Statistical Arbitrage on S&P 500 Components by Natural Language Processing (NLP)

Springboard Data Science Career Track Capstone 2, April 27th 2020 Cohort
Jiaqi Xu
12/23/2020

Thanks for Springboard mentor Kevin Glynn

The Next 20 minutes is dictated to

- Project Introduction
- Dataset Extract, Transform and Load (ETL)
- Exploration Data Analysis (EDA)
- Deep Learning Models and Prediction Results
- Trading Results and Analysis
- Future Work

Why S&P 500 Components with SEC 8-K files

- Capstone 1 – S&P 500 components, high return
- Social media, news, public files
- 8-K: additional to quarterly and annual reports

Example

On September 5, 2019, Agilent Technologies, Inc. (the "Company") entered into an underwriting agreement (the "Underwriting Agreement") with Barclays Capital Inc., J.P. Morgan Securities LLC and MUFG Securities Americas Inc., as representatives of the several underwriters named therein (the "Underwriters"), pursuant to which the Company agreed to issue and sell to the Underwriters $500 million in aggregate principal amount of its 2.750% Senior Notes due 2029 (the "Notes") in an underwritten public offering (the "Offering"). The Offering is expected to close on September 16, 2019, subject to customary closing conditions. The Underwriting Agreement contains customary representations and covenants and includes the terms and conditions of the sale of the Notes, indemnification and contribution obligations and other terms and conditions customary in agreements of this type.

Previous Related Work

- Ryan (2017) - CNN-Glove model, SEC 10-K, prediction accuracy: 62%
- Yusuf (2018) - MLP, CNN, RNN, CNN-RNN, SEC 8-K, prediction accuracy: 64.5%
- Babbe et. al (2019) - Bert + signal layer MLP, SEC 8-K, prediction accuracy: 71%
- Joseph (2020) - 267 journals predict one day price of AAPL

Goal of This Project

- MLP, CNN, RNN and CNN-RNN models : detect stock movements with immediate and delay effects (SEC 8-K)
- My prediction results vs Yusuf (2018) and Babbe et. al (2019)
- Find the best model vs S&P 500 return, challenge “semi-strong form of market efficiency”.

Dataset Extract, Transform and Load (ETL)

- List of historical S&P 500 components - Wikipedia. (up to 11/25/2020)
- Daily data of historical S&P 500 components - Python Yahoo finance module yahoo-finance-1.0.4.
- SEC 8-K Reports – (https://www.sec.gov/edgar/searchedgar/companysearch.html)

Exploration Data Analysis (EDA)

- **Dataset**: S&P 500 composites from 2005/01/01 to 2020/11/01 (15 years and 10 months data).

- **Subsets**: Split dataset into 13 subsets - each subset, 750 days/250 days – formation/ trading. Trading: non-overlapped.

Subset	form_start	form_start	trad_end	Subset	form_start	form_start	trad_end
0	1/1/2005	1/1/2008	1/1/2009	7	1/1/2012	1/1/2015	1/1/2016
1	1/1/2006	1/1/2009	1/1/2010	8	1/1/2013	1/1/2016	1/1/2017
2	1/1/2007	1/1/2010	1/1/2011	9	1/1/2014	1/1/2017	1/1/2018
3	1/1/2008	1/1/2011	1/1/2012	10	1/1/2015	1/1/2018	1/1/2019
4	1/1/2009	1/1/2012	1/1/2013	11	1/1/2016	1/1/2019	1/1/2020
5	1/1/2010	1/1/2013	1/1/2014	12	1/1/2017	1/1/2020	11/1/2020
6	1/1/2011	1/1/2014	1/1/2015				

Table 1: Formation and Trading periods for 13 non-overlapping trading batches

Exploration Data Analysis (EDA)

