Comparison of Stock and Stock Price Index Prediction Efficiency among FNN, CNN and RNN Techniques

**ABSTRACT** (summarize after all is done)

1. **Introduction**

Stock and stock price prediction has always been a difficult issue due to the fluctuation of markets. To analyze potential future values, two types of method are generally accepted by investors. One is fundamental analysis by which investors measures the intrinsic value of the stock by examining the related macro economical, industrial and corporate financial factors. The other method is technical analysis, which relies on statistical trends to predict the future movement of stock price. Technical analysis is based on the efficient market hypothesis and the assumption that history repeats itself. Invented by Malkiel and Fama in 1970, the efficient market hypothesis states that share prices reflect all information and therefore continuously beating the market is impossible. This has become the logical base of the prediction of technical analysis. Since researchers can hardly gather and evaluate all information in the markets ranging from political affairs to corporate accounting changes due to the information asymmetry, it is more reasonable using the stock price and volume that has already been efficiently measured, reflecting all information in the market. Beyond this, the hypothesis of history repeating itself makes the prediction possible.

Although linear regression model has always been a generally accepted way of technical analysis, the non-linear machine learning techniques is swiftly gaining popularity in the investment community and is assumed to have a better performance due to the applicability to non-stationary time series stock market data. Artificial neural network (ANN) is one of the models that have been most widely used to predict financial data.

This study focused on the comparison of three types of ANN model, feedforward neural network (FNN), convolutional neural networks (CNN) and recurrent neural network (RNN) with 10 parameters introduced to analyze the close price for two stock price indexes that indicate the general market and internet industry performances (HSI and HSIII) and two individual internet technology stocks (Tencent and SMIC). The prediction performance on test dataset will be measured through ADAM and RMSE approach. Both stocks and indices are highly voluminous and vehemently traded in so they reflect China’s internet industry as a whole.

The topic of HK internet industry corporation value study is rather meaningful at the current situation. Although Chinese internet tech corporation has experienced a sharp growth in the past decade, we all know that in the previous month Hongkong stock markets have seen a great decline, among which the internet industry suffers most. The reasons for the shock include but not limited to political condition, global economy and companies’ financial performance etc. At this moment of panic, investors are eager to know whether they should restore faith in the internet industry or hold short position. By utilizing technical analysis based on machine learning, we can give our own answer to this question.

1. **Literature review**

Artificial neural network is a series of algorithms that recognizing the underlying relations of data through a process that mimic human brain operation. It was firstly been used in natural science study then introduced into financial analysis. Kyoung-jae Kim and Ingoo Han (2000) proposed using ANN model to predict stock price index combining with the genetic algorithms (GAs) approach not only to improve the learning algorithms but also reduce feature space, proved to have a better prediction performance. Precious studies have shown that ANN model has a generously higher accuracy in predicting individual stock and stock price index. Patel, J., et al (2014) proposed that in predicting daily movement of Indian stock and stock price, ANN outperformed other machine learning models, namely support vector machine (SVM), random forest and Naive-Bayes on both continuous and trend input approaches. Vijha et al (2020) have also compared the performance of ANN and random forest in forecasting the close price of 4 American stocks, suggesting that ANN is more efficient under the standard of RMSE and MAPE. The comparison among different types of ANN model has also been carried out. Guresen et al (2011) compared the difference in prediction efficiency of classical ANN model MLP outperforms DAN2 and GARCH-MLP.

China’s stock market has received more and more attention since this century with more scholars try to predict the fluctuation of the stocks traded on Shanghai stock exchange and Hongkong stock exchange. What’s more, since China’s financial market faces a more complex global economic and political situation, any attempt to analyze the stock price movement using a linear regression model would face a dataset with high level of noise interruption. Cao et al (2004) has proved ANN has a better performance than Fama-French’s model in predicting stock price movement traded on Shanghai stock exchange. Due to complex regulation in IPO for mainland China financial market, it has become a praxis for Chinese high-tech companies to go public offering in Hongkong first, making the data more valuable for valuable for the research in development of China’s high-tech industry. Tsang et al (2007) proposed that NN5 model is efficient in developing a buying/selling system in Hongkong stock market. However, few of the studies have focused on the prediction of China’s internet industry, the most boosting industry in the past decade and the important driving force for economic development, making today’s research valuable.

1. **Research Data**

The study takes in less than 7 years data for total two indices (HSI and HSIII) and two stocks (Tecent and SMIC) from 17th Aug 2015 to 14th Apr 2022. Since the opening of Hang Seng Internet & Information Technology Index ("HSIII") remarks the recognition for the importance of China’s internet technology corporations in Hongkong market, datasets are selected from the first day this index is invented. All the data is attained from <<https://choice.eastmoney.com/>>. The summary for the close price of datasets is shown in Table1 (taking HSI data set as an example).

