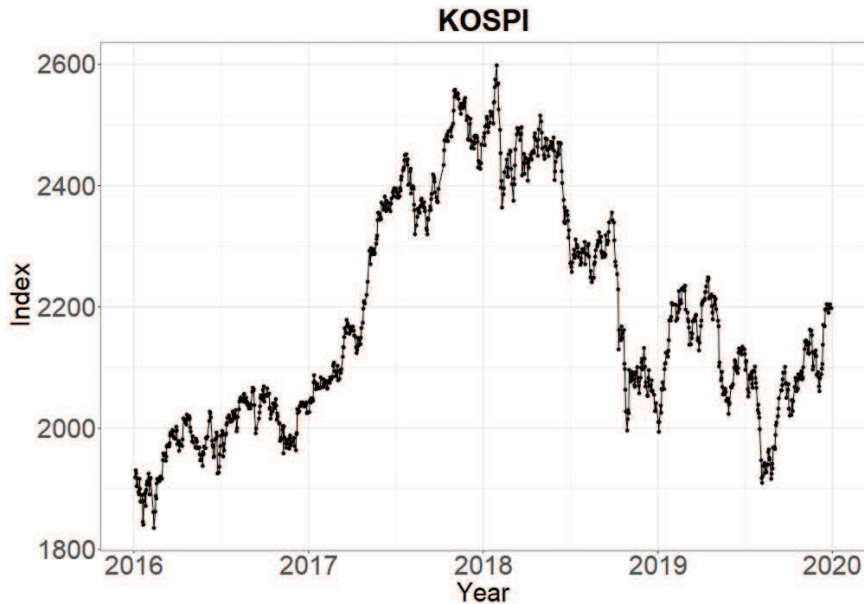


Figure 1: Korea's stock market index (KOSPI).

and Saidane (2005), and hybrid models to forecast the NASDAQ index's daily closing values. In their study, the classical ANN model and multi-layer perceptron (MLP) model which is a class of feedforward ANN perform better than other methods. Lee and Lim (2011) use ANN with fuzzy neural network (FNN) to forecast the direction of change in the daily KOSPI and KRW/USD exchange rate with several inputs, such as commodity channel index (CCI), current price position (CPP), and relative strength index (RSI). Persio and Honchar (2016) proposed a new approach based on the combination of wavelets and CNN to the S&P 500. They also compare the results of MLP, CNN, and LSTM to predict financial time series directions. Chen *et al.* (2015) predict China stock returns using LSTM by dividing the return rate into several classes.

In this study, we try to predict the daily return rate and direction of several stocks and KOSPI with various prediction models. We compare various data structures and models, such as deep learning models and machine learning models. It is known that forecasting daily return is a challenging task. However, several precedent studies show promising results. We like to check that it is possible to forecast Korean stock market data using deep learning or machine learning methods. We use 1D-CNN, LSTM, GRU, random forest (Breiman, 2001), and XGBoost (Chen and Guestrin, 2016) to make predictions. In Section 2, the dataset we used in this study is described in detail. Section 3 introduces the data structure, data splitting, and models. Section 4 shows the results of each model. Section 5 provides conclusion remarks.

## 2. Data

We use KOSPI data and individual stock data from January 4th 2016 to December 30th 2019.

We can see KOSPI data in Figure 1. KOSPI was on an upward trend from 2016 to the beginning of 2018, but has since been on a downward trend. However, there is no apparent seasonality or cyclic behavior over this period. The dataset in this study includes daily return rate and direction of KOSPI.

We use a daily return instead of index itself and the  $i^{\text{th}}$  daily return is defined as follows,

$$r_i = \frac{y_{i+1} - y_i}{y_i},$$

where  $y_i$  is  $i^{\text{th}}$  day KOSPI or stock price. We also consider the up and down direction of KOSPI (or stock) in our study, and the  $i^{\text{th}}$  day direction is calculated as follows,

$$ud_i = \begin{cases} 1, & r_i > 0, \\ 0, & r_i \leq 0, \end{cases}$$

where  $r_i$  is the  $i^{\text{th}}$  day return rate. When the value of  $ud_i$  is 1, it means that  $i^{\text{th}}$  day's stock price has risen compared to yesterday's. And when the value of  $ud_i$  is 0, it means that  $i^{\text{th}}$  day's stock price fallen compared to yesterday's.

