Application of Deep Learning in Stock Market Index Prediction

Liu Zhemin
School of Computer Science and Engineering

Assoc Prof Yeo Chai Kiat
Chen Weiling
School of Computer Science and Engineering

***Abstract* –** Stock market indices forecast is a time series prediction problem scrutinized by both the industry and academia with predictions traditionally done with Autoregressive Integrated Moving Average (ARIMA), Black Scholes etc. With recent advancements in machine learning and deep learning techniques, coupled with improvements in computing capabilities and Recurrent Neural Network (RNN) models, promising results have been shown in time-series predictions such as weather forecasting. However, deep learning techniques often require a large and representative dataset and yet, not all the features in the selected datasets might be relevant to the target function. The performance of the deep learning model might also be affected by the Curse of Dimensionality (CoD), overfitting and many other machine learning issues.

In this paper, we will focus on applying various machine learning methodologies that enhances the data representation. For instance, utilizing some domain knowledge, feature engineering is performed to derive new features (technical indicators), such as Moving Average (MA), Relative Strength Index (RSI), and Bollinger Bands (BB), from existing features. Feature transformation approaches such as the Principal Component Analysis (PCA) was also experimented on the dataset to reduce the dimensions and prevent the Curse of Dimensionality, leading to a more simplified and improvised model that abides by the Occam's razor. Initial empirical results have shown that the new models commanded slight improvements over baseline methods, with the new technical indicators as features. In the later part of the experiments, sentiment scores from Online Social Networks (OSN) were calculated and added to the dataset and observations have shown that the model outperformed baseline model.

Keywords – Stock Market Index Predictions, Sentiment Analysis, Feature Engineering, Technical Indicators

1 INTRODUCTION

Trends prediction in the financial markets such as Stocks, ETF, Forex, and Cryptocurrency has been an area of interest in both the industry and academia. Such predictions can be classified as Time Series Forecasting modeled as a sequence indexed by time. However, achieving a satisfactory performance in order not to incur a loss for the trader is challenging due to the dynamic and unpredictable external factors that are involved [1]. This results in a market condition that is volatile with random fluctuations in the stock's valuation, bringing notable risk to the trader.

The three common analysis methods used by traders to evaluate the value of a stock or currency can be broken down to three components – Fundamental Analysis (FA), Technical Analysis (TA) and lastly, Sentiment Analysis (SA) [2]. The testbed that will be used for testing and prediction is the Shanghai Shenzhen CSI 300 Index (CSI 300 Index) which benchmarks the performance of the 300 most dominant stocks being traded in both Shanghai and Shenzhen exchanges.

In this project, we developed an RNN model to attempt to learn the patterns that might exist in the temporal dataset to perform basic predictions. We then focused on applying various machine learning methodologies that enhances the data representations, mainly performing both TA and SA. New features such as MA, RSI, and BB were engineered to include several technical indicators often used by traders for valuations of stocks.

The remaining sections of the paper are organized as follows: Section 2 summarizes the relevant research and literature review conducted during the period of the research. Section 3 reviews the methods used to analyze and improve the data representations before conducting the experiments. Section 4 contains the training models experimented followed by Section 5, which discusses the results obtained. Lastly, Section 6 will conclude this paper.

2 LITERATURE REVIEW

A traditional and popular method for time series forecasting is the ARIMA technique. ARIMA is a class of statistical linear models used frequently in the past few decades, largely attributed to its accuracy, mathematical soundness, and its ability to apprehend some of the temporal structure in the time series data [3], [4].

The Black Scholes-Merton (Black Scholes) model is another commonly used model that is used to obtain a theoretical estimate on the valuation of the stock options. The extent of the usage is so wide that it led to an increased volume in stock options trading that potentially influenced the market itself [5].

However, with the recent advancements in machine learning and deep learning techniques, coupled with improvements and advancements in computing capabilities, the usage of Artificial Neural Networks (ANN) has become a promising candidate that is actively being researched on to replace the traditional

methods used to forecast and predict in various domains.

In particular, the usage of ANN has shown to be either comparable or even superior to traditional predictions models, such as the Black Scholes model [6].

