Mystery of Chinese Stock Market

Predicting the Movement of CSI 300 Index

Danfeng (Maple) Li (dl3983), Zexi Ye (zy1311), Bixing Yan (by783), Yucong (John) Hu (yh2860)

***Abstract*—In this project, we experimented and evaluated the predictive power of ARIMA, LSTM, and Xgboost tree model on the Chinese stock market, which is proxied by the benchmark CSI300. This business problem is relevant to our study because it not only helps us better understand time-series data with high variability, but it also endeavors to answer an economic question: is the Chinese stock market efficient; can we beat a heavily, governmentally intervened market by leveraging public information and predicting its next-day price movement? As we will demonstrate in our report, this is very unlikely. Even by leveraging a neural net model, LSTM, the average predictive power on price direction is 52% with a high of 63%. For future studies, we hope to apply the model on individual constituent stock of the index in order to better fit the model to the idiosyncrasy of each stock, and then aggregate them on a portfolio level to achieve better performance overall.**

I. Business Understanding

Chinese stock market is famous for its high volatility and idiosyncrasy, even by the standard of emerging markets. Traditionally, emerging markets have higher variance than developed markets, and are more prone to external shocks[1]. The characteristics mentioned above make Chinese markets ever more difficult to predict. As such, any strategy that can outperform the base rate, which in our case is equal chance of increase and decrease, or out-of-the-box models will potentially produce long term return.

In our approach, we try to tackle the issue by predicting the direction (next day upwards or downwards movement) of the market. The movement of the target variable is proxied by the benchmark: CSI 300 index - a weighted index of the 300 most valuable public companies listed in China. The reason behind the choice of target variable is that high capitalization companies are less volatile in their stock prices[2], and the CSI 300 is readily compiled daily at market close, making the target variable easier to predict. Using verifiable third-party data source will facilitate our data mining process.

The idea behind such prediction is that if we are more likely than not to be correct about next-day market performance, we can exploit these opportunities by taking buy or sell positions in our portfolio. Such inter-

[1]De Santis, G. (1997). Stock returns and volatility in emerging financial markets. Journal of International Money and finance, 16(4), 561-579.

[2]Copeland, M. M., & Copeland, T. E. (1999). Market timing: Style and size rotation using the VIX. Financial Analysts Journal, 55(2), 73-81.

day trading can beat the market in the long run if our prediction is more likely than not to be correct. Its relative performance to the market benchmark can also serve as a gauge for model evaluation.

II. Data Understanding

Our data is divided into two categories: internal Chinese factors and external factors. The goal of incorporating these data into our predictive model is to provide the model with additional historical information regarding the Chinese economy as well as the global economy. The rationale behind such practice lies in the following assumptions:

a) Stock markets effectively reflect the publics expectation about the domestic economy
b) Stock markets are subject to external shock in the international market, especially in the current context of Sino-American trade war

Therefore, we expect that these data are correlated with the direction of the stock price and, by including these features, the model will to some extent outperform the baseline model, which, based on the base rate, predicts the majority class indiscriminately. Our time span of interest falls between 01/01/2013 to 11/30/2018 because we believe a time span of 5 years offers sufficient amount of data for developing a satisfactory predictive model while captures recent spikes in volatility due to the trade war. Since we will feed our training data day by day to the predictive models, our primary data frame will consist of 2159 rows.

A. Domestic Data

For the domestic dataset, we picked a collection of macroeconomic data that are representative of the overall performance of the Chinese economy. The domestic data includes the historical daily CSI 300 index, our target variable. A detailed list of the data can be found in the Appendix at the end of this report. We pulled our data from reputable third parties website, such as the OECDs database[3], and retrieve the rest from TuShare[4], a Python package that specializes in providing real-time economic and financial data about China in the form of DataFrames. The release of these economic factors dictates the investors confidence in the national economy, affecting the equity market. Typically, the market will immediately react to a signal released by the authority or the central bank. For instance, a drop in the stock price index is expected after the Peoples Bank of China announces an increase in interest rates. Additionally, the signal may remain in effect for a short period of time, which will continuously drive up/down the stock market for days, until it is fully absorbed and priced in the market price. Therefore, we adopt a wide array of domestic economic indicators and expect a higher overall accuracy of model prediction.

