Financial Series Prediction Using Attention LSTM

Minsung Han

DSP Presentation

minsung.han@gmail.com

September 20, 2019

Introduction

About the paper

- Title: Financial Series Prediction Using Attention LSTM
- Authors: Sangyeon Kim (Graduate student, SNU) , Myungjoo Kang (Professor in Applied Math, SNU)

Structure of the paper

- Introduction
- Related Works
- Data Set
- Methodologies: MLP / 1D-CNN / LSTM / Attention Networks / Weighted Attention Networks
- Experimental Results : Best Lookback Days / Results of Various Deep Learning Models / Visualization Attention Vectors
- Conclusion

Introduction

- Predicting the trends of financial markets is one of the most important tasks for investors.
- Stock Market Data are collections of time series.
- Deep learning has been widely used in classification problems when non-linearity exists
- What about applying deep learning technique to stock market data?
- Little bit of background...
  - Technical analysis is a traditional method that uses historical stock prices and trading volumes to determine the trends of future stock movements. (Chart-ist)
  - Fundamental analysis predicts stock prices by using intrinsic values. (Warren Buffett)

Related Works

- Kohzadi[1] : used ANN and ARIMA models for forecasting commodity prices and compared the results.
- Kara[2] : applied ANNs and SVMs to predict the Istanbul Stock Exchange (ISE) National 100 Index prices.
- Qian et al. [6] : compared various models, ARIMA, SVMs, DAE (Denoising Autoencoder), and mixture of the above.
- Pyo et al. [7] : predicted the KOSPI 200 index with SVMs and ANNs using Google Trends.
- Bao et al. [13] : proposed a deep learning framework for financial time series using stacked autoencoders and long-short term memory.

Data Set

Daily Return at t+1:

$$r[t+1] = \frac{closeprice[t+1] - closeprice[t]}{closeprice[t]} \tag{1}$$

Input: daily return with other index (the currency, commodities, and global indexes) for the look back period p. x[t] = [Rt[0]; ... ;Rt[p]].

Trend at t for q:

$$trend[t,q] = \frac{closeprice[t+q] - closeprice[t]}{closeprice[t]} \tag{2}$$

Target label: if trend[t] < 0

$$y[t] = [1,0] \tag{3}$$

Otherwise

$$y[t] = [0,1] \tag{4}$$

Methodologies

MLP

- Four layers. Each hidden layer contained 64 hidden neurons.
- flattened the input matrix to create an input vector, which was used to feed our model.

1D-CNN

- we recommend using a model with two convolutional layers and three fully connected layers.
- A 1D CNN is expected to capture data locality well when a kernel slides across the input data.