We also consider some individual stock data in our study. Considering various industries, we try to select representative companies in each industry. We chose Samsung Electronics and Hyundai Motors in the KOSPI market. The reason for choosing Samsung Electronics is that it is the most representative South Korean company with the highest market capitalization. Hyundai Motors is the motor industry's leading position and global corporation. Also, we chose Nexon GT and Leeno Industrial in the KOSDAQ market. Nexon GT is the 10<sup>th</sup> largest market capitalization company among game companies and Leeno Industrial is semiconductor-related.

### 3. Method

#### 3.1. Data structure

We use a sliding window to predict a daily return rate using previous return data. This method is well known in the time-series study. Reflecting this method, we transform the data into the form shown in Figure 2. We transform the data to predict the next day's return with the last five-day returns. The number of features is equal to the training window size. For example, Figure 2 shows the data structure when the number of features is 5. The goal is to predict the next day's return with the last  $d$  day returns. In other words,  $d$  means the number of features. We compare the results when  $d = 5, 10, \text{ and } 30$  days in our study to find the best models. We also like to predict the next day's up/down directions. The data structure is equal to the one in the regression model, but only the response changes.

#### 3.2. Training and test sets

We consider walk forward optimization (Kirkpatrick and Dahlquist, 2010) to divide data into train and test set considering the temporal order. This method updates the model for each dataset as it finds the optimal parameters. After transforming the data into the form described in the previous section, we split the training data and the test data. Figure 3 shows that the shaded part represents the training set and the white part represents the test set. The date on it indicates the year and month of the  $Y$  variable. We use one year as a training set and a month as a test set. When we use test data as the next month of training data, there is a problem with some overlap between training and test data. Therefore, we set a month gap between the training set and the test set to solve this problem. We create a total of 36 data sets in our study.

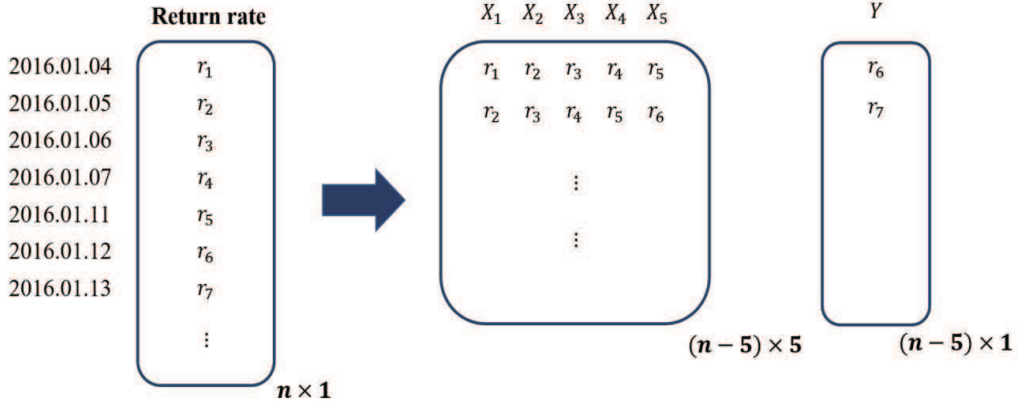


Figure 2: Data structure for regression.

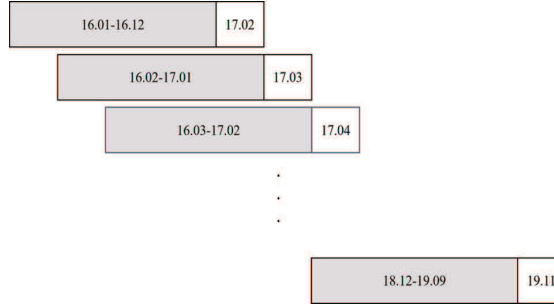


Figure 3: Walk forward optimization.