**Table 1**

The quantile and descriptive statistics summary of total data set of HSI.

|  |  |  |  |
| --- | --- | --- | --- |
| Quantile statistics | | Descriptive statistics | |
| **Minimum** | 18319.58 | **Standard deviation** | 3038.23 |
| **5-th percentile** | 20499.16 | **Coefficient of variation (CV)** | 0.12 |
| **Q1** | 23410.48 | **Kurtosis** | -0.70 |
| **median** | 25875.68 | **Mean** | 25719.50 |
| **Q3** | 28217.66 | **Median Absolute Deviation (MAD)** | 2371.54 |
| **95-th percentile** | 30515.59 | **Skewness** | -0.13 |
| **Maximum** | 33154.12 | **Sum** | 42231421.26 |
| **Range** | 14834.54 | **Variance** | 9230845.64 |
| **Interquartile range (IQR)** | 4807.18 | **Monotonicity** | Not monotonic |

This analysis used 70% of total data as the parameter selection set to carry out optimization and another 30% as the test set to compare the efficiency of model prediction. In cater to the need to solve over fitting problems in model training, parameter selection set is divided in to a validation set and a train set, each containing 35% of the total data. The sampling is carried out by taking equal proportion of data from each of the 8 years (with the percentage structure of increase and decrease case in each year also maintained). This approach of data sampling makes each dataset more representative of the fluctuation of stock price to deal the change of data distribution from year to year, gaining a better prediction accuracy. The data summary is depicted in Table 2.

**Table 2**

The mean and median for each year’s train set, validation set and test set of HSI close price.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Year** | **Close\_trn** | | **Close\_val** | | **Close\_tst** | |
| **Mean** | **Median** | **Mean** | **Median** | **Mean** | **Median** |
| **2015** | 21883.14 | 21868.39 | 22187.23 | 22140.47 | 22312.36 | 22370.04 |
| **2016** | 21509.66 | 21561.06 | 21383.50 | 21231.57 | 21418.01 | 21374.82 |
| **2017** | 25984.07 | 25808.34 | 26247.47 | 25901.10 | 26473.85 | 26849.44 |
| **2018** | 28767.33 | 28535.50 | 28893.12 | 29059.92 | 28901.02 | 29582.16 |
| **2019** | 27564.80 | 27298.28 | 27575.82 | 27614.89 | 27590.38 | 27611.47 |
| **2020** | 25269.16 | 25122.14 | 25303.86 | 24886.14 | 25337.81 | 24890.68 |
| **2021** | 27263.72 | 28050.22 | 27107.26 | 27561.34 | 26880.52 | 27723.84 |
| **2022** | 22928.78 | 23091.00 | 22918.08 | 22901.56 | 22730.99 | 22761.71 |

Technical indicators are useful in forecasting future stock movement. It is crucial to select the most explanatory indicators into the model to have the efficient performance. The 10 indicators this study employed follows the technical analysis system proposed by Kyoung-jae Kim and Ingoo Han (2020) that has been generally accepted by some of the widely cited researches in financial machine learning field (Kim, 2003, Kara et al 2011, Patel et al, 2014). Table 3 gives the formulas for each selected indicators while Table 4 summarize the detailed statistics generated from historical data.

**Table 3**

Selected indicators and their formulas.

|  |  |
| --- | --- |
| Name of indicators | Formulas |
| **Simple n(10here)-day Moving Average** |  |
| **Weighted n(10here)-day Moving Average** |  |
| **Momentum** |  |
| **Stochastic K%** |  |
| **Stochastic D%** |  |
| **Relative Strength Index (RSI)** |  |
| **Moving Average Convergence Divergence (MACD)** |  |
| **Larry William’s R%** |  |
| **A/D (Accumulation/Distribution) Oscillator** |  |
| **CCI (Commodity Channel Index)** |  |

is the closing price, the low price, the high price at time t, , exponential moving average, , smoothing factor: , is time period of k day exponential moving average, and mean lowest low and highest high in the last t days, respectively, ; ; .

