Summary

```
search"stock price prediction" in paperwithcode.com
also search "stock prediction" in paperwithcode.com to see related papers
今天才发现可以搜索"stock return prediction",这个更好
```

此文档中所有被看好的title的汇总：

- Diffusion Variational Autoencoder for Tackling Stochasticity in Multi-Step Regression Stock Price Prediction
- Stock Broad-Index Trend Patterns Learning via Domain Knowledge Informed Generative Network
- Astock: A New Dataset and Automated Stock Trading based on Stock-specific News Analyzing Model
- A Word is Worth A Thousand Dollars: Adversarial Attack on Tweets Fools Stock Predictions
- Stock Movement Prediction Based on Bi-typed Hybrid-relational Market Knowledge Graph via Dual Attention Networks
- Long Term Stock Prediction based on Financial Statements
- Price graphs: Utilizing the structural information of financial time series for stock prediction
- Trade the Event: Corporate Events Detection for News-Based Event-Driven Trading
- Stock price prediction using Generative Adversarial Networks
- Multi-Graph Convolutional Network for Relationship-Driven Stock Movement Prediction
- Enhancing Stock Movement Prediction with Adversarial Training
- Temporal Relational Ranking for Stock Prediction
- DP-LSTM: Differential Privacy-inspired LSTM for Stock Prediction Using Financial News
- S&P 500 Stock Price Prediction Using Technical, Fundamental and Text Data
- FactorVAE: A Probabilistic Dynamic Factor Model Based on Variational Autoencoder for Predicting Cross-Sectional Stock Returns

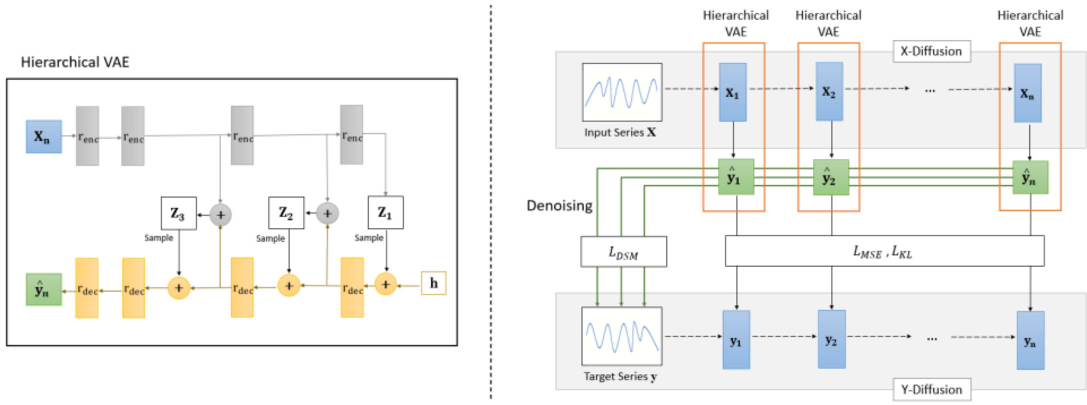
下面是一些比较被看好的文章，主要说说他们的可取之处

diffusion variational autoencoder&pytorch（还行吧，不是很了解）：

```
Diffusion Variational Autoencoder for Tackling Stochasticity in Multi-Step
Regression Stock Price Prediction
```

- **author&date:**:2023/8/18 & National University of Singapore
- **model**:diffusion variational autoencoder

- **data**:形如AAPL.csv feature:open,high,low,close,volume
- **estimation**:还行，先记录下来
- **framework**:pytorch
- **news relevant**:
- **insight**:



- **experiment**:

Table 2: Performance comparisons. The first column shows the test period year and the sequence length T. Each result represents the average MSE and standard deviation (in subscript) across 5 runs and all stocks. The best results are boldfaced.

Model		ARIMA	NBA	VAE	VAE + Adv	Autoformer	D-Va	↑ MSE	↑ SD
2016	10	$1.7054_{\pm-}$	$1.2012_{\pm0.1190}$	$1.0743_{\pm0.0255}$	$1.0752_{\pm0.0249}$	$1.0204_{\pm0.0179}$	$\mathbf{0.9040_{\pm0.0048}}$	11.41%	73.06%
	20	$1.5918_{\pm-}$	$1.1656_{\pm0.1024}$	$1.0652_{\pm0.0223}$	$1.0639_{\pm0.0202}$	$0.9920_{\pm0.0197}$	$\mathbf{0.9144_{\pm0.0047}}$	7.83%	76.16%
	40	$0.9135_{\pm-}$	$0.9713_{\pm0.0742}$	$0.9918_{\pm0.0159}$	$0.9920_{\pm0.0177}$	$0.8860_{\pm0.0241}$	$\mathbf{0.8591_{\pm0.0052}}$	3.04%	67.56%
	60	$0.8241_{\pm-}$	$0.8909_{\pm0.0373}$	$0.9390_{\pm0.0173}$	$0.9397_{\pm0.0178}$	$0.8625_{\pm0.0378}$	$\mathbf{0.8035_{\pm0.0050}}$	2.50%	71.09%
2019	10	$1.6737_{\pm-}$	$1.2266_{\pm0.1070}$	$1.1527_{\pm0.0271}$	$1.1524_{\pm0.0250}$	$1.1011_{\pm0.0180}$	$\mathbf{0.9847_{\pm0.0040}}$	10.57%	77.87%
	20	$1.2115_{\pm-}$	$1.2166_{\pm0.1161}$	$1.1156_{\pm0.0186}$	$1.1150_{\pm0.0192}$	$1.0762_{\pm0.0224}$	$\mathbf{0.9703_{\pm0.0039}}$	9.84%	79.10%
	40	$1.1189_{\pm-}$	$1.1785_{\pm0.0758}$	$1.1736_{\pm0.0168}$	$1.1744_{\pm0.0163}$	$1.1371_{\pm0.0339}$	$\mathbf{1.0453_{\pm0.0040}}$	6.58%	75.40%
	60	$1.0887_{\pm-}$	$1.1395_{\pm0.0545}$	$1.1841_{\pm0.0183}$	$1.1830_{\pm0.0180}$	$1.2066_{\pm0.0501}$	$\mathbf{1.0505_{\pm0.0043}}$	3.51%	75.93%
2022	10	$1.2857_{\pm-}$	$0.9154_{\pm0.0261}$	$1.0297_{\pm0.0223}$	$1.0196_{\pm0.0215}$	$0.9958_{\pm0.0179}$	$\mathbf{0.8653_{\pm0.0053}}$	13.10%	70.23%
	20	$1.0095_{\pm-}$	$0.9346_{\pm0.0455}$	$0.9810_{\pm0.0171}$	$0.9811_{\pm0.0179}$	$0.9529_{\pm0.0207}$	$\mathbf{0.8451_{\pm0.0037}}$	10.70%	78.41%
	40	$0.9384_{\pm-}$	$0.8739_{\pm0.0301}$	$0.9643_{\pm0.0156}$	$0.9635_{\pm0.0155}$	$0.9142_{\pm0.0227}$	$\mathbf{0.8426_{\pm0.0035}}$	7.83%	77.12%
	60	$0.8538_{\pm-}$	$0.8625_{\pm0.0195}$	$0.9436_{\pm0.0171}$	$0.9442_{\pm0.0162}$	$0.8428_{\pm0.0254}$	$\mathbf{0.8174_{\pm0.0035}}$	3.01%	78.22%

Table 4: Comparison of 10-Day Sharpe ratios. For each table, going from left to right represents the handling of epistemic uncertainty and going from top to bottom represents the handling of aleatoric uncertainty. The best results are boldfaced.

2016	Sharpe Ratio	Sharpe Ratio (Regularized)
NBA	0.0270	0.0691
Equal	0.1089	
D-Va	0.0772	**0.1174**

2019	Sharpe Ratio	Sharpe Ratio (Regularized)
NBA	0.0820	0.1767
Equal	0.2337	
D-Va	0.1197	**0.2767**

2022	Sharpe Ratio	Sharpe Ratio (Regularized)
NBA	0.0332	0.0404
Equal	0.0437	
D-Va	0.0600	**0.0645**

Handling epistemic ⟶ Handling aleatoric ⟶

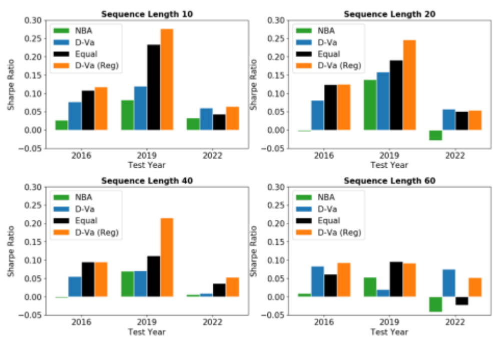


Figure 6: Comparison of T-day Sharpe ratios across the dif-

Table 5: Comparison of Sharpe ratios across benchmark multi-step prediction models, after the graphical lasso regularization is applied. The best results are boldfaced.

Model		ARIMA	NBA	Autoformer	Equal	D-Va
2016	10	0.0953	0.0691	0.1114	0.1089	**0.1174**
	20	0.0815	0.1075	0.0921	0.1243	**0.1246**
	40	0.0683	0.0717	0.0438	0.0950	**0.0953**
	60	0.0544	0.0803	0.0235	0.0614	**0.0929**
2019	10	0.2736	0.1767	0.2350	0.2337	**0.2767**
	20	0.2455	0.1443	0.1672	0.1912	**0.2467**
	40	0.2152	0.1252	0.1031	0.1113	**0.2152**
	60	0.0732	0.0826	0.0804	**0.0965**	0.0914
2022	10	0.0173	0.0404	0.0544	0.0437	**0.0645**
	20	0.0408	0.0404	0.0141	0.0505	**0.0540**
	40	0.0287	-0.0223	-0.0016	0.0365	**0.0527**
	60	-0.0329	-0.0179	-0.0630	-0.0230	**0.0524**

- **实际运行github代码**:还没做

remark:

```
论文任务是预测单只股票的价格/收益率，并在后文给出投资组合
diffusion我不太了解，但看这个是用mse作为评判指标感觉很奇怪
不过幸好后文有对投资组合的sharp ratio
这个论文的数据集是很常用的数据集，feature也很常用，感觉还行，似乎还有指数增强
```

ideas:

```
difussion model 可以考虑一下
```

link:

- **GitHub code**:https://github.com/koa-fin/dva
- **Paper with Code**:https://paperswithcode.com/paper/diffusion-variational-autoencoder-for

GAN(GRU,Attention Layer)&pytorch（看好）：

```
Stock Broad-Index Trend Patterns Learning via Domain Knowledge Informed
Generative Network
```

- **author&date:**:2023.2.27 & New Jersey Institute of Technology
- **model**:GAN(GRU,Attention Layer)
- **data**:有编码化的news_data,也有很平常的stock_data
- **estimation**:看起来逻辑清晰，思维缜密，很难得用的是acc,而且效果可以达60%,总体来看非常不错
- **framework**:pytorch
- **news relevant**:Relevant
- **insight**:

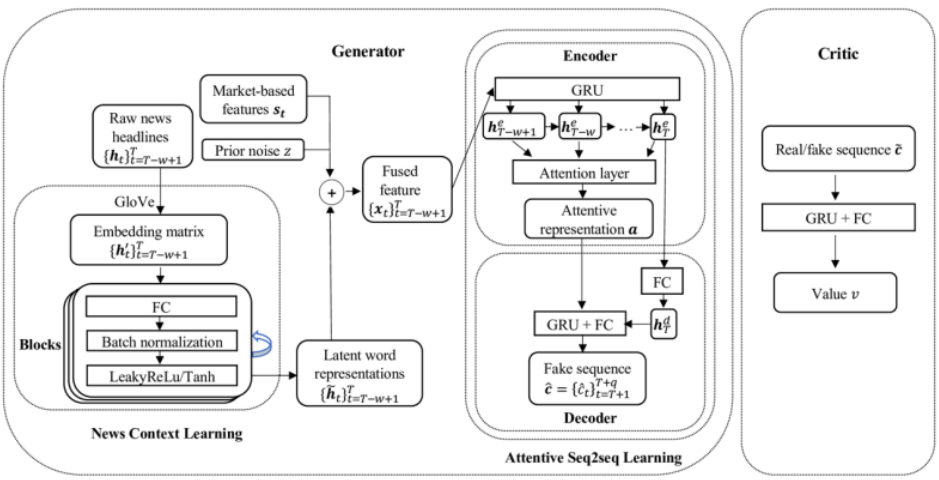


Figure 1 The Proposed *IndexGAN* Structure

- **experiment**:

Table 4. Performance comparison based on the average and standard deviation of 10 runs. The top two baselines are traditional methods. The middle four models are recent state-of-the-art benchmarks. The bottom three are based on GANs.