- Data Description

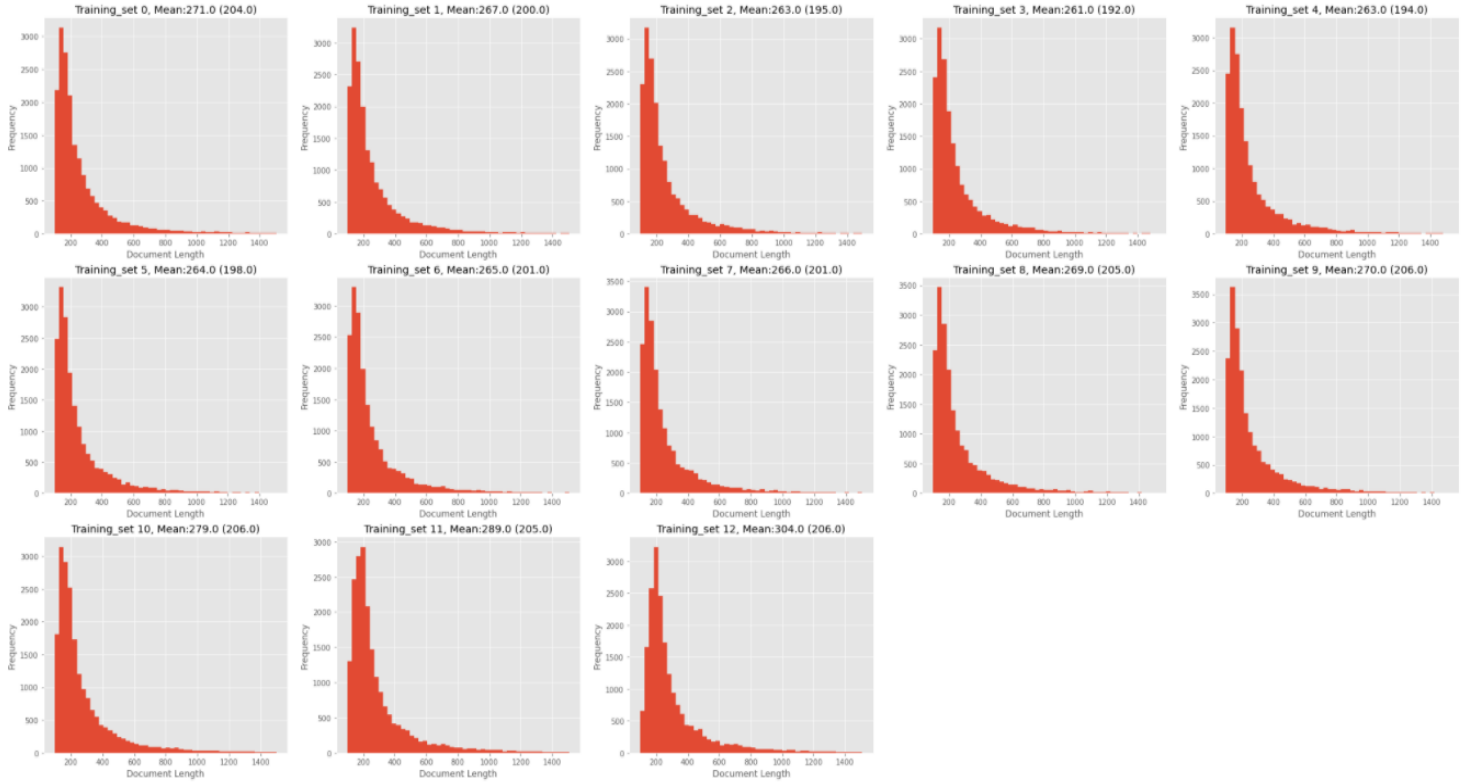


Figure 1: Length of SEC 8-K files in different training periods

Exploration Data Analysis (EDA)

Feature generation:

Input:

- $P^s = (P_t^s)_{t \in T}$ - price process of stock s, $s \in \{1, \ldots, n\}$. Simple return $R_{t,m}^s$ over m periods:

  $R_{t,m}^s = \frac{P_t^s}{P_{t-m}^s} - 1$, Where $m \in \{1\ week, 1\ month, 1\ quarter\ and\ 1\ year\}$.
- VIX (CBOE Volatility Index)
- GIS section

Output:

Case 1: Immediate effect of released files

Case 2: Delay effect of released files

- $s_{price\_change}(t) =$

$$\begin{cases} s_{close}(t) - s_{open}(t), if\ 8-K\ is\ released\ before\ normal\ trading\ hours \\ s_{open}(t+1) - s_{close}(t), if\ 8-K\ is\ released\ during\ normal\ trading\ hours \\ s_{close}(t+1) - s_{open}(t+1), if\ 8-K\ is\ released\ after\ normal\ trading\ hours \end{cases}$$

