Accurate Multivariate Stock Movement Prediction via Data-Axis Transformer with Multi-Level Contexts

Jaemin Yoo[1], **Yejun Soun**[1,2], **Yong-chan Park**[1], and **U Kang**[1,2]

[1] Seoul National University

[2] DeepTrade, Inc.

KDD 2021

Outline

- **<u>Introduction</u>**
- Proposed Method
- Experiments
- Conclusion

Stock Market Prediction

- To predict the future values of stock prices
  - The most popular task in the financial domain
- The problem is challenging but rewarding
  - **Challenging.** Stock prices have no clear patterns and make random movements
  - **Rewarding.** Even the increase of 1% of prediction accuracy results in enormous profit

Problem Definition

- **Stock movement prediction**
  - We model the problem as binary classification
  - To predict a *movement* rather than exact price
- **Definition**
  - **Given** the historical prices $\{p_t\}_{t \leq T}$ of a stock
    - We assume daily prices; p_t is the price at day t
    - T is the current index for prediction
  - **Predict** the rise ($p_{T+1} > p_T$) or fall ($p_{T+1} \leq p_T$)

Research Motivation

- Stocks are strongly correlated with each other
- Previous models can be categorized as
  - **Univariate models**
    - They treat each stock independently from the others
  - **Multivariate models with fixed correlations**
    - They take the correlations as a predefined input
    - The correlations cannot reflect the dynamic property

How can we learn dynamic correlations between stocks with no prior knowledge?

Outline

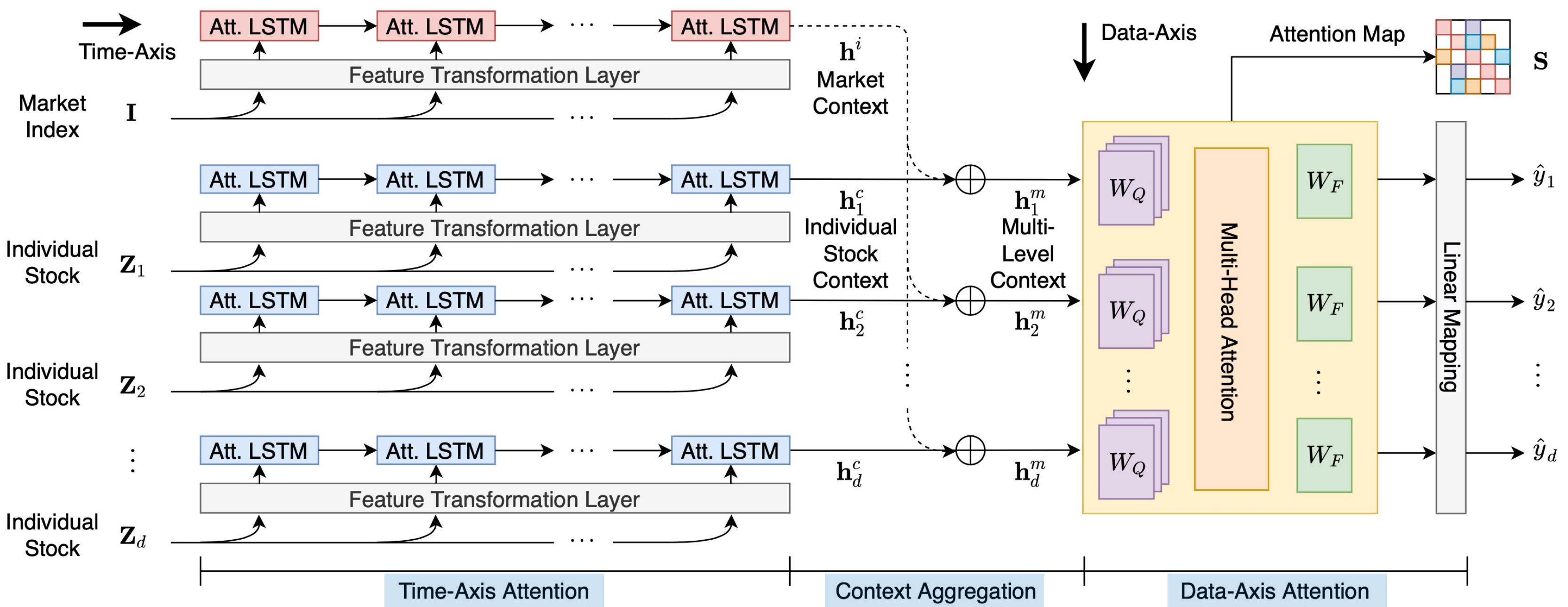
- Introduction
- <u>**Proposed Method**</u>
- Experiments
- Conclusion

Overview (1)

- We propose **DTML** for stock price prediction
  - *Data-axis Transformer with Multi-Level contexts*
- **Idea 1.** Time-axis attention
  - To summarize the historical prices of each stock
- **Idea 2.** Context aggregation
  - To combine individual contexts with a global trend
- **Idea 3.** Data-axis attention
  - To learn the stock correlations by a transformer

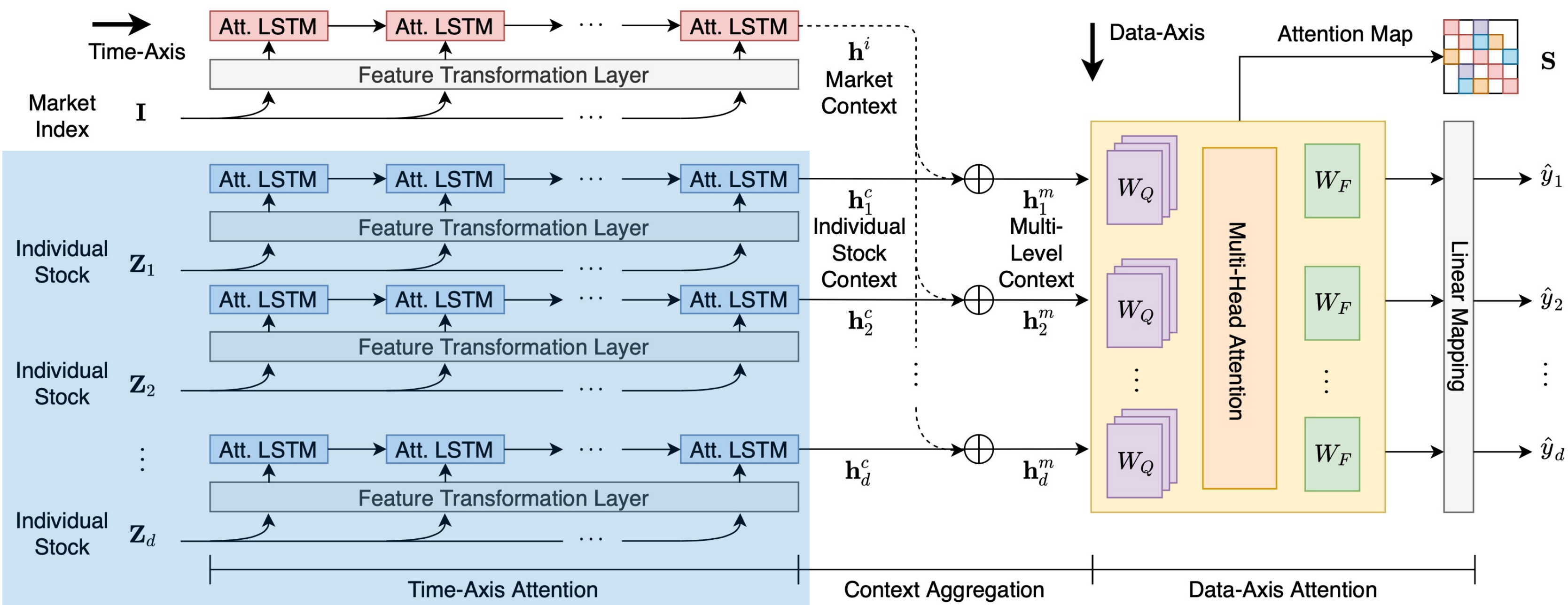
Overview (2)

- This is the overall structure of DTML
  - Three modules correspond to the three main ideas



Time-Axis Attention (1)

- **Module 1. Time-axis attention**
  - Summarizes the historical prices as a single vector



Time-Axis Attention (2)

- **Given** feature vectors $\{\mathbf{z}_{ut}\}_{t \leq T}$ of stock u
  - $\mathbf{z}_{ut}$ is made from the prices of stock u until day t
- **Feature transformation**
  - We transform each feature vector as follows:

$$\tilde{\mathbf{z}}_{ut} = \tanh(\mathbf{W}_s \mathbf{z}_{ut} + \mathbf{b}_s),$$

  - $\mathbf{W}_s$ and $\mathbf{b}_s$ are learnable weight and bias, resp.

Time-Axis Attention (3)

- **Attention LSTM**
  - We run LSTM to generate state vectors $\{\mathbf{h}_{ut}\}_t$
  - We then compute an attention score α_i such that

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

  - The state vectors are combined as $\tilde{\mathbf{h}}^c = \sum_i \alpha_i \mathbf{h}_{ui}$

Time-Axis Attention (4)

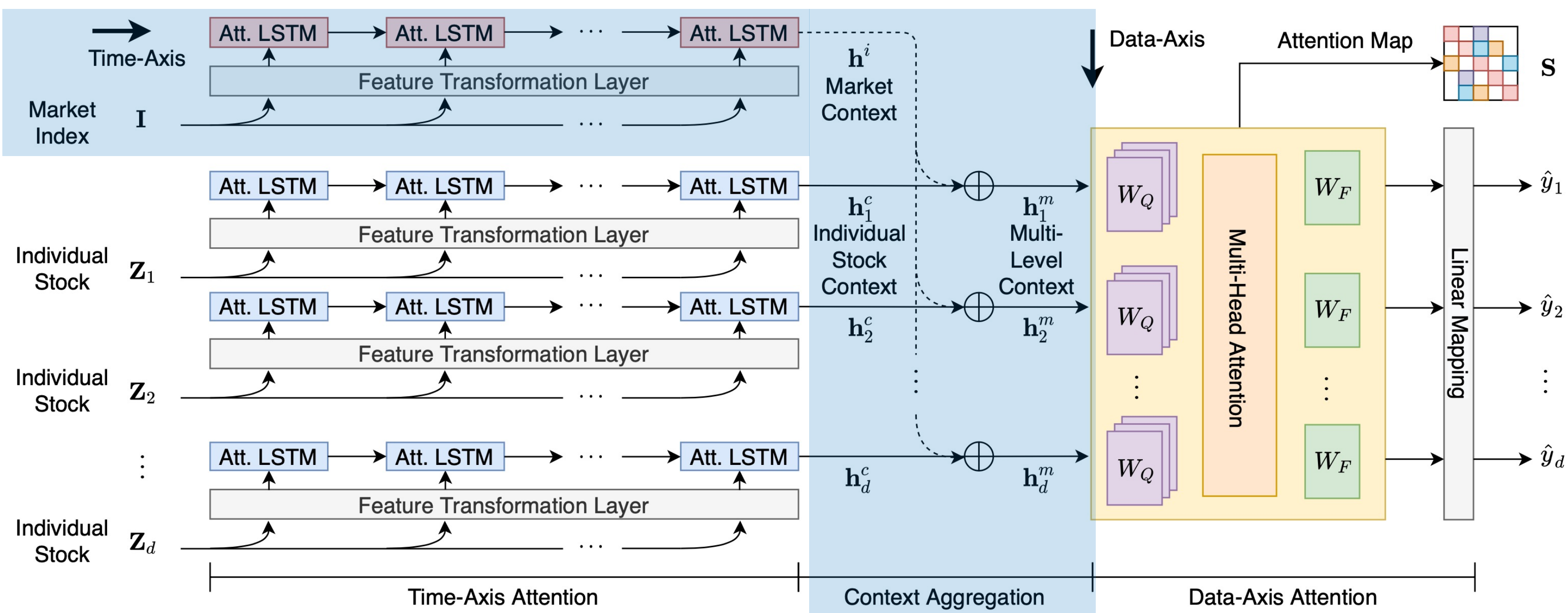
- **Context normalization**
  - Raw contexts have different scales for stocks
  - We normalize each element of $\tilde{\mathbf{h}}_u^c$ as follows:

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

    - γ_{ui} and β_{ui} are learnable parameters for each pair (u, i)

Context Aggregation (1)

- **Module 2. Context aggregation**
  - Combines individual and global context vectors



Context Aggregation (2)

- **Prepare** a global market context $\mathbf{h}^i$
  - Consider a market index i as an individual stock
    - Such as NDX100 or DJI in the US stock markets
  - Apply the time-axis attention to i and make $\mathbf{h}^i$
- **Make** a multi-level context for each stock u:

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

  - β is a parameter for balancing the two contexts

Context Aggregation (3)

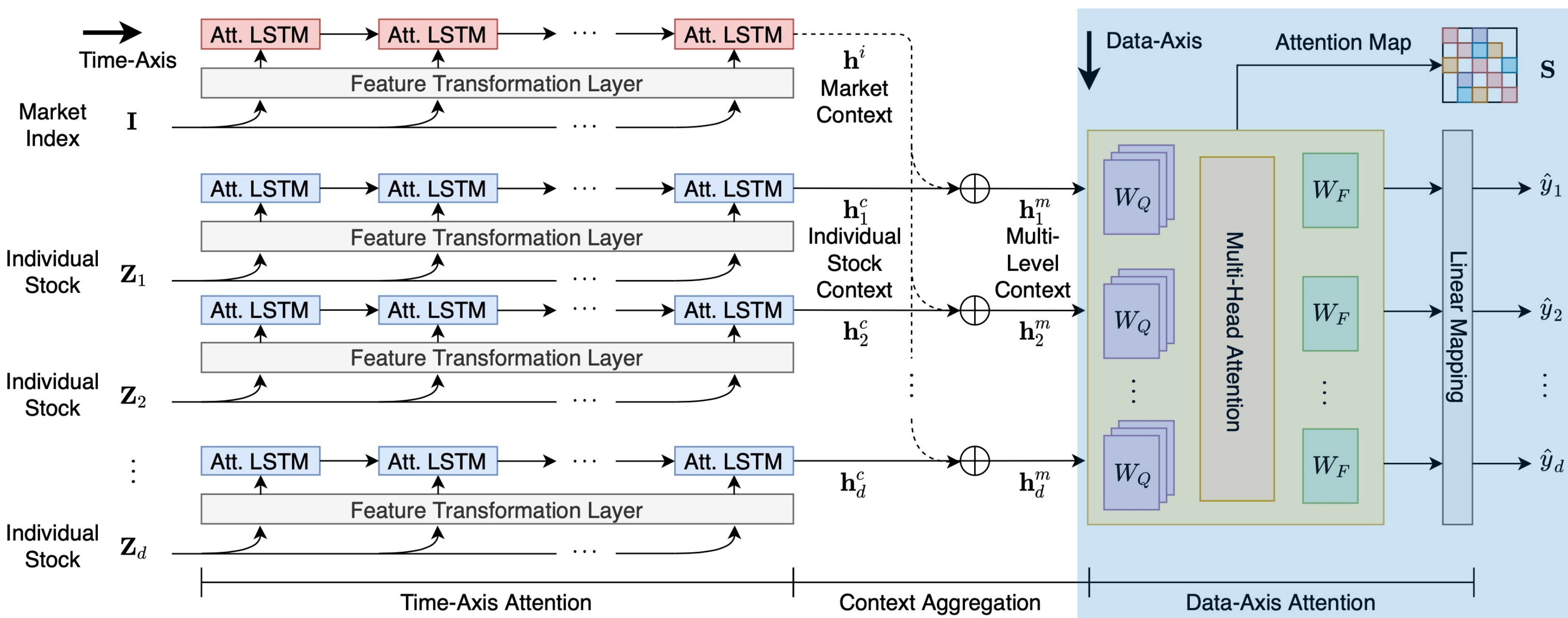
- **Effect of context aggregation**
  - Consider a simple dot-product attention as

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

  - Using multi-level contexts has several effects:
    - **T2:** A larger weight is given if the movement of a stock corresponds to the movement of the market
    - **T3:** The movement of the market works as the default value for all correlations between stocks

Data-Axis Attention (1)

- **Module 3. Data-axis attention**
  - Computes dynamic correlations between stocks



Data-Axis Attention (2)

- **Given**
  - Matrix $\mathbf{H} \in \mathbb{R}^{n \times d}$ that stacks multi-level contexts
    - n is the number of stocks, and d is the context size
- **Make** attention components $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ as

$$\mathbf{Q} = \mathbf{H}\mathbf{W}_{\mathrm{q}} \qquad \mathbf{K} = \mathbf{H}\mathbf{W}_{\mathrm{k}} \qquad \mathbf{V} = \mathbf{H}\mathbf{W}_{\mathrm{v}}.$$

- **Aggregate** the context vectors as follows:

$$\tilde{\mathbf{H}} = \mathbf{S}\mathbf{V} \quad \text{where} \quad \mathbf{S} = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\top}}{\sqrt{h}}\right).$$

Data-Axis Attention (3)

- We transform the aggregated contexts $\tilde{\mathbf{H}}$ with
  - **Nonlinear transformation**

$$\mathbf{H}_{\mathrm{p}} = \tanh(\mathbf{H} + \tilde{\mathbf{H}} + \mathrm{MLP}(\mathbf{H} + \tilde{\mathbf{H}})),$$

  - **Final prediction layer**

$$\hat{\mathbf{y}} = \sigma(\mathbf{H}_{\mathrm{p}}\mathbf{W}_{\mathrm{p}} + \mathbf{b}_{\mathrm{p}}).$$

    - σ is the logistic sigmoid function for an output

Summary

- DTML is a combination of three modules
  - **Time-axis attention** for the prices of each stock
  - **Context aggregation** with a global market trend
  - **Data-axis attention** by a transformer encoder
- DTML is trained by a gradient-based way
  - To minimize the cross-entropy loss $\mathcal{L}$ for training
  - Apply the L2 regularizer only to the last predictor
    - **Why?** To restrict the output space while not affecting the main functionality for making stock correlations

Outline

- Introduction
- Proposed Method
- **<u>Experiments</u>**
- Conclusion

Datasets

- We use six datasets in four different countries
  - Two are public datasets used in previous work
  - Four are private datasets collected in this work

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

Feature Engineering

- We use the same features as in previous work
  - Each feature $\mathbf{z}_{ut}$ summarizes the price movement of stock u until day t by simple operations

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., $z_{dk} = \frac{\sum_{i=0}^{k} \text{adj\_close}_{t-i}}{k \cdot \text{adj\_close}_t} - 1$

Evaluation

- We split each dataset into train/valid/test sets
  - The split is done by the chronological order
  - The splitting dates are the same as in prev. work
- We use two evaluation metrics
  - Simple accuracy (ACC)
    - The number of correct predictions over all predictions
  - The Matthews correlation coefficient (MCC)
    - Measures the accuracy in a more balanced manner

Questions

- We answer the questions by experiments:
  - **Q1 (Accuracy).** Does DTML outperform previous models in terms of classification accuracy?
  - **Q2 (Profit).** Does DTML make the profit on actual investment simulation in our datasets?
  - **Q3 (Correlations).** Does DTML make reasonable correlations between stocks?
  - **Q4 (Ablation study).** Does each module of DTML help improving the classification accuracy?

Q1. Prediction Accuracy

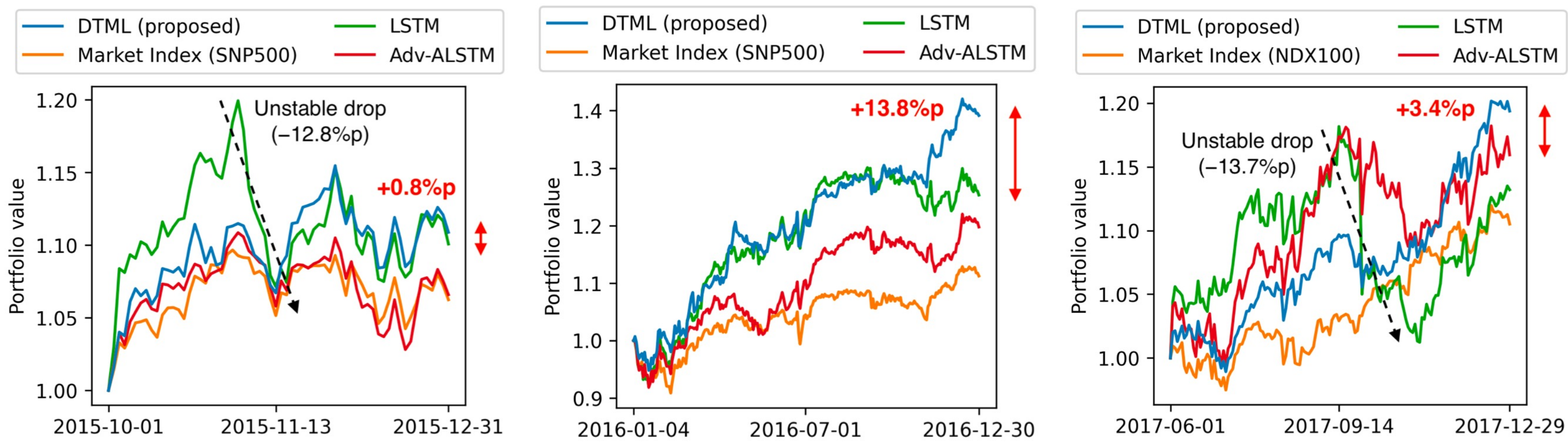
- DTML produces the highest ACC and MCC
  - The improvement is more significant with MCC

Model	ACL18 (US) ACC	ACL18 (US) MCC	KDD17 (US) ACC	KDD17 (US) MCC	NDX100 (US) ACC	NDX100 (US) 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) ACC	CSI300 (China) MCC	NI225 (Japan) ACC	NI225 (Japan) MCC	FTSE100 (UK) ACC	FTSE100 (UK) 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**

Q2. Investment Simulation

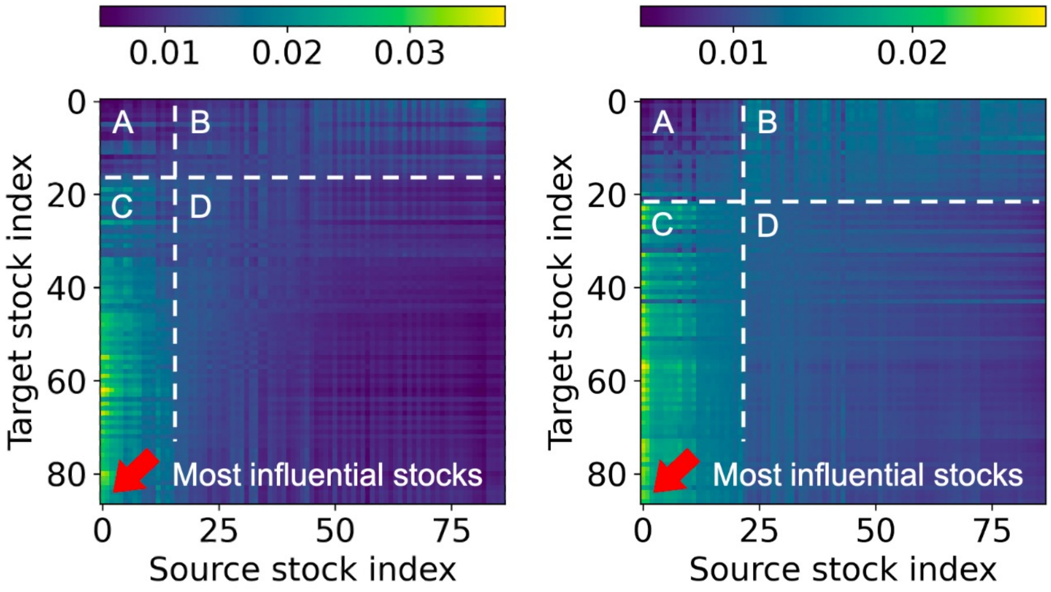
- DTML makes larger profit than the baselines
  - DTML does not suffer from unstable drops
  - The trend is the same as in the other datasets



(a) ACL18 (US) (b) KDD17 (US) (c) NDX100 (US)

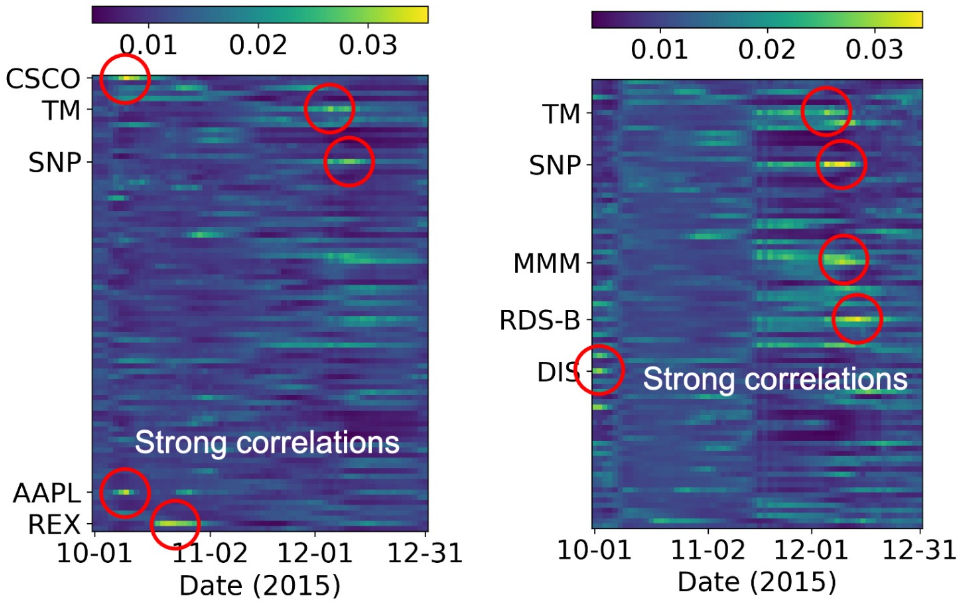
Q3. Data-Axis Attention (1)

- Attention scores indicate stock importances
  - We sort the stocks by their attention scores
  - Region C contains the most influential stocks



Q3. Data-Axis Attention (2)

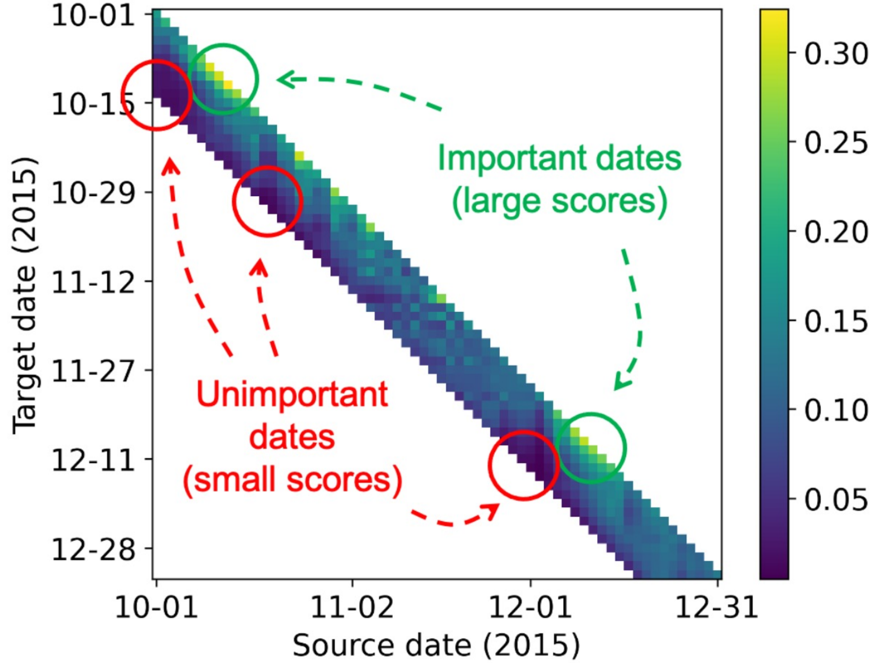
- We study the cases of Amazon and Google
  - They show smooth changes in scores with others



(a) Amazon (AMZN) **(b) Google (GOOG)**

Q3. Temporal Attention

- DTML learns also the importances of days
  - Important days have larger influences than others



Q4. Ablation Study

- We perform an ablation study for DTML
  - **TA**, **SA**, and **MC** refer to temporal attention, stock attention, and multi-level contexts, resp.
  - All three modules help improving the accuracy

Model	**ACC**	**MCC**
DTML-TA-SA-MC	0.5349 ± 0.0140	0.0828 ± 0.0246
DTML-TA	0.5574 ± 0.0163	0.1387 ± 0.0334
DTML-SA	0.5622 ± 0.0153	0.1453 ± 0.0205
DTML-MC	0.5724 ± 0.0177	0.1856 ± 0.0313
DTML	$\mathbf{0.5744 \pm 0.0194}$	$\mathbf{0.1910 \pm 0.0315}$

Outline

- Introduction
- Proposed Method
- Experiments
- **Conclusion**

Conclusion

- We propose DTML for stock price prediction
  - To learn dynamic correlations between stocks
- DTML consists of three modules
  - **Time-axis attention** for the prices of each stock
  - **Context aggregation** with a global market trend
  - **Data-axis attention** by a transformer encoder
- We perform experiments on six datasets
  - DTML achieves the state-of-the-art accuracy
  - DTML makes up to 13.8%p higher actual profit

Thank you!

DM Lab @ SNU (datalab.snu.ac.kr)

DeepTrade (deeptrade.co)