### 3.3. Models

As we mentioned in the introduction, we consider several prediction models including decision tree-based methods and deep learning methods. We decide to use random forest (RF) and extreme gradient boosting (XGBoost) for the machine learning and 1D-CNN, LSTM, GRU, and combined 1D-CNN and LSTM for deep learning methods. We use two LSTM models with the option **'return\_sequences'**. The default of **'return\_sequences'** is set to False, which returns the last output of the output sequence. On the other hand, if **'return\_sequences'** is set to true, the model returns the full sequence as the output. LSTM 1 model sets it to false and LSTM 2 model sets it to True. We also propose a deep learning model on the combination of a CNN and LSTM to predict the stock return rate on the next day. In this model, CNN layer is used for feature extraction on the input data and LSTM is used for prediction using the extracted feature.

The simple mean and weighted mean of return over the past  $d$  days are set as the baseline models. The simple mean of  $i^{\text{th}}$  daily return is defined as follows,

$$\widehat{r}_i^m = \sum_{k=1}^d \frac{r_{i-k}}{d}.$$

Table 1: RMSE of daily return rate for KOSPI

Methods	$d$		
	5	10	30
Mean	0.0084	0.0081	0.0079
W.mean	0.0092	0.0092	0.0092
1D-CNN	0.0111	0.0113	0.0153
LSTM 1	0.0074	0.0075	0.0075
LSTM 2	0.0074	0.0075	<b>0.0074</b>
1D-CNN + LSTM	0.0089	0.0086	0.0086
GRU	0.0075	0.0075	0.0075
RF	0.0079	0.0078	0.0076
XGB	0.0085	0.0085	0.0084

The weighted mean of  $i^{th}$  daily return is defined as follows,

$$\widehat{r}_i^{w.m} = \sum_{k=1}^d w_k r_{i-d+k},$$

where  $w_k = \exp(k) / \sum_{k=1}^d \exp(k)$ ,  $k = 1, \dots, d$  and  $d$  is the number of features.

For the up/down baseline models, we use the following predicted values. The predicted direction by a simple mean of  $i^{th}$  daily return is defined as follows,

$$\widehat{ud}_i^m = \begin{cases} 1, & \widehat{r}_i^m > 0, \\ 0, & \widehat{r}_i^m \leq 0. \end{cases}$$

The predicted direction by the weighted mean of  $i^{th}$  daily return is defined as follows,

$$\widehat{ud}_i^{w.m} = \begin{cases} 1, & \widehat{r}_i^{w.m} > 0, \\ 0, & \widehat{r}_i^{w.m} \leq 0. \end{cases}$$

## 4. Result

### 4.1. Regression models for daily return

To find the best model for daily return prediction, we compare the average of 36 test RMSE for each model with different feature numbers. We use the option ‘**validation\_data**’ to split the training and validation dataset. Before using this option, we randomly choose 20% as validation data from each training dataset to fix the validation dataset as a list. We find the best model with the minimum loss or maximum accuracy on the validation dataset by updating the model using the training dataset.

Table 1 shows that the LSTM 2 model shows the best performance in KOSPI with the past 30 days return as a feature set. However, the performance differences among the models are minimal. If we compare the best model to the baseline model, the performance difference is about 6%~20%. Figure 4 shows the KOSPI return rates and the predicted return rates for the baseline model (left) and the best model (right). The blue line is the observed values, and the red line is the fitted values. There is little visual difference between the two cases, and the return rate is predicted to be closer to zero in most cases. The standard deviation of the predicted value is about  $1.19 \times 10^{-3}$ , which is about seven times smaller than the standard deviation of true data. This prediction happens to almost all models in our analysis. Therefore we can see that it is very challenging to predict the return rate.

Tables 2–5 summarize the results of our models for the individual stock data. The LSTM 1 model shows the best model for Samsung electronics. LSTM 2 model is the best model for Hyundai Motors.