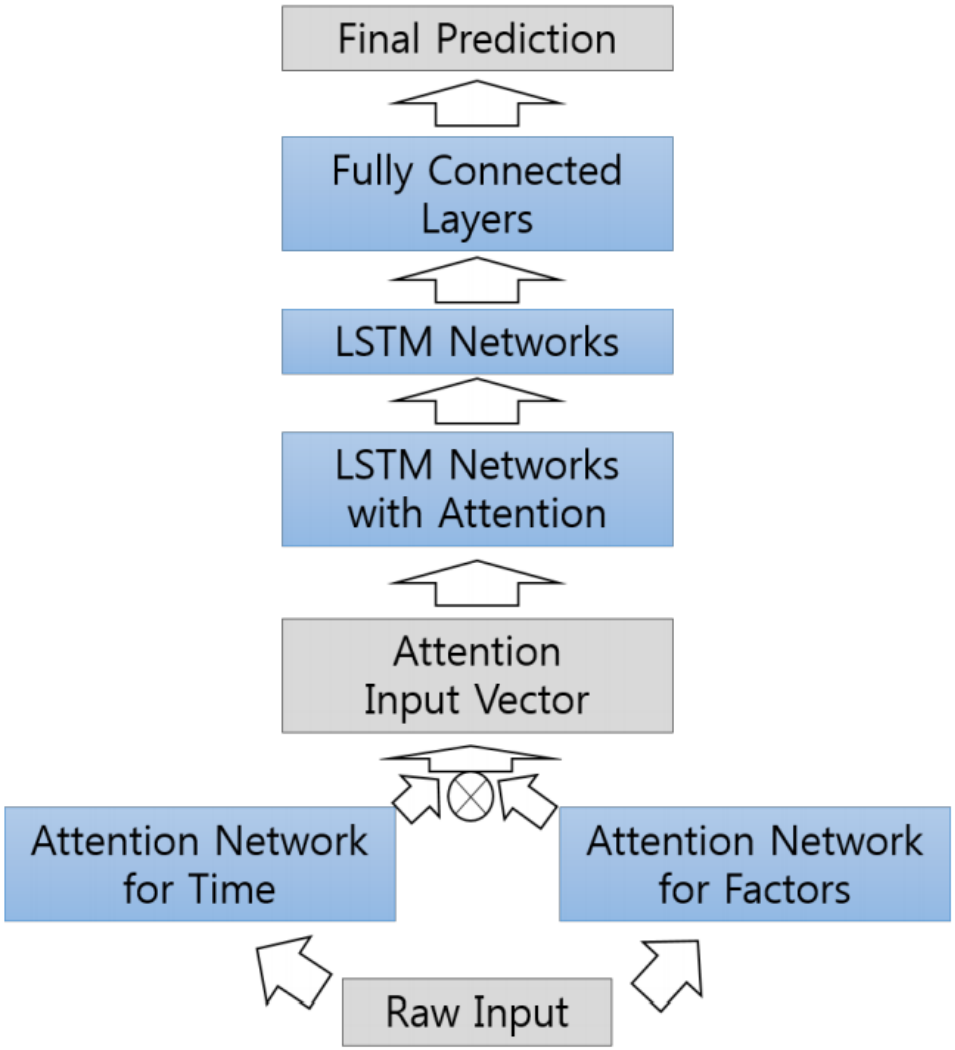
LSTM

- ?

Attention Networks

- ?

Network Sketch for (Weighted) Attention Networks



Weighted Attention Network

Original loss

$$H_{y'}(y) := -\sum_i (y'_i \log(y_i) + (1 - y'_i) \log(1 - y_i))$$

New loss; considering magnitudes for the weighted attention

$$H_{y'}^{weighted}(y) := -abs(change\_ratio_i) * \sum_i (y'_i \log(y_i) + (1 - y'_i) \log(1 - y_i))$$