B. U.S. Data

Since the global financial market has become increasingly interdependent over time, it is natural to include

[3]Real GDP Forecast. (n.d.). Retrieved Dec 5, 2018, from https://data.oecd.org/gdp/real-gdp-forecast.htm#indicator-chart

[4]TuShare. (n.d.). Retrieved Dec 5, 2018, from http://tushare.org/index.html

the data about the international market. Due to its sheer size and the power of the U.S. Dollar, American stock market wields strong influence over the market, and this is proxied by the movement of the barometer of the U.S. market - S&P 500. Therefore, a series of U.S. financial and economic data, including the historical S&P 500 indices as well as its Price-Earning Ratios, are incorporated. We retrieved all the U.S. data from Professor Robert J. Shillers personal website[5] at Yale University.

Remarkably, we decide to include President Trumps tweets in our model, based on the assumption that his tweets would influence the financial markets of both countries. In particular, in the context of the ongoing trade war between the U.S. and China, a sentiment analysis has been conducted on Trumps tweets that addressed discussions about China and trade. We believe that, a tweet that favors globalization and Sino-U.S. cooperation boosts the Chinese stock market, while one that opposes international free trade upsets the market. Hence, by taking President Trumps attitudes into account, we aim to foresee the fluctuation in the next-day market more accurately. We retrieved President Trumps original tweets on Trump Twitter Archive[6], where we searched the key words germane to our analysis consisting of "China," "trade," and "Xi," and saved it in a csv file for sentiment analysis.

[5]Shiller, R. J. (n.d.). Online Data Robert Shiller. Retrieved Dec 5, 2018, from http://www.econ.yale.edu/ ∼shiller/data/ie_data.xls

[6]Trump Twitter Archive. (n.d.). Retrieved Dec 5, 2018, from http://www.trumptwitterarchive.com/

C. Selection Bias

The issue of selection bias is somewhat tricky in this project. On one hand, since we retrieved our historical data in their entirety over the specified time frame, there was no sampling involved if we narrow down our period of interest to this particular time span, from 01/01/2013 to 11/30/2018. On the other hand, however, we deliberately picked the time interval in consideration of data availability as well as proximity to the present. Therefore, from a macroscopic perspective, the chosen data might fall victim to selection bias and fail to represent the characteristics of the features in a longer time horizon.

III. Data Preparation

A. Data Integration

One of the major challenges is that different economic and financial indicators were calibrated over different time steps. For instance, the real GDP growth rate is calculated quarterly, yet the money supply consists monthly data. To properly join all the data and form a ready-to-use data frame, we conducted following data frame merging and data cleaning steps.

1) Step 1. Date Alignment: Date Alignment A date column of from 2013-01-01 to 2018-11-30 was created and each data instance in the period would be assigned a row. To ensure consistency with the structure of the date column, we reformatted the date for each feature column so that all dates are in the YYYY-MM-DD form. Next, each feature data frame was left-joined to the

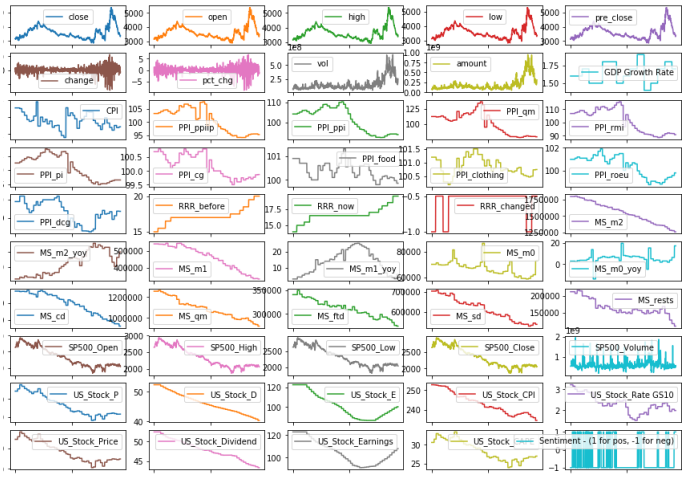

Fig. 1. Line plot of selected features & target variables. Some are repetitive or similar.

date column. By applying this to every data frame, we obtained a data frame with desired dimension.

2) Step 2. Drop Invalid Columns: We cleaned the data frame by dropping the invalid columns. Specifically, columns where all values are missing were dropped. Moreover, since a number of features have no longer been updated since the middle of our period of interest, such as the loan interest rate, we dropped these columns in favor of consistency in data length. This paved the way for the next step.

