CISC 6000: Deep Learning ProjectStock Price Prediction

Tianyi,Zhang Shuyan,Liu Yuqi,Fu

December 9, 2019

Abstract

Although stock market is considered chaotic, complex, stock price prediction is attractive in machine learning and deep learning tasks. According to sequence deep learning models such as RNN, GRU and LSTM risen in these years, we got a much more power tool in sovling predicting stock price. However, due to high complexity and high dimensionality in stock price data, denoising data becomes an signification part in stock price prediction task. Our work test several denosing methods such as wavelet transformation and apply PCA,2DPCA,2D2DPCA and Stacked Autoencoder(SAE) to reduce data dimensions. In empirical analysis, we found that 2D2DPCA with LSTM would have the best result which gets a R^2 to 0.9177. In additional, we choose random forest as baseline model, and our result shows 2D2DPCA with LSTM can perform a signification improvement over random forest.

1 Introduction

Stock price prediction has been considered as one of the most challenging tasks in the past. According to Efficient Market hypothesis, current information has an reflection on stock market but has no predictive power to future stock price changes. However, attraction of monetary returns leads lots of researchers devoting in this field. For instance, stock price researchers use fundamental and technical analysis for predictions. Fundamental analysis is a traditional approach by analyzing companies' parameters to predict the stock price. Onthe other hand, Professional Traders use price history as features to do the technical analysis.

Although many mathematical stock prediction models were developed, their accuracy are still dissatisfying. In past decades, Deep learning were taken attentions for its overwhelm performance over traditional machine learning methods, stock prediction also gets an opportunity having a breakthrough. Recurrent Neural Network (RNN) provides a method to fit time series data like stock prices. However, it faces gradient vanishing problem when using it in Long-Term Dependencies problem such as Language Modeling and Stock Price Prediction. To fix this problem, Long Short-Term Memory (LSTM) and Gated Recurrent

Units (GRU) model provide approaches getting a long memory in the model.In addition, Radial Basis Function Neural Network (RBFNN) with 2D PCA would have better performance than RNN.

Another aspect in stock price prediction is how to denoise orginal stock price features. One of methods is to use mathematics transformation such as wavelet transformation. Compared with fourier transform, wavelet transform has better performance for unsteady data such stock price time series. Another popular method is feature extracting. Since stock market often have enormous noises and complex dimensionality, feature extraction methods are commonly used in this field. In traditional methods, PCA has been proved high efficiency. Also since demands for dimension reduction for 2D array data such face recognition task. Furthermore, Stacked Autoencoder has been proved having great performance in compress and de-noise data.

2 Data Preparation

To complete this project, we used 5 datasets: IBM stock price dataset, US Dollar Index dataset, Oracle stock price dataset, Intel stock price dataset, Infosys stock price dataset. We mainly focus on IBM stock price dataset. We choose IBM daily stock price from 01/03/2000 to 12/31/2018. The IBM stock price data has features such as the date, the lowest trading price during the day, the closing bid price on a trading day, the closing price of a security at the end of a calendar period, the closing ask price on a trading day, the highest trading price during the day.

2.1 Feature Generation

Feature Engineer plays an important role in stock price prediction. In order to increase the accuracy of the result, we have to consider different factors may have an influence of the stock price. Take IBM as an example, investors, competitors, US Dollar currency, IBM's performance can be important factors affect daily stock price. Since our resource is limited, we are able to collect valid data of competitors' daily stock price and daily US dollar currency index in past 9 years to match IBM's dataset. We generated 30 new features based on our current datasets.

2.1.1 Daily Return

Based on daily closing price and open price, we generated daily return by using closing price/open price - 1. Daily return measures how much you gained or lost per day for a stock.

2.1.2 Differencing

It is important to know how a value is changing. We calculate the change in value since the prior period, which is based on daily stock price and daily volume of the stock, which are change of the price and volume since prior day and 50 days prior. Another way to measure the changing is the rate of change. We calculate the rate of change of price and volume that expressed by percent change. After we have the rate of change, we generate the changing sign of the price and volume.

2.1.3 Moving Averages

Moving average is a widely used indicator in technical analysis that helps smooth out price by filtering out the noise. It indicates the trend of the change of a value. We calculated moving average of price and log of volume in 5 days, 50 days and 200 days.

2.1.4 Percentile

Percentile transformation measures the percentile rank of traded volume for each value as compared to a trailing 200 day period.

2.1.5 Momentum

We calculate Money Flow Index, Relative Strength Index, True Strength index, Ultimate Os- cillator, Stochastic Oscillator, Williams R, Awesome Oscillator, Kaufman's Adaptive Moving Average of the daily stock price.Kaufman's Adaptive Moving Average(KAMA) is a moving average designed to account for market noise or volatility. KAMA is calculated by Efficiency Ratio(ER) and Smoothing Constant(SC). ER is the price changed adjusted for the daily volatility. SC uses the ER and two smoothing constants based on an exponential moving average. KAMA = prior LAMA + SC *(price-prior KAMA). KAMA is highly connected with stock price and its correlation of the stock price is 0.997. Williams R is a type of momentum indicator that moves between 0 and -100 and measures overbought and oversold levels. It tells a trader where the current price is relative to the highest high over the last14 periods. When this indicator is between -20 and zero the price is over bought. It is calculated by (highest high daily price - daily close price)/(highest high daily price - lowest low daily price).

2.1.6 US Dollar Index

Since IBM is a global company, We expect the change of US currency have a strong con-nection with the stock price. We collect daily US Dollar Index online in past 9 years to match IBM dataset. US Dollar Index is a measurement of the United States dollar relative to a basket of foreign currencies. The index increases when the US dollar gains value when compared to other currencies. We collect the price, open price, high price and low price of US Dollar Index, which allows us to generate 4 new features.

2.1.7 Competitors

As an information technology company, IBM has many competitors such as Intel, Oracle and infosys. A strong competitor may have a big influence on IBM's performance. We collect daily close price of these three competitors and generate 3 new features.