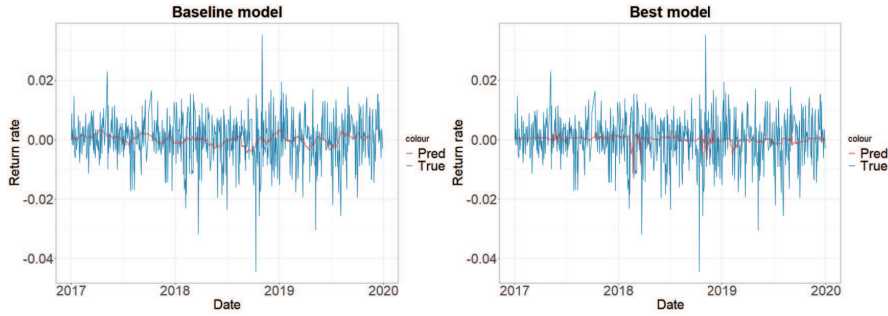


Figure 4: Base line VS best model.

Table 2: RMSE of daily return rate for Samsung Electronics

Methods	$d$		
	5	10	30
Mean	0.0172	0.0164	0.0158
W.mean	0.0186	0.0186	0.0186
1D-CNN	0.0159	0.0171	0.0205
LSTM 1	0.0154	<b>0.0154</b>	0.0154
LSTM 2	0.0155	0.0155	0.0154
1D-CNN + LSTM	0.0157	0.0160	0.0163
GRU	0.0155	0.0155	0.0154
RF	0.0163	0.0163	0.0163
XGB	0.0181	0.0181	0.0178

Table 3: RMSE of daily return rate for Hyundai Motors

Methods	$d$		
	5	10	30
Mean	0.0193	0.0186	0.0180
W.mean	0.0186	0.0216	0.0216
1D-CNN	0.0176	0.0212	0.0226
LSTM 1	0.0174	0.0173	0.0174
LSTM 2	<b>0.0173</b>	0.0173	0.0174
1D-CNN + LSTM	0.0177	0.0179	0.0177
GRU	0.0174	0.0173	0.0175
RF	0.0182	0.0181	0.0179
XGB	0.0210	0.0210	0.0198

For Nexon GT and Leeno Industrial, the GRU model shows the best model. The number of features which is the past  $d$  days in our model is a tuning parameter. The model with the past ten days shows that the best results for Samsung Electronics stock price and Nexon GT stock price, while the past five days show that the best results for Hyundai Motors and Leeno Industrial. However, the best models do not improve the performance significantly compared to the baseline models. We can see that some models performed worse than the baseline models in some cases.

#### 4.2. Classification for daily return direction

We compare the classification models by the average of 36 test set accuracies for each model with different feature numbers. We randomly choose 20% of observations for each training set as validation data to find the best models as we did in the previous section.

Table 4: RMSE of daily return rate for Nexon GT

Methods	<i>d</i>		
	5	10	30
Mean	0.0493	0.0462	0.0454
W.mean	0.0528	0.0526	0.0526
1D-CNN	0.0366	0.0363	0.0415
LSTM 1	0.0362	0.0357	0.0368
LSTM 2	0.0367	0.0365	0.0373
1D-CNN + LSTM	0.0362	0.0362	0.0373
GRU	0.0359	<b>0.0354</b>	0.0364
RF	0.0382	0.0378	0.0372
XGB	0.0451	0.0437	0.0483

Table 5: RMSE of daily return rate for Leeno Industrial

Methods	<i>d</i>		
	5	10	30
Mean	0.0221	0.0209	0.0205
W.mean	0.0254	0.0253	0.0253
1D-CNN	0.0207	0.0233	0.0272
LSTM 1	0.0191	0.0190	0.0191
LSTM 2	0.0191	0.0191	0.0190
1D-CNN + LSTM	0.0199	0.0192	0.0195
GRU	<b>0.0189</b>	0.0190	0.0195
RF	0.0198	0.0198	0.0196
XGB	0.0223	0.0220	0.0215

Table 6: Prediction accuracy of daily return direction for KOSPI

Methods	<i>d</i>		
	5	10	30
Mean	0.482	0.503	0.494
W.mean	0.487	0.486	0.486
1D-CNN	0.537	0.544	0.537
LSTM 1	0.528	0.513	0.525
LSTM 2	0.534	0.511	0.524
1D-CNN + LSTM	<b>0.551</b>	0.541	0.539
GRU	0.529	0.515	0.524
RF	0.540	0.539	0.528
XGB	0.506	0.516	0.532