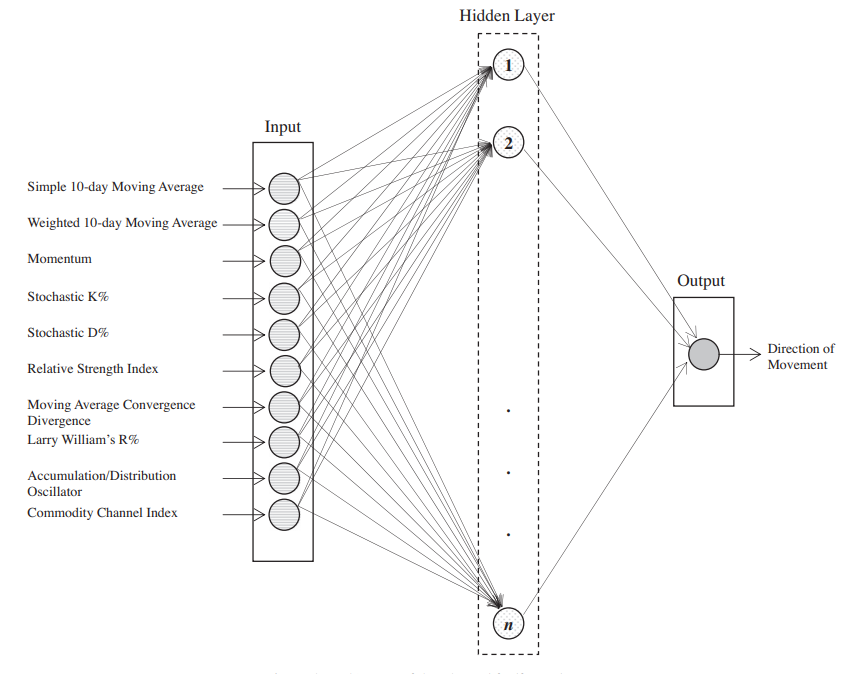
**Table 4**

Summary statistics for all 10 indicators (taking HSI as an example).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Min** | **Max** | **Mean** | **Std** |
| **SMA** | 32779.67 | 18924.34 | 25739.69 | 2998.73 |
| **WMA** | 32807.40 | -4476.05 | 25739.24 | 3003.75 |
| **Momentum** | 3269.89 | 53.55 | -2.96 | 924.44 |
| **STC K%** | 100.00 | 0.00 | 53.55 | 32.29 |
| **STC D%** | 99.20 | 1.51 | 53.52 | 27.18 |
| **MACD** | 93.22 | -1326.05 | 1.31 | 315.27 |
| **RSI** | 855.70 | 11.00 | 51.50 | 14.66 |
| **WILLR%** | 0.00 | -45.59 | -45.59 | 32.26 |
| **A/D Osc%** | 1.57E+08 | -2.39E+07 | -2.39E+07 | 4.88E+07 |
| **CCI** | 255.90 | -314.32 | -314.32 | 103.74 |

1. **Prediction models**
   1. Feedforward Neural Networks (FNN)

Feedforward neural networks is a multilayer perceptron where, as occurs in the single neuron, the decision flow is unidirectional, advancing from the input to the output in successive layers, without cycles or loops (Urso et al, 2019). As the simplest neural network model, FNN is less time consuming to construct and face fewer statistical problems like gradient explosion and gradient vanishing, but generally produce a less accurate result in prediction. In this study, a four-layer FNN model is employed, taking in 10 technical indicators as neurons in the input layers and generate a single neuron in the output layer. For each neuron in the hidden layer and output layer, ReLU is adapted as the transfer function. The structure of FNN model is depicted in Fig1. To determine the best parameter set, we introduced 10 potential number of neurons in the hidden layer (hl) and 10 potential value of epoch times (ep). The first step of parameter selection is carried out with learning rate (lr) fixed at 0.1. After the best choice of hl and ep is discovered, lr will be tested in 10 different levels in searching for the most-fit. The fluctuating parameters and potential value is given in Table 5. Since the model selection is firstly conducted on hl and ep and then lr, 900 potential model selections are introduced with 109 times of model fitting.



**Fig. 1.** Architecture of ANN model Kara et al. (2011) （这图是什么垃圾，我待会自己画）

**Table 5**

FNN parameters and their potential value tested.

|  |  |
| --- | --- |
| Parameters | Potential values |
| **Number of hidden layer 1 neurons (n)** | 10, 20…..., 100 |
| **Number of hidden layer 2 neurons (n)** | 10, 20…..., 100 |
| **Epochs (ep)** | 1000, 2000…..., 10000 |
| **Learning rate (lr)** | 0.1, 0.2…..., 0.9 |

Parameters and methods are taken to solve the over-fitting and underfitting problem. For activation function, we used ReLU whose convergence speed of SGD is proved by Krizhevsky et al to be faster than Sigmoid/Tanh, whereas model’s criterion and optimizer are selected as cross entropy and SGD, typical for classifying problems.