The dataset contains total 41 features which are generated above. Figure 1 is the descriptionof features of the current dataset.

Index number	Feature Name	Feature Description
0	'BIDLO',	The lowest daily price
1	'ASKHI'	The highest daily price
2	'PRC',	The closing price of a security
3	VOL'	Raw number of shared traded
4	RET'	The change in the total value of an investment
5	BID',	The closing bid price on a trading day
6	ASK'	The closing ask price on a trading day
7	'SHROUT'	The unadjusted number of publicly held shares
8	OPENPRC',	The first trade after a market opens
9	RETX	The capital appreciation of a security
10	daily_return',	Daily return of the stock
11	log_volume'	The log of the volume
12	changeVolume01'	The change in volume since 1 day prior
13	changePrice01'	The change in price since 1 day prior
14	changeVolume50'	The change in Volume since 50 days prior
15	changePrice50	The change in price since 50 days prior
16	MovAvofV5days'	Moving average of log of volume in 5 days
17	'Mov5Price	Moving average of price in 5 days
18	MovAvofV200days	Moving Average of log of volume in 200 days
19	'MovAvof50daysPrice	Moving average of price in 50 days
20	percentile	Percentile Ranke of traded volume
21	rateofchange	Rate of change of log of volume
22	sign',	Was volume increasing or decreasing today?
23	plus_minus	The number of up-days minus the number of down days
24	rateofchangePrice	Rate of change of the price
25	signprice	Was price increasing or decreasing today?
26	money_flow_index	Identifier of overbought or over sold conditions
27	AO	Awesome Oscillator
28	KAMA',	Kaufman's Adaptive Moving Average
29	rsi'	Relative Strength Index
30	stochastic',	Stochastic Oscillator
31	TSI'	True Strength Index
32	ultimate'	Ultimate Oscillator
33	williamsR	A momentum indicator
34	Price	DXY close price
35	Open	DXY open price
36	High	DXY high price
37	Low	DXY low price
38	PRC_intc	Daily stock price of Intel
39	PRC_orcl	Daily stock price of Oracle
40	PRC_infy	Daily stock price of Infosys

Figure 1: feature description

3 Methodology

To generate the deep stock price prediction model, this project presents a deep learning framework for financial time series using a deep learning-based forecasting scheme that integrates several data processing phases which includes feature seletcion, wavelet transform, PCA, 2DPCA, 2D2DPCA and stacked auto-encoders. The figures shows our frameworks. The frameworks consists four parts: (1) using filter method to do feature selection(2)using the wavelet transform decomposes the stock price time series to eliminate noise;(3) applica- tion of the stacked autoencoders to further clean data;(4) using feature extraction method to reduce dimensionity.(5) the use of Deep learning sequence models such as RNN,LSTM and GRU to generate the one-step-ahead output. Since data processing phase would be excessive, we use feature extraction as base-line and then try to add more method to test whether it has any improvement.

3.1 Feature Selection

Since the result is not desirable when we apply all features to the deep learning model, we decide to do the feature selection to determine the most relevant features to the target value.

3.1.1 Pearson Correlation Coefficient

To rank the importance of features, we calculate Pearson Correlation value of each feature with stock price. We visualize the correlation values in figure 3 below. The figure 4 below shows the correlation value of each feature. From the correlation result, we could know the price, moving average of price, moving average of log of volume, competitors' stock pricehave strong connections with IBM's daily stock price.