RNN is a class of ANN that is able to capture the dynamic temporal behaviors as there are connections between nodes to form a directed graph along a sequence. Due to its ability to capture the dynamic temporal behavior of the dataset, it is frequently used for time series forecasting, with results that outperform static models such as the Multilayer Perceptron (MLP) [7].

Feature Engineering is another machine learning technique commonly used in the data pre-processing pipeline. It creates new features from existing features using some domain knowledge. However, one drawback is that it might be laborious, time-consuming and often require expertise or specific domain knowledge [8]. The new features that are created can often assist the learning algorithm to focus on crucial factors to improve the performance and accuracy of the model.

Investors also often perform SA to determine the sentiment of the originator with respect to a certain topic. It is traditionally used to help a business identify and understand the social sentiment of the brand, services or product. When used on a particular stock metric, it can be used to assess the general public's sentiment towards the stock and in turn, help trader with their decision whether to buy or sell the stock.

3 DATA & REPRESENTATION

3.1 DATASET

The basic time series dataset for the CSI 300 index is first obtained from Tushare, a free and open source python financial data interface package, mainly catered for the Chinese markets. To retrieve the dataset and save it as an excel file with CSV extension, we first install the following dependencies using pip: Pandas, lxml and finally Tushare. We can then run the following python script to save it as a CSV file. The code to be used to retrieve the dataset is 'hs300' shown in Fig. 1.

```
import tushare as ts
import os


def save_dataset(start, end, code):
    path = ['data', 'raw', 'raw_stock.csv']
    cur_path = os.getcwd()
    if not os.path.exists('data/raw'):
        os.makedirs('data/raw')
    data_path = os.path.join(cur_path, *path)
    data = ts.get_hist_data(code=code, start=start, end=end)
    data.to_csv(data_path)
```

Figure 1 Python script used to retrieve the dataset

For the purpose of maintaining consistency in all the different experiments with varying feature sets, we will be using the start date and end date of 1st January 2015 and 8th March 2017 respectively. This leaves us 530 number of trading days to be used in the training process.

The basic feature set of datasets provided by tushare is shown in Table 1.

Feature		Description
(1)	Date	Date of the training instance, typically used as index
(2)	Open	Opening price for the index during the start of trading day
(3)	High	Highest price being traded for the index during the trading day
(4)	Close	Closing price for the index during the closing of the trading day
(5)	Low	Lowest price being traded for the index during the trading day
(6)	Volume	Volume of the stock for the entire trading day
(7)	Price_change	Total price change for the entire trading day
(8)	P_change	Percentage change for the trading day
(9)	MA5	Closing price MA for 5 days
(10)	MA10	Closing price MA for 10 days
(11)	MA20	Closing price MA for 20 days
(12)	V_MA5	Volume MA for 5 days
(13)	V_MA10	Volume MA for 10 days
(14)	V_MA20	Volume MA for 20 days

Table 1 Default feature set provided by tushare

3.2 DATA VISUALIZATION

After saving the raw dataset to a CSV file, data visualization is first performed to analyze if we can capture any high-level patterns from the raw dataset. For instance, if there are any visually distinct clusters in the dataset, it may potentially be holding hidden patterns such as buy, sell or hold clusters. We can then apply clustering algorithms such as Single-Linkage, K-Means or the Expectation-Maximization Clustering and add the new clusters as a feature, which can be used by the learning algorithm.

Since there are many dimensions in the dataset, it is impossible to visualize the dataset on a computer with all the features in the dataset. Hence, dimensionality reduction algorithms such as PCA and Independent Component Analysis (ICA) is first performed to reduce the number of dimensions prior to plotting them for visualization.