3) Step 3. Forward Filling the N/As by Column: Most of the features were recorded on a monthly or quarterly basis. Hence, we chronologically forward filled the NAs column and shifted each columns backwards by their time step in order to avoid data leakage; at any moment in time, only the past data are known. Further, for the column that features the Trump tweet sentiment, gaps between tweeting was forward filled assuming its potentially persistent impact on the Chinese stock market.

It is widely suggested that exponential moving average is good for filling in NA values in time series data. The method requires more a prior knowledge of our dataset and a deeper understanding of weight parameters. To avoid making more assumption about our data, however, we decide against exponential moving average after some explorations.

4) Step 4. Tweets sentiment prediction for future uses: In order to further automate our model for future uses, we built an automated Tweets cleaning and sentiment analysis program. Specifically, we used a rigorous 5-fold cross validation with grid-search on Bernoulli and logistic regression to find our best model. In terms of tokenization, we used both TF-IDF and count-based tokenization scheme by removing stop words and HTML markups as they contribute little to natural language understanding. In the end, our best model was the count based logistic regression (C=1 and L2 penalty) with an AUC score of 0.85.

B. Target Variable

In this project, the target variable is the direction of CSI 300s next-day closing price. To convert this into a classification problem, we denote the target variable by a binary variable, where 0 signals downward movement and 1 signals upward movement. We have also attempted to use the change of closing price directly. However, by plotting a simple correlation matrix, we observe that our features have limited explanatory power for the amount change in price.

C. Feature Engineering

To obtain a data frame that serves a classification model, we conducted feature engineering extensively. Firstly, for each feature column, we computed the percentage change in the current day value with respect to the previous day value row-wise in replacement of the original value. The rationale is that using the daily change in a feature is more informative in the context of directional prediction, which effectively standardized the data. Secondly, all quantitative features were normalized in preparation for fitting a Support Vector Machine at the later modeling stage. Thirdly, all features were uniformly shifted back by one day so that, within each row, the target value at a given day is aligned with the previous-day data.

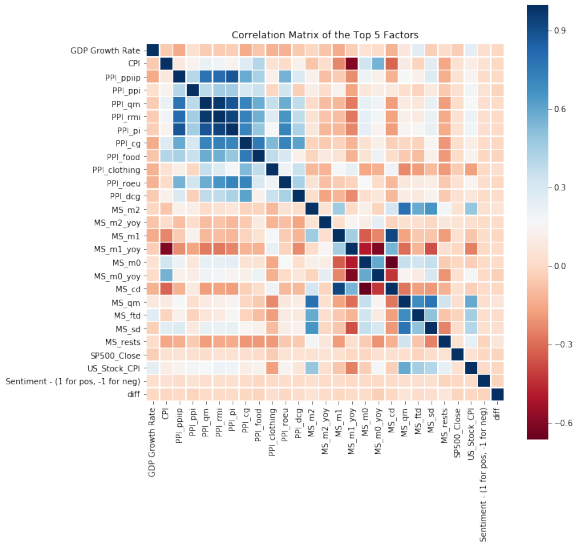

Fig. 2. Heatmap–correlation matrix of normalized features.

IV. Modeling & Evaluation

Financial modelling aims to maximize profit. To achieve this purpose, analysts always pursue a higher forecasting accuracy on the stock price. In terms of the research process, different from typical data science problems where performance is usually improved by feature engineering, in financial modeling, the higher accuracy is usually achieved by the application of more appropriate model. This is because, contrary to the typical data science problems where big data often bring problems, the financial modeling researchers usually worries about the lack of representative data due to the treacherous market condition. So, effective extract hidden information with a proper model is always the key point in a financial modeling research.

A. Algorithm

In this report, we will implement three models.

In traditional time-series analysis, **ARIMA** is the most widely applied models, famous for its simplicity and accuracy.

Recently, application of Recurrent Neural Network (**RNN**) rises quickly in the domain of sequential data. However, though this model succeeds in various area like natural language processing for its applicability, we still hard to find its position in the financial modeling. Thus, in this paper, we will analyze the possible reason and find a potential financial modeling application for an exemplar RNN model Long short-term memory (**LSTM**).

While accurate forecasting is the core of financial modeling, understanding the importance of different variable is also important for financial industry. Although the above models may wins on accuracy, it is hard to interpret the features from them. Thus, we will also

implement a boosted tree model **xgboost** to help us to understand the importance of different features.