Table 7: Prediction accuracy of daily return direction for Samsung Electronics

Methods	<i>d</i>		
	5	10	30
Mean	0.484	0.501	0.501
W.mean	0.531	0.529	0.529
1D-CNN	0.537	<b>0.544</b>	0.537
LSTM 1	0.528	0.513	0.525
LSTM 2	0.518	0.510	0.506
1D-CNN + LSTM	0.475	0.509	0.534
GRU	0.515	0.502	0.501
RF	0.464	0.523	0.484
XGB	0.480	0.526	0.501

Table 8: Prediction accuracy of daily return direction for Hyundai Motors

Methods	<i>d</i>		
	5	10	30
Mean	0.482	0.472	0.486
W.mean	0.473	0.472	0.529
1D-CNN	0.519	0.524	0.537
LSTM 1	0.563	0.567	0.525
LSTM 2	0.569	<b>0.574</b>	0.564
1D-CNN + LSTM	0.543	0.509	0.519
GRU	0.563	0.502	0.557
RF	0.522	0.523	0.554
XGB	0.529	0.526	0.519

Table 9: Prediction accuracy of daily return direction for Nexon GT

Methods	<i>d</i>		
	5	10	30
Mean	0.471	0.499	0.490
W.mean	0.492	0.492	0.492
1D-CNN	0.517	0.556	0.501
LSTM 1	0.549	0.549	0.548
LSTM 2	0.542	0.555	0.544
1D-CNN + LSTM	0.527	<b>0.595</b>	0.539
GRU	0.555	0.562	0.554
RF	0.514	0.518	0.518
XGB	0.547	0.500	0.497

Table 10: Prediction accuracy of daily return direction for Leeno Industrial

Methods	<i>d</i>		
	5	10	30
Mean	0.486	0.506	0.476
W.mean	0.487	0.487	0.487
1D-CNN	0.514	0.501	0.525
LSTM 1	0.536	0.499	0.508
LSTM 2	0.515	0.492	0.510
1D-CNN + LSTM	0.536	0.494	0.537
GRU	0.503	0.491	0.518
RF	0.531	<b>0.549</b>	0.486
XGB	0.524	0.511	0.506

Table 6 shows that the 1D-CNN with LSTM model gives the best performance with the past five days return as a feature set from KOSPI data. If we compare the best model to the baseline model, the performance difference is about 5%~7%. However, it does not increase the accuracy that much. Tables 7–10 show the average of the test set accuracies for the individual stock data. Most of the baseline models show 50% accuracy. The 1D-CNN model is the best model for Samsung Electronics. The LSTM 2 model for Hyundai Motors is the most accurate. For Nexon GT, the 1D-CNN with LSTM model shows the best model. The RF model is the best prediction model for Leeno Industrial. For all individual companies, the models with the past ten days show the best accuracy. Among all individual companies, the model for Nexon GT achieves the best result which is the average of test set accuracies 0.595 and accuracy for whole test data is 0.592. This result show improvement in accuracy of about 10% over the baseline model. The binomial test is used to compare the accuracy ( $\pi$ ) of the best performing methods with the random guess. We use the following null and alternative



hypotheses,

$$\begin{aligned} H_0 : \pi &= 0.5, \\ H_1 : \pi &\neq 0.5. \end{aligned}$$

This test is performed by determining whether the confidence interval for  $\pi$  contains 0.5. The sample size of the test set is 733 and the best accuracy for Nexon GT is 0.592. For large samples, we can approximate the binomial using the normal distribution by the central limit theorem. We calculate a 95% confidence interval on the parameter  $\pi$ .

$$\begin{aligned} p \pm z_{\frac{1-0.05}{2}} \sqrt{\frac{p(1-p)}{n}} &= 0.592 \pm 1.96 \sqrt{\frac{0.592(1-0.592)}{733}}, \\ &= (0.557, 0.628). \end{aligned}$$

Since it does not contain 0.5, the null hypothesis is rejected at a significance level of 95%. Therefore, we conclude that the best performing method is significantly different from the random guess.

As we can see from the previous section for the regression models, some models performed worse than the baseline models in some cases. We also consider predicted direction using the predicted values of regression models. If the predicted value is greater than 0, the predicted direction is defined up, otherwise it is defined down. The best accuracy for KOSPI comes out to 0.547. Despite these different attempts to predict the classification, the performance is not good in other datasets.