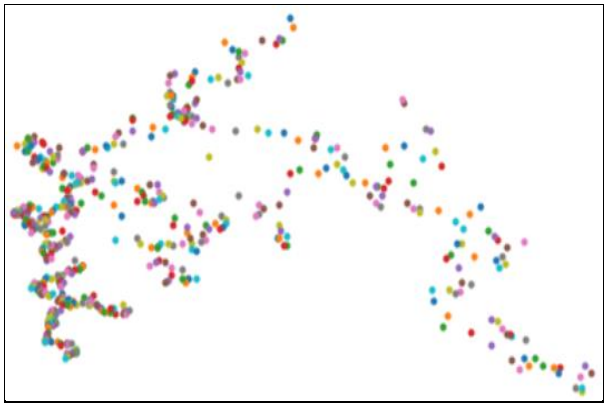

Figure 2 Visualization of dataset after PCA

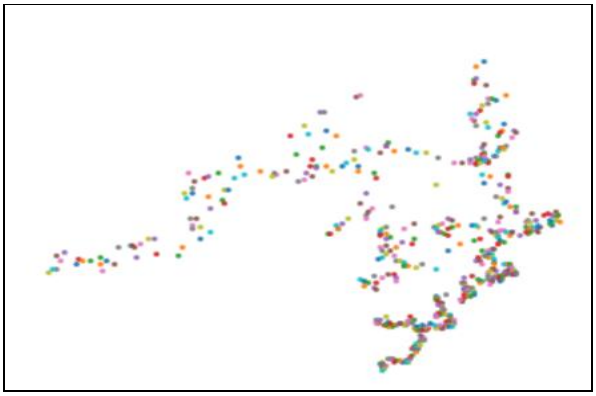

Figure 3 Visualization of dataset after ICA

Fig. 2 and Fig. 3 show the visualization of the dataset after using the dimensionality reduction algorithm, PCA, and ICA respectively. We can see that there is one densely clustered area with the rest of the points being spread out randomly. Unfortunately, this is not enough to be used to cluster the points and use them as a new feature.

3.3 FEATURE ENGINEERING

Technical indicators are often calculated from the dataset using both temporal and sequential difference between each training instances with the goal to provide signals on the potential movements of the market. In this section, we will be performing feature engineering to complement the existing feature set with some of the common technical indicators used by traders to evaluate the value of the stocks or any signals before making any decisions to buy, sell or hold the stocks.

3.3.1 Moving Average

MA and Volume Moving Average (VMA) in terms of stocks are often referred to as the MA of the closing price and the volume respectively.

$$MA_n = \frac{\sum_{i=1}^{n} close_i}{n} \quad (1)$$

$$V\_MA_n = \frac{\sum_{i=1}^{n} volume_i}{n} \quad (2)$$

Where,
n = Number of periods in days

Equations (1) and (2) show the formulae to calculate MA and VMA respectively. Although it is included in the basic feature set, calculating for MA is important as it can give rise to more complex technical indicators such as BB, Exponential Moving Average (EMA), Moving Average Convergence Divergence (MACD) etc. The period n is also crucial as using the correct period will better reflect the market volatility and increase the performance of the model.

3.3.2 Relative Strength Index

The RSI is a widely used momentum oscillator that is utilized to measure both the rates and movements of price changes. It takes the range of 0 to 100 and determines whether a particular index is overbought or oversold. If the RSI value is above 70, it is considered to be overbought and if the value falls below 30, it is considered to be oversold.

$$RSI = 100 - \frac{100}{1 + \text{Smoothed RS}} \quad (3)$$

$$\text{Avg Gain} = \frac{\sum_{i=1}^{n} [gain>0]_i}{n} \quad (4)$$

$$\text{Avg Loss} = (-1)\frac{\sum_{i=1}^{n} [gain<0]_i}{n} \quad (5)$$

$$1^{st}\ RS = \frac{\text{Avg Gain}}{\text{Avg Loss}} \quad (6)$$

$$\text{Smoothed RS} = \frac{\frac{(\text{Prev Avg Gain} \times (n-1) + \text{Current Gain})}{n}}{\frac{(\text{Prev Avg Loss} \times (n-1) + \text{Current Loss})}{n}} \quad (7)$$

Where,
n = Number of periods in days

Unlike normal RSI, Equations (3) to (7) use the smoothed Relative Strength (RS) to calculate the RSI. The main difference in the calculations of the first RS and subsequent RS is in terms of the average gain and average loss used. Smoothed RS uses the previous day's average gain and loss, together with the current gain for smoothing purposes. This prevents the RSI from over fluctuating from just one day of high price change.