Model	DJIA			S&P500		
	Acc (%)	F1	MCC	Acc (%)	F1	MCC
EMA	51.83	0.483	0.058	53.93	0.511	0.087
ARIMA	53.15	0.479	0.091	50.53	0.475	0.085
DA-RNN	56.23 ±1.12	0.642 ±0.007	0.104 ±0.021	56.71 ±1.53	0.513 ±0.010	0.092 ±0.013
StockNet	57.90 ±0.84	0.632 ±0.008	0.111 ±0.031	55.37 ±1.35	0.585 ±0.009	0.091 ±0.019
A-LSTM	58.12 ±1.05	0.674 ±0.010	0.145 ±0.018	57.13 ±1.20	0.596 ±0.011	0.143 ±0.022
Trans	58.87 ±1.01	0.676 ±0.009	0.179 ±0.027	58.30 ±1.34	0.610 ±0.011	0.176 ±0.031
DCGAN	56.79 ±0.09	0.622 ±0.003	0.082 ±0.014	55.85 ±1.01	0.495 ±0.008	0.121 ±0.010
GAN-FD	57.02 ±1.09	0.607 ±0.011	0.094 ±0.023	56.93 ±1.15	0.591 ±0.012	0.105 ±0.024
S-GAN	57.93 ±1.08	0.632 ±0.010	0.099 ±0.020	57.42 ±1.07	0.601 ±0.012	0.112 ±0.018
IndexGAN	**60.85 ±0.95**	**0.713 ±0.006**	**0.208 ±0.024**	**60.00 ±1.37**	**0.616 ±0.014**	**0.199 ±0.027**

remark:

```
非常规范，还在github里放了自己最终学习的，模型的参数
正确率一看就很真实，eg:他的对照组的真确率可以到57%（LSTM）这个数看起来就很正常
```

ideas:

```
或许可以把他的GRU进行改进？GAN我目前不太了解，之后要去学习一下
```

link:

- **GitHub code**:https://paperswithcode.com/paper/stock-broad-index-trend-patterns-learning-via
- **Paper with Code**:https://paperswithcode.com/paper/stock-broad-index-trend-patterns-learning-via

Transformer&pytorch（看好）：

```
Astock: A New Dataset and Automated Stock Trading based on Stock-specific News Analyzing Model
```

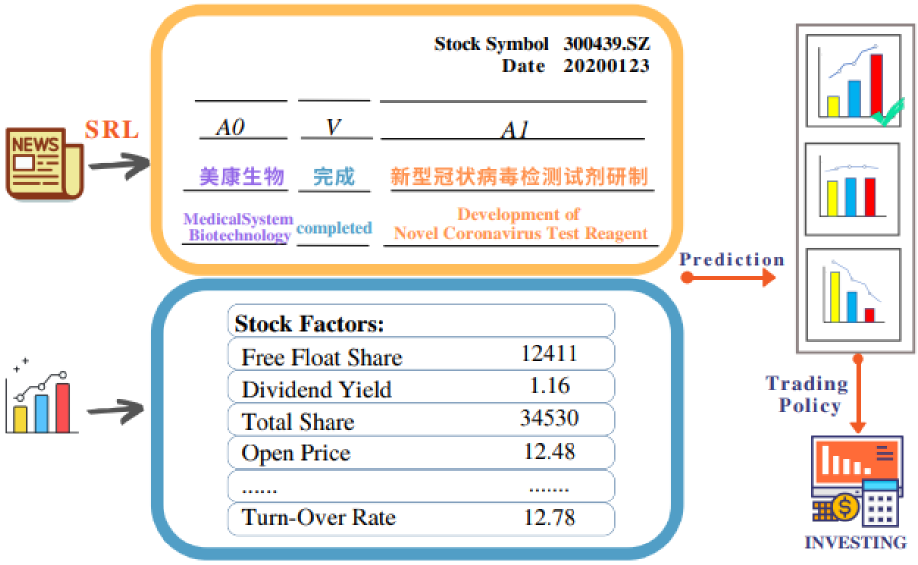
- **author&date:**:2022.06.14 & (学校似乎一般而且为什么有6个共同一作？)

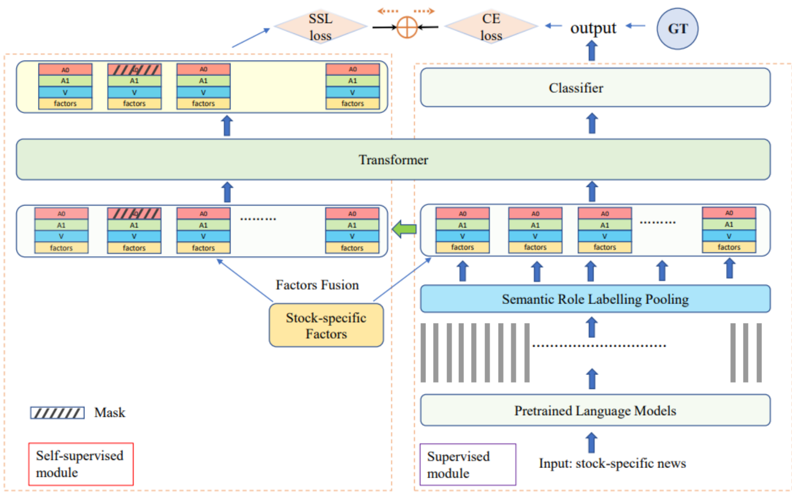
Jinan Zou[1] *, **Haiyao Cao**[1] *, **Lingqiao Liu**[1], **Yuhao Lin**[1], **Ehsan Abbasnejad**[1], **Javen Qinfeng Shi**[1] †,

[1]Australian Institute for Machine Learning, University of Adelaide

- **model**:Transformer
- **data**:github上的代码感觉他没处理好,高开低收都放一列了
- **estimation**:accuracy效果很好，很赞，又是基于transformer的，如果他数据和结果可以复现的话这个很值得考虑
- **framework**:pytorch
- **news relevant**:相关

- **insight**:


SRL
NEWS
Stock Symbol 300439.SZ
Date 20200123
A0
V
A1
美康生物
完成
新型冠状病毒检测试剂研制
MedicalSystem Biotechnology
completed
Development of Novel Coronavirus Test Reagent
Stock Factors:
Free Float Share 12411
Dividend Yield 1.16
Total Share 34530
Open Price 12.48
....
Turn-Over Rate 12.78
Prediction
Trading Policy
INVESTING


SSL loss
CE loss
output
GT
Classifier
Transformer
A0
A1
V
factors
Factors Fusion
Stock-specific Factors
Semantic Role Labelling Pooling
Pretrained Language Models
Mask
Self-supervised module
Supervised module
Input: stock-specific news

- **experiment**:

Table 2: The stock movement classification performance(%) of in-distribution evaluation on our scheme and others demonstrates the effectiveness of our self-supervised SRL method. ✔ indicates that the model adopted this Semantic role's pooling information. - indicates that the method does not adopt this semantic role's pooling. ✗ indicates that semantic role's pooling is masked.

Model	Resource	Semantic Role			Accuracy	F1 Score	Recall	Precision
		A0	V	A1				
StockNet [Xu and Cohen, 2018] ACL 18'	News	-	-	-	46.72	44.44	46.68	47.65
HAN Stock [Hu *et al.*, 2018] ICWSDM 18'	News	-	-	-	57.35	56.61	57.20	58.41
Bert Chinese [Devlin *et al.*, 2019]NAACL 19'	News	-	-	-	59.11	58.99	59.20	59.07
ERNIE-SKEP [Tian *et al.*, 2020] ACL 20'	News	-	-	-	60.66	60.66	60.59	61.85
XLNET Chinese [Cui *et al.*, 2020]EMNLP 20'	News	-	-	-	61.14	61.19	61.09	61.60
RoBERTa WWM Ext [Cui *et al.*, 2020]EMNLP 20'	News	-	-	-	61.34	61.48	61.32	61.97
	News + Factors	-	-	-	62.49	62.54	62.51	62.59
Our SRLP	News	✔	✔	✔	61.76	61.69	61.62	61.87
	News + Factors	✔	✔	✔	64.79	64.85	64.79	65.26
Our Self-supervised SRLP	News	✗	✔	✗	61.07	61.11	61.11	61.11
	News	✗	✔	✔	62.36	62.32	62.43	62.64
	News	✔	✔	✗	62.42	62.46	62.44	62.62
	News	✗	✗	✔	62.15	62.15	62.15	62.59
	News	✔	✗	✗	61.34	61.23	61.46	61.30
	News	✔	✗	✔	62.97	63.05	62.93	63.47
Our Self-supervised SRLP with Factors	News + Factors	✗	✔	✗	64.59	64.62	64.63	64.65
	News + Factors	✗	✔	✔	66.82	66.81	66.90	66.82
	News + Factors	✔	✔	✗	65.54	65.53	65.62	65.50
	News + Factors	✗	✗	✔	65.34	65.21	65.43	65.43
	News + Factors	✔	✗	✗	65.27	65.35	65.24	65.77
	News + Factors	✔	✗	✔	**66.89**	**66.92**	**66.95**	**66.92**

5.4 Profitability Evaluation in Real-world

Table 4: The comparison of profitability test on Maximum Drawdown(%), Annualized Rate of Return(%), and Sharpe Ratio Rate(%) with strong baselines, XIN9, CSI300 and our proposed method from 1/1/2021 to 12/11/2021.