Apart from cross validation, two other methods are also introduced to solve the Underfitting and overfitting problems, namely dropout method, batch normalization for layers.

When a complex feedforward neural network is trained on a small dataset, it is prone to overfitting. According to Hinton (2012), to prevent this, the performance of the neural network can be improved by preventing the feature detectors from working together, which is the basic line of dropout method. Since we only have 1642samples in dataset but 10 indexes, dropout method is introduced in modeling.

Layer Normalization (LN), which normalizes all neurons in an intermediate layer, is introduced to prevent over-fitting and enhance prediction accuracy, including normalizing the input data before modeling and normalizing data output from each layer before entering the activation function.

For deep learning, a network structure that contains many hidden layers will face the problem of covariate shift due to the constantly changing of parameters during training process, which will lead to underfitting of the model. According to Sergey Ioffe and Christian Szegedy (2015), this problem can be delt with batch normalization (BN). The basic idea of ​​BN is actually quite intuitive: because the activation input value of the deep neural network before the nonlinear transformation, as the network depth deepens or during the training process, its distribution gradually offsets or changes. The reason why the training convergence is slow is generally that the overall distribution gradually approaches the upper and lower limits of the value range of the nonlinear function, which leads to the disappearance of the gradient of the low-level neural network during backpropagation, making the training of the deep neural network slower. BN is to use a certain normalization method. The skewing distribution of the input value of the element is forcibly pulled back to a standard normal distribution with a mean of 0 and a variance of 1 so that the input value falls in a range where the nonlinear activation function is generally more sensitive. In tis way, the gradient becomes larger and the problem of gradient disappearance is prevented, meanwhile a faster learning convergence can speed up the training process.

* 1. Recurrent Neural Networks (RNN)

Comparing with FNN, recurrent neural network (RNN) is more suitable for time series dataset analysis. In FNN, data is processed independently as datapoint, without necessary connection with previous data. RNN has the concept of ‘memory’ that helps them store the states or information of previous inputs to generate the next output of the sequence. In this experiment, a four-layer RNN model is constructed to predict the index/stock price movement, with the input data extend from 10 technical indexes to additional n period previous movements of the close price. The number of backtracking periods is obtained by best parameter selection experiment. The experiment also includes the selection of best number of hidden layers, best epoch for training and best learn rate with the same process in FNN modeling.

The RNN models face the underfitting and overfitting problems just as the FNN models, so cross validation, dropout method and batch normalization for layers are also introduced. What’s more, since the memory process of RNN tend to learn the feature from the close period while the early period’s influence keeps decreasing, we used Long Short-term Memory (LSTM) to improve the memory capability of model when dealing with long-period dataset. First introduced by Sepp Hochreiter and Jürgen Schmidhuber in 1997, LSTM model is equipped with forget gate on each layer to select useful information while forgetting the useless. This additional path of information transfer ensures the features from early data being learnt effectively.

To deal with the gradient explosion problem, ReLU weight initialization is carried out.

1. **Results**

We used accuracy and F1-score to analyze the performance of model prediction. Accuracy is the most straightforward metric to measure a classification model, but its flaws are also obvious. For dataset whose target classification distribution is skewed, a model which is unsensitive to the certain kind of classification will also show a high accuracy. Therefore, F1-scrore is introduced. Precision reflects the model's ability to distinguish negative samples. The higher the Precision, the stronger the model's ability to distinguish negative samples; Recall reflects the model's ability to identify positive samples. The higher the Recall, the stronger the model's ability to identify positive samples. The F1 score is a combination of the two. The higher the F1 score, the more robust the model. The formula of accuracy and F1-score are as follows.

Table 6 gives the statistical details of FNN’s prediction performance obtained from the best parameter combination for 4 datasets while Fig3 depicts the confusion metrics of each model. It can be learnt that FNN model using dropout and batch normalization methods is generally robust enough in stock index and individual stock prediction with both Accuracy and F1-Score reaching a level of 83%, which is almost 10% higher than Patel, J., et al. (2014) ‘s prediction performance of FNN model for Indian stock and stock index without (an average accuracy of 74.94% and F1-Score of 76.59% reached).

**Table 6**

Performance of model prediction for 4 datasets and their average.

|  |  |  |
| --- | --- | --- |
| **Index** | **FNN** | |
| **Accuracy** | **F1-Score** |
| **HSI** | 81.34% | 82.51% |
| **HSIII** | 84.99% | 85.82% |
| **Tencent** | 82.35% | 83.68% |
| **SMIC** | 82.35% | 78.62% |
| **Average** | 82.76% | 82.66% |



**Fig. 3.** Confusion Metrics for FNN Models

1. **Conclusion and discussion**