3.3.3 Bollinger Bands

The BB consists of 2 bands, plotted in two standard deviations both below and above the simple MA. Both the standard deviation and band vary as the volatility increases and decreases.

$$\text{Upper Band} = MA_n + SD_n \times 2 \quad (8)$$

$$\text{Lower Band} = MA_n - SD_n \times 2 \quad (9)$$

Where,
MA_n = Simple MA for n days

SD_n = Standard Deviation for n days

When the band narrows, it signals a period of low volatility. Likewise, when the band widens, it signals a period of high volatility. Another popular belief is that the BB is signaling an overbought market when the price moves towards the upper band and it signals an

oversold market when the price moves towards the lower band.

3.4 SENTIMENT ANALYSIS

With the rise in the popularity of OSN such as Twitter, promising results have been shown by incorporating SA into the feature set for the learning algorithm to consider the public's sentiment towards the stock index [9].

The equivalent of Twitter for the Chinese market is Sina Weibo (SWB), which bears similar results for the Chinese stock index, outperforming baseline methods without sentiment [10].

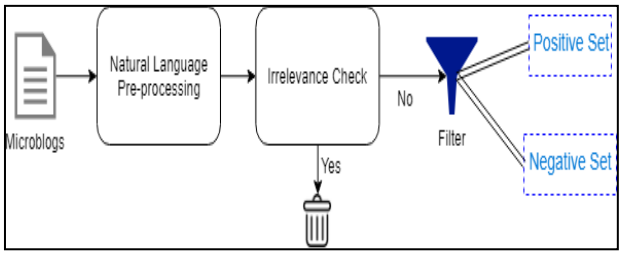

Figure 4 Processing pipeline for microblog

For this project, we first create a set of 100 SWB accounts that potentially have an influence on the index to a certain extent. One criterion for the account is that they have their identities officially verified by the platform. After the set is created, we will then retrieve all microblogs that are published by them during the period of concern as mentioned earlier (1st January 2015 and 8th March 2017). The process is then followed by some microblog preprocessing using Natural Language Processing (NLP) methodologies, such as tokenization, stop word and punctuation removals coupled with dropping of microblogs that are irrelevant such as greetings and requests for a repost.

After obtaining the pre-processed microblogs, we then filter out these microblogs that caused some volatility in the market the next day and split each microblog into 2 distinct sets, s_1 and s_2. The overview of the process is shown in Fig. 4. We then calculate the probability of each keyword w_i, given that it belongs to set s [10].

$$P(w_i|\, s) = \frac{1}{total_s} \sum_{i=1}^{n_s} freq(w_i, m_i) \qquad (10)$$

Where,

$total_s$ = Frequency of words w_i in the class s_1 or s_2

n_s = Total number of microblog in the set s_1 or s_2

$freq(w_i, m_i)$ = Frequency of word w_i in microblog m_i

For each microblog m, we then calculate the probability that it will fall into each set by:

$$P(m \in s) = P(s) \prod_{i=1}^{n_w} P(w_i|\, s) \qquad (11)$$

Where,

$P(S)$ = Probability of class s, e.g. $total_{s1} / (total_{s1}+total_{s2})$

n_w = number of words in each microblog

The influence of each microblog m can then be calculated by:

$$P(m) = \frac{P(m \in s_1)}{P(m \in s_1) + P(m \in s_2)} \qquad (12)$$

4 TRAINING

4.1 MODEL

Gated Recurrent Unit (GRU) and Long Short Term Memory (LSTM) are the most common recurrent units for RNN. However, GRU is found to be comparable to LSTM. In particular, GRU has shown superior performance with a dataset that is smaller [11]. Since we have a small dataset of 530 training instance, it is less computationally intensive and logical to use GRU as our recurrent unit.

4.2 TRAINING

In this project, we adopted a RNN with 2 hidden layers and GRU as the recurrent units. We then used a period where p = 10 days to train and predict the closing price of a particular day. We choose a value of 10 for p because later parts of the experiments suggested that it is a suitable period lookup due to the volatility of the index. When predicting for a more volatile market, a lower value of p might achieve a better result.