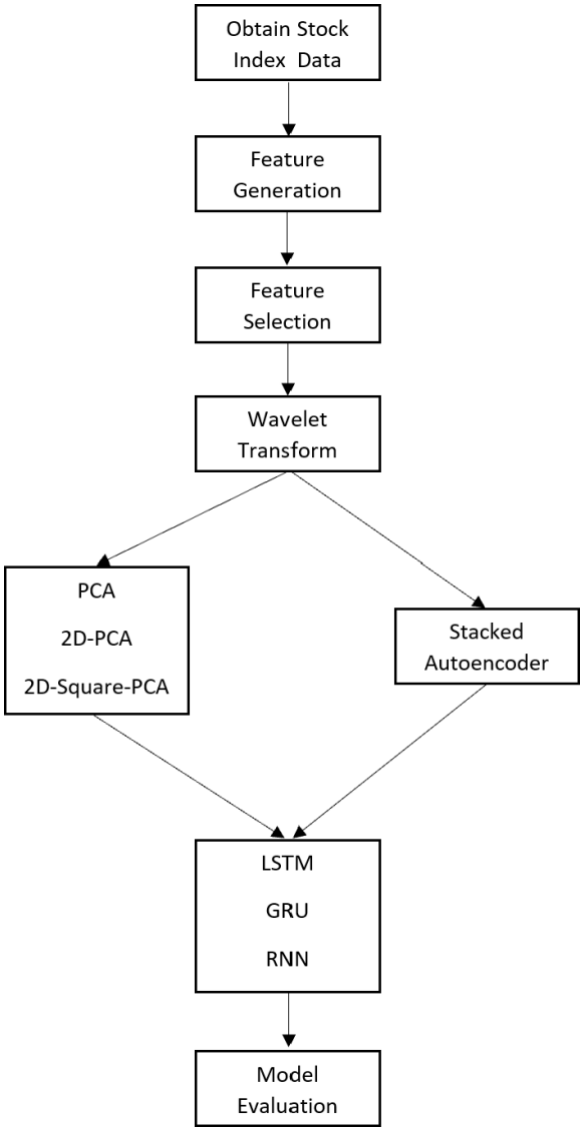

Figure 2: Workflow

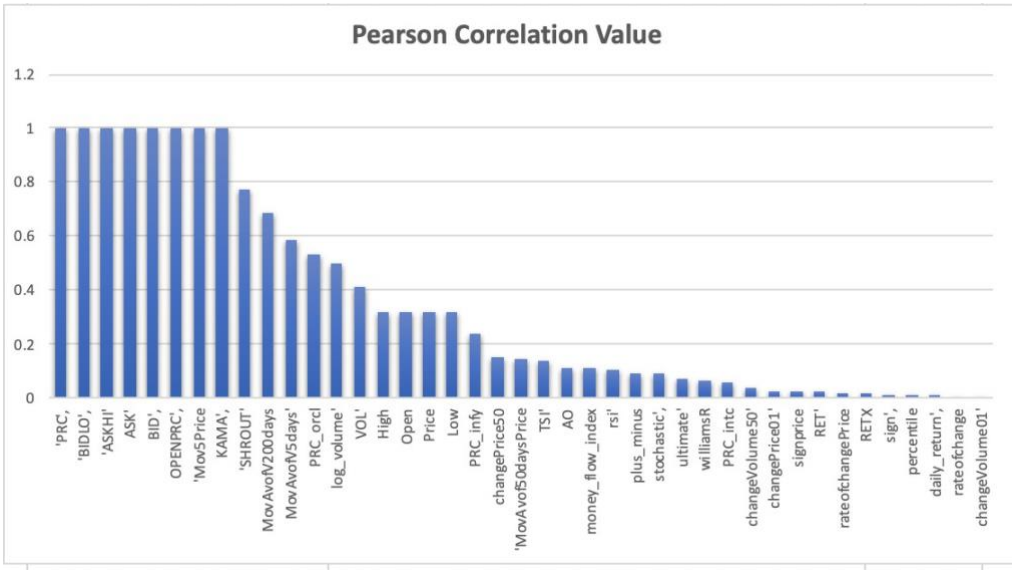

Figure 3: Pearson Correlation Histogram

Index number	Feature Name	Pearson Correlation Value
2	'PRC',	1
0	'BIDLO',	0.999629652
1	'ASKHI'	0.99962333
6	ASK'	0.999336422
5	BID',	0.999324336
8	OPENPRC',	0.999251863
17	'Mov5Price	0.998558994
28	KAMA',	0.99651043
7	'SHROUT'	0.772579414
18	MovAvofV200days	0.68326685
16	MovAvofV5days'	0.584247987
39	PRC_orcl	0.533620151
11	log_volume'	0.496594285
3	VOL'	0.413232948
36	High	0.320714533
35	Open	0.31890388
34	Price	0.318439801
37	Low	0.317239499
40	PRC_infy	0.235934054
15	changePrice50	0.148717003
19	'MovAvof50daysPrice	0.147680062
31	TSI'	0.138296587
27	AO	0.114203199
26	money_flow_index	0.108695098
29	rsi'	0.101821994
23	plus_minus	0.091435813
30	stochastic',	0.088677588
32	ultimate'	0.068641749
33	williamsR	0.065786819
38	PRC_intc	0.060469248
14	changeVolume50'	0.03659781
13	changePrice01'	0.024596306
25	signprice	0.024420221
4	RET'	0.021011073
24	rateofchangePrice	0.020103794
9	RETX	0.019749114
22	sign',	0.014199181
20	percentile	0.011133134
10	daily_return',	0.010996574
21	rateofchange	0.002064497
12	changeVolume01'	0.000880255

Figure 4: Pearson Correlation Result

3.1.2 Best combination of features

We use backward elimination of wrapper method to find the best combination of features. We start with all the features and removes the least significant feature at each iteration which improves the model. From the figure 5, the Mean Squared Error has a very big decline when we drop the 27 least significant features. Therefore, we keep 14 features in the end, which

are: Closing price of IBM, lowest price of IBM, highest price of IBM, closing ask price of IBM, closing bid price of IBM, open price of IBM, Moving average of price in 5 days, KAMA, number of held shares of IBM, Moving average of volume in 200 and 5 days, daily stock priceof Oracle, the log of volume and number of shares traded.

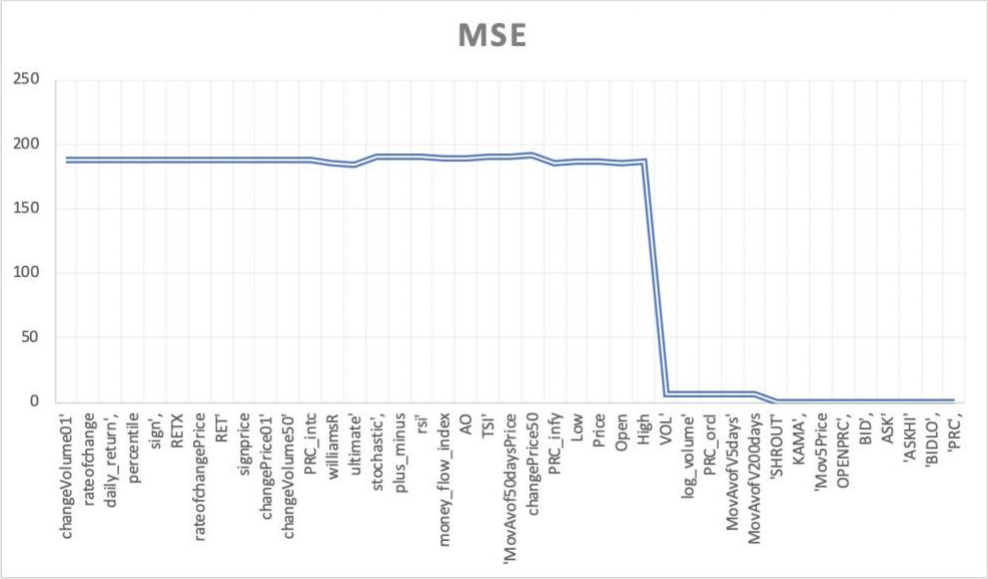

Figure 5: Features combination result

3.2 Discrete Wavelet Transform

Due to the complexity of the stock market dynamics, stock price data is often filled withnoise that might distract the machine learning algorithm from learning the trend and struc- ture. Hence, it is in our interest to remove some of the noise, while preserving the trends and structure in the data. We thought Wavelet Transforms may be a good choice to preserve the time factor of the data.

