Prediction of Stock Prices
(A Machine Learning Approach)

Kirtika Gupta[#1], Paras Goyal[#2], K C Tripathi[#3], M L Sharma[#4]
[#]Department of Information Technology, Maharaja Agrasen Institute of Technology, Delhi- India

Abstract: We employ both random forests and LSTM networks and In this paper, we propose a novel way to minimize the risk of investment in stock market by predicting the returns of a stock using a class of powerful machine learning algorithms known as ensemble learning. Some of the technical indicators such as Relative Strength Index (RSI), stochastic oscillator etc. are used as inputs to train our mode as training methodologies to analyze their effectiveness in forecasting out-of-sample directional movements of constituent stocks of the S&P 500 from January1993 till December 2018 for intraday trading. We introduce a multi-feature setting consisting not only of the returns with respect to the closing prices, but also with respect to the opening prices and intradayreturns. As trading strategy, we use Krauss et al. (2017) and Fischer & Krauss (2018) as benchmark and, on each trading day, buy the 10 stocks with the highest probability and sell short the 10 stocks with the lowestprobability to outperform the market in terms of intraday returns – allwith equal monetary weight. Our empirical results show that themulti-feature setting provides a daily return, prior to transaction costs, of 0.64% using LSTM networks, and 0.54% using random forests. Hencewe outperform the single-feature setting in Fischer & Krauss (2018) and Krauss et al. (2017) consisting only of the daily returns with respect to the closing prices, having corresponding daily returns of 0.41% and of the 0.39% with respect to LSTM and random forests, respectively.

Keywords: Random Forest, Machine Learning, Forecasting

1. Introduction

In the last decade, machine learning methods have exhibited distinguished development in financial time series prediction. Huck (2009) and Huck (2010) construct statistical arbitrage strategies using Elman neural networks and a multi-criteria-decision method. Takeuchi & Lee (2013) evolve a momentum trading strategy. Moritz & Zimmermann (2014) apply random forests to construct a trading decision. Tran et al (2018), and Sezer & Ozbayoglu (2018) use neural networks for predictingtime series data. Borovykh et al. (2018) and Xue et al. (2018) employ convolutional neural networks, and Siami-Namini & Namin (2018) use long short-term memory networks (LSTM).

In my work, I use the results in Krauss et al. (2017) and Fischer and Krauss (2018) as benchmark.I introduce a multi feature setting consisting not only of the returns with respect to the closing prices, but also with respect to the opening prices and intraday returns to predict for each stock, at the beginning of each day, the probability to outperform the market in terms of intraday returns. As a data set I use all stocks of the S&P 500 from the period of January 2005 until December 2018. I employ both random

forests on the one hand and LSTM networks (more precisely CuDNNLSTM) on the other hand as training methodology and apply the same trading strategy as in Krauss et al. (2017) and Fischer & Krauss (2018). My empirical results show that the multi-feature setting provides a daily return, prior to transaction costs, of 0.64% for the LSTM network, and 0.54% for the random forest, hence outperforming the single-feature setting in Fischer & Krauss (2018) and Krauss et al. (2017), having corresponding daily returns of 0.41% and of 0.39%, respectively.

Data Source: Yahoo Finance

2. Related Work

The use of prediction algorithms to determine future trends in stock market prices contradict a basic rule in finance known as the Efficient Market Hypothesis (Fama and Malkiel (1970)). It states that current stockprices fully reflect all the relevant information. It implies that if someone were to gain an advantage by analyzing historical stock data, then the entire market will become aware of this advantage and as a result, the price of the share will be corrected. This is a highly controversial and often disputed theory. Although it is generally accepted, there are many researchers who have rejected this theory by using algorithms that can model more complex dynamics of the financial system.

Several algorithms have been used in stock prediction such as SVM, Neural Network, Linear Discriminant Analysis, Linear Regression, KNN and Naive Bayesian Classifier. Literature survey revealed that SVM has been used most of the time in stock prediction research.