B. Evaluation frameworks and metrics

For certain quantitative finance research, a precise price prediction can help analysts forecast the market conditions, while for most quantitative trading research, traders are more interested in the price trend, on which they can directly build their trading strategy. Mathematically, this means we should focus on two kinds of problem: regression (price) and classification (direction of price change).

For regression analysis on numerical value of price, we will follow the tradition and use standard deviation as the evaluation metric. For the direction of price change, which is a classification problem, since our sample price almost has an equal probability to decrease (labelled as 0, 49.9%) and increase (labelled as 1, 50.1%), it is reasonable to use the accuracy as our evaluation metric. Further, with regards to a confusion matrix, the unit cost of a false positive and false negative is virtually the same because they have the same underlying asset. In other words, whether we are in a long or short position, the monetary cost to incorrectly predicting a rise or drop in stock price, given we are only concerned with its direction of movement, is the same.

Confusion Matrix	Predicted: **Up**	Predicted: **Down**
Actual: **Up**	Long (buy) the stock	Loss on potential upside by selling too early
Actual: **Down**	Loss on potential downside by selling too late	Sell (short) the stock

Fig. 3. Confusion Matrix

C. Baseline model and its improvement process

In the traditional time series analysis, ARIMA is the most widely applied model. Thus, we choose it as our baseline model. The ARIMA model has three parameters (p,i,q) which we will optimized with following procedures:

First, we plot the price and its autocorrelation and partial autocorrelation function, as shown below: As

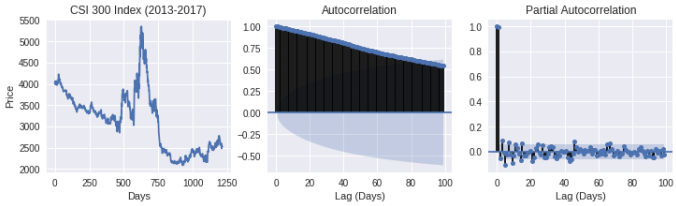

Fig. 4. Stock Price and its Autocorrelation and Partial Auotcorrelaation Fucntion

demonstrated in the first plot from the left, the price is not stationary. Therefore, we cannot use ARIMA (ARIMA with integral parameter i=0) model to predict price directly. This conclusion is also supported by the Autocorrelation and Partial Autocorrelation data. Thus, we further studied the price difference and its Autocorrelation (AC) and Partial Autocorrelation (PAC), as shown below: From the shape the the price difference and its

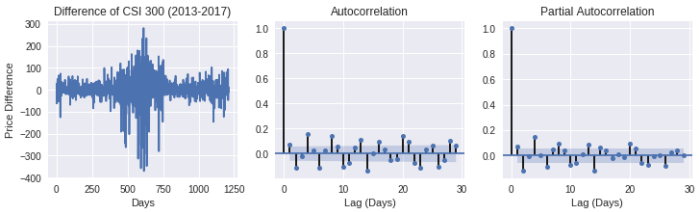

Fig. 5. Stock Price Difference and its Autocorrelation and Partial Auotcorrelaation Fucntion

AC and PAC plot. We can see the price difference is a stationary process and as such the integral parameter i should be 1. The ARIMA parameter (p,q) is identified by

the maximum likelihood principle. From peak positions in the AC and PAC of the price difference, we know the candidate of p and q should be chosen from [0, 2 , 4, 6, 8, 10, 13, 14, 20] and [0, 2, 4, 6, 8]. From our programming calculation, we know the (2,2) has maximum probability. Thus our model is ARIMA(2,1,2). The value and its statistic information of the training set are shown as followed:

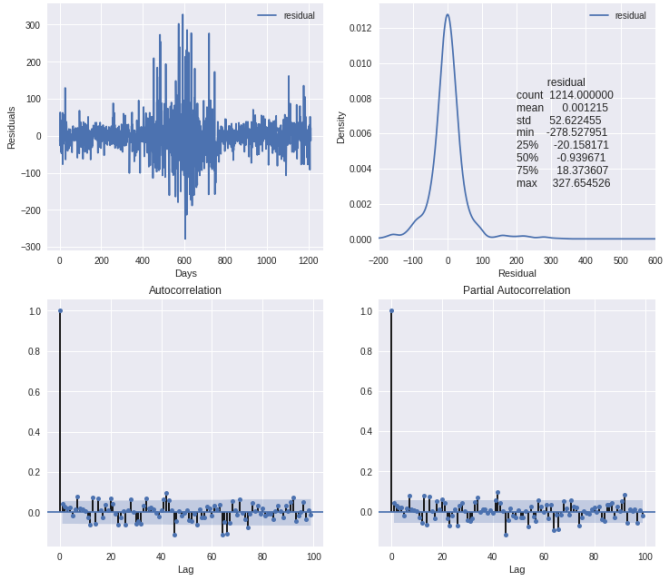

Fig. 6. Model Fitting Residues and its statistical information.