Wavelet analysis helps to analyse localized variations of signal within a time series. Both the dominant modes of variability and their variation time can be captured by decomposing a time series into time-scale or time-frequency space.Discrete Wavelet Transform (DWT) can decompose the signal in both time and frequency domain simultaneously.

The decomposition of signal using DWT is shown in Figure 1. DWT operates on the prin- ciple of convolution. Various kinds of wavelets are available, for instance, Haar, Daubechies, Morlet and Mexican Hat. Morlet and Daubechies wavelets have applications in image pro- cessing and are subject to aliasing problem. Mexican Hat wavelets are expensive to compute. Haar wavelets are advantageous for time series analysis since they are capable of capturing

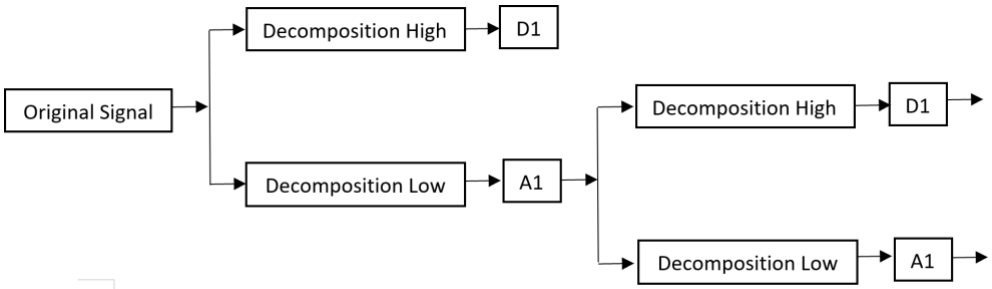

Figure 6: Decomposition of Signal using

DWTfluctuations between adjacent observations.

3.3 Principal Component Analysis

principal components analysis (PCA) can be used to simplify a data. We find that the eigen- vectors of the covariance matrix with the largest eigenvalues correspond to the dimensions that have the strongest correlation in the data itself. Then we can only use first significationn compositions to represents whole data. Although PCA perform well for one-dimension data, it spend lots of time for image data. 2DPCA and 2D2DPCA are designed for dimen- sion reduction for 2dimension images, but they also show great performance in time-series data such as stock price.

3.4 2D Principal Component Analysis

By using 2DPCA, the image matrix does not need to be previously transformed into a vector. Therefore 2DPCA can construct the covariance matrix accurately and spend less timne to determine the corresponding eigenvectors. When using PCA in stock price, we firstly need to flatten n window size feature and then apply PCA transformation. However, in 2DPCA transformation, we define each input is a matrix like an image, then 2DPCA can be applied in stock series. For examples, we have 60 features and decide window size as 60, our input data will be (60,60). In 2DPCA method, we can decide to choose 20 top eigenvectors and then input data would be reduced to (60,20)

3.5 2-Directional 2-Dimensional PCA

2-Directional 2-Dimensional PCA (2D2DPCA) is an improve of 2DPCA. In previous 2DPCA, what we consider are eigenvectors in one direction. However, in 2D2DPCA it calculation top n signification eigenvectors in both rows and cols. In 2D2DPCA, we need to decided top n eigenvectors horizontally and top m eignvectors vertically. Therefore, input data will be

transformed to (n,m) dimensions. Figures show how 2D2DPCA applied on face recognition. As we can see, using 2D2DPCA, we can reduce data dimension from (218,178) to (50,50) without loss lots of information.

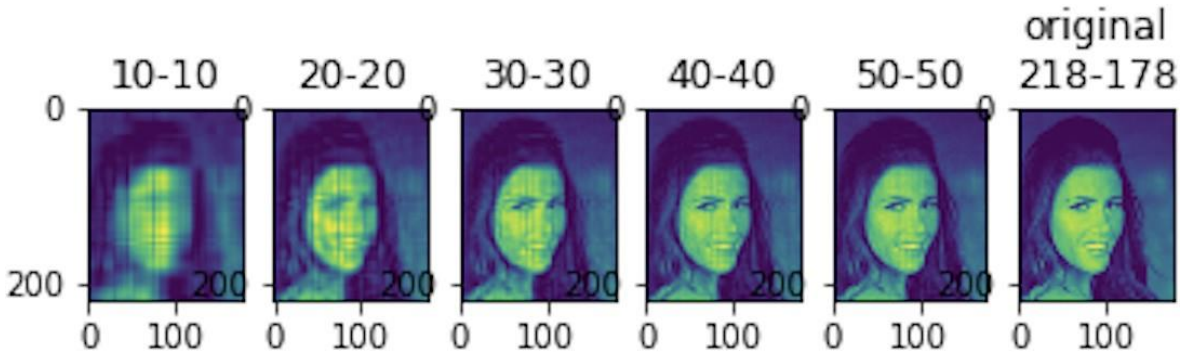

Figure 7: 2D2DPCA

3.6 Stacked Auto-encoders for Compression

Autoencoders map the data they are fed to a lower dimensional space by combining thedata's most important features. It encodes the original data into a more compact represen- tation. It also decides how the data is combined, hence the auto in Autoencoder. We referto these encoded features as latent variables.

4 Empirical Analysis

4.1 Type of Wavelet Filter and Level of Decomposition

MODWT is used to decompose the stock price to overcome the limitation of DWT (refer to Section 2.1). The type ofwavelet filter chosen here is "Haar". Haar wavelet helps to capture fluctuations between adjacent observations and is known as differencing filter. Italso eliminates the problem of aliasing. The level of decomposition to be selecteddepends on the number of data points available. When the level of decomposition increases, the datawill be smoothened to a larger extent, thus, leading to loss of information. Hence, the level of decomposition chosen here is 9.