Multiple algorithms were chosen to train the prediction system. These algorithms are Logistic Regression, Quadratic Discriminant Analysis, and SVM. These algorithms were applied to next day model which predicted the outcome of the stock price on the next day and long term model, which predicted the outcome of the stock price for the next n days. The next day prediction model produced accuracy results ranging from 44.52% to 58.2%. Dai and Zhang (2013) have justified their results by stating that US stock market is semi-strong efficient, meaning that neither fundamental nor technical analysis can be used to achieve superior gain. However, the long-term prediction model produced better results which peaked when the time window was 44. SVM reported the highest accuracy of 79.3%. In Xinjie (2014), the authors have used 3 stocks (AAPL, MSFT, AMZN) that have time span available from 2010-01-04to 2014-12-10. Various technical indicators such as RSI, on balance Volume, Williams %R etc. are used as features. Out of 84 features, an extremely randomized tree algorithm was implemented as described in Geurts and Louppe (2011), for the selection of the most relevant features. These features were then fed to an rbf Kernelized SVM for training. Devi, Bhaskaran and Kumar (2015) has proposed a model which uses hybrid cuckoo search with support vector machine. The literature survey helps usconclude that Ensemble learning algorithms have remained unexploited in the problem of stock market prediction. We will be using an ensemble learning method known as Random Forest to build our predictive model. Random forest is a multitude of decision 2 May 3, 2018 Applied Mathematical Finance main trees whose output is the mode of the outputsfrom the individual trees.

3. METHODOLOGY

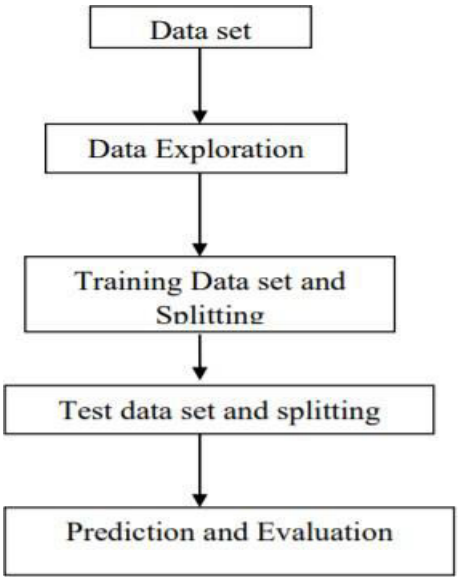

Fig. 1: Methodology

Our methodology is composed of five steps. In the first step, we divide our raw data into study periods, where each study period is divided into a training part (for in-sample-trading), and a trading part (for out-of-samplepredictions). In the second step, we introduce our features, whereas in thethird step we set up our targets. In the fourth step, we define the setup of our two machine learning methods we employ, namely random forest andCuDNNLSTM.
Finally, in the fifth step, we establish a trading strategy for the trading part.

Dataset creation with non-overlapping testing period.

We follow the procedure of Krauss and divide the dataset consisting of 29 years starting from January 1990 till December 2018, using a 4-year window and 1-year stride, where each study period is divided into a training period of approximately 756 days (≈ 3 years) and a trading period of approximately 252 days (≈ 1 year). As a consequence, we obtain 26 study periods with non-overlapping trading part.

Features selection

Technical Indicators are important parameters that are calculated from time series stock data that aim to forecast financial market direction. They are tools which are widely used by investors to check for bearish or bullish signals. Thetechnical indicators which we have used are listed below:

3.1 Relative Strength Index

The formula for calculating RSI is:

$RSI = 100 - (100/ (1 + RS))$

RS = Average Gain Over past 14 days Average Loss Over past 14 days RSI is a popular momentum indicator which determines whether the stock is overbought or oversold. A stock is said to be overbought when the demandunjustifiably pushes the price upwards.
This condition is generally interpreted as a sign that the stock is overvalued and the price is likely to go

down. A stock is said to be oversold when the price goes down sharply to a level below its true value. This is a result causeddue to panic selling. RSI ranges from 0 to 100 and generally, when RSI is above 70, it may indicate that the stock is overbought and when RSI is below 30, it may indicate the stock is oversold.