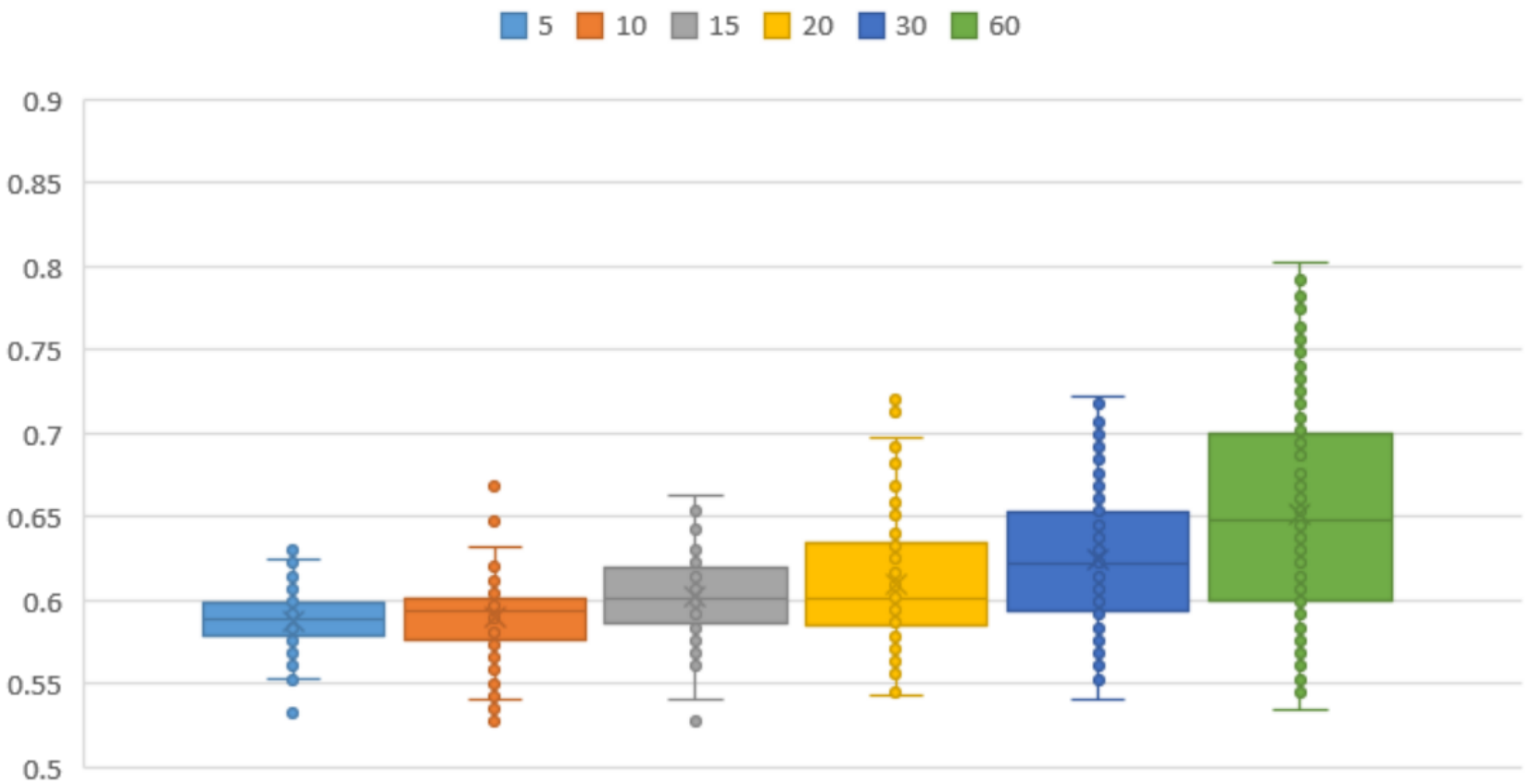
Performance Measure

$$hit\ ratio = \frac{\sum_{i=1}^{N} prediction_i}{N},$$

$$prediction_i = \begin{cases} 1 & if\ prediction_i \cdot real_i > 0 \\ 0 & otherwise \end{cases}$$

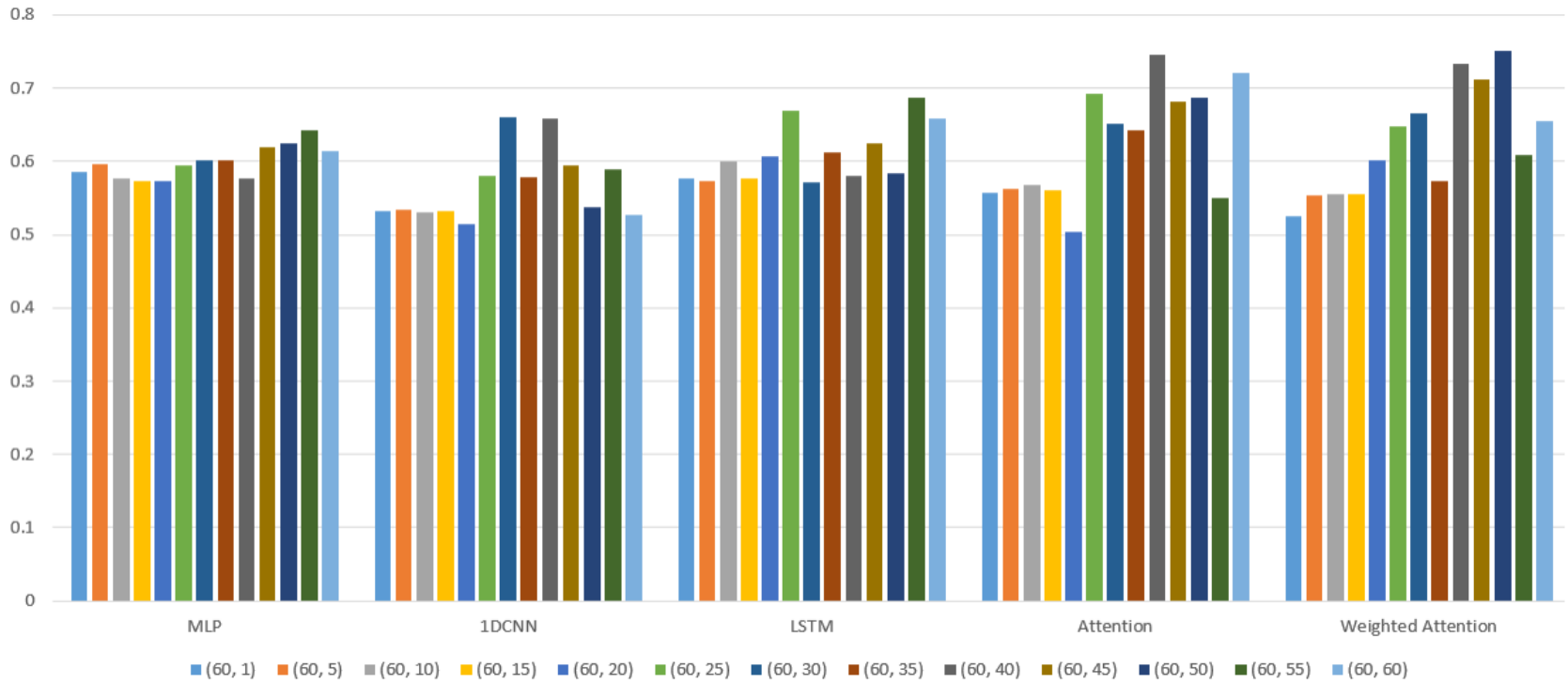
Best lookback days

Hit ratios for various deep learning models for each lookback day



Performance of Deep Learning Models

Fix lookback day=6. Hit ratios for various deep learning models.