- $Index_{price\_change}(t) =$

$$\begin{cases} index_{close}(t) - index_{open}(t), if\ 8-K\ is\ released\ before\ normal\ trading\ hours \\ index_{open}(t+1) - index_{close}(t), if\ 8-K\ is\ released\ during\ normal\ trading\ hours \\ index_{close}(t+1) - index_{open}(t+1), if\ 8-K\ is\ released\ after\ normal\ trading\ hours \end{cases}$$

- $spread_{price\_change}(t) = s_{price\_change}(t) - index_{price\_change}(t)$
- $Y_t^s = \begin{cases} up\ (2), & if\ spread_{price\_change}(t) > 0.01 \\ stay\ (1), & if\ |spread_{price\_change}(t)| \leq 0.01 \\ down\ (0), & if\ spread_{price\_change}(t) < -0.01 \end{cases}$

Exploration Data Analysis (EDA)

Feature Description (Case 2)

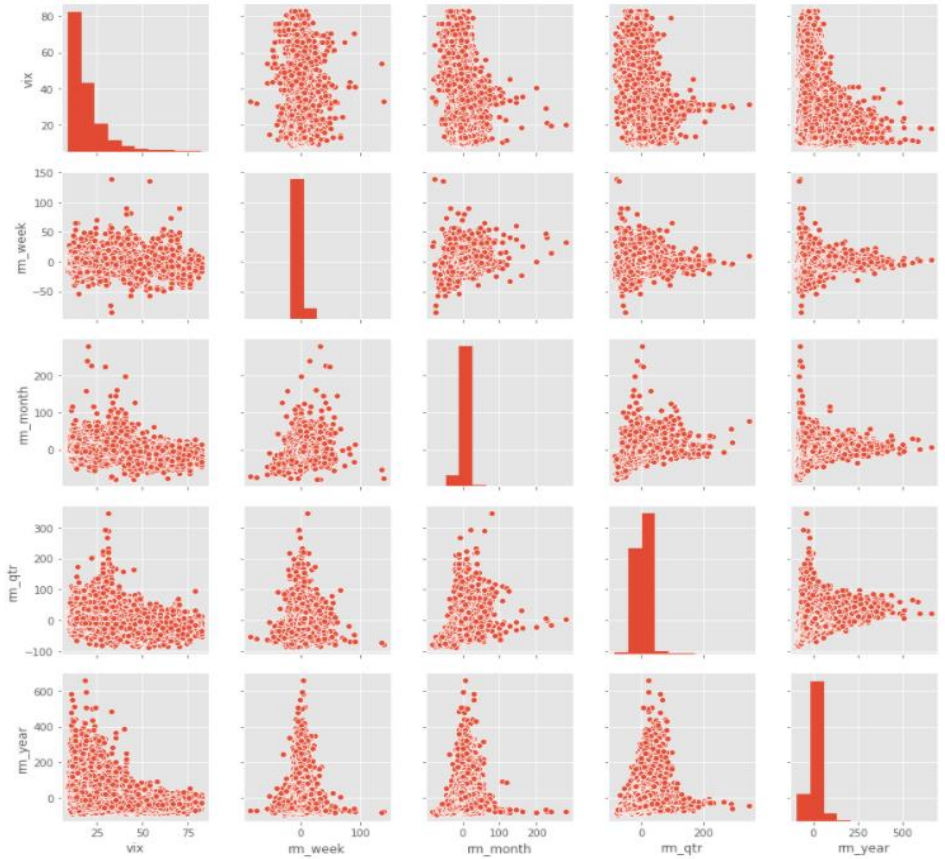


Figure 2: Heatmap correlation between new created features vix, 1-week lag (rm_week), 1-month lag (rm_month), 1-quarter lag (rm_qtr), 1-year lag (rm_year) of stock prices for case 2