3.2 Stochastic Oscillator

The formula for calculating Stochastic Oscillator is
$\%K = 100 * ((C - L14) / (H14 - L14))$
Where, C = Current Closing Price
L14 = Lowest Low over the past 14 days H14 = Highest High over the past 14 days

Stochastic Oscillator follows the speed or the momentum of the price. As a rule, momentum changes before the price changes. It measures the level of the closing price relative to low-high range over a period of time.

3.3 Moving Average Convergence Divergence

$MACD = EMA12(C) - EMA26(C)$
$SignalLine = EMA9\ (MACD)\ (9)$

Where,
MACD = Moving Average Convergence DivergenceC = Closing Price series
EMAn = n day Exponential Moving Average

EMA stands for Exponential Moving Average.
When the MACD goes below the SingalLine, it indicates a sell signal. When itgoes above the SignalLine, it indicates a buy signal.

Target selection
Model training specification
Model specification for Random forest

- Number of decision trees in the forest = 1000
- Maximum depth of each tree4 = 10

4. Result and Analysis

The empirical results show that our multi-feature setting consisting not only of the returns with respect to the closing prices, but also with respect to the opening prices and intraday returns, outperforms the single feature setting of Krauss et al. (2017) and Fischer & Krauss (2018), both with respect to random forests and LSTM. We refer to ”IntraDay” for our setting and ”NextDay” for thesetting in Krauss et al. (2017) and Fischer & Krauss (2018) in Tables 1–3.

Metric	3-Feature IntraDay LSTM	3-Feature IntraDay RF	1-Feature NextDay LSTM	1-Feature NextDay RF	1-Feature IntraDay LSTM	1-Feature IntraDay RF
Time per epoch (in sec)	33.1	-	166	-	13.8	-
Training time (in min)	24.21	7.21	112.3	2.59	10.4	2.56
Decision making time (in sec)	0.086924	0.419563	0.180778	0.380040	0.036128	0.374121

Table 1: Time comparison

Metric	3-Feature IntraDay LSTM	3-Feature IntraDay RF	1-Feature NextDay LSTM	1-Feature NextDay RF	1-Feature IntraDay LSTM	1-Feature IntraDay RF	SP500 Index
Mean (long)	0.00332	0.00273	0.00257	0.00259	0.00094	0.00104	0.00033
Mean (short)	0.00312	0.00266	0.00158	0.00130	0.00180	0.00187	0.00000
Mean return	0.00644	0.00539	0.00414	0.00389	0.00274	0.00290	0.00033
Standard error	0.00019	0.00020	0.00024	0.00023	0.00021	0.00021	0.00014
Minimum	-0.1464	-0.1046	-0.1713	-0.1342	-0.1565	-0.1487	-0.0903
Quartile 1	-0.0017	-0.0028	-0.0052	-0.0051	-0.0054	-0.0050	-0.0044
Median	0.00559	0.00462	0.00352	0.00287	0.00242	0.00221	0.00056
Quartile 3	0.01433	0.01306	0.01294	0.01161	0.01086	0.01036	0.00560
Maximum	0.14101	0.14153	0.19884	0.28139	0.13896	0.16064	0.11580
Share > 0	0.69663	0.65857	0.60598	0.59479	0.58405	0.58937	0.53681
Std. deviation	0.01572	0.01597	0.01961	0.01831	0.01713	0.01683	0.01133
Skewness	0.15599	0.28900	0.36822	1.41199	-0.1828	0.12051	-0.1007
Kurtosis	9.71987	8.32627	10.8793	19.8349	10.1893	11.7758	11.9396
1-percent VaR	-0.0352	-0.0364	-0.0492	-0.0432	-0.0461	-0.0448	-0.0313
1-percent CVaR	-0.0519	-0.0528	-0.0712	-0.0592	-0.0678	-0.0660	-0.0451
5-percent VaR	-0.0157	-0.0170	-0.0234	-0.0208	-0.0214	-0.0197	-0.0177
5-percent CVaR	-0.0284	-0.0297	-0.0401	-0.0345	-0.0377	-0.0357	-0.0270
Max. drawdown	0.22345	0.19779	0.42551	0.23155	0.35645	0.43885	0.56775
Avg return p.a.	3.84750	2.75103	1.68883	1.53806	0.91483	1.00281	0.06975
Std dev. p.a.	0.24957	0.25358	0.31135	0.29071	0.27193	0.26719	0.17990
Down dev. p.a.	0.17144	0.17301	0.21204	0.18690	0.19270	0.18530	0.12970
Sharpe ratio	6.34253	5.20303	3.22732	3.23339	2.39560	2.59188	0.24867
Sortino ratio	62.7403	49.6764	27.8835	30.0753	19.0217	21.2964	1.77234

Table 2: Average performance metrics of the simulations before transaction cost

Metric	3-Feature IntraDay LSTM	3-Feature IntraDay RF	1-Feature NextDay LSTM	1-Feature NextDay RF	1-Feature IntraDay LSTM	1-Feature IntraDay RF	SP500 Index
Mean (long)	0.00232	0.00173	0.00157	0.00159	-0.0000	0.00004	0.00033
Mean (short)	0.00212	0.00166	0.00058	0.00030	0.00080	0.00087	0.00000
Mean return	0.00444	0.00339	0.00214	0.00189	0.00074	0.00090	0.00033
Standard error	0.00019	0.00020	0.00024	0.00023	0.00021	0.00021	0.00014
Minimum	-0.1484	-0.1066	-0.1733	-0.1362	-0.1585	-0.1507	-0.0903
Quartile 1	-0.0037	-0.0048	-0.0072	-0.0071	-0.0074	-0.0070	-0.0044
Median	0.00359	0.00262	0.00152	0.00087	0.00042	0.00021	0.00056
Quartile 3	0.01233	0.01106	0.01094	0.00961	0.00886	0.00836	0.00560
Maximum	0.13901	0.13953	0.19684	0.27939	0.13696	0.15864	0.11580
Share > 0	0.63129	0.59319	0.54279	0.53006	0.51534	0.50810	0.53681
Std. deviation	0.01572	0.01597	0.01961	0.01831	0.01713	0.01683	0.01133
Skewness	0.15599	0.28900	0.36822	1.41199	-0.1828	0.12051	-0.1007
Kurtosis	9.71987	8.32627	10.8793	19.8349	10.1893	11.7758	11.9396
1-percent VaR	-0.0372	-0.0384	-0.0512	-0.0452	-0.0481	-0.0468	-0.0313
1-percent CVaR	-0.0539	-0.0548	-0.0732	-0.0612	-0.0698	-0.0680	-0.0451
5-percent VaR	-0.0177	-0.0190	-0.0254	-0.0228	-0.0234	-0.0217	-0.0177
5-percent CVaR	-0.0304	-0.0317	-0.0421	-0.0365	-0.0397	-0.0377	-0.0270
Max. drawdown	0.39227	0.40139	0.87232	0.92312	0.97263	0.87123	0.56775
Avg return p.a.	1.94325	1.27179	0.63060	0.53901	0.16046	0.21146	0.06975
Std dev. p.a.	0.24957	0.25358	0.31135	0.29071	0.27193	0.26719	0.17990
Down dev. p.a.	0.17144	0.17301	0.21204	0.18690	0.19270	0.18530	0.12970
Sharpe ratio	4.32307	3.21553	1.60856	1.49969	0.54218	0.70557	0.24867
Sortino ratio	39.0873	27.9810	12.9274	12.7950	3.98288	5.34773	1.77234

