



Paper Review : Accurate Multivariate Stock Movement Prediction via Data-Axis Transformer with Multi-Level Contexts.

Tae-Won Lee,
leetee95@gmail.com
2021. 11. 02.

TO DO list

- 지난주

- Accurate Multivariate Stock Movement Prediction via Data-Axis Transformer with Multi-Level Contexts 논문에 관한 조사.

- 구할 수 있는 데이터셋 list up

- 이번주

- Dataset에 대한 목록과 상세한 정보

- 데이터셋 보유

- 논문 구현

- 성능향상에 대한 해결책

Introduction

- Univariate time series는 **하나의 time-dependent variable**를 가지고 있는데 비해 Multivariate time series 은 최소한 **두개 이상의 time-dependent variable**을 가지고 있음.



< Multivariate time series 의 예시 >

- Time series prediction은 **전력량 예측, 주가예측, 수요 예측** 등의 문제에서 이용되는 중요한 문제임.
- Univariate time series data를 이용한 예측은 예측 시점 **이전의 정보(Seasonality, Tendency, cycle)**를 이용해 다음 시점의 예측을 진행하지만, Multivariate time series는 연관성이 있다고 판단되는 time series data간의 **Correlation**이용해 다음 값들을 예측함.

Introduction

- 대부분의 주식 종목들은 각각의 sector에 속함.
- Previous paper
 - sector의 **고정된 list**를 가지고 있고, sector의 **pre-trained list에 의존함**.
 - 시간에 따라 자연적으로 변하는 주식 상관의 **동적인 속성을 잃음**.
 - sector에 없는 주식이나, sector가 모호한 주식에 대해 예측 모형이 적용되지 않음.
 - 예측의 품질이 **sector 정보의 품질에 크게 의존됨**.
- Univariate time series data를 이용한 예측은 예측 시점 **이전의 정보(Seasonality, Tendency, cycle)**를 이용해 다음 시점의 예측을 진행하지만, Multivariate time series는 연관성이 있다고 판단되는 time series data간의 **Correlation**이용해 다음 값들을 예측함 .

Related works

- **Correlated Stock prediction**

Statistical method

Vector Autoregression (VAR) [1]

$$y_t = c + A_1 y_{t-1} + A_2 y_{t-2} + \dots + A_p y_{t-p} + e_t$$

- VAR은 확률적 프로세스 모델의 한 유형으로 Multivariate time series 를 이용하여 단일 변수 Autoregression 모델을 일반화 함.

- 통계적인 모델의 단점은 선형적인 정보만을 선택적으로 학습한다는 단점이 있음.

Others

[5]

- 주가의 패턴을 포착하고, Markov random field를 사용하여 연관성 정보로 이용함.

[6]

- 주식 사이의 상관을 Graph 형태로 따로 학습함.

- 위 방법의 주된 제한점은 pre-defined correlation에 의존한다는 점임.

Neural network method

DA-RNN [2]

- Multivariate time series의 Correlation을 Attention 구조를 활용하여 학습하는 초기의 모델
- 금융 시계열에서 Close 값 만을 이용하는 단점이 있음.

Spectral temporal GNN [4]

- Graph neural network로 Multivariate time series의 Correlation을 학습.
- 금융 도메인에 적용이 되지 않음

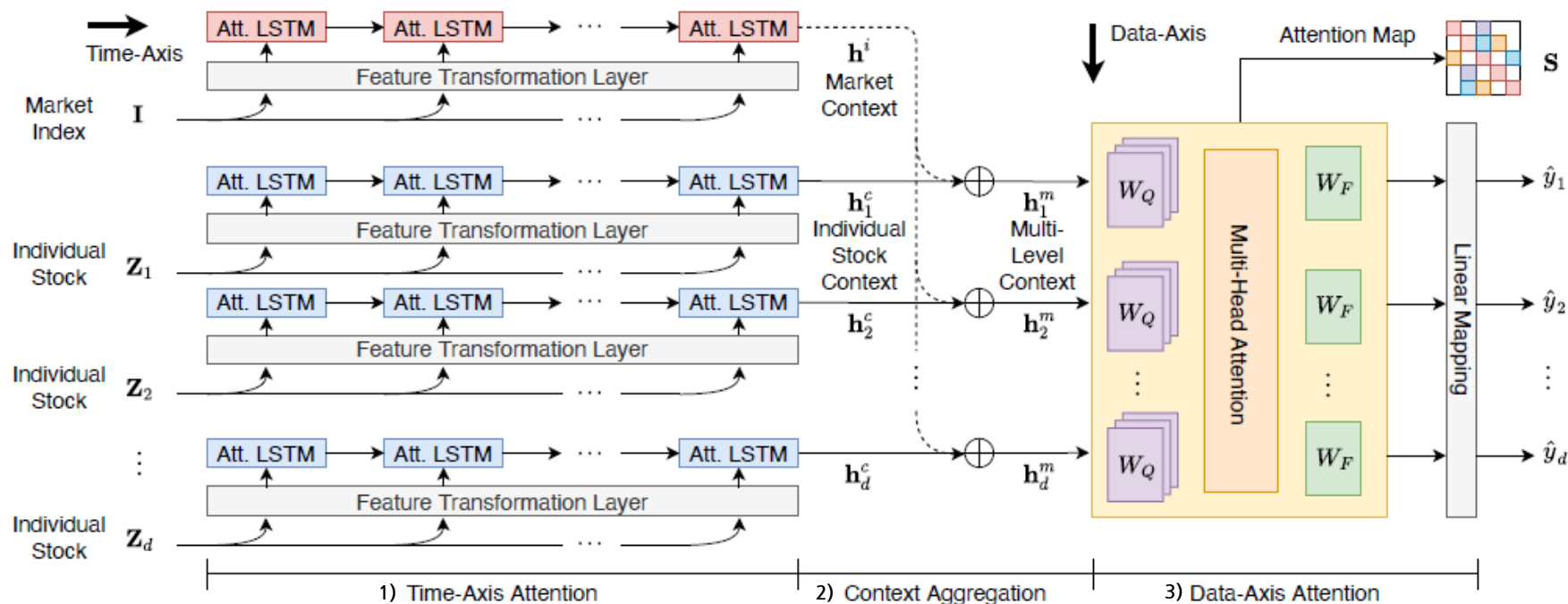
Transforemr encoder [7] [8] [9]

- 전력 소비량, 감염병 전파, 주가 예측 등의 업무를 Transforemr encoder를 활용하여 상관성을 학습한 예측 진행.

- 신경망 기반 방식은 비선형적 정보를 함께 학습할 수 있음.
- Transformer 방식은 기존 RNN의 Gradient vanishing 문제를 피할 수 있음.

Experiment Setting

- Model Structure



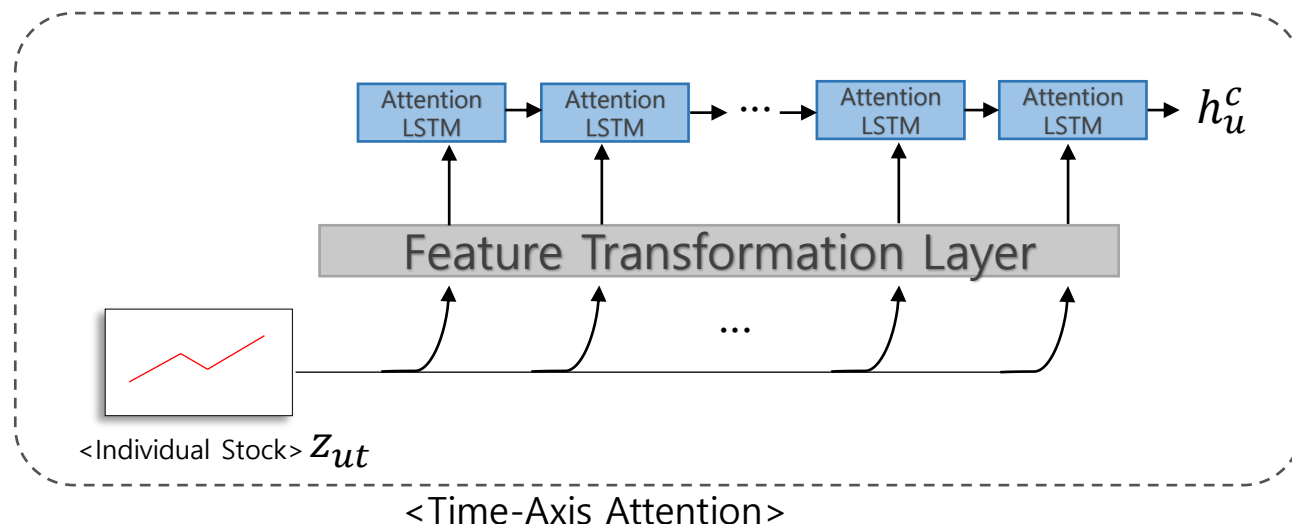
- 전체 모델 구조는 크게 3가지로 나뉨
 - 1) Time-Axis Attention
 - 2) Context Aggregation
 - 3) Data-Axis Attention

Experiment Setting

- Model Structure - 1) Attentive Context Generation

- $\{z_{ut}\}_{t \leq T}$ of length l , where u and t are stock and time indices respectively

- 위의 z_{ut} 에서 포괄적인 context vector인 h_u^c 를 얻는것이 Time-Axis Attention단계의 목표



- Feature Transformation

$$\tilde{z}_{ut} = \tanh(\mathbf{W}_s \mathbf{z}_{ut} + \mathbf{b}_s), \text{ where } \mathbf{W}_s \in \mathbb{R}^{h \times l} \text{ and } \mathbf{b}_s \in \mathbb{R}^h$$

- 위 계산은 Attention LSTM 구조에 학습시키기 전 계산복잡도의 증가 없이 복수의 Time step을 aggregation하기 위한 과정

Experiment Setting

- Model Structure - 1) Attentive Context Generation

- Attention LSTM

$$\alpha_i = \frac{\exp(\mathbf{h}_i^\top \mathbf{h}_T)}{\sum_{j=1}^T \exp(\mathbf{h}_j^\top \mathbf{h}_T)}$$

<Attention score>

$$\tilde{\mathbf{h}}^c = \sum_i \alpha_i \mathbf{h}_i$$

<Context vector>

- Context vector는 attentionLSTM을 통과하여 최종적으로 계산됨
 - AttentionLSTM은 Vanila LSTM과 비교해 효과적으로 정보를 전달함.

- Context Normalization

$$h_{ui}^c = \gamma_{ui} \frac{\tilde{h}_{ui}^c - \text{mean}(\tilde{h}_{ui}^c)}{\text{std}(\tilde{h}_{ui}^c)} + \beta_{ui}$$

- 각각의 주식의 정보를 요약한 Context vector는 서로 다른 값의 범위를 가지고 있기 때문에 Aggregation 단계 전에 Normalization을 진행.
 - 감마와 베타는 학습가능한 파라미터이다.

Experiment Setting

- Model Structure - 2) Multi-Level Context Aggregation

- Multi-Level Contexts

$$\mathbf{h}_u^m = \mathbf{h}_u^c + \beta \mathbf{h}^i$$

- $\mathbf{h}_u^m$ 은 개별 주식의 context vector이고, $\mathbf{h}^i$ 는 Market Index의 context vector임.
 - β 는 Index context vector가 어느정도 영향을 미칠지 결정할 수 있는 hyper parameter이다.

- The Effect of Global Contexts

$$\mathbf{h}_u^{m\top} \mathbf{h}_v^m = \mathbf{h}_u^{c\top} \mathbf{h}_v^c + \beta \mathbf{h}^{i\top} (\mathbf{h}_u^c + \mathbf{h}_v^c) + \beta^2 \mathbf{h}^{i\top} \mathbf{h}^i$$

- Global Contexts의 효과를 알아보기 위해 가장 단순한 Attention인 dot product를 시행.
 - 오른쪽 항에서 두번째 term을 보면, 주식들과 시장 움직임의 상관성에 더 큰 가중치를 부여함.
 - 오른쪽 항에서 세번째 term을 보면, 시장 움직임의 강도를 나타낼 수 있음.

Experiment Setting

- Model Structure - 3) Data-Axis Self-Attention

- Self-Attention

$$Q = HW_q \quad H \in \mathbb{R}^{d \times h}, \quad W \in \mathbb{R}^{h \times h} \\ K = HW_k \quad V = HW_v$$

$$\tilde{H} = SV \quad \text{where} \quad S = \text{softmax}\left(\frac{QK^T}{\sqrt{h}}\right)$$

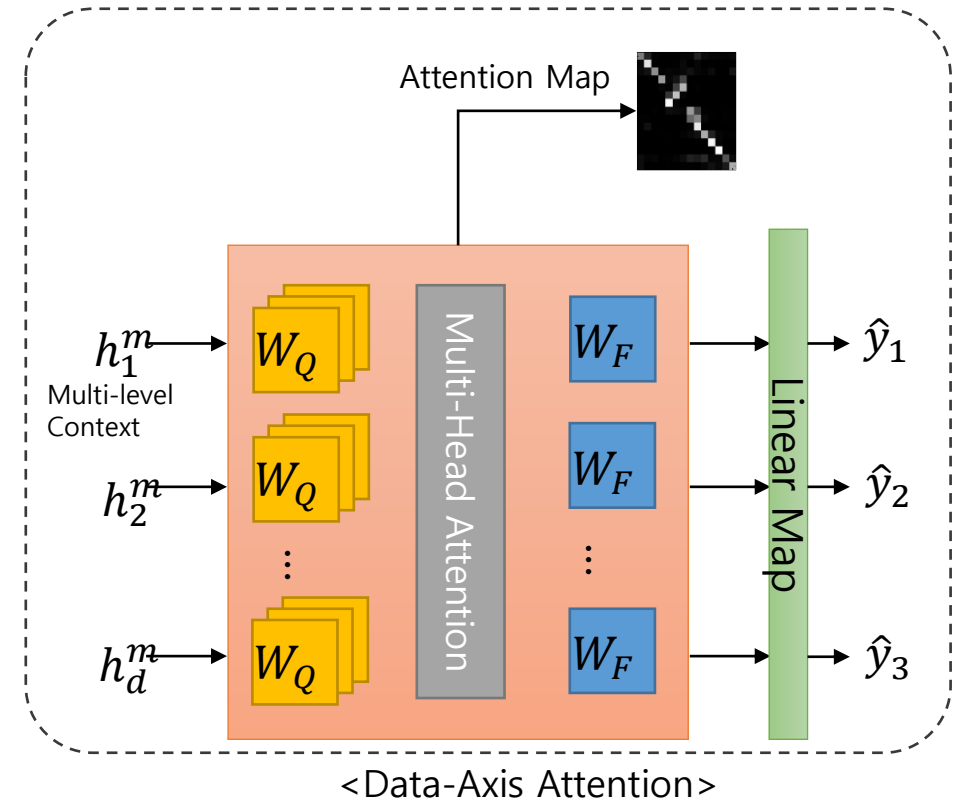
- d is the number of stocks and h is the length of context vectors

- W is learnable weight

- Nonlinear Transformation

$$H_p = \tanh(H + \tilde{H} + \text{MLP}(H + \tilde{H}))$$

- The hidden layer size of MLP is $4h$



Experiment Setting

- Model Structure - 3) Data-Axis Self-Attention

- Transformer encoder를 사용하는 장점.

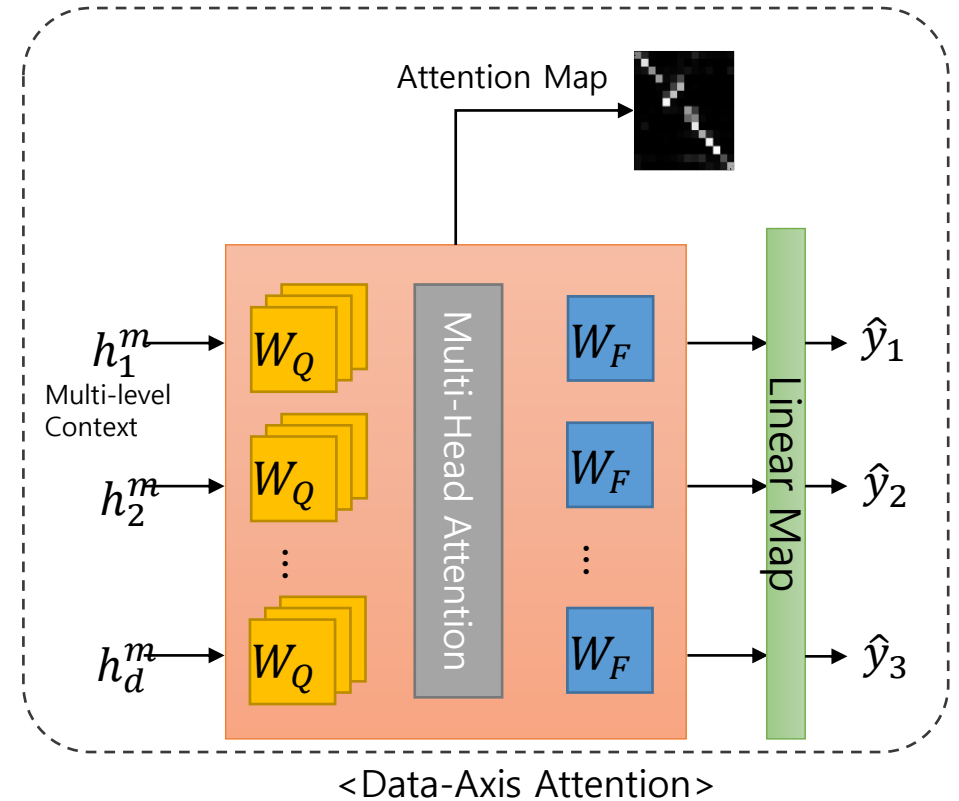
- 주식시장의 정보 확산에 대비해 다른 Query와 Key를 학습함으로써 **비대칭적인 Attention score**를 학습할 수 있음.

- AttentionLSTM이 최근의 지역적인 정보에 초점을 두는 것과는 대비되는 것으로, Transformer encode는 지역적인 정보에 초점을 두지 않음

- Final Prediction

$$\hat{y} = \sigma(H_p W_p + b_p).$$

- 각 값들을 확률로 해석하고, 직접적인 방향예측에 적용하기 위해 logistic sigmoid를 사용함.



Experiment Setting

- Training with Selective Regularization

- training

$$\mathcal{L}(\mathcal{X}, y) = -\frac{1}{d} \sum_u (y_u \log \hat{y}_u + (1 - y_u) \log(1 - \hat{y}_u)).$$

- 본 논문의 제안된 Structure인 DTML은 주식 예측값과 실제 값 사이의 Cross Entropy Loss를 최소화하는 방향으로 Training함
 - $\mathcal{X} \in \mathbb{R}^{w \times d \times l}$ is an input tensor of the current time step, and y is the true stock movements

- Selective Regularization

$$\mathcal{L}_{\text{reg}}(\mathcal{X}, y) = \mathcal{L}(\mathcal{X}, y) + \lambda(\|\mathbf{W}_p\|_F^2 + \|\mathbf{b}_p\|_2^2).$$

- Regularization은 deep learning trainin시 overfitting을 줄이기 위해 많이 사용되지만. λ 의 최적값을 찾는 데 어려움이 있음
 - 마지막 Predictor의 parameter에만 페널티를 부여하는 방식을 사용
 - 그 결과로 output space는 제한하지만, 표현력은 보존함.

Experiment Setting

- Experiment Setting

- Datasets

Dataset	Country	Stocks	Days	From	To
ACL18 ¹	US	87	504	2014-01-01	2015-12-31
KDD17 ¹	US	50	2,518	2007-01-01	2016-12-31
NDX100	US	95	1,259	2013-01-01	2017-12-31
CSI300	China	219	1,119	2015-06-01	2019-12-31
NI225	Japan	51	856	2016-07-01	2019-12-31
FTSE100	UK	24	1,134	2014-01-01	2018-06-30

- ACL18 and KDD17 are public datasets
 - 다음날의 움직임을 예측하는 것을 목표로 했고
 - 상승($y=1$) or 하락($y=0$) 으로 분류됨.

Experiment Setting

- Experiment Setting

- Feature Vectors

Features	Calculation
z_{open}	$z_{\text{open}} = \text{open}_t / \text{close}_t - 1$
z_{high}	$z_{\text{high}} = \text{high}_t / \text{close}_t - 1$
z_{low}	$z_{\text{low}} = \text{low}_t / \text{close}_t - 1$
z_{close}	$z_{\text{close}} = \text{close}_t / \text{close}_{t-1} - 1$
$z_{\text{adj_close}}$	$z_{\text{adj_close}} = \text{adj_close}_t / \text{adj_close}_{t-1} - 1$
z_{d5}, z_{d10} z_{d15}, z_{d20} z_{d25}, z_{d30}	e.g., $z_{dk} = \frac{\sum_{i=0}^k \text{adj_close}_{t-i}}{k \cdot \text{adj_close}_t} - 1$

- z_{ut} 는 주식 u의 t 시점(날짜)의 트렌드를 묘사함.

- 위 값들은 종가에 대한 open, high, low 등의 상대적인 값들을 의미함

- Evaluation

- Accuracy (ACC)
 - Matthews correlation coefficient (MCC)

- Hyperparameters

Window size (w)	10, 15
market context weight	0.01, 0.1, 1
number of epochs	100, 200
learning rate	0.001, 0.0001
strength λ of selective regularization	1
dropout rate to	0.15
optimizer	Adam

Experiment Result

- Prediction Accuracy

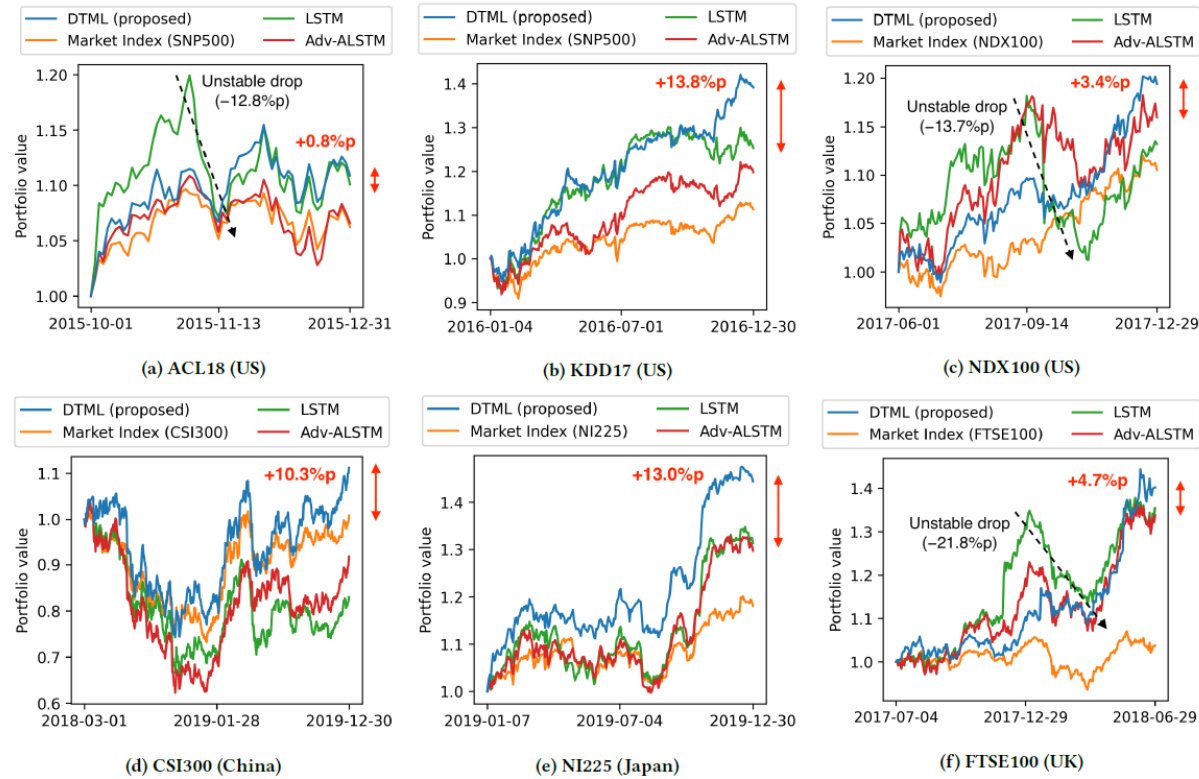
Model	ACL18 (US)		KDD17 (US)		NDX100 (US)	
	ACC	MCC	ACC	MCC	ACC	MCC
LSTM [24]	0.4987 ± 0.0127	0.0337 ± 0.0398	0.5118 ± 0.0066	0.0187 ± 0.0110	0.5263 ± 0.0003	0.0037 ± 0.0049
ALSTM [31]	0.4919 ± 0.0142	0.0142 ± 0.0275	0.5166 ± 0.0041	0.0316 ± 0.0119	0.5260 ± 0.0007	0.0028 ± 0.0084
StockNet [31]	0.5285 ± 0.0020	0.0187 ± 0.0011	0.5193 ± 0.0001	0.0335 ± 0.0050	0.5392 ± 0.0016	0.0253 ± 0.0102
Adv-ALSTM [9]	0.5380 ± 0.0177	0.0830 ± 0.0353	0.5169 ± 0.0058	0.0333 ± 0.0137	0.5404 ± 0.0003	0.0046 ± 0.0090
DTML (proposed)	0.5744 ± 0.0194	0.1910 ± 0.0315	0.5353 ± 0.0075	0.0733 ± 0.0195	0.5406 ± 0.0037	0.0310 ± 0.0193

Model	CSI300 (China)		NI225 (Japan)		FTSE100 (UK)	
	ACC	MCC	ACC	MCC	ACC	MCC
LSTM [24]	0.5367 ± 0.0038	0.0722 ± 0.0050	0.5079 ± 0.0079	0.0148 ± 0.0162	0.5096 ± 0.0065	0.0187 ± 0.0129
ALSTM [31]	0.5315 ± 0.0036	0.0625 ± 0.0076	0.5060 ± 0.0066	0.0125 ± 0.0139	0.5106 ± 0.0038	0.0231 ± 0.0077
StockNet [31]	0.5254 ± 0.0029	0.0445 ± 0.0117	0.5015 ± 0.0054	0.0050 ± 0.0118	0.5036 ± 0.0095	0.0134 ± 0.0135
Adv-ALSTM [9]	0.5337 ± 0.0050	0.0668 ± 0.0084	0.5160 ± 0.0103	0.0340 ± 0.0201	0.5066 ± 0.0067	0.0155 ± 0.0140
DTML (proposed)	0.5442 ± 0.0035	0.0826 ± 0.0074	0.5276 ± 0.0103	0.0626 ± 0.0230	0.5208 ± 0.0121	0.0502 ± 0.0214

- 본 논문에서 제안된 DTML 알고리즘이 이전에 높은 정확성을 보였던 Adv-ALSTM에 비해 더 나은 일반화가 가능했고, 정확성에서 좋은 성과를 보임.

Experiment Result

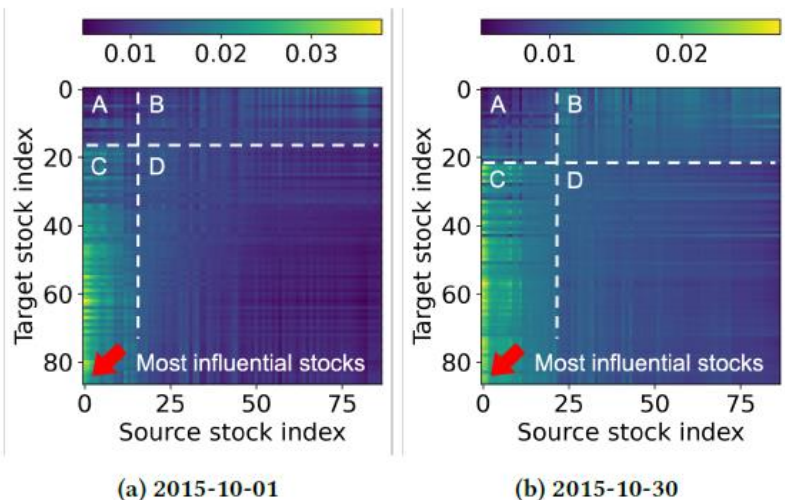
- Invest simulation



- SNP500 is used for the ACL18 and KDD17 datasets 데이터셋을 비교모델 중 최상의 성과와 비교한 결과 13.8%p 더 높은 성과를 보였고
- 나머지 시뮬레이션에서도 좋은 결과를 보임으로써, 비교모델과 비교해 더 일반화된 모델임을 알 수 있음.

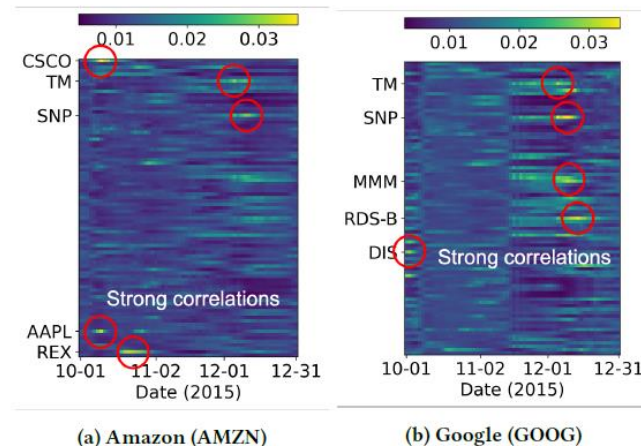
Experiment Result

- Interpreting Attention Maps



- 위의 Attention Map을 분석한 결과, c에 해당하는 소수의 주식이 대부분의 주식 예측 결과에 영향을 미치는 것을 알 수 있음.
- B영역은 대부분의 주식이 다른 주식의 예측에 영향을 주는 것을 알 수 있음.

<전체 주식간의 attention score>

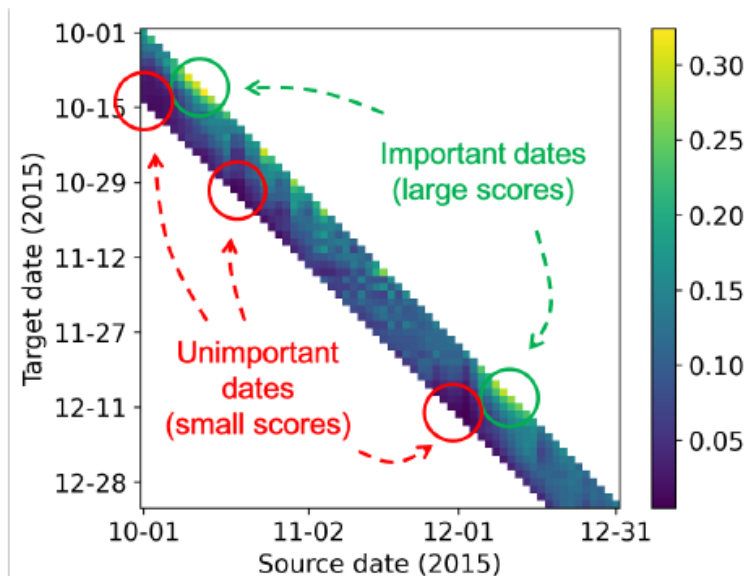


- 위의 Attention Map을 분석한 결과 시간에 따라 특정 주식에 대한 가중치가 달라짐을 알 수 있음.
- 이전에는 연관 지을 수 없었던 다른 주식과의 상관성을 알 수 있음.

<아마존과 구글 주식의 다른 주식에 대한 attention score>

Experiment Result

- Interpreting Attention Maps



<구글 주식의 target data 와 source data에 관한 attention score>

- 초록색은 Attention 값이 큰 부분을 나타내고, 붉은색 부분은 Attention 값이 적은 부분을 나타냄
- 위의 Attention Map은 본 DTML 모델이 다른 주식 뿐만 아니라 해당 주식 자체의 정보 또한 이용함을 알 수 있음.

Conclusion

- DTML

- 상술된 다양한 데이터셋과 평가 metric 에서 비교모델들에 비해 전반적으로 좋은 성능을 보임.
- 비교 모델에 비해 일반적으로 성능이 좋은 모델임을 알 수 있음.

- Issue of DTML

- 일반적인 시계열 값을 넣지 않고 Context를 만들어 넣은 이유에 대한 의문점.
- Context Normalization에서 학습 가능한 Parameter를 추가해야 되는지에 대한 의문점.

Reference

- 1 James H. Stock and Mark W. Watson. 2001. Vector Autoregressions. *Journal of Economic Perspectives*. 101-115. [10.1257/jep.15.4.101](https://doi.org/10.1257/jep.15.4.101)
- 2 Jaemin Yoo, Yejun Soun, Yong-chan Park and U Kang. 2021. Accurate Multivariate Stock Movement Prediction via Data-Axis Transformer with Multi-Level Contexts. *ACM SIGKDD International Conference on Knowledge discovery and data mining*. 2037–2045. <https://doi.org/10.1145/3447548.3467297>
- 3 Yifeng Zhang, Ka-Ho Chow and S.-H. Gary Chan. 2019. DA-LSTM: A Long Short-Term Memory with Depth Adaptive to Non-uniform Information Flow in Sequential Data. *Neural and Evolutionary Computing*. arXiv:1903.02082
- 4 Defu Cao, Yujing Wang, Juanyong Duan, Ce Zhang, Xia Zhu, Congrui Huang, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, and Qi Zhang. 2020. Spectral Temporal Graph Neural Network for Multivariate Time-series Forecasting. *NeurIPS*.
- 5 Chang Li, Dongjin Song, and Dacheng Tao. 2019. Multi-task Recurrent Neural Networks and Higher-order Markov Random Fields for Stock Price Movement Prediction. *KDD*.
- 6 Wei Li, Ruihan Bao, Keiko Harimoto, Deli Chen, Jingjing Xu, and Qi Su. 2020. Modeling the Stock Relation with Graph Network for Overnight Stock Movement Prediction. *IJCAI*.

Dataset survey

- Available Dataset

- from pandas_datareader import data as pdr
pdr.get_data_yahoo

- 위 방식을 사용해 야후 파이낸스의 모든 주식정보를 가져올 수 있음.

- Open, High, low, Close, Adj_close, Volume에 대한 정보를 불러옴

- S&P 500등의 지수 정보와 애플, 구글 등의 개별주식 정보 또한 업로드 가능

Date	Open	High	Low	Close	Adj Close	Volume
2018-01-02	2683.729980	2695.889893	2682.360107	2695.810059	2695.810059	3367250000
2018-01-03	2697.850098	2714.370117	2697.770020	2713.060059	2713.060059	3538660000
2018-01-04	2719.310059	2729.290039	2719.070068	2723.989990	2723.989990	3695260000
2018-01-05	2731.330078	2743.449951	2727.919922	2743.149902	2743.149902	3236620000
2018-01-08	2742.669922	2748.510010	2737.600098	2747.709961	2747.709961	3242650000
...	...	...	...	...	...	...
2021-09-23	4406.750000	4465.399902	4406.750000	4448.979980	4448.979980	2833290000
2021-09-24	4438.040039	4463.120117	4430.270020	4455.479980	4455.479980	2772090000
2021-09-27	4442.120117	4457.299805	4436.189941	4443.109863	4443.109863	3032870000
2021-09-28	4419.540039	4419.540039	4346.330078	4352.629883	4352.629883	3495970000
2021-09-29	4362.410156	4385.569824	4355.080078	4359.459961	4359.459961	2753800000

<직접 데이터를 불러온 예시>

Dataset survey

- Available Dataset
- Public Datasets(<https://github.com/fulifeng/Adv-ALSTM>)
 - KDD 17
 - ACL 18

- Non-Public Datasets
 - NDX100
 - CSI300
 - NI225
 - FTSE100

-> Non-Public Datasets은 본문의 저자가 수집한 데이터셋으로 **상세한 종목명이 명기되지않음**.

Table 2: Summary of datasets.

Dataset	Country	Stocks	Days	From	To
ACL18 ¹	US	87	504	2014-01-01	2015-12-31
KDD17 ¹	US	50	2,518	2007-01-01	2016-12-31
NDX100	US	95	1,259	2013-01-01	2017-12-31
CSI300	China	219	1,119	2015-06-01	2019-12-31
NI225	Japan	51	856	2016-07-01	2019-12-31
FTSE100	UK	24	1,134	2014-01-01	2018-06-30

Dataset survey

- KDD 17 (2007-01-01~2016-12-31)

- 주식 데이터 코드 목록 : 'AAPL', 'AMZN', 'BA', 'BAC', 'BHP', 'BRK-B', 'CHL', 'CMCSA', 'CVX', 'D', 'DCM', 'DIS', 'DOW', 'DUK', 'EXC', 'GE', 'GOOGL', 'HD', 'INTC', 'JNJ', 'JPM', 'KO', 'MA', 'MMM', 'MO', 'MRK', 'MSFT', 'NGG', 'NTT', 'NVS', 'ORCL', 'PEP', 'PFE', 'PG', 'PTR', 'RDS-B', 'RIO', 'SO', 'SPY', 'SYT', 'T', 'TM', 'TOT', 'UNH', 'UPS', 'VALE', 'VZ', 'WFC', 'WMT', 'XOM', 총 50개.

- ACL 18 (2014-01-01~2015-12-31)

- 주식 데이터 코드 목록 : 'AAPL', 'ABB', 'ABBV', 'AEP', 'AGFS', 'AMGN', 'AMZN', 'BA', 'BABA', 'BAC', 'BBL', 'BCH', 'BHP', 'BP', 'BRK-A', 'BSAC', 'BUD', 'C', 'CAT', 'CELG', 'CHL', 'CHTR', 'CMCSA', 'CODI', 'CSCO', 'CVX', 'D', 'DHR', 'DIS', 'DUK', 'EXC', 'FB', 'GD', 'GE', 'GOOG', 'HD', 'HON', 'HRG', 'HSBC', 'IEP', 'INTC', 'JNJ', 'JPM', 'KO', 'LMT', 'MA', 'MCD', 'MDT', 'MMM', 'MO', 'MRK', 'MSFT', 'NEE', 'NGG', 'NVS', 'ORCL', 'PCG', 'PCLN', 'PEP', 'PFE', 'PG', 'PICO', 'PM', 'PPL', 'PTR', 'RDS-B', 'REX', 'SLB', 'SNP', 'SNY', 'SO', 'SPLP', 'SRE', 'T', 'TM', 'TOT', 'TSM', 'UL', 'UN', 'UNH', 'UPS', 'UTX', 'V', 'VZ', 'WFC', 'WMT', 'XOM', 총 87개.

-> 데이터 저장 완료 + 추후에 위의 코드로 데이터 불러오기 가능

Dataset survey

- KDD 17 (2007-01-01~2016-12-31)
Train data = 2007-01-03 ~ 2015-01-01,
validation data = 2015-01-02 ~ 2016-01-03
Test data = 2016-01-04 ~ 2016-12-31

- ACL 18 (2014-01-01~2015-12-31)
Train data = 2014-01-02 ~ 2015-08-02,
validation data = 2015-08-03 ~ 2015-09-30
Test data = 2015-10-01 ~ 2015-12-31

논문 구현 진행상황

- 완료

- (주식 index list, input sequence length, output sequence length, 기간)를 입력할 경우, batch 로 만들어 여러 주식의 입력 묶음으로 만듦.
- 전체 training, validation, test 등의 실험과정의 frame
- 입력을 넣을 model (AttentionLSTM, Transformer encoder)

- 향후 과제

- 논문에서 요구된 모델 입력 전 feature vector에 대한 구현
- index vector와 stock vector의 Aggregation 단계
- 본 논문에서 소개된 Transformer의 결과물인 다양한 Attention Map의 출력

논문 개선 방안.

- 긴 기간의 정보를 뽑아내는 것

- 본 논문에서의 input length는 10으로써 긴 기간의 정보를 뽑아내는 데 무리가 있음
- 본 논문의 모델 입력 전 feature vector를 만드는 과정에는

$$\tilde{\mathbf{z}}_{ut} = \tanh(\mathbf{W}_s \mathbf{z}_{ut} + \mathbf{b}_s), \text{ where } \mathbf{w}_s \in \mathbb{R}^{h \times l} \text{ and } \mathbf{b}_s \in \mathbb{R}^h$$

- Linear transformation 과정이 존재함
- 해당 Linear transformation 과정에 여러 sequence interval (1일, 5일, 20일, 250일 등)을 가진 데이터를 입력하여 긴 정보를 입력.