During the training process, the weights and parameters of the GRU units are first initialized randomly. The prediction is first done using the initialized weights to compute an output. The output is then compared with the ground truth and we will perform backpropagation with gradient descent to adjust the weights and parameters in order to optimize the loss function

4.3 DIMENSIONALITY REDUCTION

With the use of smaller datasets, coupled with noise and a stock market that is dynamic and unpredictable, having too many features compared to the number of training instances can often result in CoD. Hence, we performed PCA, which obtains the principal components that have the largest possible variance to reduce the dimensions to a specified amount and subsequently compare the results with the results obtained from the raw feature set.

5 RESULTS

5.1 PERFORMANCE METRICS

The performance metrics used to evaluate the model in this experiment are both the accuracy (14) and the Root Mean Squared Error (RMSE) (13).

$$RMSE = \sqrt{\frac{1}{N} \sum_{t=1}^{N} (y_{pred} - y_{true})} \qquad (13)$$

$$accuracy = \frac{pred_{correct}}{\text{Total number of predictions}} \times 100 \qquad (14)$$

Equation (14) shows the formula for calculating the accuracy of the model. First, the class of each training instances is labeled as either buy or sell, depending on whether there is a rise or dip from the closing price of

the index for the next day. If the predicted closing price for the next day will be lower than the current closing price, we will label the training instance as a sell. Conversely, if the predicted closing price for the next day will be higher than the current closing price, we will label it as a buy.

Each experiment with distinct feature set is then run for 100 times before aggregating and the results and the average accuracy and RMSE will be shown in the next few sections.

5.2 WITHOUT SENTIMENT ANALYSIS

For this and the next section, we will denote feature set f_{norm} as the feature set that contains all the basic features which are provided in most public dataset denoted as feature numbers 2-8 in Table 1. The more notable results will be shown in Tables 2 and 3.

Features set	Accuracy (%)	RMSE (%)
f_{norm}	52.88	2.90
f_{norm}, MA10	53.37	2.87
f_{norm}, V_MA10	53.05	2.89
f_{norm}, MA10, V_MA10	53.00	2.88
f_{norm}, RSI	52.72	2.96
f_{norm}, BB	53.33	2.89
Close	51.13	3.02
Close, MA10	53.52	2.95
Close, BB	**53.86**	**2.90**
f_{norm}, RSI, ma10 (PCA:5)	**53.61**	**2.86**

Table 2 Results for feature sets without sentiment

As we can see from Table 2, the accuracy of just the close feature is 51.13%. Using the basic feature in most public dataset indeed improved the performance of the model both in terms of accuracy and RMSE. However, it was observed that using {f_{norm}, MA10}, the model performed poorer than purely using {Close, MA10}, which is evident that there may be noise in the dataset or the model might be affected by CoD. The best performance in terms of accuracy and RMSE are both highlighted in Table 2 and they are the {Close, BB} and {f_{norm}, *RSI*, MA10} with PCA to reduce the dimension to 5 respectively.

5.3 SENTIMENT ANALYSIS

Table 3 shows the results of the feature set with the calculated sentiment values included in the training.

Features set	Accuracy (%)	RMSE(%)
f_{norm}, Sentiment	64.05	2.23
f_{norm}, Sentiment, MA10, BB, RSI	60.95	2.33
f_{norm}, Sentiment, MA10, BB, RSI (PCA: 5)	61.39	2.31
Close, Sentiment, MA10	64.61	2.207
Close, Sentiment, BB	62.5	2.261
Close, Sentiment, MA10, V_MA10	65.06	2.21
Close, Sentiment	**65.51**	**2.0419**

Table 3 Results for feature set with sentiment

After including the sentiment value in the feature set for RNN, we can see a significant improvement in the performance of the model. However, as we start to include more features such as technical indicators commonly used by traders, the performance starts to decrease. The performance of the same set of features {f_{norm}, Sentiment, Ma10, BB, RSI}, but with dimensions reduced, achieved better performance in terms of both accuracy and RMSE, indicating that the model might be affected by CoD. In general, the model with just the closing price and sentiment values, together with some of the technical indicators outperformed the feature set with all of the basic features included by a public dataset. The overall winner is the feature set {Close, Sentiment} with just 2 dimensions. One possible explanation is that by just using {Close, Sentiment}, we avoid the noises that might be included in some of the features that exist in the dataset, at the same time, avoiding the CoD and abiding by Occam's razor.