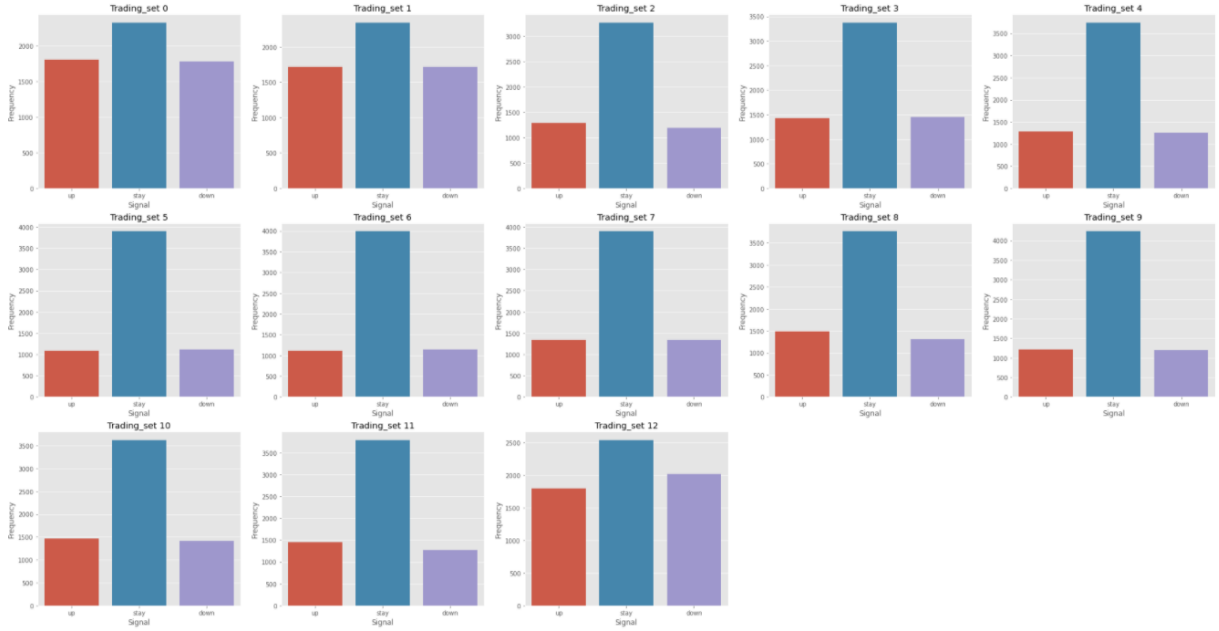
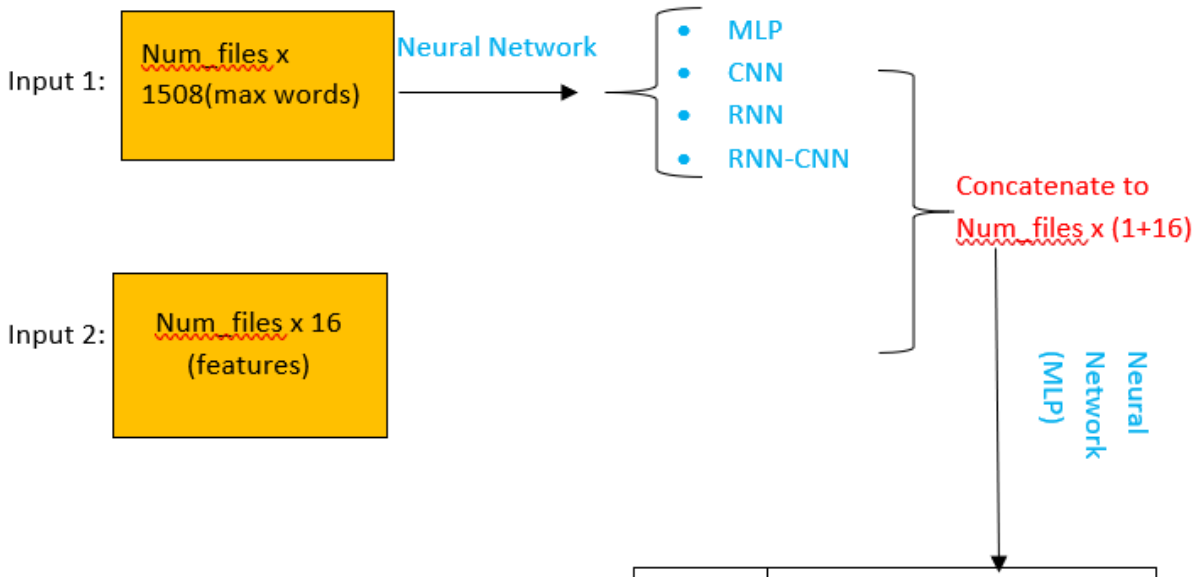


Figure 3: Frequency of signals in trading sets

Embedding Words – GloVe 100d, (https://nlp.stanford.edu/projects/glove/)