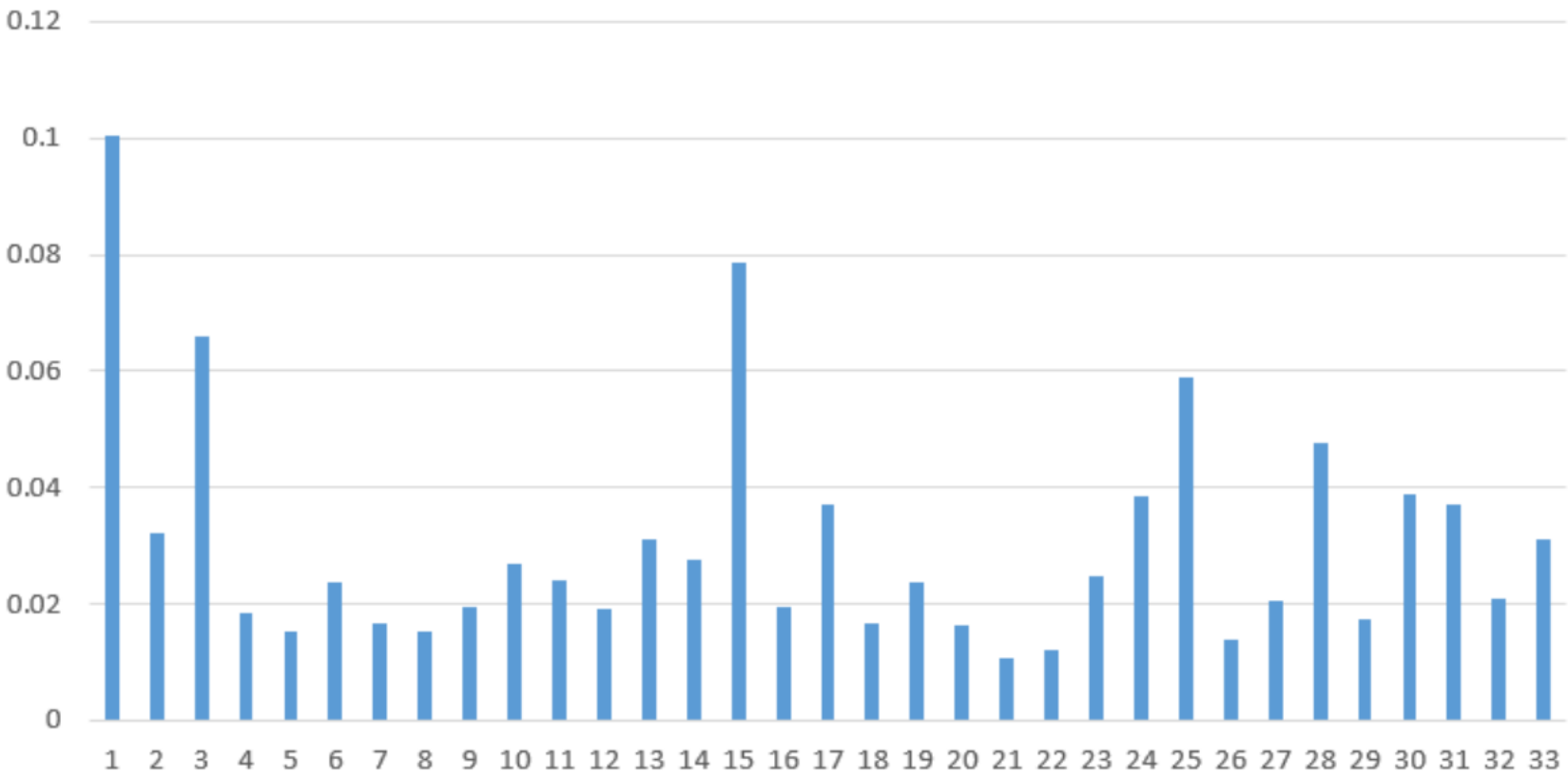


The Best Model

- Weighted Attention Network (modified loss function)
- Lookback day=60 (12 weeks)
- 40 predicting days