The process is as follows:

1. The data is transformed using Wavelet transform.

2. Coefficients that more than a full standard deviation away are removed (out of all the coefficients)

3. Inverse transform the new coefficients to get the denoised data.

Here,we obtained nine wavelet co-efficients and one scaling co-efficient.The MODWT de-composition of the series is shown in Figure 2 and Figure 3. In Figure 2, the first row represents the original time series and the second row represents the scaling co-efficient. This is the smoothed component whose pattern is quite similar to the original series. This represents the trend of the movement of the stock index. In Figure 3, rows from 1 to 9 represent the wavelet coefficients, respectively. row 1 is the high frequency component.

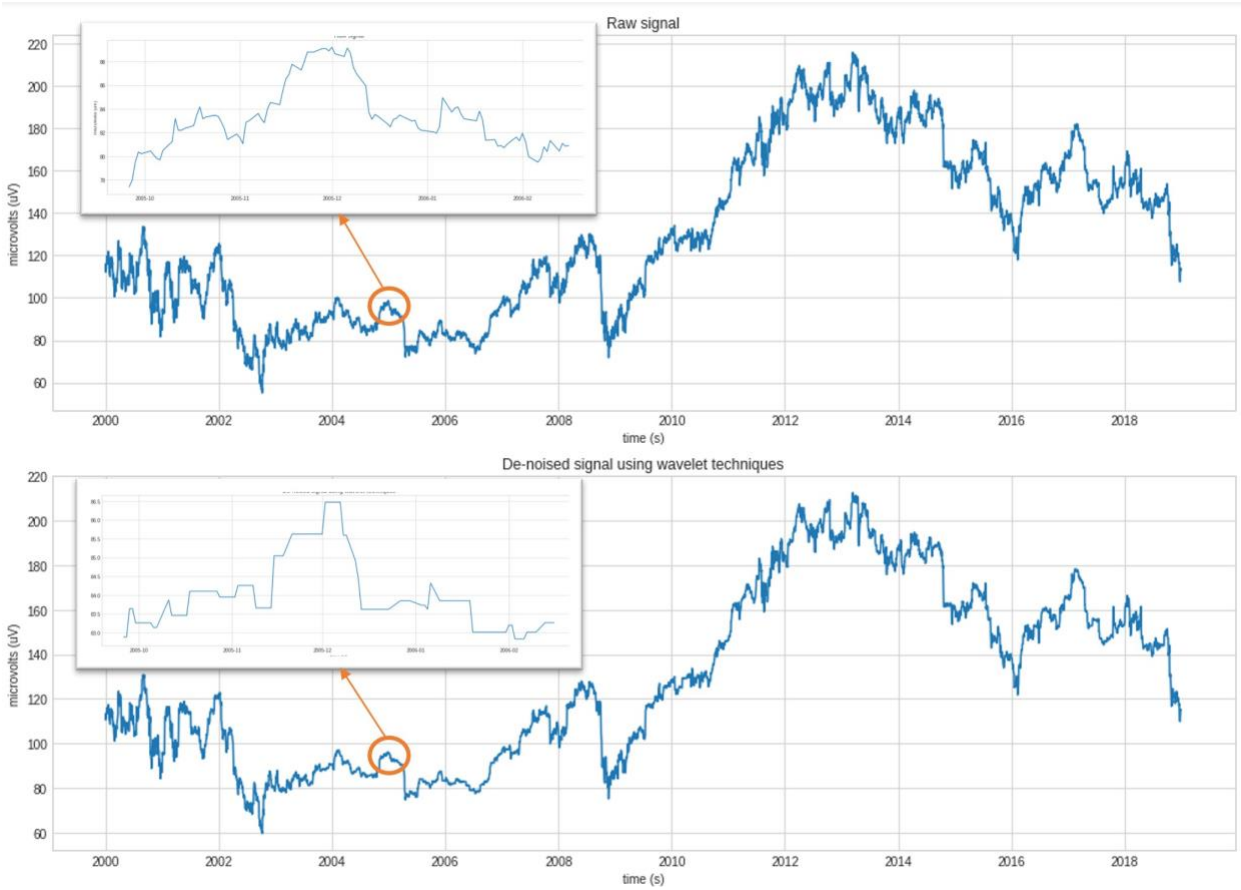

Figure 8: Wavelets of Stock Index

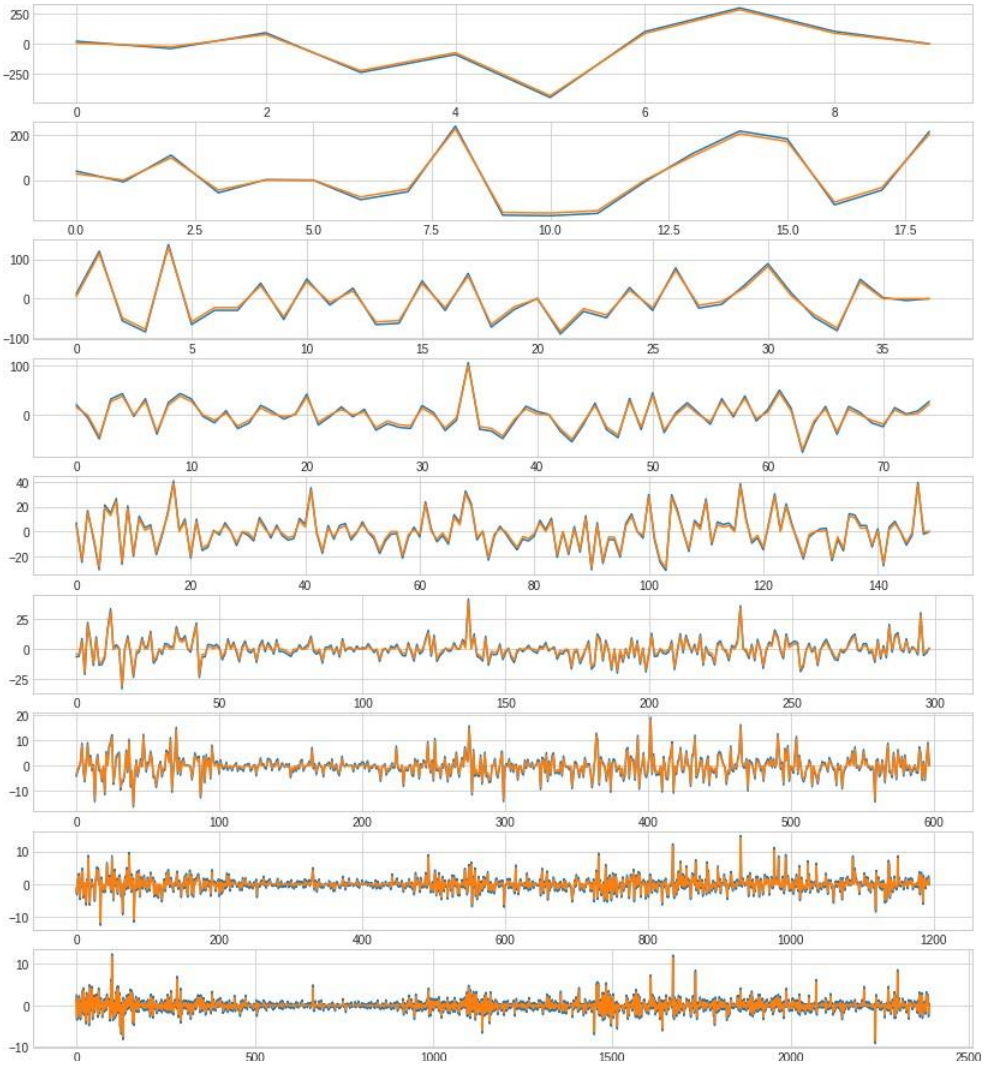

Figure 9: Wavelets Coefficients of Stock Index

4.2 Type of Auto-encoder to reduce the dimensionality

In this project, we explored the potential of 4 different types of autoencoders to capture the dynamic information of stock market prices in a lower and traceable dimension space.

1st model: a simple multi-layer percepetron (MLP) autoencoder

