[Link to Research Paper]
Luca Di Persio and Oleksander Honchar in “Artificial Neural Networks Approach to the Forecast of Stock Market Price Movements” have suggested an approach to predict price movement polarity for a given trading day using prior 30 days of close data.

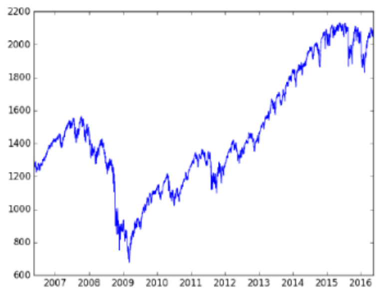

Figure 1: S&P500 index data from 2006 to 2016.

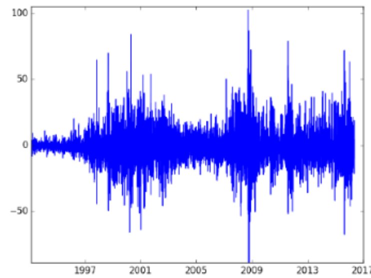

Figure 3: S&P500 index daily returns data.

Figure 1 and 3 above are taken from their paper and outlines their data processing methodology: they take price action data(daily closes) from 1950 to 2016 for the S&P 500 index, representing over 16,000 data points. They then normalize this data to have zero-mean and unit variance.

Then they feed the input data in 4 different architectures; 2 of which I replicate in the accompanying Google Colab environment. The MLP and CNN architectures from the paper are:

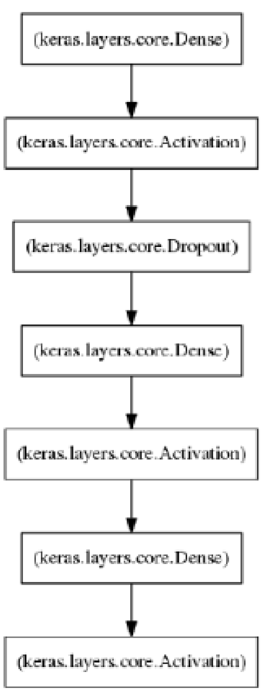

Figure 4: MLP architecture.

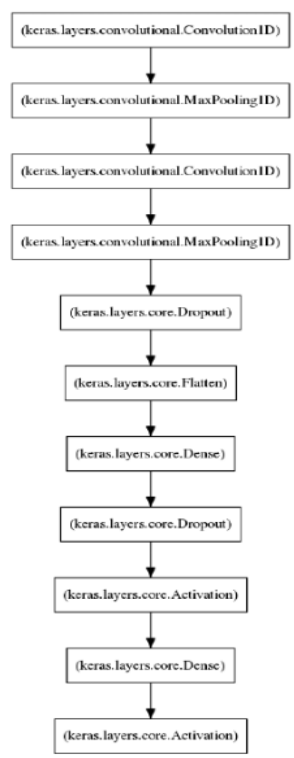

Figure 5: CNN architecture.

The MLP architecture has 2 hidden layers with 250 and 100 neurons, respectively. They defined the activation function as ReLu and included a drop out layer after the initial hidden layer. The CNN architecture includes 2 hidden convolutional layers, with 2 hidden layers for the MLP layer and drop out layers following the 1st of each type of hidden layer. Other parameters for the CNN are:

- Number of filters = 64;
- Pool length = 2;
- Subsample length = 1;
- Activation function ReLU.

I replicate these models given the limited information on all details and obtained the following performance:

Model	Test Loss	Test Accuracy
MLP	0.7082	0.5238
CNN	0.6919	0.5255

Compared to the paper's performance:

Model	Test Loss	Test Accuracy
MLP	0.58	0.6
CNN	0.56	0.59

Discrepancies between my replication and original model may have been caused by mismatched hyper parameters including drop out settings, batch size, learning rate, and optimizer selection, which were undisclosed.

See accompanying Colab notebook for detailed classification report with precision and recall metrics.

The methodologies outlined by their paper show promising results. I want to build on top of their solution to achieve potentially higher performance by using a narrower window of input data, representing 3-5 days of prior price movement, coupled with a more intricate feature engineering strategy to group labels as not only 1 day price movements, but as compound movements that span multiple days. Ultimately, the model I use will take as input 3-5 day price action (some combination of OHLCV) and predict the price movement for any given window following the input price pattern.