Deep Learning models



Layers	Design
1	(None,32), relu, dropout (0.2)
2	(None,32), relu
3	(None,32), relu
4	Sigmoid

Figure 4: Design of two inputs and neural networks

Result from Formation (Train/Test) Window – case 2

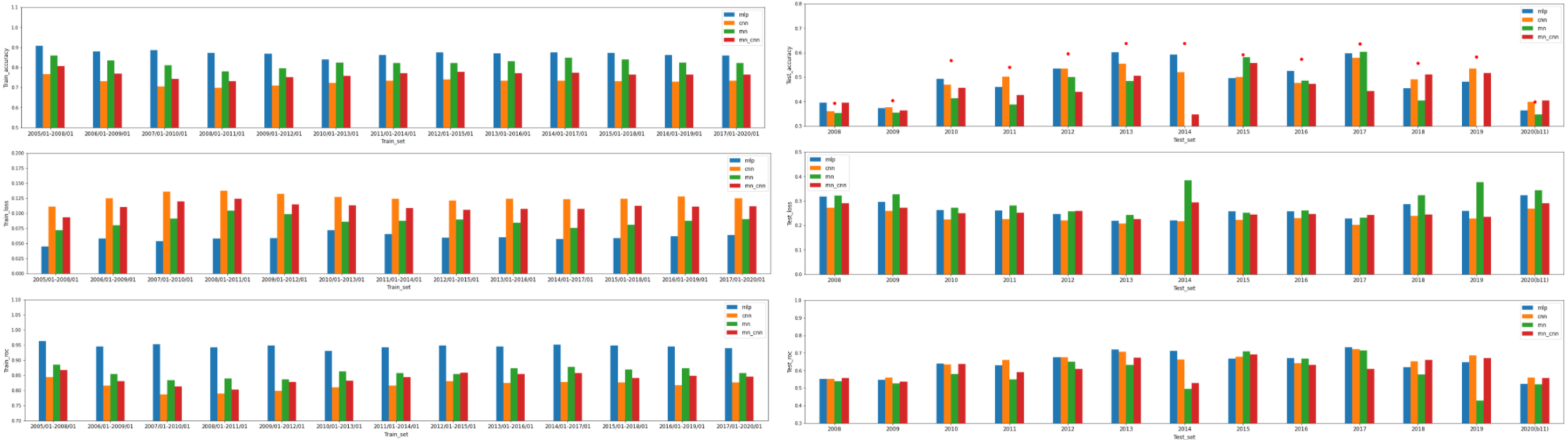


Figure 5. Train and Test Accuracy for deep learning methods for case 2

- - Train subsets: MLP best for all years
- - Test subsets: MLP best – year 2008, 2010, 2012, 2013, 2014 and 2016; CNN best: year 2011, 2019 ; RNN best: year 2015, 2017; CNN-RNN best: year 2018, 2020 Oct

Trading window with Momentum

For each of trading dataset

- Sorting all stocks over trading period in descending order with the prediction probabilities by “up” and “down” signals.

- Long the highest (“expected winner”) k probabilities for “up”, short S&P 500 index, and short the highest (“expected loser”) k probabilities for “down”, long S&P 500 index simultaneously.

Trading Results of Momentum (3 stocks/pair)