From Figure 6, we know that residue is mostly white noise with a small standard deviation. Thus, the model is acceptable. The fitting result of the test set is plotted as below:

The accuracy on price movement is 50.5%. This accuracy may be not significant for other data science topics. But in stock price prediction, even 0.1% probability can make great profit. And normally, most applied successful traditional time series model also just has a accuracy around 50%. Thus, we believe the baseline model is effective.

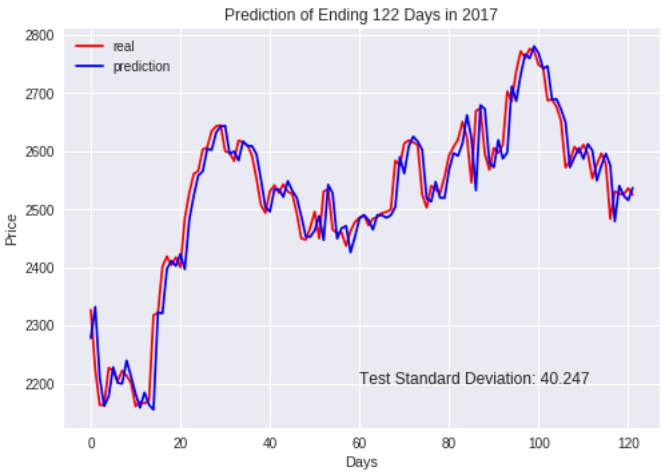

Fig. 7. Price Prediction of the optimized ARIMA model

D. LSTM model and its improvement process

LSTM model has risen in popularity lately. Here, we will evaluate its performance on the stock market. By passing the engineered features to the LSTM model, we can optimize the network parameters with a standard grid search cross-validation process.

Aside from its technical parameters, one important structural parameter of LSTM to decide by data scientists is the number of previous days input in the model. From the ARIMA process above, we can know the price could be correlated with the economic condition within two weeks. Thus, we run the LSTM with input of previous information from 2-14 days. The exact number of days is determined by the value of standard deviation. The plot of the calculation will be discussed later; and the prediction result of the optimized model is shown as below:

The standard deviation on the prediction of the price value is 39.13 (Just 3% improvement compared to the ARIMA baseline). On the other hand, the accuracy of

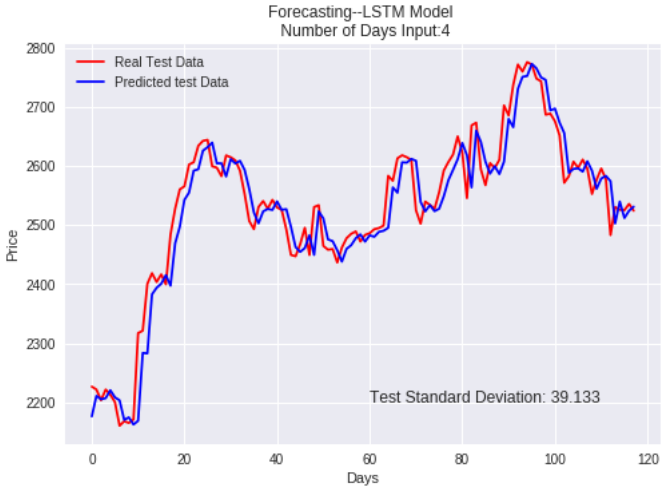

Fig. 8. Price prediction of the optimized LSTM model.

the sign of price change is 63.1% (Approximately 30% improvement compared to the ARIMA baseline).

E. Xgboost model and evaluation on the features

So far, although we could achieve a fair precision on the exact price and very high accuracy on the price trend, we still do not have an evaluation on the importance of features. Mostly, this is because ARIMA model is too simple to include the most features and LSTM is too much of a black box to for explanation, too.