Model	Maximum Drawdown↓	Annualized Rate of Return↑	Sharpe Ratio↑
XIN9	-15.85	-15.38	-32.01
CSI300	-14.40	-9.34	-32.99
StockNet[Xu and Cohen, 2018]	-7.40	-22.42	-177.65
HAN Stock[Hu *et al.*, 2018]	-7.38	-13.50	-55.84
RoBERTa WWM Ext[Cui *et al.*, 2020]	-3.83	1.35	-16.31
Self-supervised SRLP(V masked) with Factors	**-3.60**	**13.85**	**40.93**

remark:

```
github上的star很多，是基于transformer很好，从论文来看效果非常好，正确率可以到66%，同时似乎加了一些新闻的信息；
当然，他这个数据处理我不是很很懂？为啥把高开低收都放一列？（是他在读入数据的时候就是这么干的，有可能是这很可能是一个制表符分隔的 .tsv 文件（Tab-Separated Values），但有可能被保存为 .csv 扩展名）
```

link:

- **GitHub code**:https://paperswithcode.com/paper/astock-a-new-dataset-and-automated-stock
- **Paper with Code**:https://paperswithcode.com/paper/astock-a-new-dataset-and-automated-stock

Title（还行）：

```
A Word is Worth A Thousand Dollars: Adversarial Attack on Tweets Fools Stock Predictions
```

remark:

```
论文投至ACL和NAACL,但论文里研究的东西我完全看不懂（没花时间看）
似乎是类似NLP预测网络攻击，主要是他的github上的star很少，而且他22年就发表了，好像用了一些RNN
总体而言类似于基于twitter文本分析预测股价
```

data:有基本的股票数据，也有这样的文本数据

{"text": ["summary", "of", "yesterdays", "webcast", "featuring", "$aapl", "<ticker>", "$goog", "<ticker>"], "created_at": "Wed Jan 01 03:59:03 +0000 2014", "user_id_str": "1938270918"}
{"text": ["summary", "of", "yesterdays", "webcast", "featuring", "$aapl", "<ticker>", "$goog", "<ticker>"], "created_at": "Wed Jan 01 03:29:29 +0000 2014", "user_id_str": "1933063572"}
{"text": ["itv", "will", "boost", "apple", "$aapl"], "created_at": "Wed Jan 01 18:08:47 +0000 2014", "user_id_str": "23059499"}
{"text": ["users", "are", "more", "intelligent", "than", "and", "hit the cell", "owners", "$aapl", "<ticker>"], "created_at": "Wed Jan 01 01:52:31 +0000 2014", "user_id_str": "23954327"}
{"text": ["summary", "of", "yesterdays", "webcast", "featuring", "$aapl", "<ticker>", "$goog", "<ticker>"], "created_at": "Wed Jan 01 01:18:36 +0000 2014", "user_id_str": "1937591882"}
{"text": ["<number>", "wrapup", "and", "trading", "set", "review", "part", "in", "$aapl", "$bp", "<ticker>", "$csco", "<ticker>", "$cvx", "$goog", "<ticker>", "<ticker>", "$intc", "$ngg"], "created_at": "Wed Jan 01 10:52:20 +0000 2014", "user_id_str": "23059499"}
{"text": ["apple", "screwed", "up", "big", "time", "$amzn", "$aapl"], "created_at": "Wed Jan 01 15:01:12 +0000 2014", "user_id_str": "23669783"}
{"text": ["summary", "of", "yesterdays", "webcast", "featuring", "$aapl", "<ticker>", "$goog", "<ticker>"], "created_at": "Tue Dec 31 23:10:08 +0000 2013", "user_id_str": "1864753100"}

link:

- **GitHub code**:https://github.com/yonxie/advfintweet
- **Paper with Code**:https://paperswithcode.com/paper/a-word-is-worth-a-thousand-dollars-2

GRU+Graph+attention&pytorch（一般吧）：

```
Stock Movement Prediction Based on Bi-typed Hybrid-relational Market Knowledge Graph via Dual Attention Networks
```

data:新闻标题和股票数据

author:

YU ZHAO, Southwestern University of Finance and Economics, China
HUAMING DU, Southwestern University of Finance and Economics, China
YING LIU, Southwestern University of Finance and Economics, China
SHAOPENG WEI, Southwestern University of Finance and Economics, China
XINGYAN CHEN, Southwestern University of Finance and Economics, China
FUZHEN ZHUANG, Beihang University, China
QING LI, Southwestern University of Finance and Economics, China
JI LIU, AI Lab, Kwai Inc., USA
GANG KOU*, Southwestern University of Finance and Economics, China

model:GRU+Graph+Dual Attention

framework:pytorch

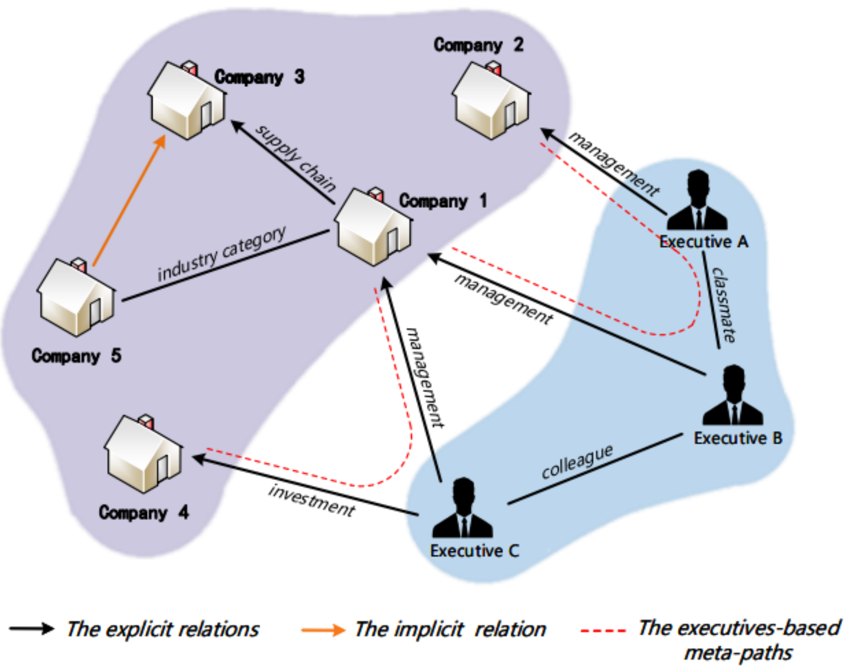
experiment:

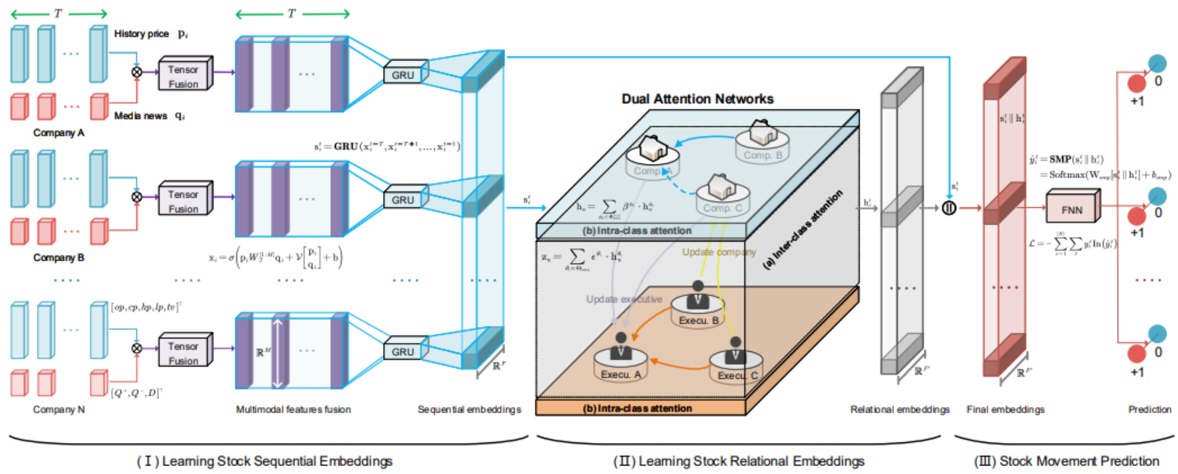
Table 5. Stock prediction results of different models.

Methods	CSI100E Accuracy	CSI100E AUC	CSI300E Accuracy	CSI300E AUC
LSTM [Hochreiter and Schmidhuber 1997]	51.14	51.33	51.78	52.24
GRU [Cho et al. 2014]	51.66	51.46	51.11	52.30
GCN [Kipf and Welling 2017]	51.58	52.18	51.68	51.81
GAT [Velickovic et al. 2018]	52.17	52.78	51.40	52.24
RGCN [Schlichtkrull et al. 2018]	52.33	52.69	51.79	52.59
HGT [Hu et al. 2020]	53.01	52.51	51.70	52.19
MAN-SF [Sawhney et al. 2020a]	52.86	52.23	51.91	52.48
STHAN-SR [Sawhney et al. 2021]	52.78	53.05	52.89	53.48
AD-GAT [Cheng and Li 2021]	54.56	55.46	52.63	54.29
DANSMP (ours)	**57.75**	**60.78**	**55.79**	**59.36**

insight:

Stock Movement Prediction Based on Bi-typed Hybrid-relational MKG via Dual Attention Networks. 3





!

remark:

```
文章思路很有趣，但很多地方我看不懂。github上的代码很专业，甚至有rawdata和preprocess的data和code
```

link:

- **GitHub code**:https://github.com/trytodoit227/dansmp
- **Paper with Code**:https://paperswithcode.com/paper/stock-movement-prediction-based-on-bi-typed-1

LSTM&tf（很一般）：

```
（没啥参靠价值，这人就是简单调包了，但是把任务做得投机取巧，accuracy乍一眼一看很高）
Long Term Stock Prediction based on Financial Statements
```

- **author&date:**:2021 & from stanford

- **model**:LSTM
- **data**:daily_stock_data with some yearly statistics & 年度财务报表
- **estimation**:

  ```
  效果很好，就是没什么创新，单纯就是用了LSTM，我也不知道你为啥效果这么好？
  个人感觉他投机取巧了，他的任务改成了预测年化的收益率的大体分类，分了5类
  但是没有后续了，没有shrp-ratio之类的指标，对我没什么参考价值
  ```

(a) annual percentage change smaller than -50%;
(b) annual percentage change greater or equals to -50% and smaller than 0%;
(c) annual percentage change greater or equals to 0% and smaller than 50%;
(d) annual percentage change greater or equals to 50% and smaller than 100%;
(e) annual percentage change greater or equals to 100%;

- **framework**:tensorflow
- **news relevant**:不相关
- **experiment**:

Model	Structure	Training set accuracy	Dev set accuracy	Test set accuracy
A	LSTM256, Dropout, Dense256, Dropout, Softmax5	0.95	0.89	0.86
B	LSTM512, Dropout, Dense512, Dropout, Softmax5	0.97	0.72	0.71
C	LSTM128, Dropout, Dense128, Dropout, Softmax5	0.82	0.59	0.58
D	LSTM256, Dense256, Softmax5	0.97	0.63	0.61
E	LSTM256, Dropout, LSTM256, Dropout, Dense256, Dropout, Softmax5	0.91	0.70	0.66
F	LSTM256, Dropout, Dense256, Dropout, Dense256, Dropout, Softmax5	0.89	0.68	0.61

Table 1: Comparison of accuracy between models with different structures.

remark:

```
鸡肋
```

link:

- **GitHub code**:https://paperswithcode.com/paper/long-term-stock-prediction-based-on-financial
- **Paper with Code**:https://paperswithcode.com/paper/long-term-stock-prediction-based-on-financial

Transformer&pytorch（看起来非常不错）：

```
Price graphs: Utilizing the structural information of financial time series for stock prediction
```

- **author&date:**:2021.7 & 北航
- **model**:类transfomer+timeseries_graph_embedding
- **data**:daily_stock_data
- **estimation**:

  ```
  模型和accuracy看起来很不错，真的很值得借鉴学习
  文中只用了VG建立图，但事实上，也可以用神经网络建立图，这个很值得我思考和学习
  ```

- **framework**:pytorch

- **news relevant**:不相关
- **insight**:

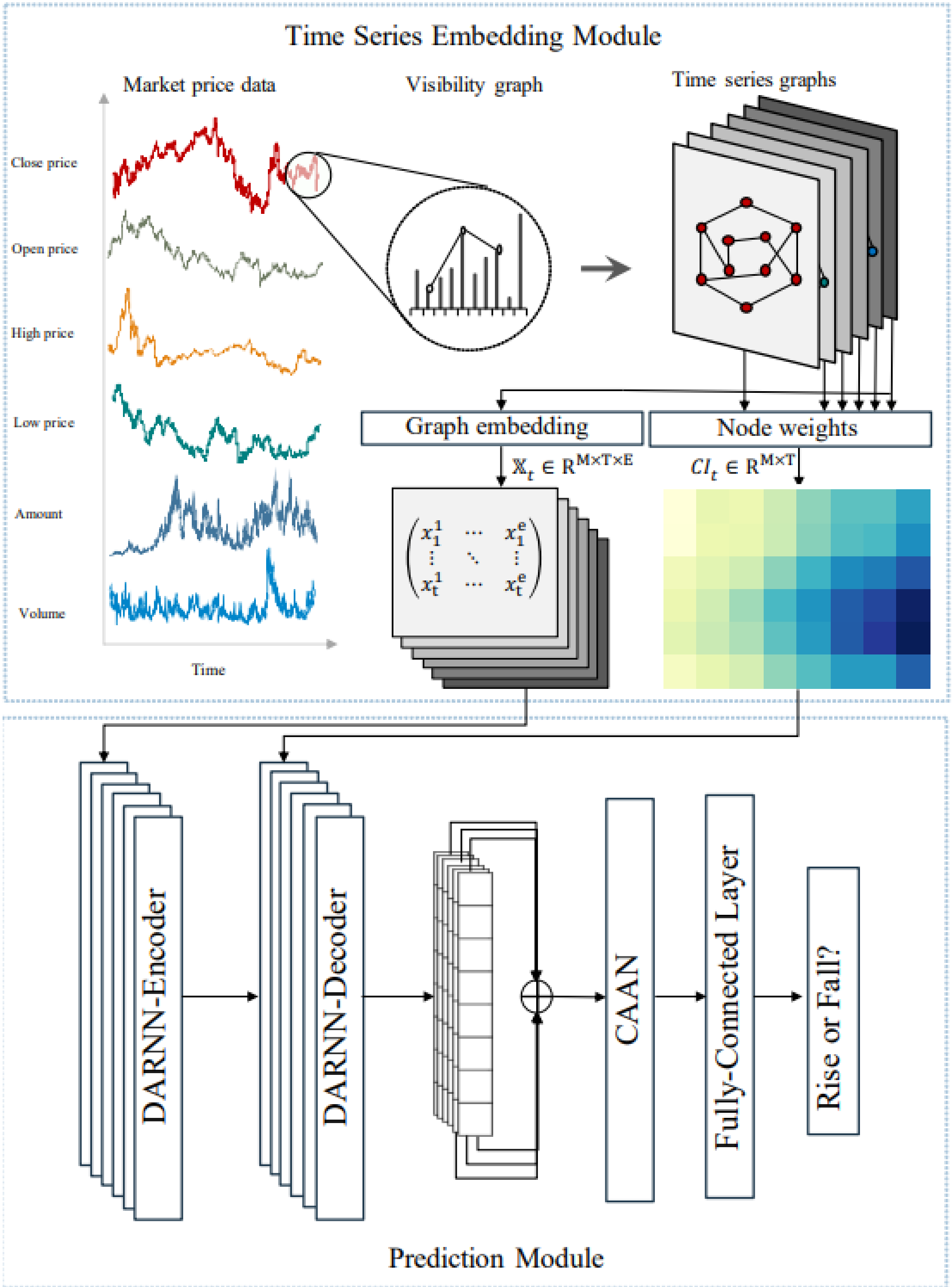


Figure 1: **Computational flow of the proposed framework.** Let $P_t \in \mathbb{R}^{M\times T}$ denote the history price data for a stock at time t, the node vectors after embedding is $\mathbb{X}_t \in \mathbb{R}^{M\times T\times E}$ and node weights is $CI_t \in \mathbb{R}^{M\times T}$, where M is number of stock quote data (six here), T is the length of the lookback window and E is the embedding size of graph nodes.

- **experiment**:

  Table 2: **Results (%) of our proposed framework and the baselines.** All models predict price trend labels at the next time step. The best-performing results are highlighted with boldface. Our proposed framework outperforms all the state-of-the-art baselines on the test accuracies.

	2019(S1)				2019(S2)				2019(S3)				2019(S4)			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
ARMA	50.15	54.96	42.81	48.13	50.75	46.61	42.51	44.46	49.89	44.26	39.91	41.97	50.07	49.40	41.77	45.26
GARCH	50.28	54.90	44.68	49.26	50.66	46.74	45.74	46.23	50.43	45.05	41.76	43.34	50.36	49.75	42.24	45.69
LSTM	57.94	64.88	52.26	57.89	59.88	58.98	44.04	50.43	56.71	52.23	35.33	42.16	54.75	55.97	44.92	49.84
DARNN	60.87	68.27	54.69	60.73	62.03	61.48	48.29	54.09	60.62	58.10	61.33	59.67	61.54	68.34	63.31	65.73
DARNN-SA	64.32	71.72	58.63	64.52	66.23	66.60	54.40	59.89	65.47	65.00	59.09	61.9	65.63	72.34	68.92	70.59
MFNN	61.21	68.28	60.81	64.33	63.00	65.30	51.40	57.52	62.74	67.68	58.75	62.9	64.69	67.19	57.52	61.98
CA-SFCN	65.51	72.82	60.10	65.85	67.21	67.61	**73.52**	70.44	66.10	**77.81**	68.32	**72.76**	67.30	70.24	73.38	71.78
Our framework	**67.48**	**75.24**	**61.45**	**67.65**	**68.46**	**69.81**	71.67	**70.73**	**68.34**	67.86	**73.77**	68.09	**67.91**	**77.51**	**73.78**	**75.60**

Notes. Precision, recall and the F1 measure are metrics calculated in the upward direction.

remark:

```
github里的组成看起来很规范，想法也很不错，accuracy很好，其中对照组的LSTM的accucary可以到57%，这让他的数据看起来很可信
其中timeseries graph的建立和graph embedding很值得借鉴
方法：VG(Visible graph)论文里这段全是废话不用看。
我问了gpt,他的回答如下:
```

https://chat.openai.com/share/d661a86a-c8ef-481c-ae98-9531a5cacc4d

```
简单来说就是如果两个向量满足了某种相关性就建立连个股票之间的边
文中只用了VG建立图，但事实上，也可以用神经网络建立图，这个很值得我思考和学习
```

link:

- **GitHub code**:https://github.com/BUAA-WJR/PriceGraph
- **Paper with Code**:https://paperswithcode.com/paper/price-graphs-utilizing-the-structural

Transfomer&pytorch（还行）：

```
Trade the Event: Corporate Events Detection for News-Based Event-Driven Trading
```

- **author&date:**:2021 & 被ACL接收
- **model**:transformer & 似乎有domain adaptation我没仔细看
- **data**:EDT dataset, 数据有点大，目前还没下载下来，别忘了看看！！！！！！！
- **estimation**:

  ```
  虽然没给出accuracy,但好像我一开始很看重这篇文章，但至于为什么有点忘了，可能好不容易看到一个transformer+nlp且比较靠谱的吧？
  ```

- **framework**:pytorch
- **news relevant**:相关

- **insight**:

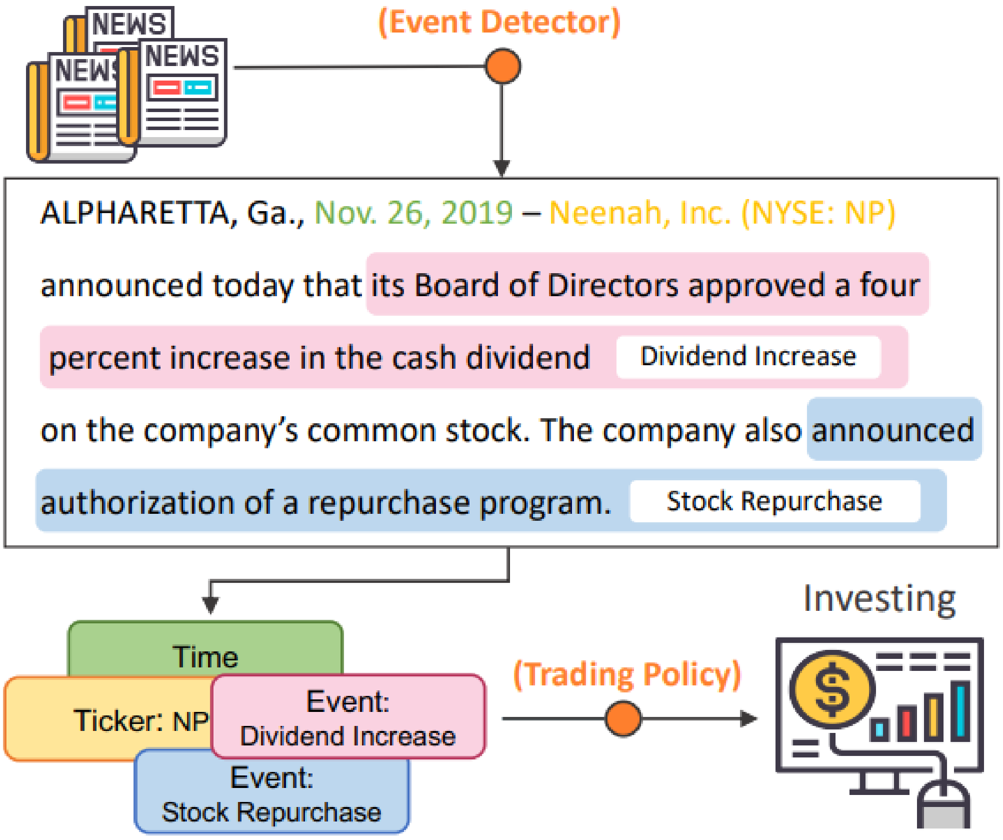
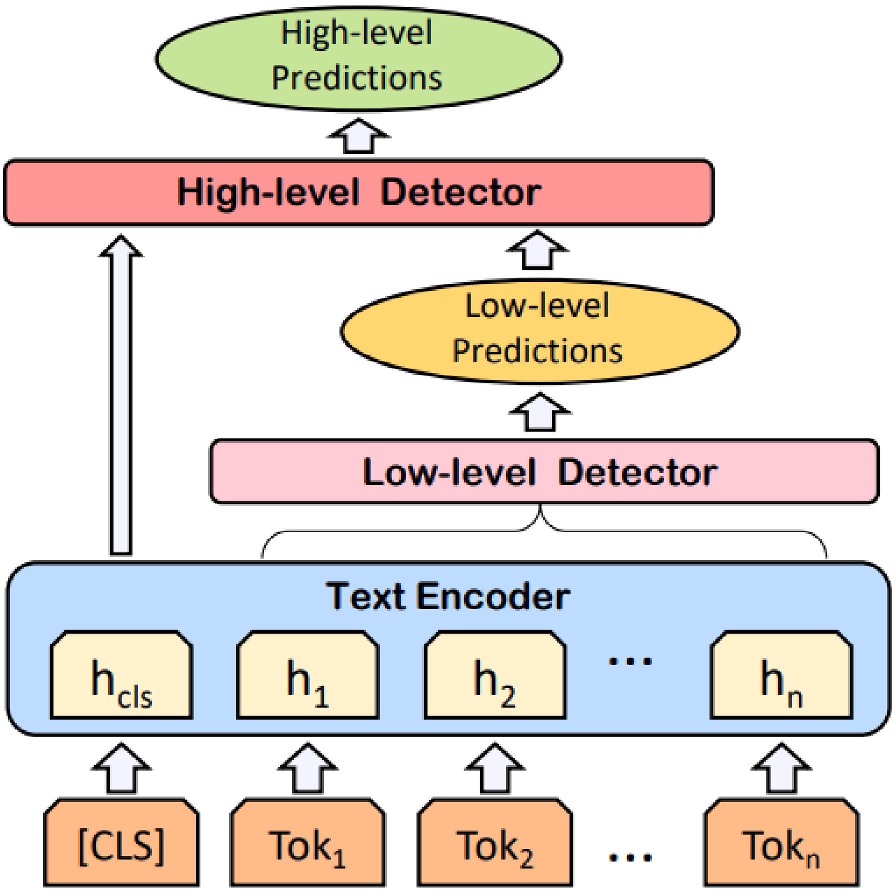


Figure 1: Overview of the event-driven trading strategy



- **experiment**:

Model	TAE (1)			TAB (1)	
	Win Rate	Ave. Return	Exc. Returns	Ave. Return	Num. Trans.
Vader (Gilbert, 2014)	52.8‖24.3%	0.06%	-$8116	1.72%	4327
BERT-SST (0.995)	54.3‖26.9%	0.45%	$3743	2.94%	2378
BERT-SST (0.9)	52.9‖26.1%	0.41%	$44049	3.31%	15663
Sentence (Jacobs et al., 2018)	**55.5**‖30.9%	1.37%	$54064	7.21%	2881
BERT-CRF	53.7‖33.8%	1.60%	$83120	8.87%	3533
Bi-level Detection (ours)	54.5‖**34.2%**	**1.74%**	**$84443**	**9.11%**	3118

Table 2: This table shows the 1-day trading result, in which we start each transaction at the news article's publish time and end the transaction after 1 trading day. **Win Rate** stands for the overall winning rate (rate of transactions that have a return over 0) ‖ big win rate (rate of transactions that have a return over 1%). **Ave. Return** stands for the average return on each transaction. **Exc. Return** stands for the total excess returns over the market when starting with $10000 and invest $2000 to each detected trading signal. **Num. Trans.** stands for the number of transactions (valid trading signals) of each model.