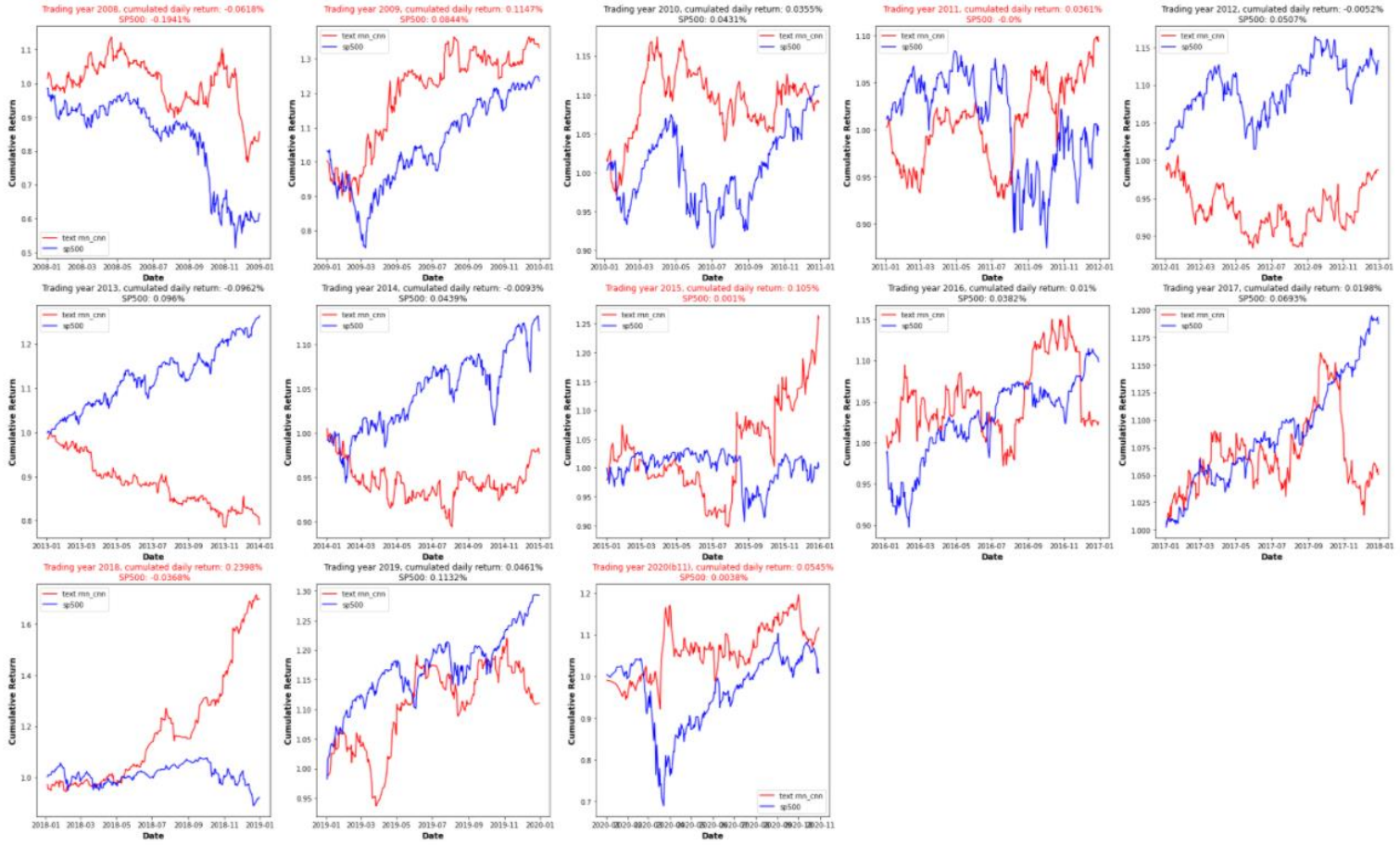


Figure 6: Cumulated Daily Trading Returns vs S&P 500 for 3 stocks/pair with CNN-RNN method