The model used is super simple but the comparison between the input and the output reveal the ability of the network to abstract few important features such as peaks and lows. Interestingly, we can see that the some of the outputs are almost identical between each other even though the inputs are reasonably different.

2nd model: stacked deep autoencoder

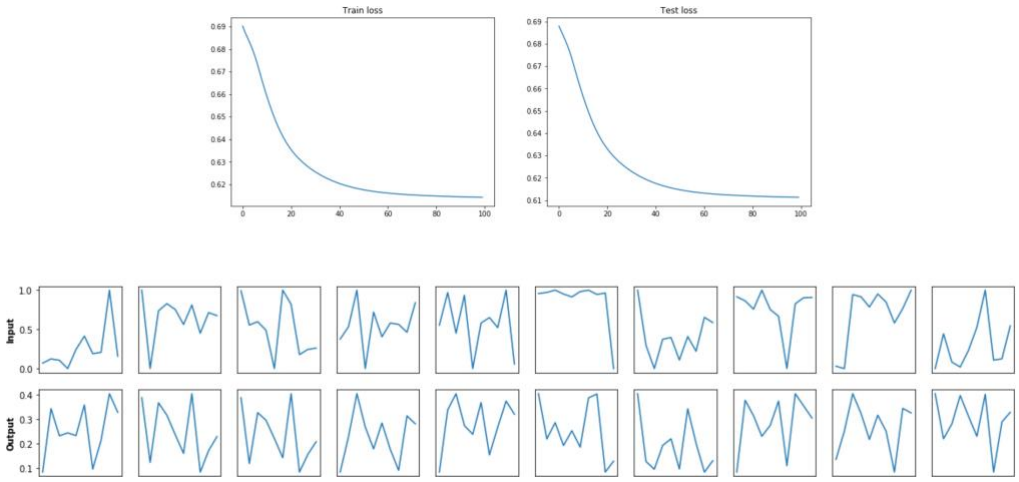

Figure 10: simple multi-layer percepetron

In essence, stacked deep autoencoders get very good at compressing data and reproduc-ing it back again. What we are interested in is the compression part, as it means the information required to reproduce the data is in some way encoded in the compressed form. This suggests that these compressed data can be in some way the features of the data thatwe are trying to extract features out from.

3rd model: 1D convolutional autoencoder

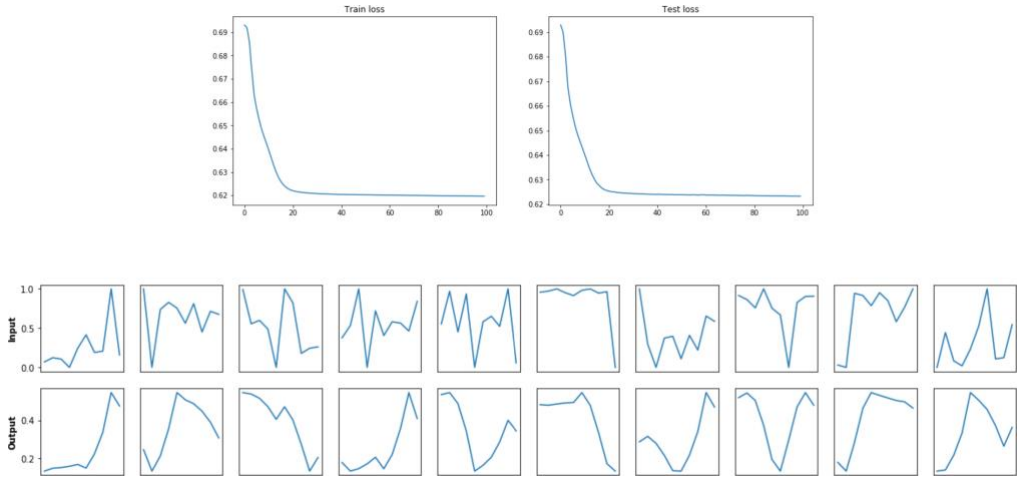

Figure 11: deep autoencoder

In this model we are using convolutions with kernel size of 3 and the idea is that these convolutions should look at patterns occurring in groups of 3 returns. However, the input/output examples show a different type of plots, the majority of them containing onesingle low or high peak unlike the previous results, which were much more variant in themiddle range.

