# Introduction

Predicting Financial Instrument prices is as much difficult as it is rewarding. There is huge data generated in Capital Markets and leveraging these predictions can be made. Analyzing their behaviour and extracting useful insights can help traders. Traditional ML models have shown considerable performance in this task with SVMs, random forest ensembles, leading the scope. But with the improvement of our computational abilities deep learning have started to take the forefront with MLPs, ANNs, RNNs, AEs, LSTMs and RBMs. CNNs are highly useful in auto-extracting features which is in itself a difficult task.

This paper “***CNNpred: CNN-based stock market prediction using a diverse set of variables****”* by ***Ehsan Hoseinzade***and***Saman Haratizadeh***published *in Faculty of New Sciences and Technologies, University of Tehran, Tehran, Iran* on Mar 20, 2019 tries to predict next day’s stock prices using market data from the past as well as different markets as well using CNNs.

Technical indicators including Moving Averages, OHLC, exchange rates, correlated commodity prices and macroeconomic factors are some of the numerous possible variables to predict stock prices. To correctly choose the right slew of features to build the model is difficult, to say the least, even for a financial expert. Thus, using CNNs seem a powerful alternative to automatic feature selection. Since CNNs are ideally used for computer vision the convolutional layers available generally are not optimal for this use case. Although based on previous work, this experiment develops a new CNN specifically to predict financial stock indices as S&P500, NASDAQ, NYSE, DJI and RUSSELL. This experiment is unique as it uses a wide range of financial variables and creates a 3d CNN to predict markets.

The author analyzed and compared previous works in this field,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Author/year | Target data | Variables set | Feature extraction | Prediction method |
| Kara et al., 2011 | Borsa Istanbul BIST 100 Index | technical indicator | ANN | ANN SVM |
| Patel et al., 2015 | 4 Indian stocks & indices | technical indicator | ANN | ANN-SVM RF-NB |
| Qiu et al., 2016 | Nikkei 225 index | Financial indicator macroeconomic data | ANN | GA + ANN SA+ANN |
| Qiu & Song, 2016 | Nikkei 225 index | technical indicator | ANN | GA + ANN |
| Nelson et al., 2017 | Brazil Bovespa 5 stocks | technical indicator | LSTM | LSTM |
| Di Persio & Honchar, 2016 | S&P 500 index | price data | MLP-RNN-CNN wavelet + CNN | MLP RNN CNN |
| Moghaddam et al., 2016 | NASDAQ | price data | ANN-DNN | ANN-DNN |
| Arévalo et al., 2016 | AAPL Inc. | 3 extracted features | DNN | DNN |
| Zhong & Enke, 2017 | S&P 500 index | various variables | PCA | ANN |
| Chong et al., 2017 | Korea KOSPI 38 stock returns | price data | PCA-RBM AE | DNN |
| Gunduz et al., 2017 | Borsa Istanbul BIST 100 stocks | technical indicator temporal variable | Clustering+CNN | CNN |
| This paper | U.S. 5 major indices | various variables | 3D representation of data + CNN | CNN |

# Anatomy of CNN

CNNs typically have several layers which can be categorized into input layer, convolutional layer, pooling layer, fully connected layer, and output layer. The convolutional layer acts as a filter identifying recurring patterns in the data. These filters get attributed their own weights and fed to an activation function. Next the pooling layer subsamples the data preventing overfitting and drastically reducing training time. Finally, there is a fully-connected/MLP layer that is responsible to for generating the outputs. Also, in this experiment random dropouts (of neurons) were configured to prevent overfitting.

CNN has many hyperparameters including number of layers, filters in each layer, dropout rate, size and shape of filters, initial representation of input data. Traditional computer vision 3\*3 or 5\*5 filters are not suitable for financial data, so new filters were constructed using interpretations of variables. A new architecture CNNpred, was introduced in this paper for stock prediction. Depending upon representation of input data, it is of 2 types 2d CNN and 3d CNN

The 2d CNNpred is based on the philosophy that given a market’s history the future trends can be predicted and the architecture remains same for different markets. Thus, it is trained with data from several markets and future contracts, commodities prices, forex rates etc. The input data is fed as a 2-dimensional tensor. There are 2 convolutional layers with 3\*1 filters inspired from popular candlestick patterns, pooling layers of 2\*1, and sigmoid activation function as the output can be interpreted as probability.

The 3d CNNpred is based on the philosophy that different models are required for different markets even though they may use data, variables from other markets as well. The input data is fed as 3-dimensional tensor. Filters of 1\*1 and 3\*1 shape that can extract high level features through the depth of input tensor. The 3rd dimension in this case over 2d CNN would be the different markets. The fully connected layer takes 104 features and produces the final output similar to 2d CNNpred.

The feature extraction filters work mainly on two levels- Daily and durational (mid-long term). The features extracted from 82 variables corresponding to each stock indices are then flattened to 1-dimensional vector and fed to the NN. The model is trained on a moving history of past 60 days.

The 82 variables are principally primitive variables, technical indicators, world stock market indices, the exchange rate of U.S. dollar to the other currencies, commodities, data from big companies of the U.S. markets, future contracts etc. The data has been gathered within Jan 2010 and Nov 2017. Training data is initial 60%, validation data is 20% and test data is last 20%. The data has been gaussian normalized. Evaluation metric is Macro-Averaged-F-Measure. All activation function except the last layer is RELU

# Results and Discussion

Compared to following baseline algorithms:

1. PCA + ANN (Zhong & Enke, 2017), PCA as dimension reduction and ANN as classifier
2. Technical (Kara et al., 2011), Technical indicators and ANN as classifier
3. CNN-cor (Gunduz et al., 2017), A CNN with mentioned structure in the paper

The 2d CNNpred and 3d CNNpred performed statistically significantly better, in terms of Macro-Averaged-F-Measure. Even in trading simulations with up predictions as long positions and down predictions as short positions (0.1% brokerage fee) the strategy yields consistently about 30% profits. This is further proven by good Sharpe ratio and CEQ return. This shows promise as possible trading algorithm after some refinement in features.