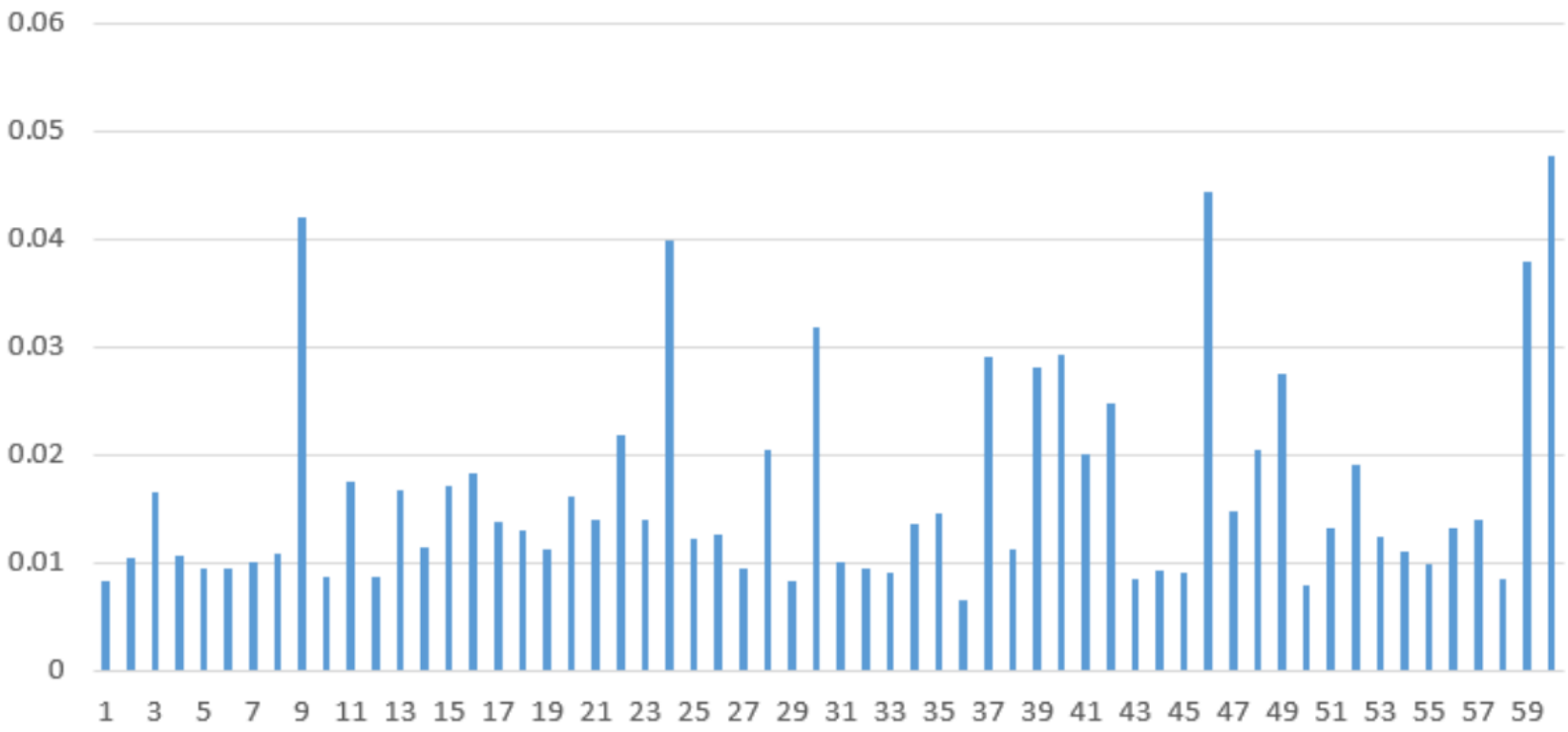
	Dataset	Attention Networks	Weighted Attention Networks
Positive	0.602	0.709	0.825
Negative	0.398	0.723	0.671
Total	1.0	0.715	0.763
Earn points	1525.20	1090.718	1257.462

Attention Vector of Factors



- The dollar currency index / SP 500 global index

Attention Vector of Days



- t-1 / t-15

Conclusion

- Tested various deep learning models for predicting the trends of the KOSPI 200 index.
- Found that lookback days with 60 trading days returned higher hit ratios with various models.
- When 60 trading days considered as lookback days, using 40 trading days as prediction days returned the highest hit ratio with the attention networks model.
- The highest earn points were returned when we used weighted attention networks, as loss functions were minimized when improved at higher change ratios.

Thank you