4th model: LSTM autoencoder

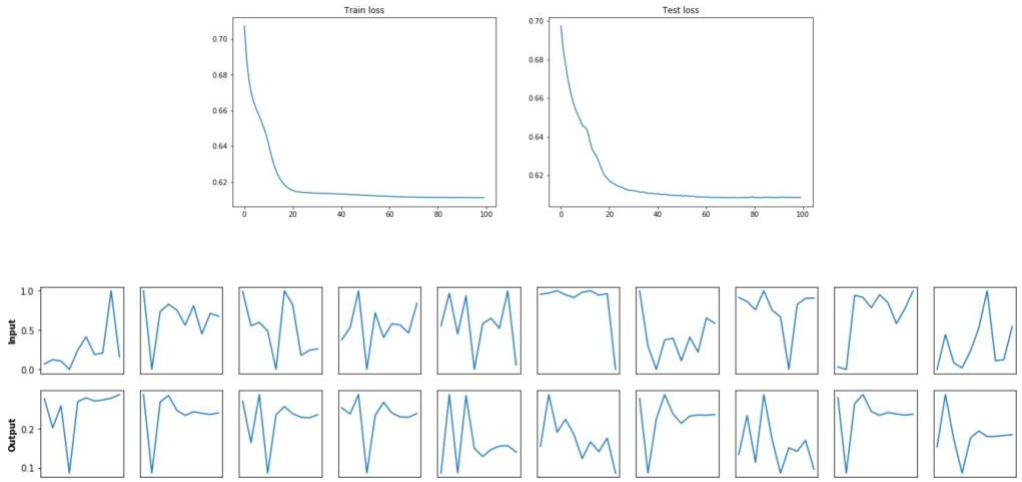

Figure 12: 1D convolutional autoencoder

While recurrent neural networks such as Long-Short Term Memory (LSTM) models are particularly suitable to tackle time series, we can see that their performance as autoencodersis very poor.

In conclusion, we decide use stacked deep autoencoder to reduce the dimensionality of stock index time series. After training for 300 epochs, the loss decreased to around 0.002. Then, we used stacked deep autoencoder model to encode the stock index data into features.

5 Results and Conclusions

5.1 Error Measure

In the experiment, the performance of the stock prediction models were measured by three indicators: Mean Square Error (MSE), Mean Absolute Error (MAE) and R Square (R2). MSE is one of the most common indicators for comparing prediction accuracy. As shown in Figure 8,we compared the errors of three prediction algorithms: SimpleRNN, LSTM and

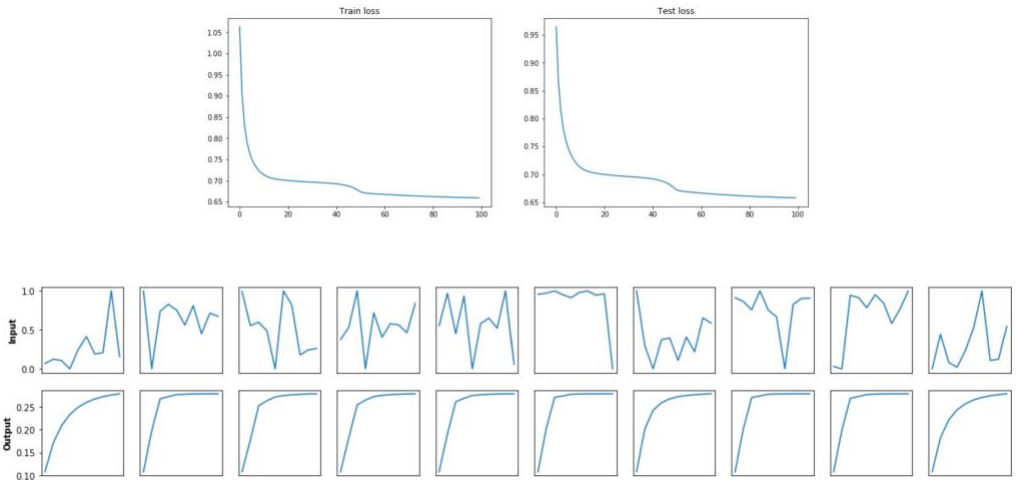

Figure 13: LSTM autoencoder

GRU. We compared the predict performance with five feature extraction methods: Baseline, PCA, 2D-PCA, 2D-Sqaure-PCA and Stacked Autoencoder.

5.2 Experimental feature selection and exaction

After parameter tuning process, we found that best output dimension for PCA,2DPCA and 2D2DPCA are 200,11 and (16,10) respectively. For SAE, we use four shallow encoder-encoder structure and decide output dimension as 100.
As shown in Figure 9, we plotted 15 figures. We compared the errors of three prediction algorithms: SimpleRNN, LSTM and GRU. We compared the predict performance with five feature extraction methods: Baseline, PCA, 2D-PCA, 2D-Sqaure-PCA and Stacked Autoencoder. We build a baseline model only use recurrent neural network phase without feature selection phase and feature extraction phase. We predicted the stock for two years, thus there are 944 data points.The vertical axis represents the stock price of IBM company. In fact, the stock price pattern is similar. As shown in Figure 5,2D-Square-PCA feature extraction with LSTM had better prediction performance most of the time and matched well with the test stock price values. The predictive performance of 2D-Square-PCA feature extraction proposed in was better than that of PCA, 2D-PCA and SAE. Despite the prediction performances of the other three methods are not good as 2D-Sqaure-PCA, the overall prediction performances have significant improvement over the baseline model. So this is a strong proof proves the feature selection and extraction phase are effective.