In various machine learning models, decision tree algorithms are most famous for their interpretability. Hence, even though it may not be as accurate as ARIMA and LSTM, its interpretability could offer learning opportunities on the importance of different features. The parameters like depth, split and child weight are also optimized with a standard grid search, cross validation process. The prediction of optimized model is shown as below:

Although the standard deviation of the xgboost is worse than both of the ARIMA model and the LSTM

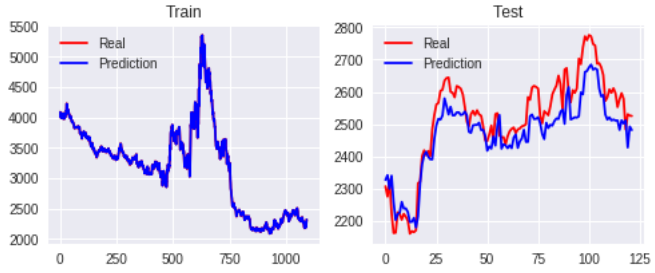

Fig. 9. Price Prediction of the optimized xgboost model.

model. The accuracy of the sign of price change is 53.1% which is 2% higher than ARIMA baseline, which can also bring much profit in real application. This means the xgboost could be used to predict the price trend even though it is not the best model. However, we could still roughly evaluate the importance of features from this model, which is just our purpose to implement it.

For the xgboost tree, the feature importance is evaluated by f score. This is because the F score of a feature indicate how many times the feature is split on. The higher the f score is, the more important the corresponding feature is. The f scores of different features are shown as following:

feature	SP500_Volume	SP500_Open	SP500_Low	SP500_Close	SP500_High	CPI	PPI_ppiip	GDP Growth Rate	PPI_clothing	PPI_cg
f score	2732	1803	1052	975	932	269	206	197	145	129

feature	MS_m1_yoy	PPI_dcg	MS_m0_yoy	PPI_pi	PPI_food	MS_rests	PPI_roeu	PPI_qm	US_Stock_Earnings	MS_m0	US_Stock_E
f score	127	114	110	96	86	81	78	70	68	62	62

Fig. 10. f-score of selected features.)

The evaluated features are S&P500 prices, CPI index, PPI index and GDP etc. Unsurprisingly, S&P500 has the highest f score, partly because it monitors daily change in the market. Other features that are monthly or quarterly collected have less explanatory power. In this case, PPI and money supply have limited predictive power.

F. Discussion

In this part, we will discuss the performance of different models on the prediction on price and its trend. Since the implement of arima and xgboost is very standardized. We just plot their best performance. For the LSTM, we will plot the detailed number of input day dependent metric values.

For the price prediction, we use standard deviation as the evaluation metric. To demonstrate the improvement of these models on forecasting performance, we also added a previous day baseline. The purpose of this baseline originates from the observation that the prediction looks like a delay of previous real price. So we have a question: whether the prediction is just a copy of previous days price? To this end, we calculated the standard deviation in the case if we use the present price as prediction.

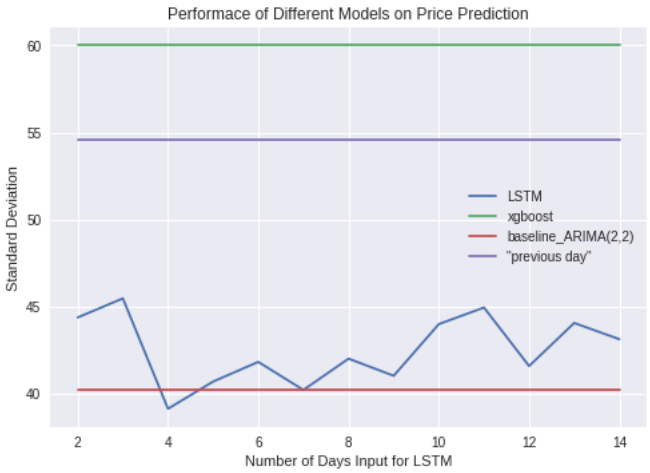

Fig. 11. Model performance on price prediction

Clearly, the ARIMA and LSTM do capture some hidden information in the dataset and thus could achieve more accurate prediction. However, in terms of the numeric price value, the xgboost performs rather disappointingly. This may be due to the nature of the tree algorithms. Usually, tree algorithms perform better in classification problems

Such reasoning is supported by the price trend prediction, the xgboost performs much better than the traditional ARIMA method. To make the improvement of our machine learning on forecasting result more explicit, we introduced a random guess baseline (randomly give 0 or 1) and only increase baseline (only give 1).