Cumulative Return (CNN-RNN) with Stocks K=1, 2, 3, 4, 5, all/pair vs S&P 500

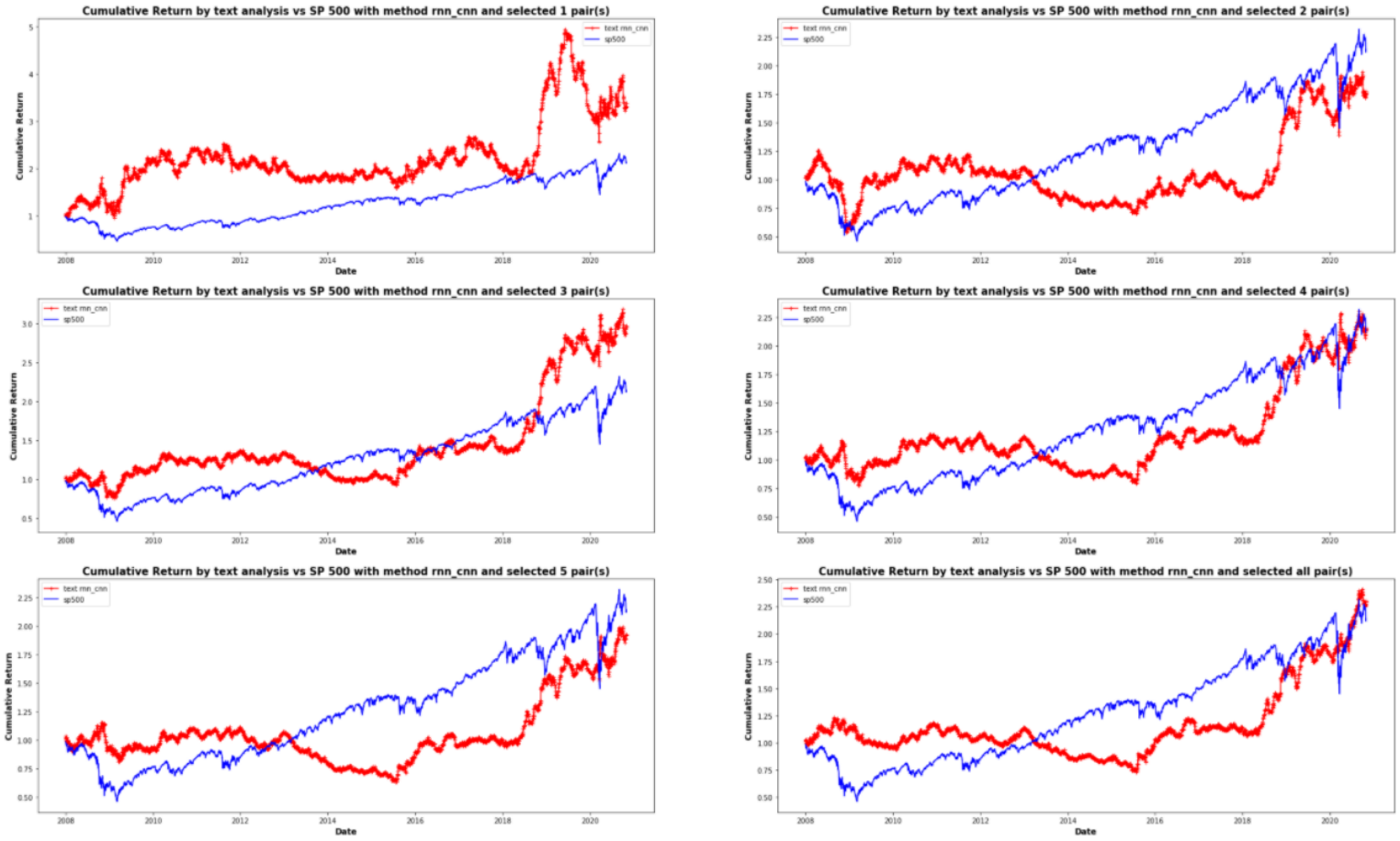


Figure 7:Cumulative return with CNN-RNN method in 15 years and 10 months by 1,2,3,4, 5 and all possible stocks/pair

Conclusion

By CNN-RNN method

- 2008, 2009, 2011, 2015, 2018 and 2020 Oct : beat S&P 500.
- 2010, 2012, 2014, 2016, 2017 and 2019: worse than S&P 500.
- Few literatures related to this kind of research with trading process, so we cannot compare the results to others
- Select appropriate methods can decrease the probability of the hypothesis in capstone 1 - “profits are declining in recent years”.
- Confirms “there is a severe challenge to the semi-strong form of market efficiency”

Future Work

- Use more accurate S&P 500 historical components
- Use other SEC files - 10-Q or 10-K.
- Other GloVe files - 50, 200, 300d.
- BERT, GPT2 or GPT3 models
- Other financial and text datasets
- More criteria, check bad performance
- Use ensemble methods
- Add transaction fee and leverage.

Reference

- Patty Ryan (2017). Stock Market Predictions with Natural Language Deep Learning.
- Yusuf Aktan (2018). Using NLP and Deep Learning to Predict Stock Price Movements.
- Mark Babbe et al. (2019). BERT is the Word: Predicting Stock Prices with Language Models.
- Trist'n Joseph (2020). An investigation into NLP using sentiment analysis to predict Apple stock price movements.

Thank You!

Jiaqi Xu

Email: jiaqiperson@gmail.com

LinkedIn: https://www.linkedin.com/in/jiaqixu1

ResearchGate: https://www.researchgate.net/profile/Jiaqi_Xu10

Project report: https://github.com/jiaqixu/Springboard/blob/master/Capstone/Capstone2/Capstone2_Final_report.pdf