## 5. Conclusion remarks

In this paper, we study the performance of deep learning and machine learning methods to predict the daily return rate and direction. Comparing various prediction methods, we can see that deep learning methods show better performance than machine learning methods in forecasting market values. However, the difference between deep learning and machine learning methods are insignificant. For example, the RMSE for best deep learning for KOSPI return rate is 0.0074, and that of the best machine learning method is 0.0076. The prediction accuracy for the best deep learning method for KOSPI is 0.551, and that of the best machine learning method is 0.540. In regression, results show that the LSTM and GRU achieve better forecasting performance than all other models. There is no clear winner for the classification cases.

However, it is not significantly improved over the baseline model. Although we do not include other prediction models' results, we considered different training and test duration setup. we also used the signs of the predicted values of the regression models in our study. However, the performance is worse than that of the classification model. Deep learning methods show good performances in many fields especially for image and text data but it is very hard to find a deep learning model with good performance for the stock data in terms of prediction accuracy. Gu *et al.* (2020) shows that even the prediction model with  $R^2$  about 2% can make significant change in financial market. Our next study will be the construction of the optimal model in financial market in terms of profit.

## References

- Bahdanau D, Cho K, and Bengio Y (2016). *Neural Machine Translation by Jointly Learning to Align and Translate*, arXiv:1409.0473.
- Breiman L (2001). *Machine Learning*, **45**, 5–32.

- Chen K, Zhou Y, and Dai F (2015). A LSTM-based method for stock returns prediction: a case study of China stock market. *2015 IEEE International Conference on Big Data (Big Data)*, 2823–2824.
- Cho K, Merriënboer BV, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, and Bengio Y (2014). *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*, arXiv:1406.1078.
- Covington P, Adams J, and Sargin E (2016). Deep neural networks for YouTube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, 191–198.
- Ghiassi M and Saidane H (2005). A dynamic architecture for artificial neural networks. *Neurocomputing*, **63**, 397–413.
- Graves A, Jaitly N, and Mohamed AR (2013). Hybrid speech recognition with deep bidirectional lstm. *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, 273–278.
- Graves A, Mohamed AR, and Hinton G (2013). Speech recognition with deep recurrent neural networks. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 6645–6649.
- Gu S, Kelly B, and Xiu D (2020). Empirical asset pricing via machine learning, *2020 The review of Financial Studies*, **33**, 2223–2273.
- Guresen E, Kayakutlu G, and Daim TU (2011). Using artificial neural network models in stock market index prediction. *Expert Systems with Applications*, **38**, 10389–10397.
- Hannun A *et al.* (2014). *Deep Speech: Scaling Up End-to-End Speech Recognition*, arXiv:1412.5567.
- He K, Zhang X, Ren S, and Sun J (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.
- Kirkpatrick C and Dahlquist J (2010). *Technical Analysis: The Complete Resource for Financial Market Technicians*, FT press.
- Krizhevsky A, Sutskever I, and Hinton GE (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, **25**.
- Lee SH and Lim JS (2011). Forecasting kospi based on a neural network with weighted fuzzy membership functions, *Expert Systems with Applications*, **38**, 4259–4263.
- Lee SM, Yoon SM, and Cho H (2017). Human activity recognition from accelerometer data using CNN. *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 131–134.
- Persio LD and Honchar O (2016). Artificial neural networks architectures for stock price prediction: comparisons and applications. *International Journal of Circuits, Systems and Signal Processing*, **10**, 403–413.
- Simard PY, Steinkraus D, and Platt JC (2013). Best practices for convolutional neural networks applied to visual document analysis. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition*, **2**, 958–963.
- Voulodimos A, Doulamis N, Doulamis A, and Protopapadakis E (2018). Deep learning for computer vision: a brief review. *Computational Intelligence and Neuroscience*, **2018**, 1–13.
- Young T, Hazarika D, Poria S, and Cambria E (2018). Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine*, **13**, 55–75.
- Yu S, Jia S, and Xu C (2017). Convolutional neural networks for hyperspectral image classification, *Neurocomputing*, **219**, 88–98.