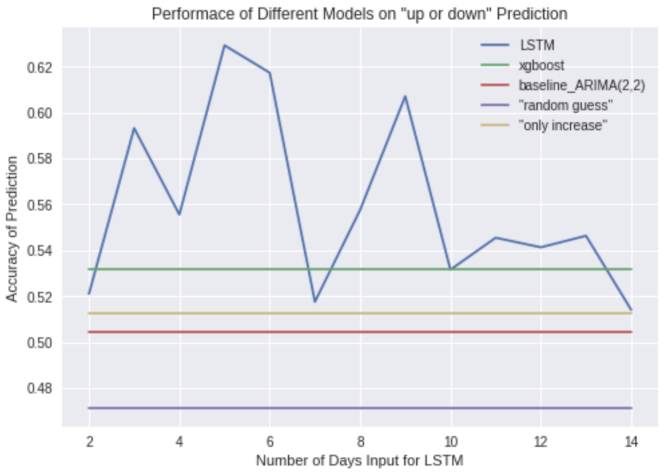

Fig. 12. Model performance on the price trend prediction

From the Figure 11, all of the machine learning model has a better performance than random guess. This assures us that these models do works in the stock price forecasting. In those models the LSTM outperforms other models. The optimized input information is 5 trading days which is just one week, which is reasonable because it is just a small cycle in the context of investment planning.

From the analysis above, we can see that, in the domain of numerical prediction, although the LSTM has the best performance, its improvement is not significant and it would require a complex feature engineering pro-

cess. Thus, we recommend to keep the classical ARIMA model. However, in the field of trading, which places more emphasis on the prediction of price trend, we suggest the use of the LSTM model to replace the ARIMA model. A significant improvement can be achieved by the adoption of the the LSTM model. However, if a company is more interested in the interpretability of models, and want accountability in the decision-making process, xgboost serves as a reasonable candidate as well.

G. Application in business problems

Since the price and price trend forecasting is important in many business problems, our model has great application potential. For example, our models could be adopted by investors and market researchers like investment banks, hedge funds and financial technology companies to maximize the profit and reduce risk. Also, since we are predicting an index, government agencies could also make economic policies based on our price trend prediction. For instance, how would an interest rate spike or external economic shock affect the health of the stock market?

V. DEPLOYMENT

Business knowledge of the subject matter cannot be overemphasized. Our approach here focuses on the technical analysis on the Chinese equity market. However, between the two schools of thought of fundamental analysis and technical analysis, industry practitioners almost always employ a combination of both when it comes to stock picking.

If we look instead at the up and down direction prediction of our models, the performance of the LSTM model is relatively satisfactory compared to that of the baseline ARIMA and xgboost. The average prediction accuracy over different time spans of 53% is not quite enough for an enterprise trading platform, but we can deliver it as an advisory online tool or robo-adviser. The rise of FinTech and online investment platforms should be perfect candidates for our model. The Trump tweet sentiment analysis is also automated. Whenever Trump writes a relevant post, the system makes a sentiment prediction and incorporates it as part of the model. In the end, if anyone does deploy our model, its efficacy can be readily tested by comparing investment returns based on prediction and that based on simply following the benchmark. This should be evaluated consecutively over an agreed-upon time period. Observing performance over longer time span can reduce accidental lucky prediction, or noise.

A. Concept drift, market efficiency dilemma, and their mitigation

Due to the industry cyclicality and seasonality of the equity market, concept drift will inevitably occur for companies deploying our model. Shifts between the emerging market and developed market, equity market and fixed income market, and business cycles will almost certainly undermine our model performance in the long run.

To mitigate such risks, frequent re-training of the model is necessary. Factors to specifically account for

such shifts can also be included in the model. Additionally, as mentioned above, fundamental analysis on the economys and markets health can also guide us in our management of the model. Additionally, depending on clientele, model interpretability and transparency could be a concern. Pension and sovereign funds will almost certainly demand accountability in the decision-making process. Ethically, such investors need to be assured of the integrity of the data used in the model, and what kind, if at all, private information is allowed. For private investors, however, the end might very well justify the means.

The most fundamental assumption underpinning modern economic theory is that an ideal market is efficient. In reality, however, this is rarely the case and arbitrage opportunities exist. As demonstrated in our best model, opportunities to beat the market does exist, especially if the model is consistent in its prediction in the long run. What's more, studies have shown that the market is otherwise efficient at disseminate information[7]. As such, any successful model should be a carefully guarded business secret, and continuously re-trained so as to retain its comparative advantage of being private information over the market benchmark, which is public information.