5.3 Conclusion

To test models' performance ,we chose random forest which is a traditional machine learning representative as our baseline model. As shown in table 2, the RSquare of random forest is

Table 1: Evaluation result

Algorithm	Methods	MSE	MAE	RSquare
RNN	None	0.0197	0.1218	0.3199
	PCA	0.0029	0.0423	0.8998
	2DPCA	0.0169	0.1084	0.4153
	2D2DPCA	0.0052	0.0585	0.8208
	SAE	0.0062	0.0612	0.7863
GRU	None	0.0188	0.1143	0.3495
	PCA	0.0025	0.0405	0.9144
	2DPCA	0.0039	0.0475	0.8657
	2D2DPCA	0.0026	0.0399	0.9094
	SAE	0.0059	0.061	0.7944
LSTM	None	0.0209	0.1171	0.2783
	PCA	0.0031	0.0456	0.8938
	2DPCA	0.0055	0.0575	0.8086
	2D2DPCA	0.0024	0.0367	0.9177
	SAE	0.0058	0.0582	0.8005

0.7723

As our research found that random forest performance strongly well in stock prediction.

Table 2: Evaluation result compared to RandomForest

Algorithm	MSE	MAE	RSquare
PCA-GRU	0.0025	0.0405	0.9144
2D2DPCA-GRU	0.0026	0.0399	0.9094
2D2DPCA-LSTM	0.0024	0.0367	0.9177
RandomForest	0.0066	0.0606	0.7723

According our above experimental feature selection error measure matrix result shows that PCA-GRU,2D2DPCA-GRU and 2D2DPCA-LSTM are best three models. So we compared these three models with baseline random forest model.Table 2 shows that all these threemodel is taken a better performance.

For SAE method to data de-noising, although it is not as good as PCA method, it can get a better results compared with traditional RandomForest. Also, performance of SAE highly depends on hyper-parameter choices, with more experiment, its result would be better.

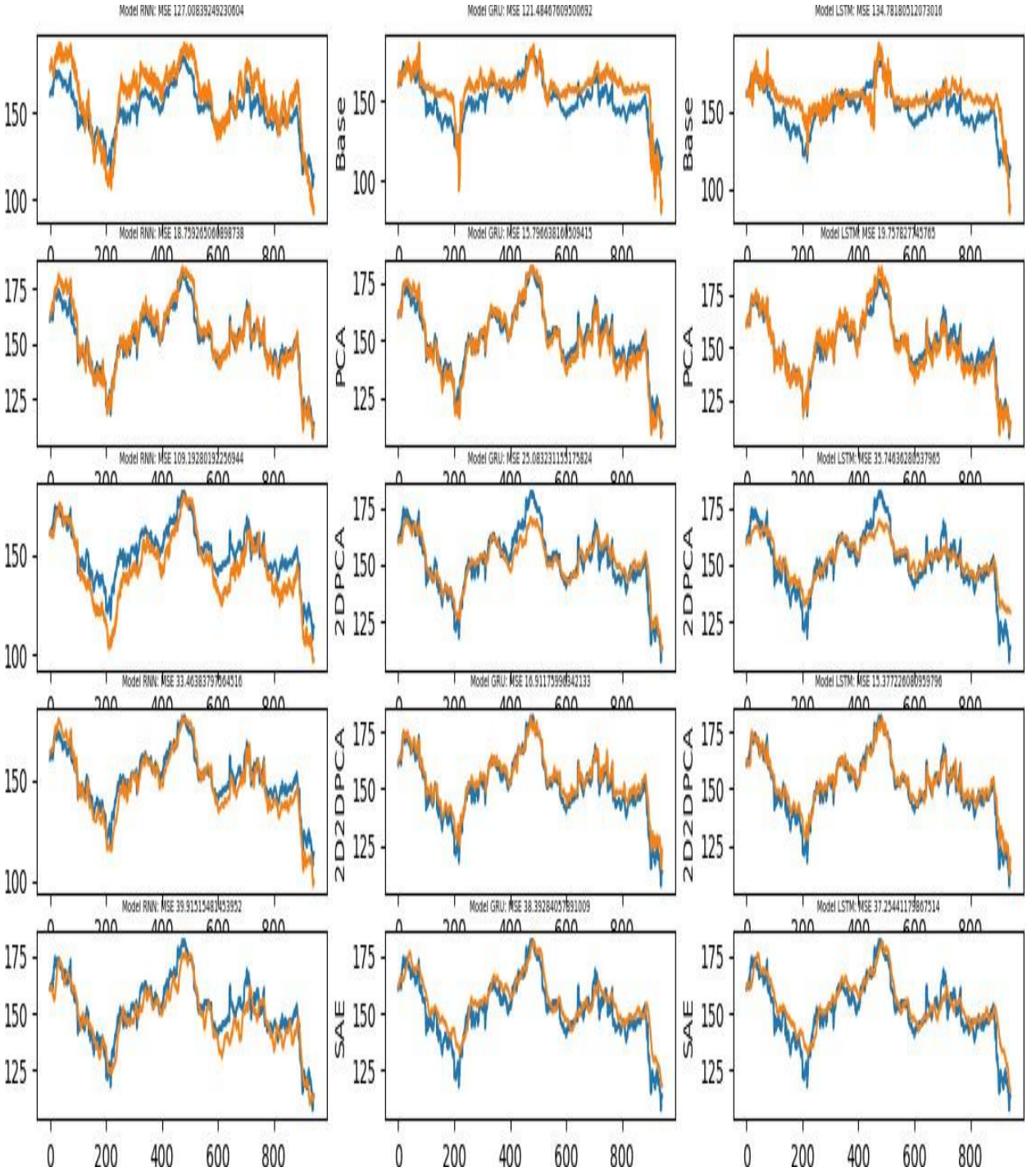

Figure 14: Error Measure Result