6 CONCLUSION

In this paper, we have discussed some of the challenges in stock market indices prediction due to the dynamic and unpredictable external factors involved. As stock market prediction is heavily researched in both the industry and academia, we then discussed some of the conventional methods used followed by promising, state of the art models like RNN.

We also performed feature engineering by using some domain knowledge to arrive at the common technical indicators that traders use day to day to make informed decisions, such as the volatility of the stocks, and whether the stock is overbought or oversold. After creating the new features and using them as inputs for the GRU model, it was observed that feature engineering and dimensionality reduction only brought slight improvements to the performance of the model. One possible explanation is that the Neural Network attempted to learn directly from the input space, decreasing the importance of feature engineering compared to conventional machine learning models.

In the later part of the project, we observed that performing SA brings major improvements to the

performance of the model, outperforming baseline models. This does not come as a surprise as the public sentiment towards the stock will generally determine whether the stock is overbought or oversold, which in turn determines the performance of the stock.

7 FUTURE WORK

Possible future work includes creating more complex technical indicators and the application of feature selection algorithms to select a subset of features for the learning algorithm. We could also have utilized more training instances during the training process. Lastly, we would like to incorporate ensemble learning by building several base models and predicting the closing price of the index using the weighted average.

ACKNOWLEDGMENT

We wish to acknowledge the funding support for this project from Nanyang Technological University under the Undergraduate Research Experience on CAmpus (URECA) programme.

The author of the paper would like to thank Chen Weiling for her contributions in some of the coding and technical aspect of this project

REFERENCES

[1] S.-S. Liaw, "Stock Markets are Unpredictable, but Can be Exploitable", *Research & Reviews: Journal of Statistics and Mathematical Sciences*, 2016.

[2] J. Wagner, "3 Types of Forex Analysis", *Dailyfx.com*, 2014. [Online]. Available: https://www.dailyfx.com/forex/education/trading_tips/chart_of_the_day/2014/02/13/3_methods_of_forex_analysis.html. [Accessed: 19- May- 2018].

[3] G. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model", *Neurocomputing*, vol. 50, pp. 159-175, 2003.

[4] J. Contreras, R. Espinola, F. Nogales and A. Conejo, "ARIMA models to predict next-day electricity prices", *IEEE Transactions on Power Systems*, vol. 18, no. 3, pp. 1014-1020, 2003.

[5] K. Ronnie Sircar and G. Papanicolaou, "General Black-Scholes models accounting for increased market volatility from hedging strategies", *Applied Mathematical Finance*, vol. 5, no. 1, pp. 45-82, 1998.

[6] J. Bennell and C. Sutcliffe, "Black-Scholes Versus Artificial Neural Networks in Pricing FTSE 100 Options", *SSRN Electronic Journal*, vol. 12, no. 4, pp. 243-260, 2004.

[7] T. Barbounis, J. Theocharis, M. Alexiadis and P. Dokopoulos, "Long-Term Wind Speed and Power Forecasting Using Local Recurrent Neural Network Models", *IEEE Transactions on Energy Conversion*, vol. 21, no. 1, pp. 273-284, 2006.

[8] A. Ng, "Machine Learning and AI via Brain simulations", *forum.stanford.edu*. [Online]. Available: https://forum.stanford.edu/events/2011/2011slides/plenary/2011plenaryNg.pdf. [Accessed: 24- May- 2018].

[9] T. Rao, S. Srivastava, "Analyzing Stock Market Movements Using Twitter Sentiment Analysis", *International Conference on Advances in Social Networks Analysis and Mining*, pp. 119-123, 2012.

[10] W. Chen, Y. Zhang, C. K. Yeo, C. T. Lau, B. S. Lee "Stock market prediction using neural network through news on online social networks", *IEEE International Smart Cities Conference*, 2017.

[11] J. Chung, C. Gulcehre, K. Cho, Y. Bengio "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modelling", *arXiv: 1412.35555 [cs.NE]*, 2014.