Table 3: Average performance metrics of the simulations after transaction cost

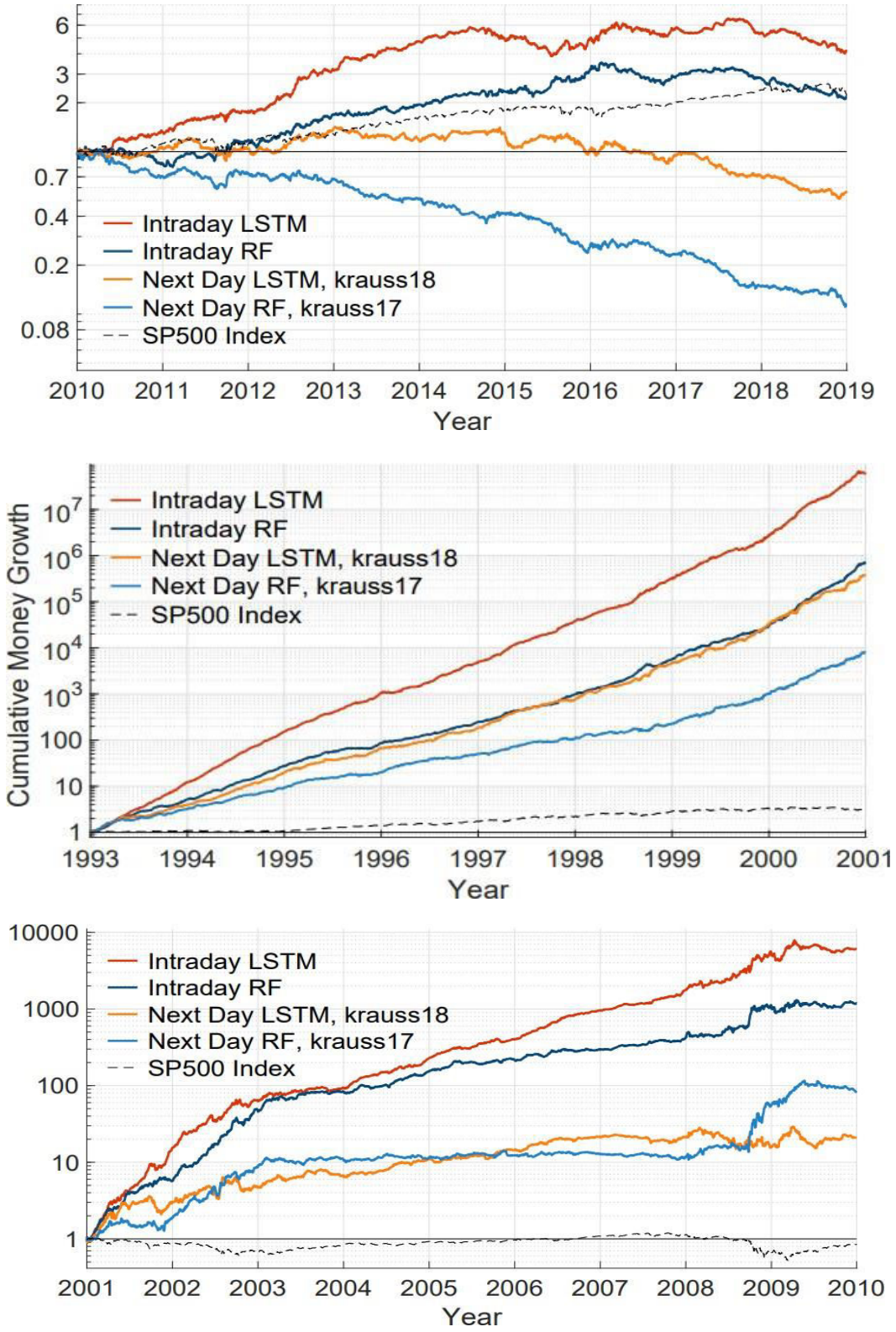

Fig. 2: Cumulative Money Growth with Initial Investment, after DeductingTransaction Costs

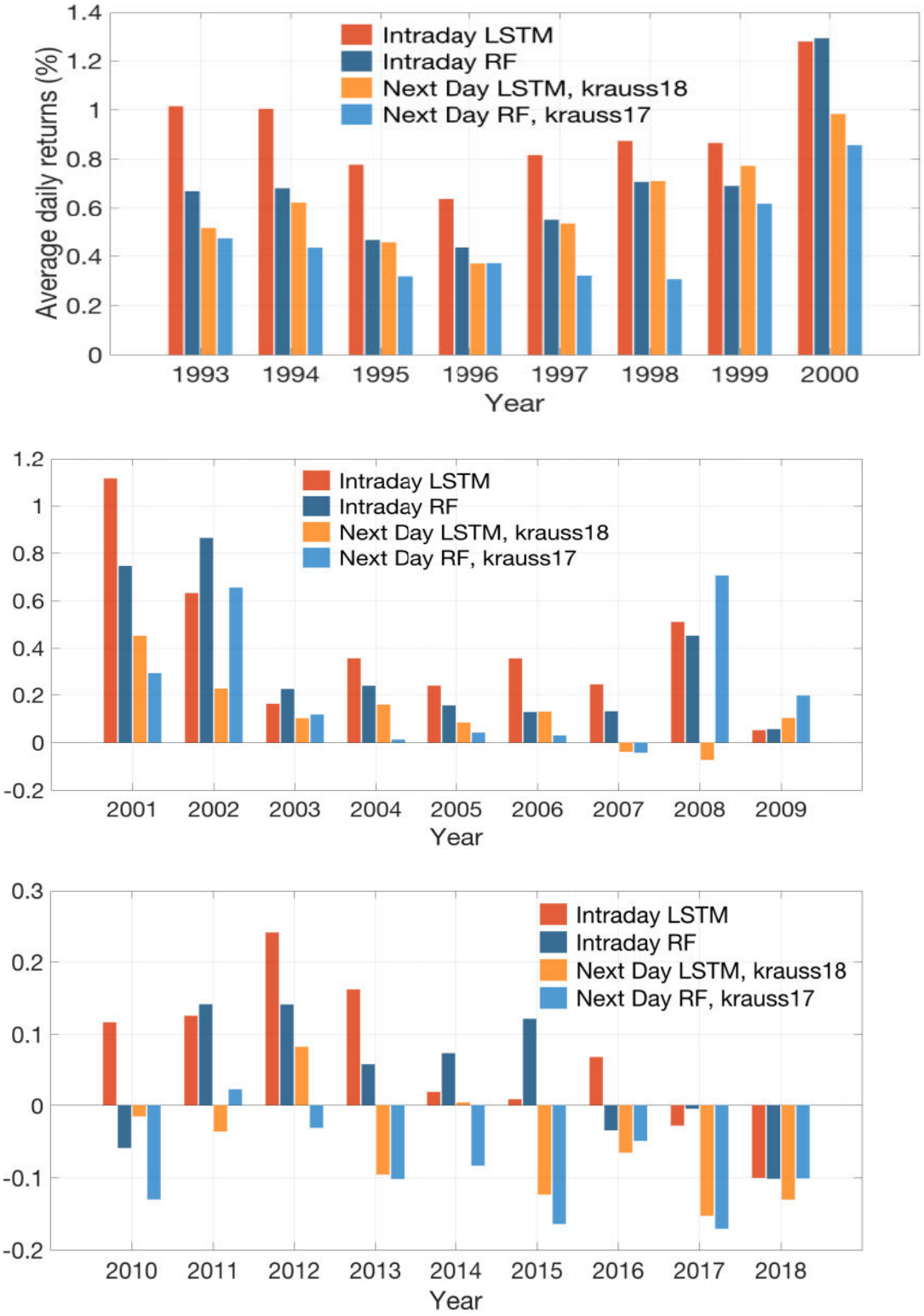

Fig. 3: Average of daily Mean Returns after deducting Transactioncosts

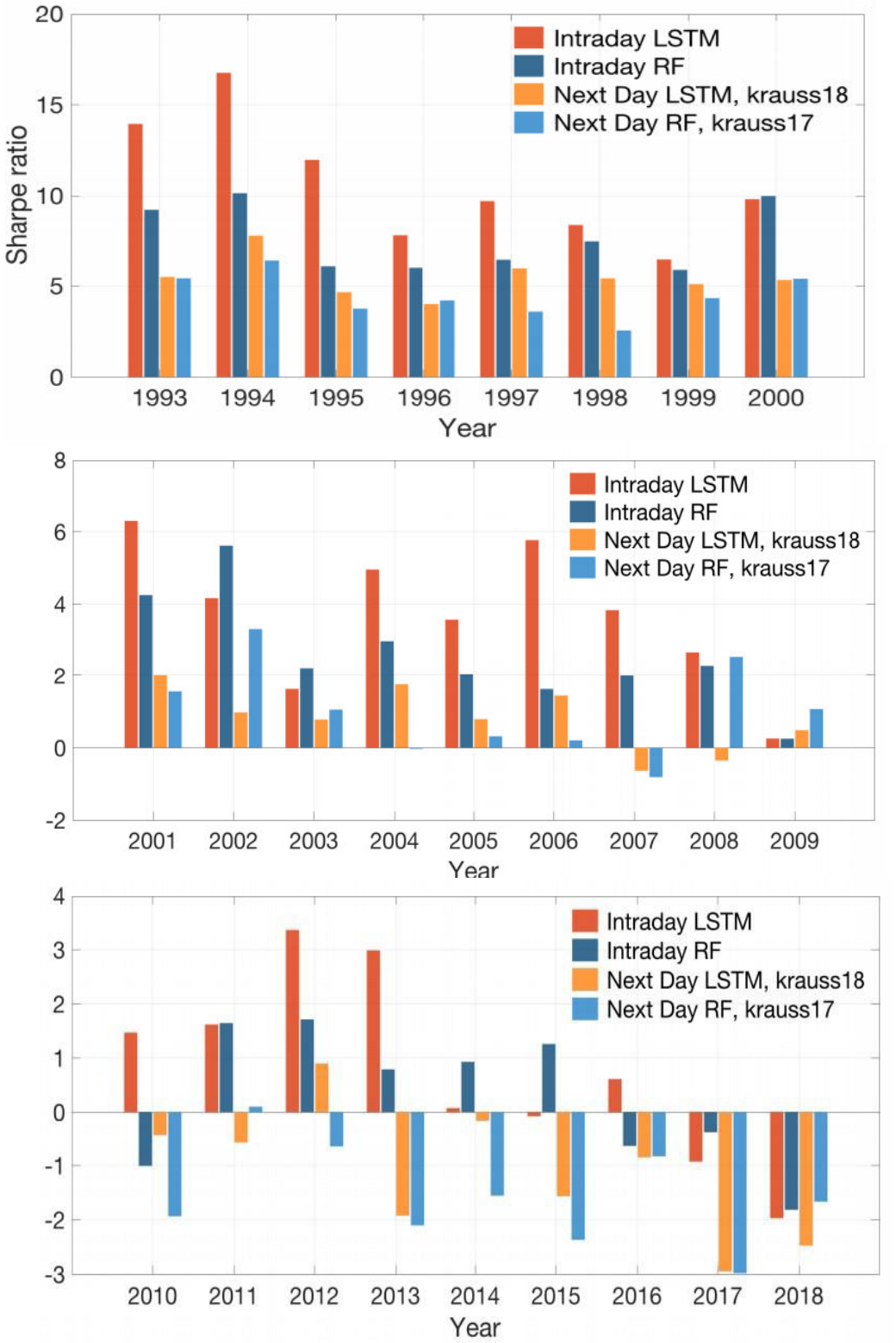

Fig. 4: Annualized Sharpe Ratio, after deducting transaction cost

5. References

[1] https://in.finance.yahoo.com/quote/yhoo/history/

[2] Braun, S. (2018). LSTM benchmarks for deep learning frameworks. preprint, arXiv:1806.01818. Breiman, L. (2001). Random forests. Machine learning, 45, 5–32.

[3] Chetlur, S., Woolley, C., Vandermersch, P., Cohen, J., Tran, J., Catanzaro, B., & Shelhamer, E. (2014). cuDNN: Efficient primitives for deep learning. preprint, arXiv:1410.0759.

[4] Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research, 270, 654–669.

[5] Ho, T. K. (1995). Random decision forests. In Proceedings of 3rd international conference on document analysis and recognition (pp. 278–282). IEEE volume 1.

[6] Huck, N. (2009). Pairs selection and outranking: An application to the S&P 100 index. European Journal of Operational Research, 196, 819–825. Huck, N. (2010). Pairs trading and outranking: The multi-step-ahead forecasting case. European Journalof Operational Research, 207, 1702–1716.

[7] Krauss, C., Do, X. A., & Huck, N. (2017). Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. European Journal ofOperational Research, 259, 689–702.

[8] Moritz, B., & Zimmermann, T. (2014). Deep conditional portfolio sorts: The relation between past and future stock returns. In LMU Munich and Harvard UniversityWorking paper .

[9] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. et al. (2011). Scikit-learn: machine learning in python. Journal of machine learning research, 12, 2825–2830.

[10] Schmidhuber, J., & Hochreiter, S. (1997). Long short-term memory. Neural Comput, 9, 1735–1780.

[11] Sezer, O. B., & Ozbayoglu, A. M. (2018). Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach. Applied Soft Computing, 70, 525–538.

[12] Siami-Namini, S., & Namin, A. S. (2018). Forecasting economics and financial time series: ARIMA vs. LSTM. preprint, arXiv:1803.06386.

[13] Takeuchi, L., & Lee, Y.-Y. A. (2013). Applying deep learning to enhance momentum trading strategies in stocks. In Technical Report. Stanford University.

[14] ran, D. T., Iosifidis, A., Kanniainen, J., & Gabbouj, M. (2018). Temporal attention-augmented bilinear network for financial time-series data analysis. IEEE transactions on neural networks and learning systems, 30, 1407–1418.

[15] Xue, J., Zhou, S., Liu, Q., Liu, X., & Yin, J. (2018). Financial time series prediction using ℓ2,1RF-ELM. Neurocomputing, 277, 176–186. 8

[16] Widom, J. (1995). Research problems in data warehousing. In Proceedings of the fourth international conference on information and knowledge management, CIKM '95 (pp. 25- 30). New York, NY, USA: ACM. 10.1145/221270.221319.

[17] R. Gencay,"Linear, non-linear and essential foreign exchange rate prediction with simple technical trading rules," Journal of International Economics, vol. 47, no.! pp. 91-107 1999.

[18] A. Timmermann and C. W Granger,"Efficient market hypothesis and forecasting," International Journal of Forecasting, vol. 20, no.! pp. 15- 27, 2004.

[19] D. Bao and Z. Yang, "Intelligent stock trading system by turning point confirming and probabilistic reasoning," Expert Systems with Applications, vol.34,no. 1,pp. 620-627,2008.

[20] Haoming Li, Zhijun Yang and Tianlun Li (2014). Algorithmic Trading Strategy Based On Massive Data Mining. Stanford University.

[21] Avellaneda, M., & Lee, J.-H. (2010). Statistical arbitrage in the US equities market. Quantitative Finance, 10, 761–782. Borovykh, A., Bohte, S., & Oosterlee, C. W. (2018). Dilated convolutional neural networks for time series forecasting. Journal of Computational Finance, Forthcoming.