Model	TAE (2)			TAB (2)	
	Win Rate	Ave. Return	Exc. Returns	Ave. Return	Num. Trans.
Vader (Gilbert, 2014)	53.7‖36.9%	0.38%	-$2551	3.11%	4327
BERT-SST (0.995)	**53.8**‖38.4%	0.62%	$8479	4.72%	2378
BERT-SST (0.9)	52.3‖37.2%	0.46%	$12802	3.93%	15663
Sentence (Jacobs et al., 2018)	52.7‖39.9%	1.24%	$24673	9.49%	2881
BERT-CRF	51.3‖39.9%	1.39%	$52891	11.27%	3533
Bi-level Detection (ours)	52.3‖**40.8%**	**1.56%**	**$59375**	**11.53%**	3118

Table 3: This table shows the 2-day trading result, in which we start each transaction at the news article's publish time and end the transaction after 2 trading day.

remark:

```
文章看起来还可以，但他的数据集太大了，压缩包都1.5GB，感觉可能会对我的复现和训练带来麻烦
```

link:

- **GitHub code**:https://paperswithcode.com/paper/trade-the-event-corporate-events-detection
- **Paper with Code**:https://paperswithcode.com/paper/trade-the-event-corporate-events-detection

GAN(GRU+CNN)&pytorch（一般）：

```
Stock price prediction using Generative Adversarial Networks
```

- **author&date:**:journal of computer science 2021 & 哥大
- **model**:GAN GRU CNN
- **data**:

  ```
  AAPL_daily_stock_data,with features such as S&P 500 index, NASDAQ Composite index, U.S. Dollar index, etc
  ```

- **estimation**:思路似乎可以，但是实验结果很不喜欢，看起来就很奇怪
- **framework**:pytorch
- **news relevant**:不相关

- **insight**:

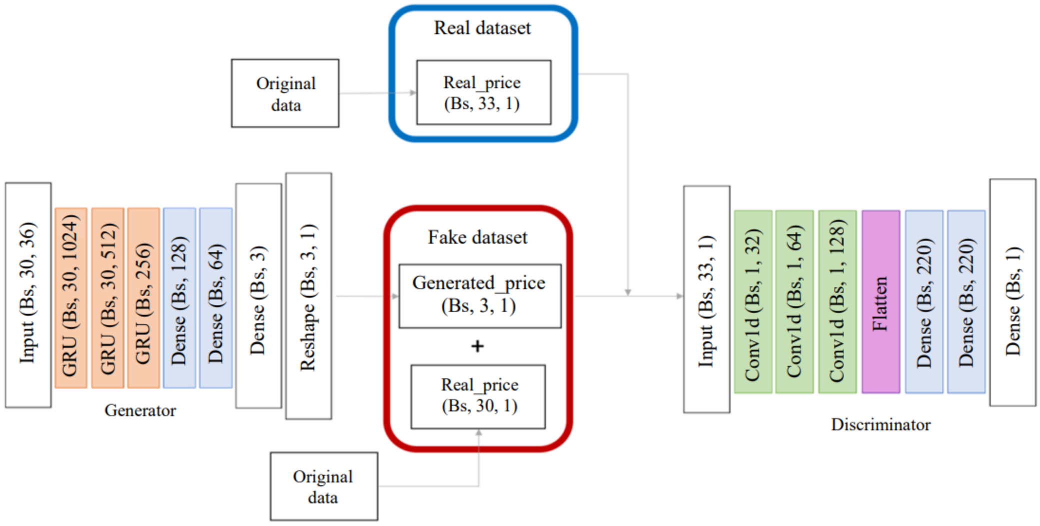


Fig. 1: GAN Architecture

- **experiment**:

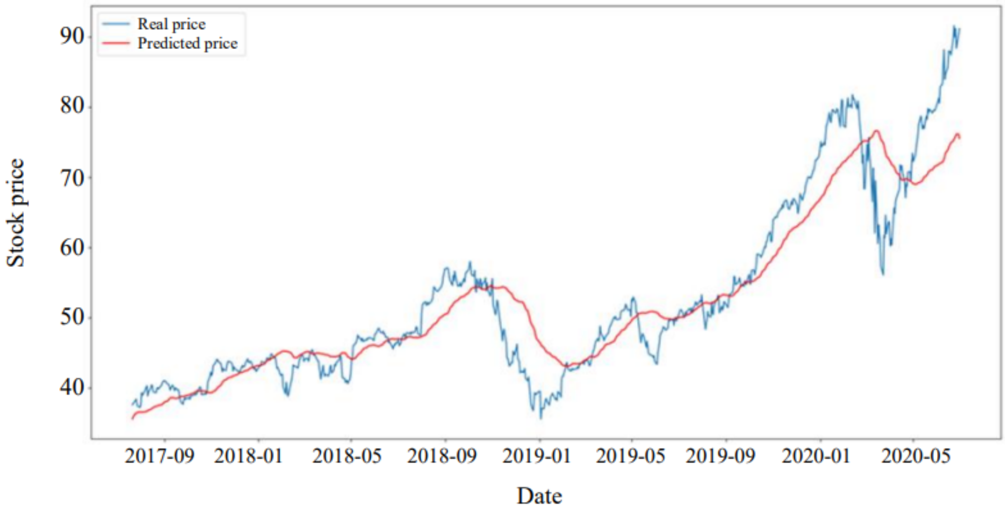


Fig. 9: WGAN-GP test data plot

remark:

```
abstract中提到：用GRU GAN作为generator，为什么用CNN作为判别器来，我其实没太懂
不过后来我问GPT它告诉我这个从理论上是可行的，
这个文章是对抗网络的思路可以借鉴
同时他对数据加入的一些人工特征也给我了一些启发，
加入大盘的指数或许可以反应一些总体涨跌的信息，理论上可以帮助训练
```

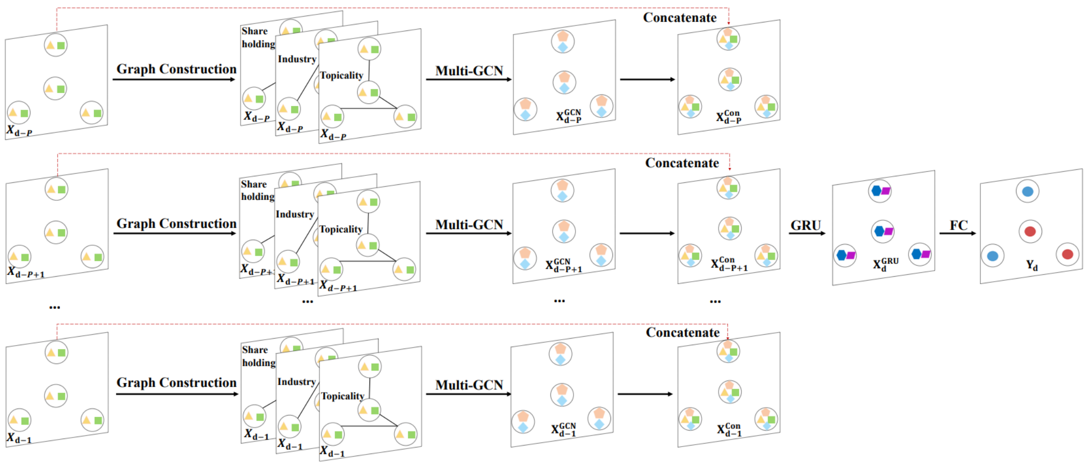
link:

- **GitHub code**:https://github.com/ChickenBenny/Stock-prediction-with-GAN-and-WGAN
- **Paper with Code**:https://paperswithcode.com/paper/stock-price-prediction-using-generative

GRU+GCN&tf（还行）：

```
Multi-Graph Convolutional Network for Relationship-Driven Stock Movement Prediction
```

- **author&date:**:2020.5 & 中科院大学
- **model**:GCN+GRU
- **data**:daily_stock_data+price/volume_moving avg5 10+(p)return
- **estimation**:GCN+时序模型的思路确实很不错哦
- **framework**:tensorflow
- **news relevant**:不相关
- **insight**:



- 建立图的方法：首先根据金融领域知识将股票之间的多个关系编码为图，并利用GCN基于这些预定义图提取交叉效应
- **experiment**:

TABLE II
THE EXPERIMENTAL RESULTS

Input Feature	Models	CSI300 Accuracy	CSI300 Precision	CSI300 Recall	CSI300 F1	CSI300 MCC	CSI500 Accuracy	CSI500 Precision	CSI500 Recall	CSI500 F1	CSI500 MCC
Historical Records	LR	0.5145	0.9746	0.5133	0.6724	0.0228	0.5149	0.9723	0.5148	0.6732	0.0117
	SVM	0.5197	0.9498	0.5165	0.6691	0.0412	0.5253	0.9662	0.5202	0.6763	0.0636
	RF	0.5375	0.9298	0.5271	0.6728	0.0957	0.5433	0.9900	0.5294	0.6899	0.1587
	ANN	0.5191	0.9724	0.5158	0.6740	0.0463	0.5202	0.9900	0.5170	0.6792	0.0576
	LSTM	**0.5435**	0.9756	0.5291	0.6861	0.1443	**0.5461**	0.9662	0.5318	0.6860	0.1384
Historical Records & Corporation Relationships	GCN-S	0.5472	0.9609	0.5317	0.6845	0.1421	0.5463	0.9675	0.5423	0.6950	0.0717
	GCGRU-S	0.5505	0.9321	0.5346	0.6795	0.1338	0.5521	0.9635	0.5458	0.6969	0.0938
	GCGRU-I	0.5598	0.9561	0.5392	0.6895	0.1739	0.5678	0.9814	0.5540	0.7082	0.1655
	GCGRU-T	0.5628	0.9512	0.5412	0.6899	0.1782	0.5751	0.9837	0.5581	0.7122	0.1916
	GCGRU-D	0.5602	0.9442	0.5402	0.6871	0.1667	0.5697	0.9844	0.5549	0.7097	0.1756
	Multi-GCGRU	**0.5754**	0.9603	0.5484	0.6981	0.2171	**0.5885**	0.9894	0.5658	0.7199	0.2377

TABLE III
MULTI-GCGRU WITH DIFFERENT LAG SIZES

Length	CSI300 ACC	CSI300 MCC	CSI500 ACC	CSI500 MCC
3-days	0.5623	0.1513	0.5752	0.1742
5-days	0.5754	0.2171	0.5885	0.2377
7-days	**0.5790**	0.2196	**0.5901**	0.2821
9-days	0.5769	0.1869	0.5783	0.1965
11-days	0.5705	0.1378	0.5691	0.1221

remark:

```
GCN+GRU的想法是很不错，并且这篇文章讨论的是accuracy
但是实验数据里面RNN的accuracy为什么这么低
另外，这篇文章用的是tensorflow，
btw,这篇文章建立图的方式可以借鉴，我记得之前也看过GCN+RNN的模式
这篇文章的feature启发：price/volume_moving avg5 10
```

link:

- **GitHub code**:https://github.com/start2020/Multi-GCGRU
- **Paper with Code**:https://paperswithcode.com/paper/multi-view-graph-convolutional-networks-for

Adversarial A-LSTM&tf（不错）：

```
Enhancing Stock Movement Prediction with Adversarial Training
```

- **author&date:**:2018.10 & USTC的老师
- **model**:Adversarial Attention Lstm
- **data**:

  ```
  ACL18 dataset
  ACL18 contains historical data from Jan-01-2014 to Jan01-2016 of 88 high-trade-volume-stocks in NASDAQ and NYSE markets
  KDD17 dataset
  KDD17 contains a longer history ranging from Jan-01-2007 to Jan-01-2016 of 50 stocks in U.S. markets
  ```

  -**estimation**:效果看起来一般但是这毕竟是18年的，这个文章的思路很值得学习，效果一般的原因在remark里
- **framework**:tf
- **news relevant**:不相关

- **insight**:

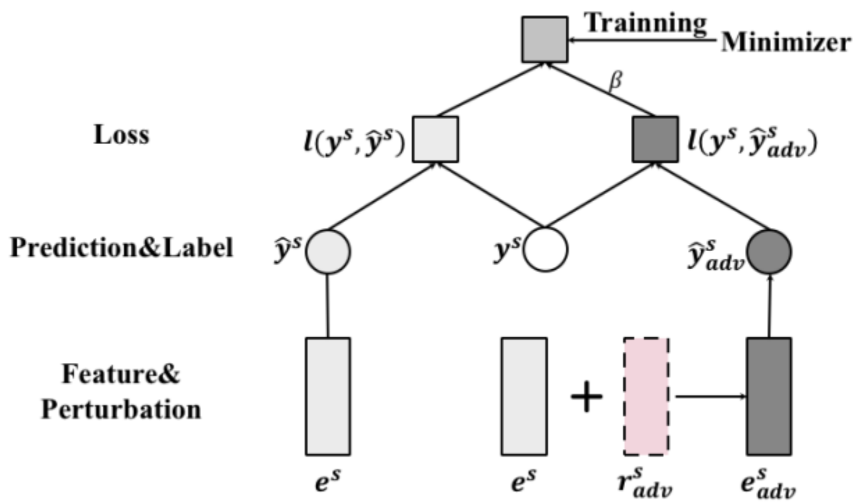


Figure 3: Illustration of the *Adversarial Attentive LSTM*.

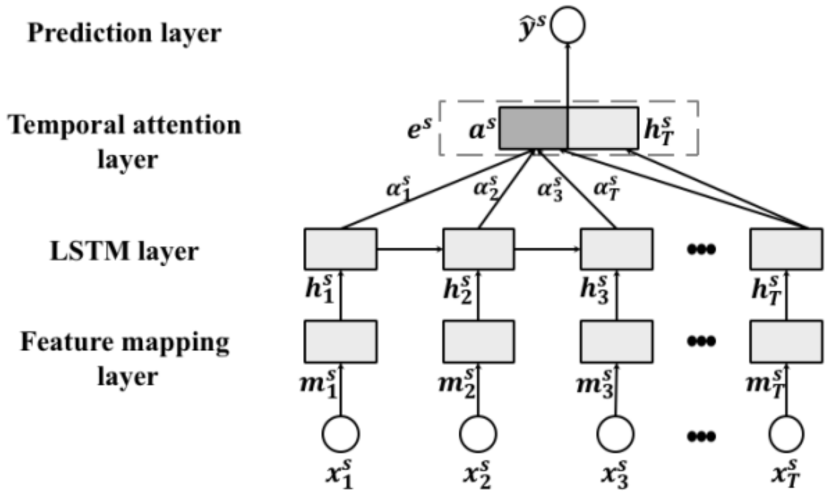


Figure 2: Illustration of the *Attentive LSTM*.

- **experiment**:

Table 2: Performance comparison on the two datasets.

Method	ACL18		KDD17	
	Acc	MCC	Acc	MCC
MOM	47.01±—	-0.0640±—	49.75±—	-0.0129±—
MR	46.21±—	-0.0782±—	48.46±—	-0.0366±—
LSTM	53.18±5e-1	0.0674±5e-3	51.62±4e-1	0.0183±6e-3
ALSTM	54.90±7e-1	0.1043±7e-3	51.94±7e-1	0.0261±1e-2
StockNet	54.96±—	0.0165±—	51.93±4e-1	0.0335±5e-3
Adv-ALSTM	**57.20**±—	**0.1483**±—	**53.05**±—	**0.0523**±—
RI	4.02%	42.19%	2.14%	56.12%

RI denotes the relative improvement of **Adv-ALSTM** compared to the best baseline. The performance of **StockNet** is directly copied from [Xu and Cohen, 2018].

Table 3: Performance of **Rand-ALSTM** on the two datasets.

Datasets	Acc	MCC
ACL18	55.08±2e0	0.1103±4e-2
KDD17	52.43±5e-1	0.0405±8e-3

remark:

```
思路很好，值得借鉴阅读论文。毕竟是这个领域比较老的文章
为什么效果一般？可能是因为数据集的问题，
一般而言我们对某一只股票数据训练一个用于预测他自己的首收益率的模型
如果把很多只股票放在一起，其之间的相互作用关系没有被考虑进来，简单地训练时序模型估计不太行
可行需要GCN等来辅助提取信息
```

link:

- **GitHub code**:https://github.com/yuxiangalvin/Stock-Move-Prediction-with-Adversarial-Training-Replicate
- **Paper with Code**:https://paperswithcode.com/paper/enhancing-stock-movement-prediction-with

GCN+LSTM&tf&pytorch（给出股票之间的关系度量）：

```
Temporal Relational Ranking for Stock Prediction
```

- **author&date:**:2018.9 USTC老师
- **model**:重点是股票之间的关系和排名的度量等：GCN+LSTM
- **data**:NYSE
- **estimation**:思路非常值得借鉴
- **framework**:pytorch & tf
- **news relevant**:不相关，但是和行业相关
- **insight**:

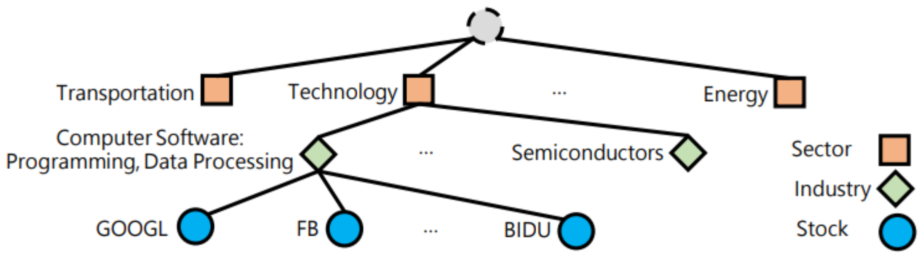


Fig. 3. Illustration of ICB's sector-industry classification hierarchy.

remark:

```
1）为股票定制深度学习模型排名
2）以时间敏感的方式捕获股票关系
这两点很重要，给以进行推广
```

link:

- **GitHub code**:https://paperswithcode.com/paper/temporal-relational-ranking-for-stock
- **Paper with Code**:https://paperswithcode.com/paper/temporal-relational-ranking-for-stock

LSTM+VADER+DP&tf（结果很离谱，但思想不错）：

```
DP-LSTM: Differential Privacy-inspired LSTM for Stock Prediction Using Financial
News
```

- **author&date:**:2019.12 哥大 & 北理工
- **model**:ARMA+LSTM+VADER+'Differential Privacy'
- **data**:daily_stock_data+news(数据压缩包太大了，还没下下来看)
- **estimation**:思路不错，论文结果离谱或者我没看懂
- **framework**:tf
- **news relevant**:相关
- **insight**:ARIMA+LSTM+VADER+DP
- **experiment**:

Method	Mean MPA
LSTM without news	0.978305309
LSTM with news	0.978366682
DP-LSTM	**0.981582666**

Table 1: Predicted Mean MPA results.

Evaluation

idate our DP-LSTM based on the S&P 500 stocks
PA) to evaluate the proposed methods, which is de

$$\text{MPA}_t = 1 - \frac{1}{L}\sum_{\ell=1}^{L}\frac{|X_{t,\ell} - \hat{X}_{t,\ell}|}{X_{t,\ell}},$$

remark:

```
文章的结果是accuracy可以到0.9几，很离谱，
感觉他是不是混淆了mean prediction accuracy和accuracy
虽然我也不知道MPA为什么这么定义，但我感觉这个加入新闻的思路很值得借鉴
不过感觉和news沾边的数据量真的好大
```

link:

- **GitHub code**:https://github.com/Xinyi6/DP-LSTM-Differential-Privacy-inspired-LSTM-for-Stock-Prediction-Using-Financial-News
- **Paper with Code**:https://paperswithcode.com/paper/dp-lstm-differential-privacy-inspired-lstm

LSTM+RF&tf（还行）：

```
S&P 500 Stock Price Prediction Using Technical, Fundamental and Text Data
```

- **author&date:**:2021.8 & 南卡
- **model**:LSTM+Random Forest
- **data**:

  ```
  weekly historical prices, finance reports, and text information from news items
  ```

- **estimation**:

  ```
  accuracy非常好，思路清晰，但是你用的是weekly的数据....
  ```

- **framework**:tf
- **news relevant**:相关
- **experiment**:

Author	model refresh and predicting period	Predicting	Predicted accuracy	Percentage of time stock price increased in test set	Improved accuracy compared to model only predicting upwards
Jiang et al.[21] for S&P 500 index	no refresh, 09/01/2012-04/01/2019	closing on next month	69.17%	67.35%	1.82%
Yu an Yan[43] for S&P 500 index	yearly update, 01/01/2010-12/29/2017	closing on next day	58.07%	54.59%	3.48%
Ding et al.[12] for S&P 500 index	no refresh, 02/22/2013- 11/21/2013	closing on next day	64.21%	59.68%	4.53%
Gorenc Novak and Velušček[17] for 370 S&P stocks	20 days update, 10/27/2005-06/14/2013	high on next day	61.16%	57.07%	4.09%
Our model for S&P 500 index	monthly & yearly mixed 01/01/2003-12/31/2019	closing on next week	66.18%	60.65%	**5.53%**
Our model for 518 S&P stocks	monthly & yearly mixed 01/01/2003-12/31/2019	closing on next week	62.12%	55.49%	**6.63%**

Table 11: Comparison of the improvement in DA among different models. Our model for the S&P 500 index was fitted using the same ensemble method but with the median of the individual stock model outputs.

remark:

```
结果很好，可惜是weekly的数据，
news+fundamental+technical的数据准备很值得借鉴
只是用的是tf，要是用pytorch就好了
```

link:

- **GitHub code**:https://github.com/Shanlearning/SP-500-Stock-Prediction
- **Paper with Code**:https://paperswithcode.com/paper/s-p-500-stock-price-prediction-using

VAETransfomer&pytorch（不错）：

```
FactorVAE: A Probabilistic Dynamic Factor Model Based on Variational Autoencoder
for Predicting Cross-Sectional Stock Returns
```

- **author&date:**:2022 & 清华
- **model**:DFM + VAE + Transfomer
- **data**:daily_stock_data+rank选股任务
- **estimation**:

> 看起来不错，思想可以学习学习，可以试试在个股上的正确率怎么样

- **framework**:pytorch
- **news relevant**:不相关
- **insight**:

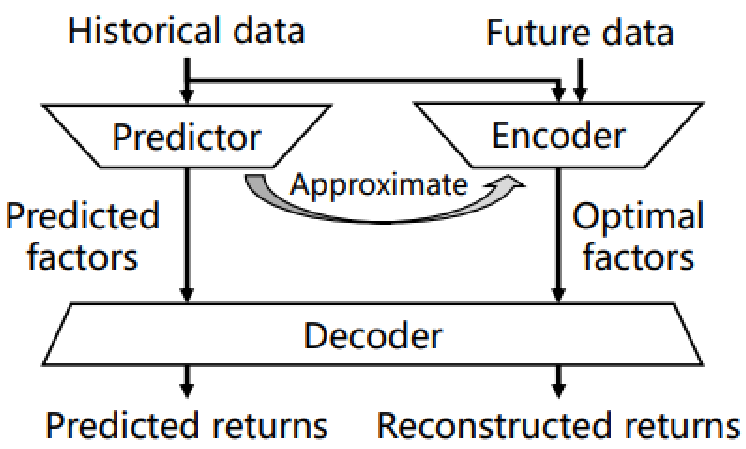


Figure 1: Brief illustration of FactorVAE

- **experiment**:

Methods	m=50		m=100		m=200	
	Rank IC	Rank ICIR	Rank IC	Rank ICIR	Rank IC	Rank ICIR
GRU	0.031(0.005)	0.184(0.029)	0.030(0.004)	0.234(0.030)	0.031(0.004)	0.282(0.032)
ALSTM	0.027(0.004)	0.162(0.022)	0.028(0.007)	0.210(0.045)	0.026(0.005)	0.237(0.041)
GAT	0.029(0.008)	0.166(0.043)	0.023(0.011)	0.176(0.085)	0.025(0.009)	0.215(0.071)
Trans	0.034(0.007)	0.201(0.040)	0.034(0.006)	0.259(0.043)	0.033(0.003)	0.302(0.023)
SFM	0.037(0.007)	0.220(0.042)	0.038(0.004)	0.294(0.035)	0.038(0.003)	0.342(0.036)
linear	0.018(0.005)	0.138(0.075)	0.018(0.005)	0.147(0.044)	0.018(0.004)	0.176(0.042)
CA	0.038(0.008)	0.215(0.046)	0.039(0.004)	0.284(0.034)	0.039(0.003)	0.328(0.027)
FactorVAE-port	0.043(0.005)	0.241(0.022)	0.039(0.003)	0.272(0.005)	0.041(0.004)	0.328(0.011)
FactorVAE	**0.053(0.007)**	**0.299(0.039)**	**0.056(0.002)**	**0.384(0.044)**	**0.050(0.008)**	**0.399(0.063)**

Table 2: The robustness of the compared methods.

remark:

> 其实我没看明白这个`future_data`是怎么放进去的
> 这个文章改进`Transformer`的思路很值得学习

link:

- **Paper with Code**:https://paperswithcode.com/paper/factorvae-a-probabilistic-dynamic-factor

##

下面都是不被看好的文章，但还是记录一下为什么不行或者他的优点

Title（很久远的文章，看看理论吧）：

```
Automatic Relevance Determination in Nonnegative Matrix Factorization with the β-Divergence
```

- **data**:'the swimmer dataset'
- **framework**:pytorch但是2011年

remark:

```
一种具有beta散度的半正定矩阵分解，
或许在我日后需要给文章加点数学的时候用到
```

link:

- **Paper with Code**:https://paperswithcode.com/paper/automatic-relevance-determination-in

Title（不好，但论文的结构很可取）：

```
Stock Price Prediction Based on Natural Language Processing
```

- **author**:2022 & 贸大、西财
- **experiment**:

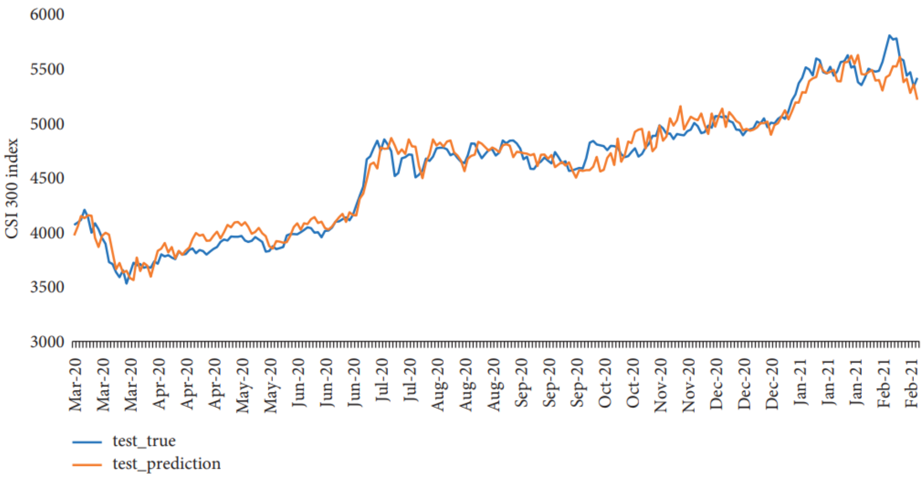


FIGURE 7: The trend chart of the relationship between LSTM model predicted values of Generated vocabulary and CSI 300 index true values. The red and blue lines represent the true and predicted values, respectively ($n = 243$).

- **framework**:Mindspore5

remark:

```
用RMSE作为指标是反面教材，但文章的结构和写作很清晰，很值得借鉴
另外，文章的框架是华为的MindSpore，我也是第一次见，有时间可以学习一下
```

link:

- **Paper with Code**:https://paperswithcode.com/paper/stock-price-prediction-based-on-natural

Title（不看好，但网络设计思路可取）：

```
Particle Filter Recurrent Neural Networks
```

- **author**:2019.3 & 新国立
- **framework**:pytorch
- **data**:原问题是机器人相关的，对也股票涨跌进行了预测，就是结果离谱
- **insight**:

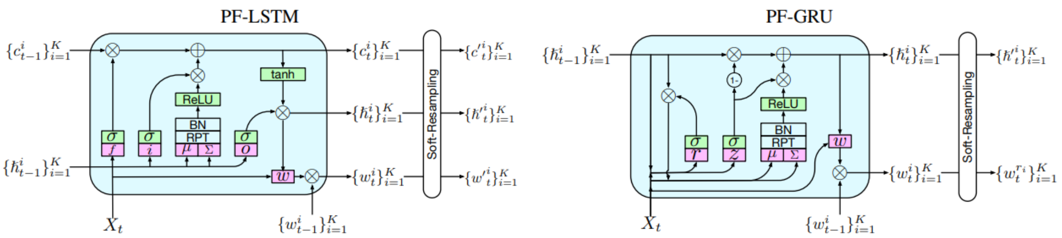


Figure 3: **PF-LSTM and PF-GRU network architecture**. Notation (1) green box: activation (2) pink box: learned function (3) *RPT*: reparameterization trick (4) *BN*: batch normalization.

- **experiment**:

Table 2: Prediction Accuracy (%)

	LPA	AREM	GAS	MR	R52	UID
LSTM	91.7	99.1	76.6	73.1	81.1	93.2
PF-LSTM	**100**	99.1	89.9	78.3	89.1	95.3
LSTM-best	98.3	100	81.2	73.1	81.1	99.1
PF-LSTM-best	**100**	**100**	**94.1**	**82.2**	**91.3**	**99.6**
GRU	97.8	98.4	76.8	75.1	84.2	96.1
PF-GRU	98.1	99.1	**83.3**	76.2	87.2	94.1
GRU-best	97.8	**100**	80	75.2	84.2	99
PF-GRU-best	**99.2**	**100**	**83.3**	**79.6**	**89.1**	**99.5**
SOTA	91.3	94.4	80.9	**83.1**	**93.8**	-

AREM, and GAS. For NASDAQ, R52 and MR, SOTAs (Qin et al. 2017; Zhou et al. 2016; Cardoso-Cachopo 2007) use complex network structure designed specifically for the task, thus they work better than PF-RNNs. Future work may investigate PF-RNNs with larger, more complex network components, which would provide a more fair comparison with the SOTA.

remark:

```
文章构建模型的思路可取，大体是在RNN里加入了一些权重的东西
但是结果离谱，accuracy可以到0.9几，我觉得可能是他的分类标准和一般的不同
```

link:

- **Paper with Code**:https://paperswithcode.com/paper/particle-filter-recurrent-neural-networks

Title（和ai关系不大，传统alpha因子相关）：

```
Trader Company Method: A Metaheuristic for Stock Selection
```

remark:

```
日本人搞的传统交易算法的开发，对毕设没什么帮助有兴趣可以开阔开阔眼界
```

link:

- **Paper with Code**:https://paperswithcode.com/paper/trader-company-method-a-metaheuristic-for

Title（不看好，想水论文要跟这个学）：

```
Multi-step-ahead Stock Price Prediction Using Recurrent Fuzzy Neural Network and
Variational Mode Decomposition
```

remark:用了离散余弦变换处理数据
link:

- **Paper with Code**:https://paperswithcode.com/paper/multi-step-ahead-stock-price-prediction-using

Title（不看好，大模型相关）：

- **data**:eg:有ACL18 dataset等
- **remark**:

  基于LLM的微调的大模型，有点意思，先记录一下，图一个乐
  link:
- **GitHub code**:https://github.com/chancefocus/pixiu
- **Paper with Code**:https://paperswithcode.com/paper/pixiu-a-large-language-model-instruction-data

Title（拿不准）：

```
Detach-ROCKET: Sequential feature selection for time series classification with
random convolutional kernels
```

- **author**:2023.9 瑞典皇家理工学院
- **data**:他没有给出dataset,只给了一个py文件，加载data
- **model**:ROCKET+特征选择' Detach-ROCKET'
- **experiment**:

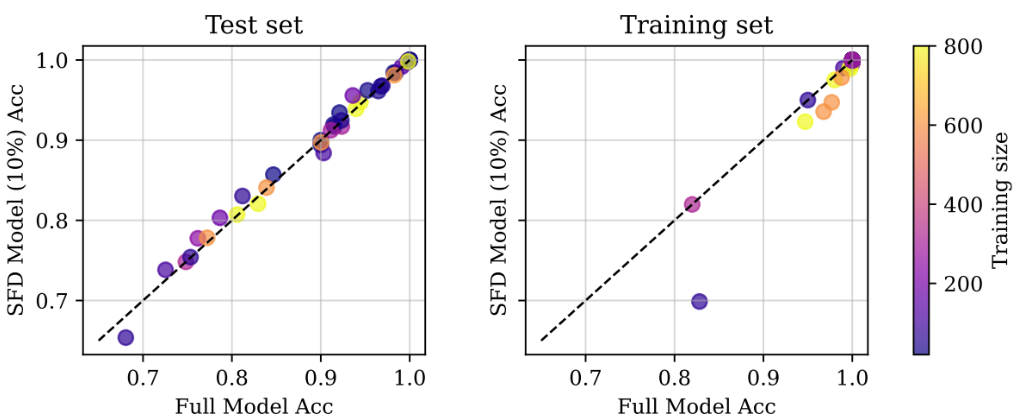


Figure 3: Accuracy (Acc) comparison between the pruned models with 10% of retained features and the full ROCKET model, for both test and training sets. Each point depicts the average accuracy of one of the 42 datasets over 25 realizations. The color indicates the training size of the dataset, with the color range saturated at 800 instances for enhanced contrast.

remark:

```
论文提出改进的ROCKET(RadOm Convolutional KErnel Transform)方法，用于时间序列分类
说他是RNN更好的模型，这个思路不错，我可以借鉴一下
```

link:

- **GitHub code**:https://github.com/gon-uri/detach_rocket
- **Paper with Code**:https://paperswithcode.com/paper/detach-rocket-sequential-feature-selection