For future studies, we can apply the model on an individual stock basis to achieve better overall performance. We can also predict whether the market is going over a certain price, using the model as a resistance line or support line.

[7]Merton, R. C. (1987). A simple model of capital market equilibrium with incomplete information. The journal of finance, 42(3), 483-510.

B. Epilogue: A Few Insights

Apart from the technical discussion in the sections above, our test results offer some significant insights into a more fundamental economic problem as well. The relatively underwhelming performance of the LSTM and xgboost model, albeit both of which are our best, confirms the hypothesis of an efficient stock market. Since the features fed into the xgboost model are universally accessible to the public, an efficient stock market immediately reacts to the newly released signal through price adjustment, eliminating arbitrage opportunities. Hence, we conclude that models that purely rely on public data will hardly be profitable in practice.

VI. Appendix A: Contribution

Yucong (John) Hu - Retrieved and labeled President Trumps original tweets; performed sentiment analysis using 5-fold cross validation logistic regression. Tuned hyperparameters and achieved an accuracy of 87%. Merged features and target datasets and performed preliminary data cleaning and feature selection. Ran 10-fold cross validation model using SVM for market movement prediction; tuned hyperparameters. Wrote Business Understanding and Risks & Mitigation section. Proof-read report.

Zexi Ye - Labeled President Trumps original tweets. Cleaned the tweets for sentiment analysis. Retrieved the quarterly GDP growth rate data. Merged the features into a monolithic data frame and performed date alignment. Created the Table of Variables. Wrote the Data Understanding and Data Preparation sections of the report.

Danfeng (Maple) Li - Collected proposed feature variables such as CPI, PPI, US stock market index. Selected, preprocessed and normalized feature variables from cleaned dataset. Tried an exponential moving average method of filling NaN. Performed a few data visualization including feature plotting, scatter plot and correlation matrix heatmap. Wrote deployment section of the report. LaTeX the document.

Bixing Yan - Design and code the ARIMA analysis process; design and code the LSTM analysis process; design and code the Xgboost analysis process. Design the evaluation framework and metrics. Achieved a standard deviation of price prediction as low as 39 (6% of the amplitude) and an accuracy of price trend prediction as high as 63%. Wrote Business Modeling & Evaluation section.

`Code:`

1) ARIMA, LSTM: https://colab.research.google.com/drive/1il57F0UZQAiPxlSKIjAbKhzQ0lUtIIpE
2) xgboost: https://colab.research.google.com/drive/1-zc3YD0CnQKBqWnJPPnlhW3GpU4MLDA6#scrollTo=ULOB9zrHymZf

VII. Appendix B: Variable List

Variable Name	Time Step	Notes (All values are percentage change compared to previous day)
diff	Daily	(Target) current-day price minus previous-day price
GDP Growth Rate	Quarterly	Real Gross Domestic Product growth rate
CPI	Monthly	Consumer Price Index
PPI_ppiip	Monthly	Producer Price Index (final industrial product)
PPI_ppi	Monthly	Producer Price Index (factor)
PPI_qm	Monthly	Producer Price Index (quarry & mining)
PPI_rmi	Monthly	Producer Price Index (raw materials)
PPI_pi	Monthly	Producer Price Index (production)
PPI_cg	Monthly	Producer Price Index (consumer goods)
PPI_food	Monthly	Producer Price Index (food)
PPI_clothing	Monthly	Producer Price Index (clothing)
PPI_roeu	Monthly	Producer Price Index (regular commodities)
PPI_dcg	Monthly	Producer Price Index (durable consumer goods)
MS_m2	Monthly	Money Supply (M2)
MS_m2_yoy	Monthly	Money Supply (M2 year over year)
MS_m1	Monthly	Money Supply (M1)
MS_m1_yoy	Monthly	Money Supply (M1 year over year)
MS_m0	Monthly	Money Supply (M0)
MS_m0_yoy	Monthly	Money Supply (M0 year over year)
MS_cd	Monthly	Money Supply (current deposit)
MS_qm	Monthly	Money Supply (quasi-money)
MS_ftd	Monthly	Money Supply (fixed-time deposit)
MS_sd	Monthly	Money Supply (savings deposit)
MS_rests	Monthly	Money Supply (other deposit)
SP500_close	Daily	Standard & Poor 500 Index Price
US_Stock_CPI	Monthly	United States Consumer Index Price
Sentiment (1 for pos, -1 for neg)	Variable	Sentiment label of Trumps tweet (NOT A PERCENTAGE CHANGE)