Title（不看好）：

```
Learning Multiple Stock Trading Patterns with Temporal Routing Adaptor and
Optimal Transport
```

- **data**:他是写了两个py文件，我也没看懂他的data从哪来的
- **insight**:

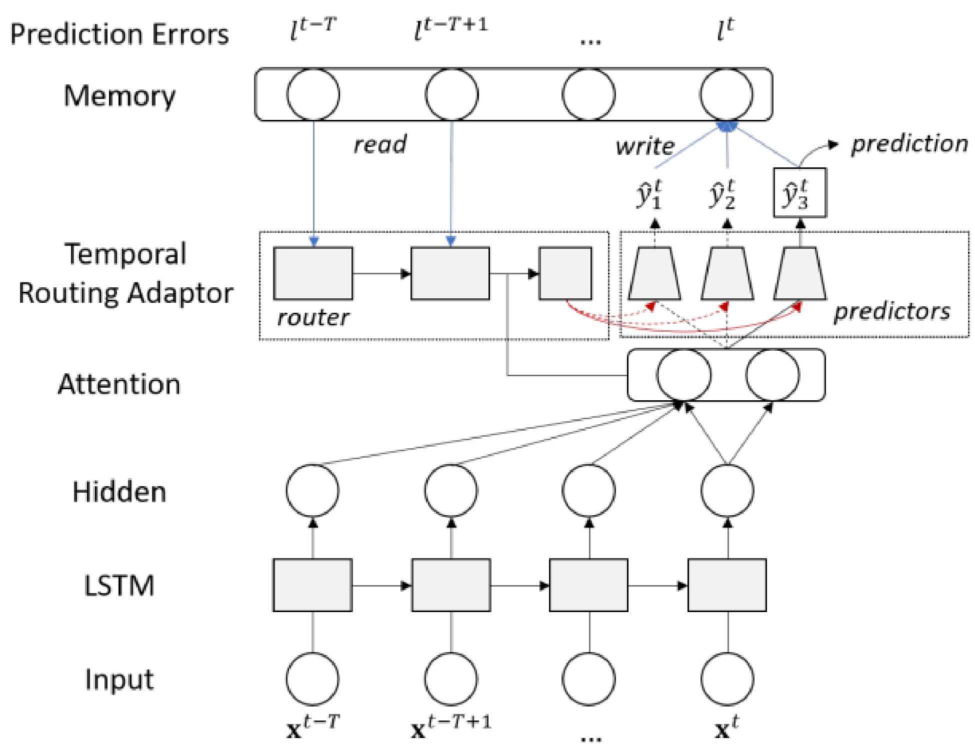


Figure 4: An example to use Temporal Routing Adaptor (TRA) as an extension module on Attention LSTM. The router uses both latent representation from Attention LSTM and temporal prediction errors stored in memory to select the best predictor for the current sample.

- **experiment**:

Table 1: Stock ranking and portfolio performance of the compared methods. For MSE and MAE, we also report the standard deviation of 5 random seeds as $(\cdot)$. ↑ means the larger the better while ↓ means the smaller the better.

Method	Ranking Metrics				Portfolio Metrics			
	MSE (↓)	MAE (↓)	IC (↑)	ICIR (↑)	AR (↑)	AV (↓)	SR (↑)	MDD (↓)
Linear	0.163	0.327	0.020	0.132	-3.2%	16.8%	-0.191	32.1%
LightGBM	0.160(0.000)	0.323(0.000)	0.041	0.292	7.8%	15.5%	0.503	25.7%
MLP	0.160 (0.002)	0.323 (0.003)	0.037	0.273	3.7%	15.3%	0.264	26.2%
SFM	0.159 (0.001)	0.321 (0.001)	0.047	0.381	7.1%	14.3%	0.497	22.9%
ALSTM	0.158 (0.001)	0.320 (0.001)	0.053	0.419	12.3%	**13.7%**	0.897	**20.2%**
Trans.	0.158 (0.001)	0.322 (0.001)	0.051	0.400	14.5%	14.2%	1.028	22.5%
ALSTM+TS	0.160 (0.002)	0.321 (0.002)	0.039	0.291	6.7%	14.6%	0.480	22.3%
Trans.+TS	0.160 (0.004)	0.324 (0.005)	0.037	0.278	10.4%	14.7%	0.722	23.7%
ALSTM+TRA (Ours)	**0.157 (0.000)**	**0.318 (0.000)**	**0.059**	**0.460**	12.4%	14.0%	0.885	20.4%
Trans.+TRA (Ours)	**0.157 (0.000)**	0.320 (0.000)	0.056	0.442	**16.1%**	14.2%	**1.133**	23.1%

remark:

```
时间序列模型加了一个：类似attention的东西，我要是想水论文就跟着这个的思路走
这个的代码是基于微软的qlib平台
此外，作者挺实诚，把自己的困惑写在github上了
对于这些困惑：我或许也会遇到，或者这些也是很好的点子
```

Todo

- [x] 发现模型层、控制层是比较容易剥离的，比较麻烦的是数据层
- [x] QLib.init() 这个函数没有也可以。其中传入的 `provider_uri`，`region` 并没有造成什么影响。
- [x] tra 部分，测试的时候，gumbel softmax 加权求和的方式与训练的时候不同
- [ ] 搞清楚 memory 机制，memory 写入的 index 是输入区间的右端点，但读出的时候，读的是左边 39 天的误差。另外在训练、验证、测试的过程中 memory 都一直在变，会不会造成影响。在训练过程中，memory.clear() 会造成什么影响？
- [ ] 训练过程，数据的格式是 (B, 60, 16); 测试过程中，数据的格式是 (800, 60, 16), index 是怎么对上去的？
- [ ] label 为什么是一个浮点数，代表什么？似乎是代表 ranking 的累计百分比

link:

- **GitHub code**:
- **Paper with Code**:

Title（不看好吗，你没代码，效果固然好）：

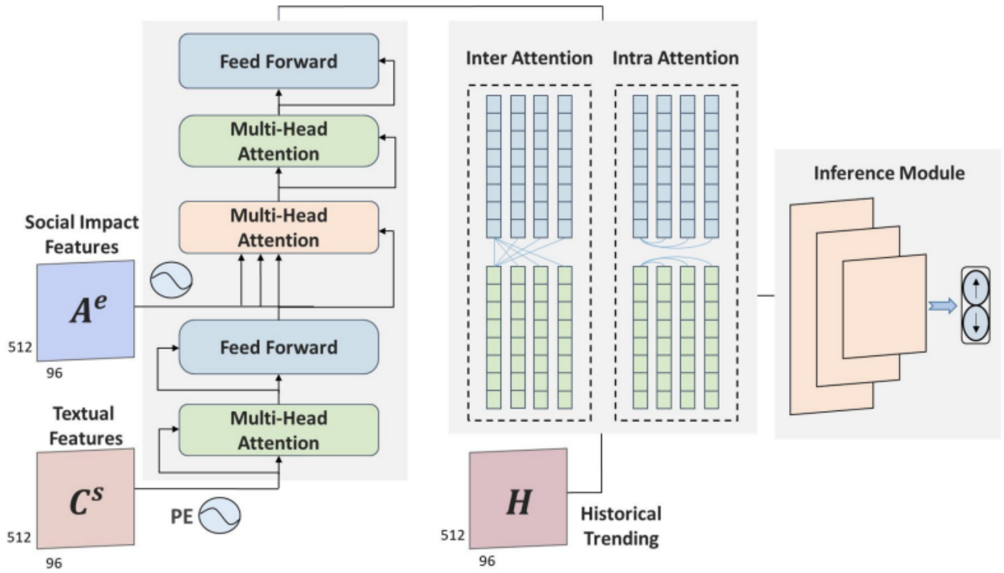
```
（你accuracy非常好，但是你github上没有code只有data，你的构建model的思路值得我好好学习借鉴）
Multi-modal Attention Network for Stock Movements Prediction
```

- **data**news+很奇怪的stock_data
- **experiment**:

Model	Acc.	MCC
RF	53.13%	0.0129
HAN	55.96%	0.0447
StockNet	57.35%	0.0621
Adv-LSTM	57.93%	0.0672
MHACN	58.84%	0.0721
CapTE	59.87%	0.0976
Ours-MMAN-oH	56.67%	0.0583
Ours-MMAN-oC	59.49%	0.0737
Ours-MMAN-nA	60.06%	0.0850
Ours-MMAN-nH	60.46%	0.0937
Ours-MMAN	**61.20%**	**0.1193**

Table 2: Performance of baselines and MMAN variations in accuracy and MCC.

insight:



remark:

```
没代码，model是Transformer的改版，数据集很奇怪，但是效果很好，accuracy可以到61%
```

link:

- **GitHub code**:https://github.com/HeathCiff/Multi-modal-Attention-Network-for-Stock-Movements-Prediction
- **Paper with Code**:https://paperswithcode.com/paper/multi-modal-attention-network-for-stock

Title（不看好）：

```
Attention-based CNN-LSTM and XGBoost hybrid model for stock prediction
```

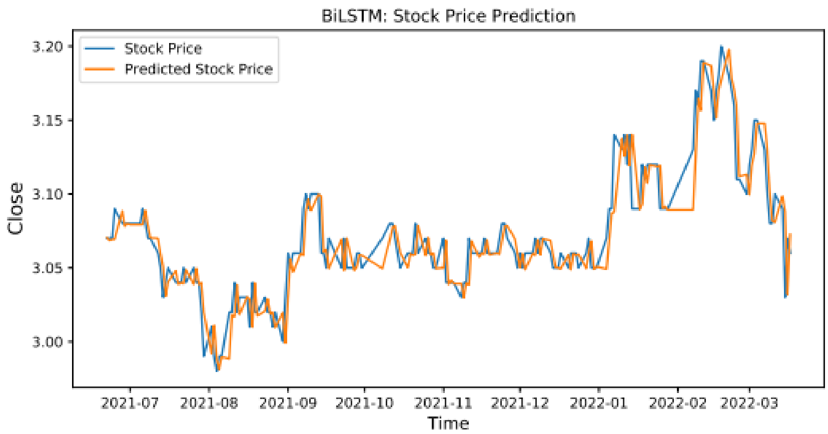
- **data**:形如

  ts_code,trade_date,open,high,low,close,pre_close,change,pct_chg,vol,amount,turnover_rate,volume_ratio,pe,pb,ps,total_share,float_share,free_share,total_mv,circ_mv

- **abstract**:

  ARIMA处理数据，cnn-lstm是总体，xgboost微调，预测股票价格，框架tensorflow

- **experiment**:这个预测股价基本就是胡闹了



remark:

```
预测股价看起来就很不靠谱，但看在你github上star比较多我还是记录一下，要是不选这个也记录个理由
CNN-LSTM+XGBoost的结构我还是第一次见，有水文章嫌疑
CNN-LSTM是胡闹吗？gpt答：不是，CNN-LSTM是CNN和LSTM的结合，CNN用于提取局部特征，LSTM用于提取序列特征
另外我从直观上来讲，你这个cnn作用于feature是高开低收的股票数据没有任何理由
```

link:

- **GitHub code**:https://github.com/zshicode/attention-clx-stock-prediction
- **Paper with Code**:https://paperswithcode.com/paper/attention-based-cnn-lstm-and-xgboost-hybrid
- **同一个作者的文章（关于GCN和股票预测）**：https://paperswithcode.com/paper/differential-equation-and-probability

##

下面是Summary的模板：

Title:

- **author&date:**:
- **model**:
- **data**:
- **estimation**:
- **framework**:
- **news relevant**:
- **insight**:
- **experiment**:

remark:

link:

- **Paper with Code**:

Title（不看好）：

- **data**:形如
- **experiment**:

remark:

link:

- **Paper with Code**: