Improved Predictive Deep Temporal Neural Networks with Trend Filtering

Youngjin Park[1] Deokjun Eom[2] Byoungki Seo[1] Jaesik Choi[2,3]*

[1] Ulsan National Institute of Science and Technology, UNIST

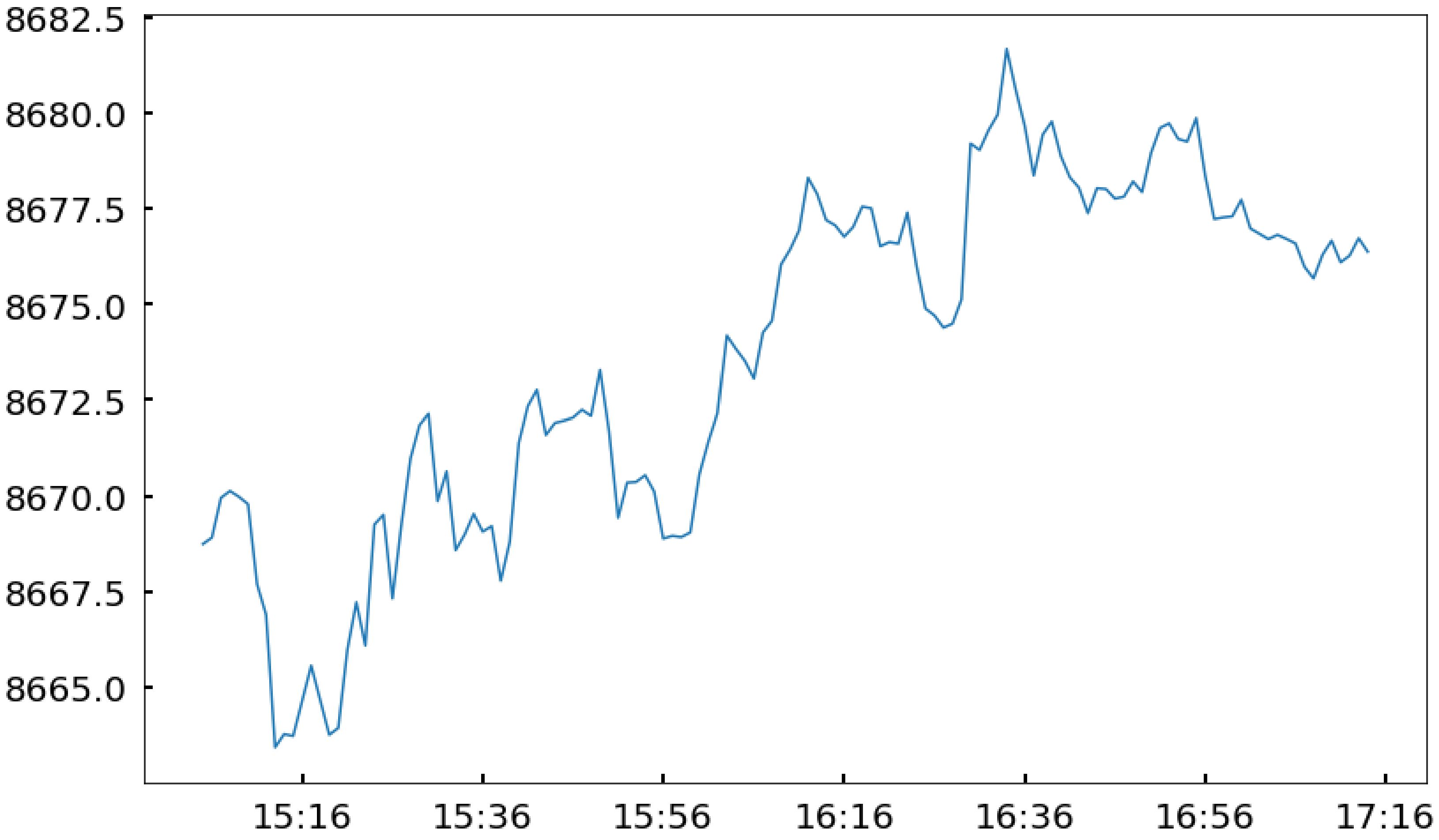
[2] Korea Advanced Institute of Science and Technology, KAIST

[3] INEEJI

* Speaker

Problem

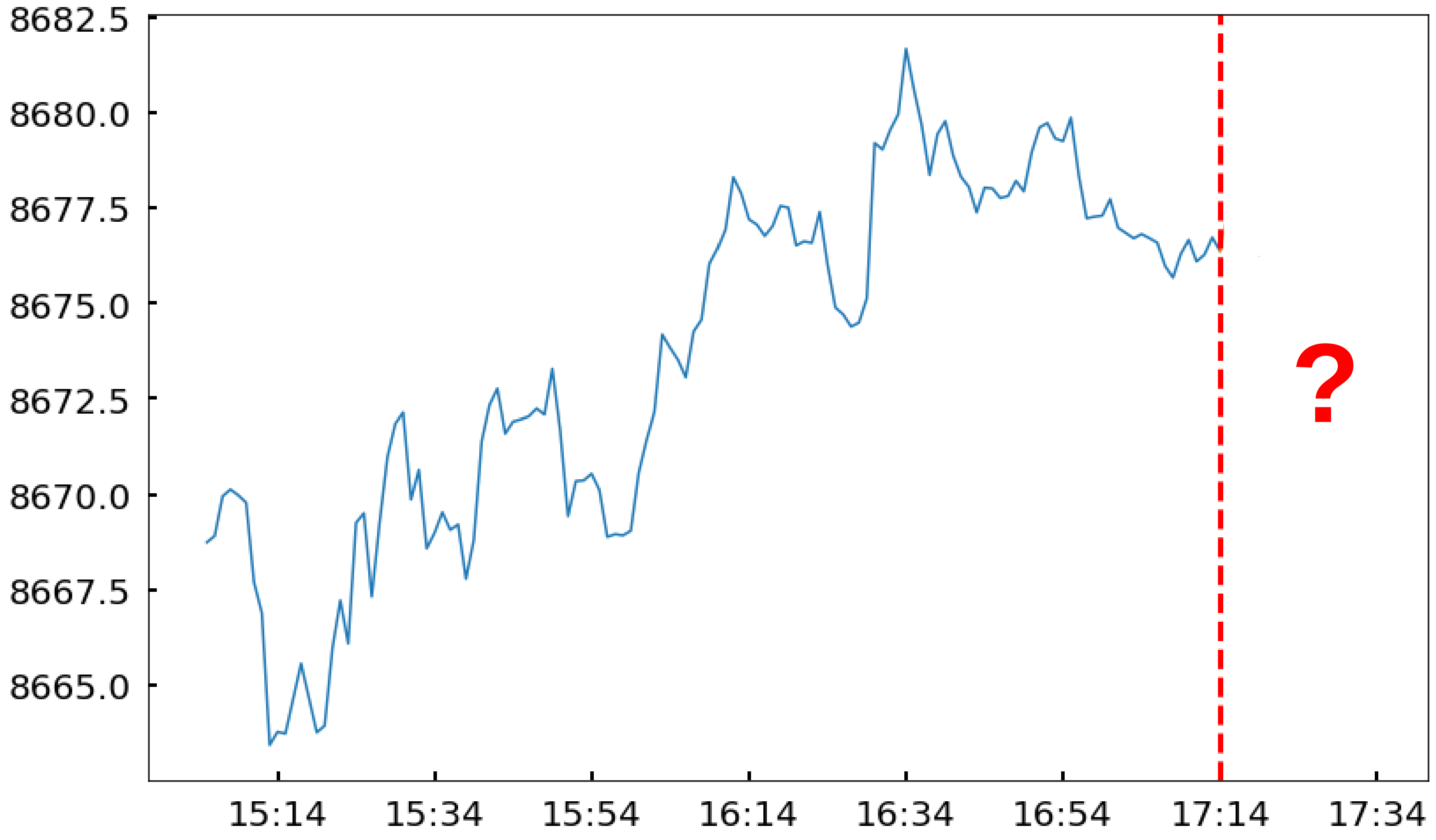
Prediction with noisy multivariate time series



NASDAQ 100 index

Problem

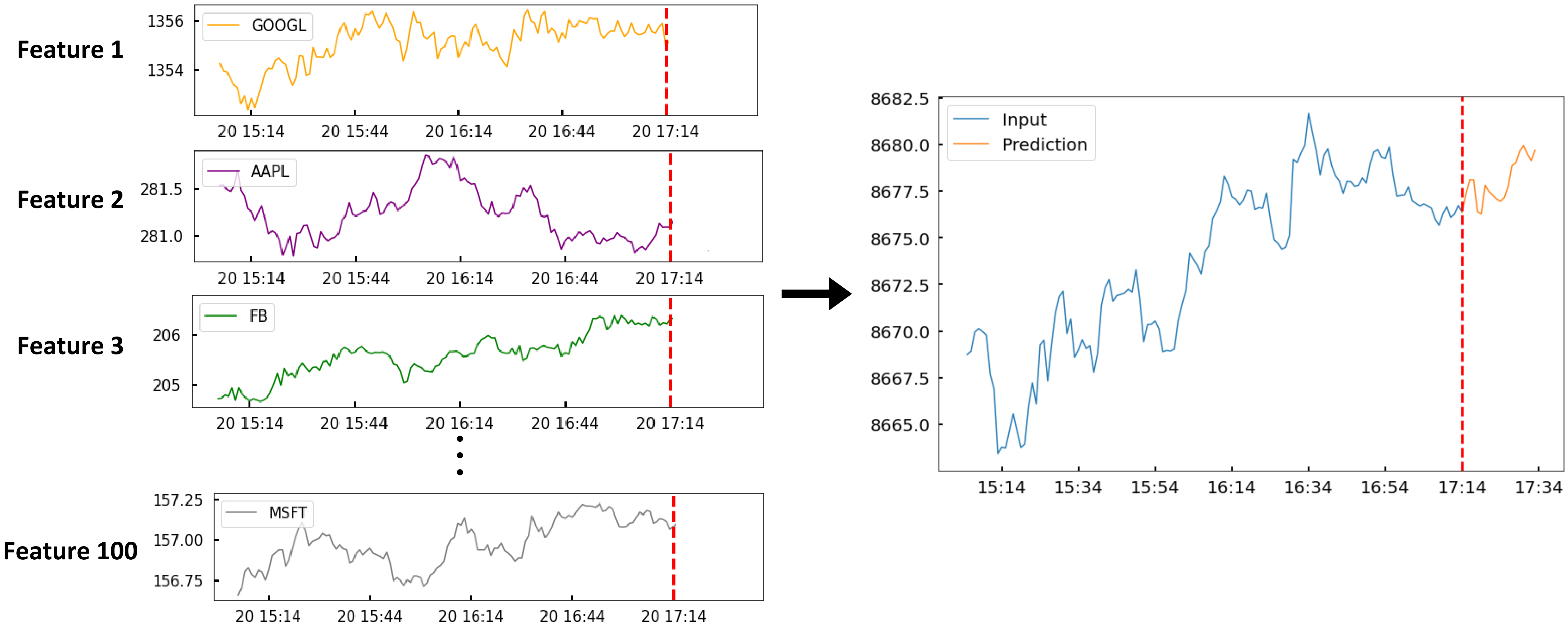
Prediction with noisy multivariate time series



NASDAQ 100 index

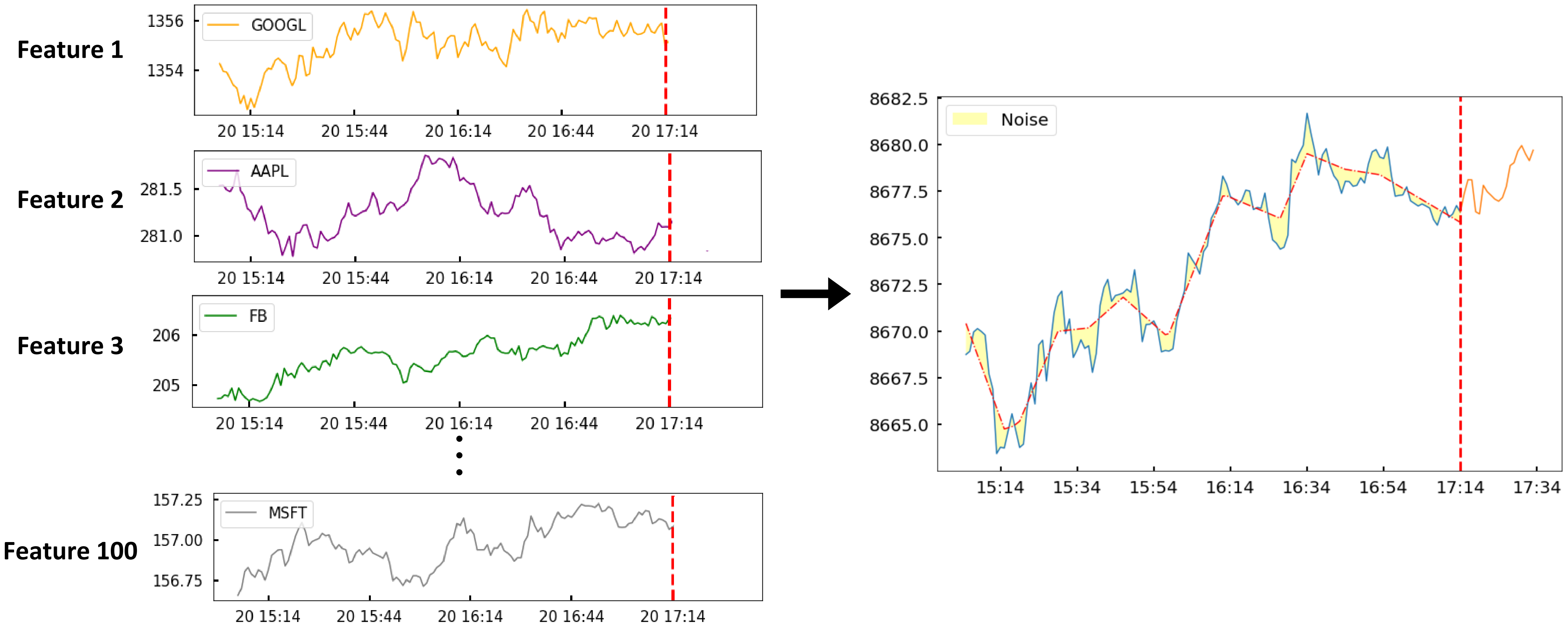
Problem

Prediction with noisy multivariate time series



Problem (in this paper)

Prediction with noisy multivariate time series



Deep Temporal Neural Networks with Trend Filtering

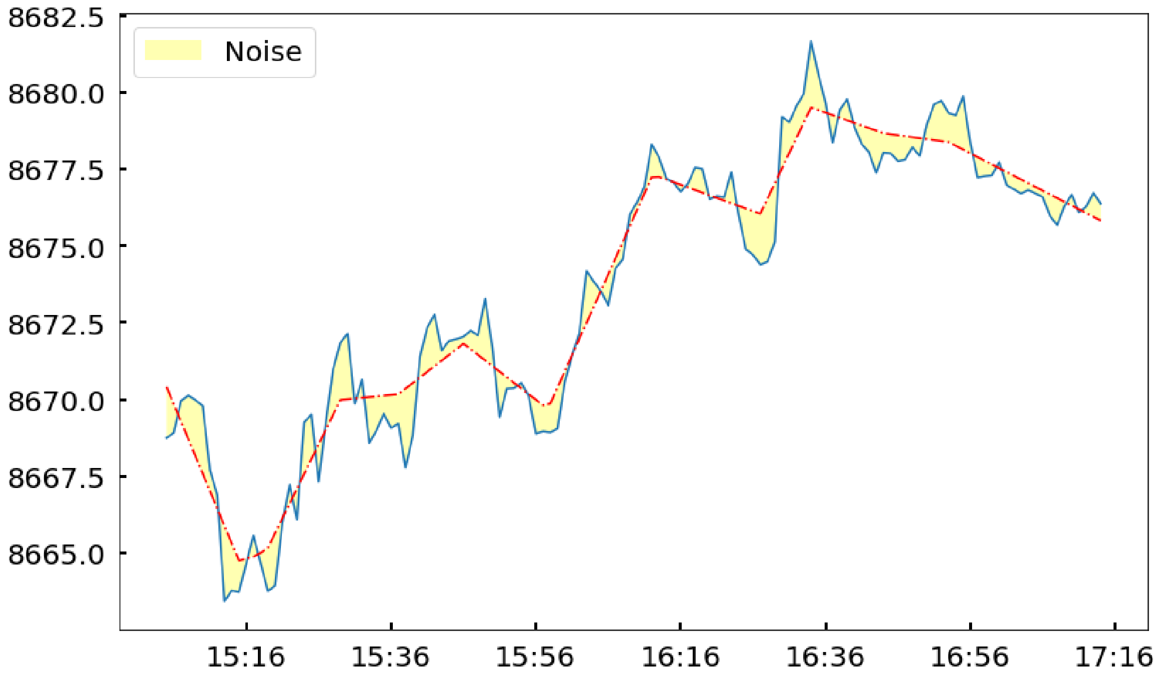
Deep Temporal Neural Networks

+

L1 Trend Filtering

Motivation

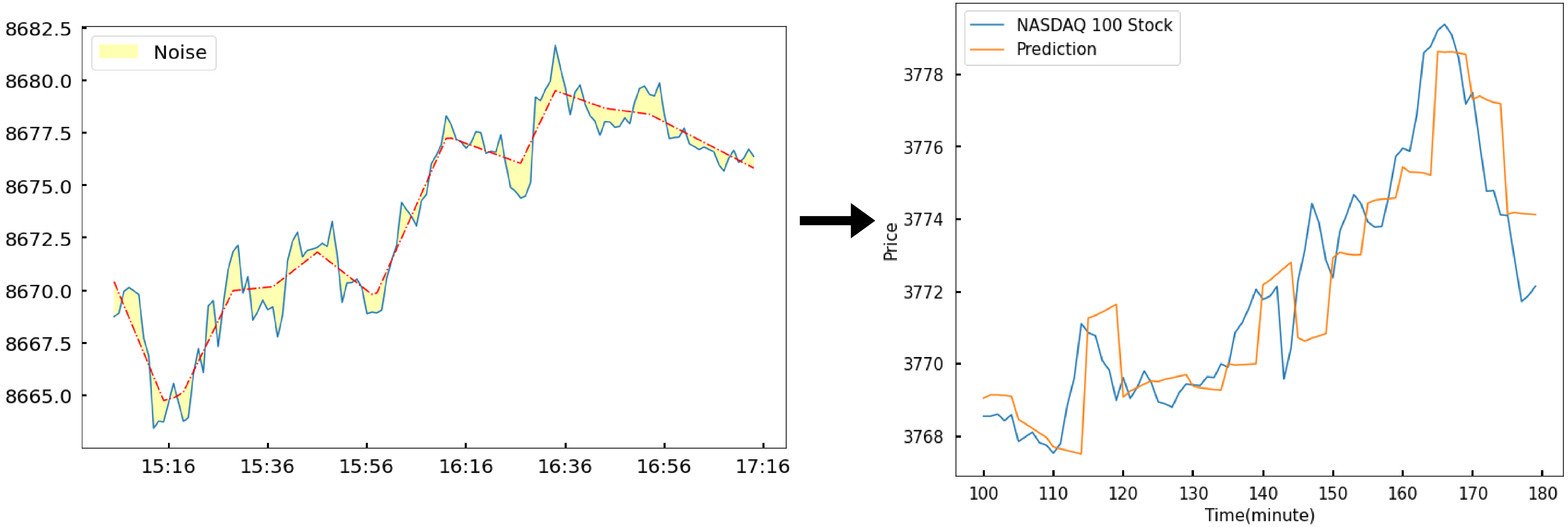
Because of noise in time series, designing a good model is not easy task.



Motivation

Because of noise in time series, designing a good model is not easy task.

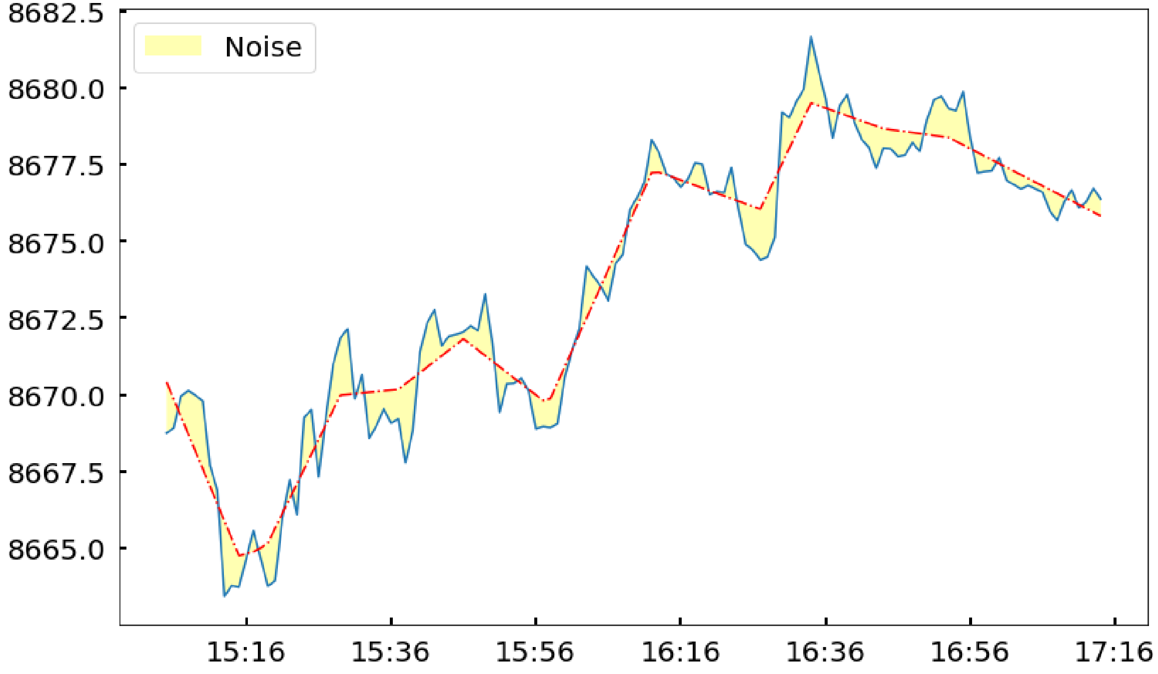
Noisy data makes the performance worse (lagging, overfitting).

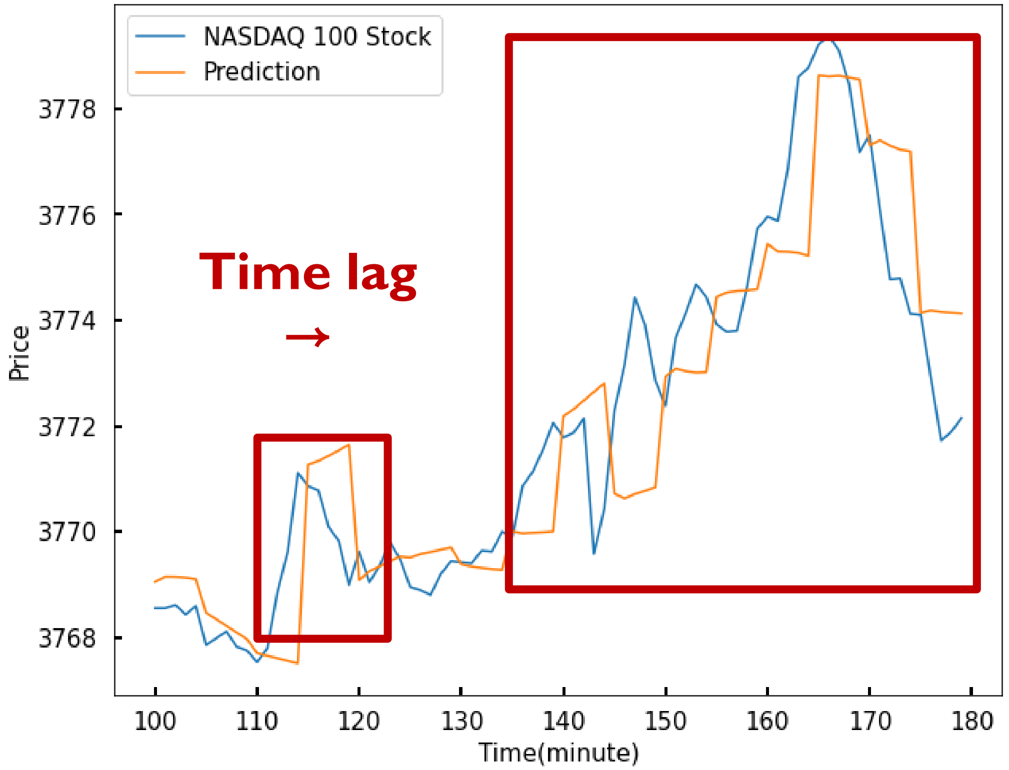


Motivation

Because of noise in time series, designing a good model is not easy task.

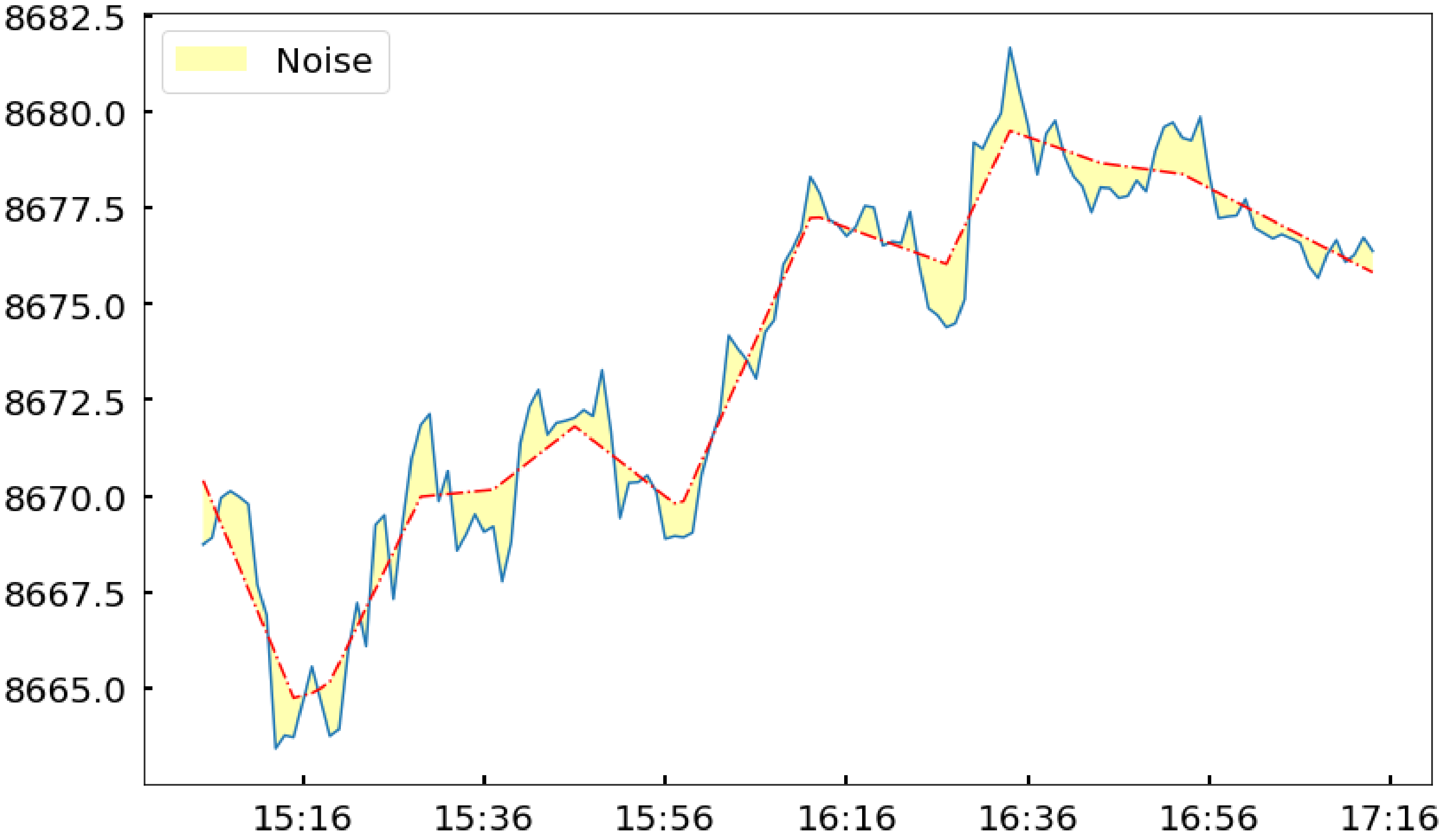
Noisy data makes the performance worse (lagging, overfitting).





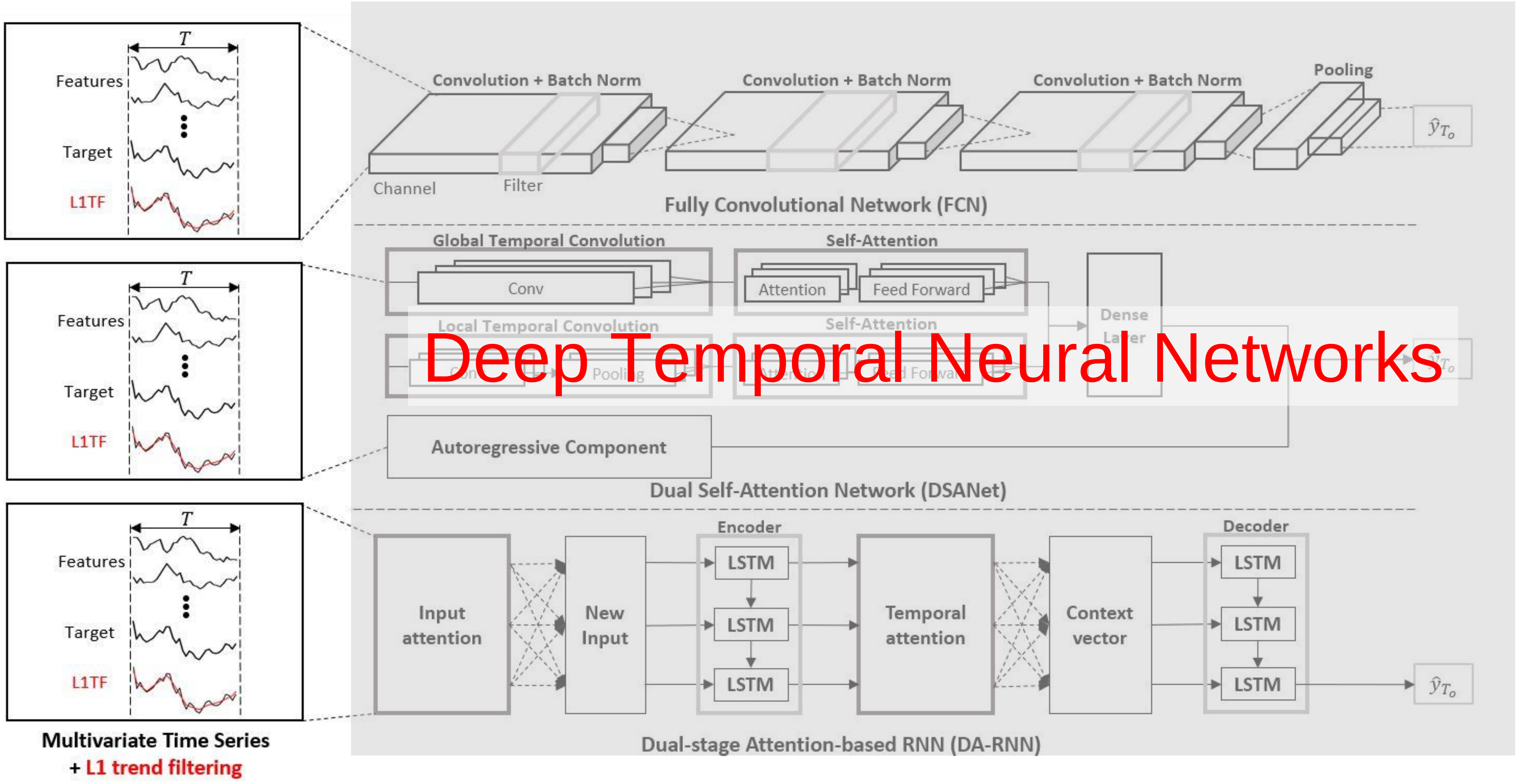
Motivation

Noise is defined as a local change towards the different trend of a long-term trend.



Our Approach

additional feature with L1 Trend Filtering (L1TF)* of original input data.



* Kim, Seung-Jean, et al. “L1 trend filtering." SIAM review 51.2 (2009): 339-360.

L1 trend filtering

L1 trend filtering* converts noisy data into a piecewise linear fashion.

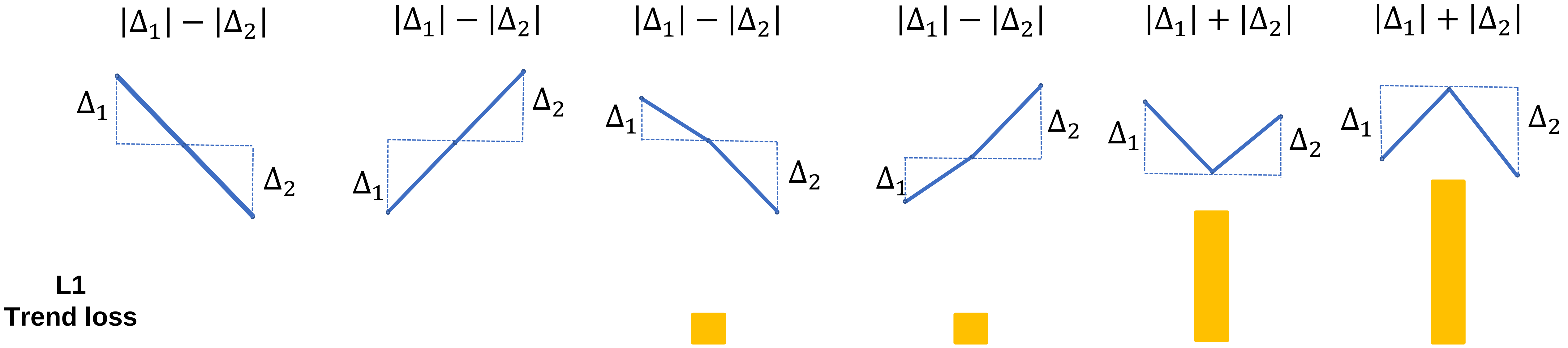
- Objective function

$$\frac{1}{2}\sum_{t=1}^{n}(\boldsymbol{x_t} - y_t)^2 + \lambda\sum_{t=2}^{n-1}|y_{t-1} - 2y_t + y_{t+1}|, \qquad t = 1, \ldots, n.\ \lambda \geq 0.$$

x_t : univariate time series

y_t : trend component x_t

λ : regularization parameter



L1 trend filtering

L1 trend filtering* converts noisy data into a piecewise linear fashion.

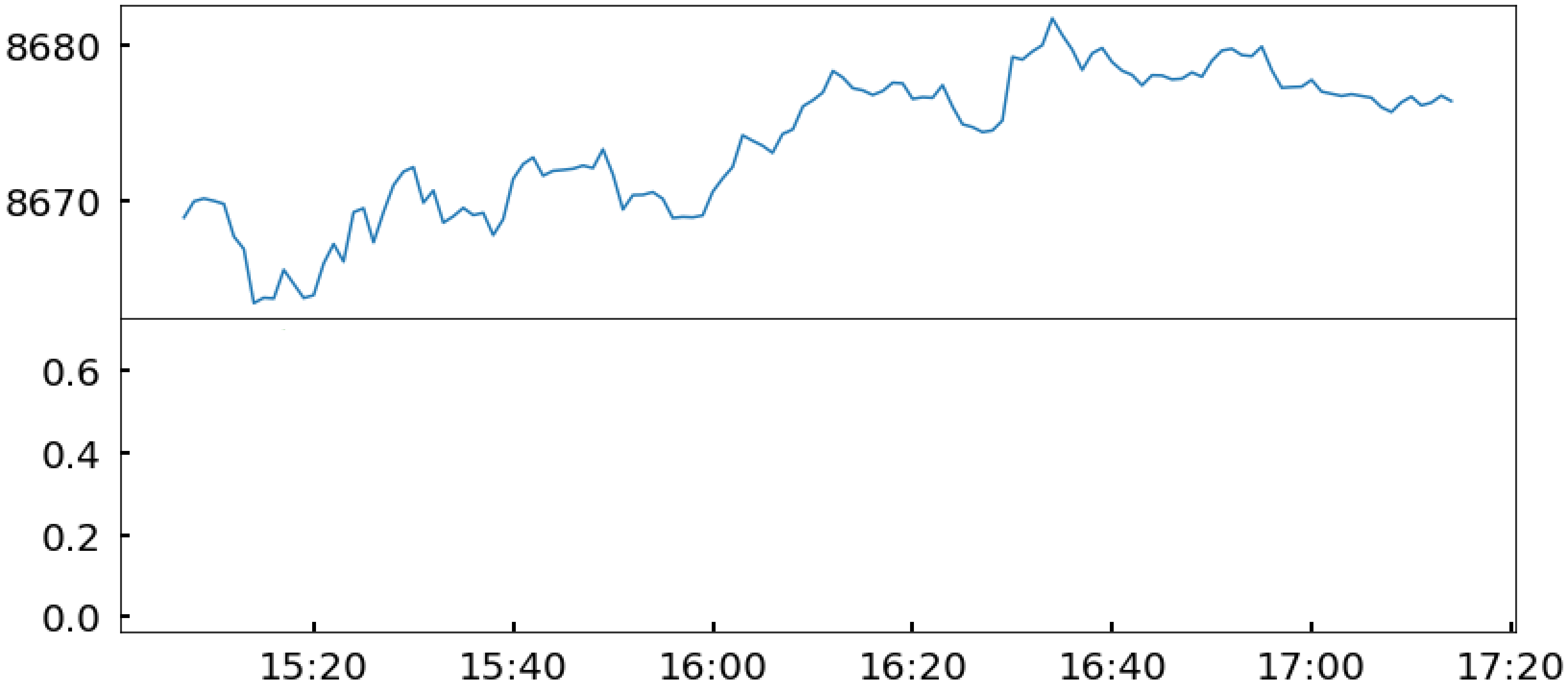
- Objective function

$$\min_{y_{1:n}} \frac{1}{2}\sum_{t=1}^{n}(x_t - y_t)^2 + \lambda \sum_{t=2}^{n-1}|y_{t-1} - 2y_t + y_{t+1}|, \qquad t = 1, \dots, n.\ \lambda \geq 0.$$

x_t : univariate time series

y_t : trend component in x_t

λ : regularization parameter



L1 trend filtering

L1 trend filtering* converts noisy data into a piecewise linear fashion.

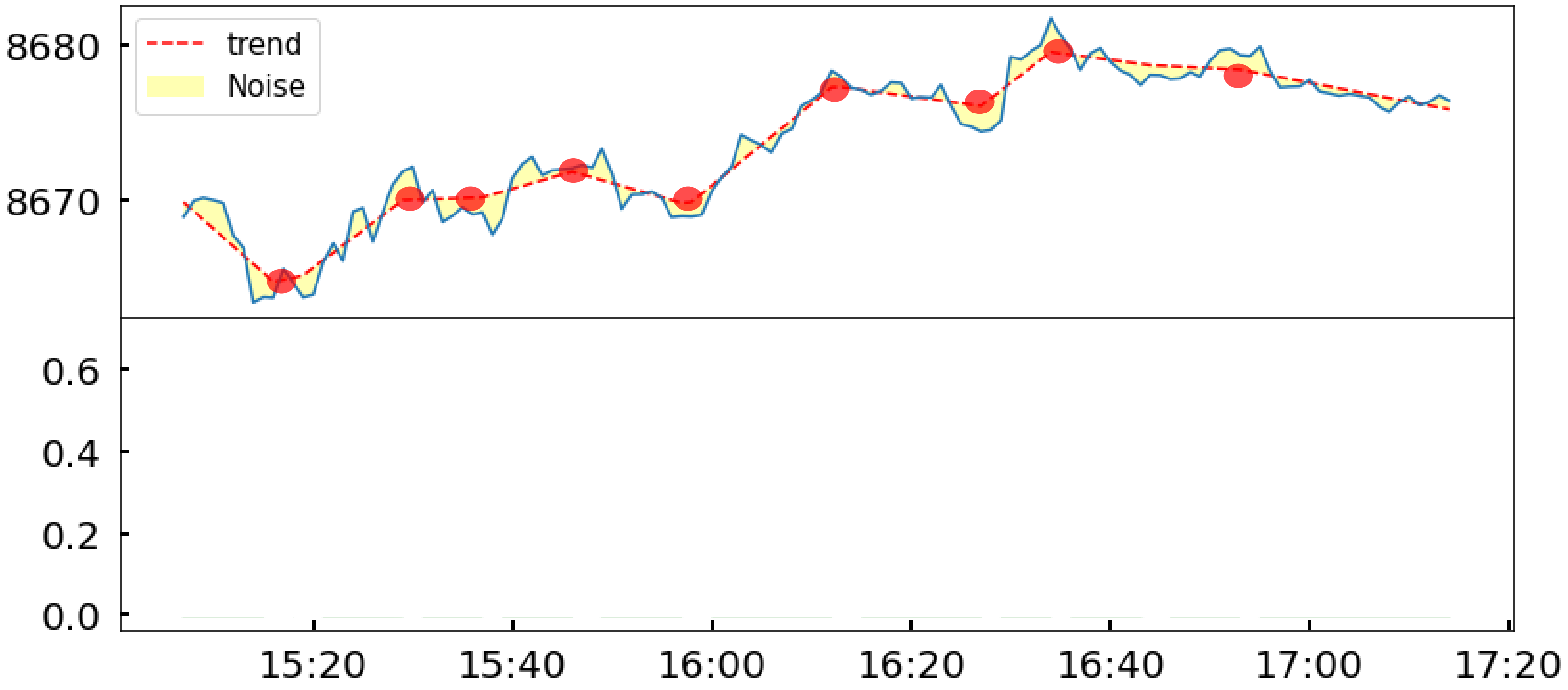
- Objective function

$$\min_{y_{1:n}} \frac{1}{2}\sum_{t=1}^{n}(x_t - y_t)^2 + \lambda \sum_{t=2}^{n-1}|y_{t-1} - 2y_t + y_{t+1}|, \qquad t = 1, \dots, n.\ \lambda \geq 0.$$

x_t : univariate time series

y_t : trend component in x_t

λ : regularization parameter



L1 trend filtering

L1 trend filtering* converts noisy data into a piecewise linear fashion.

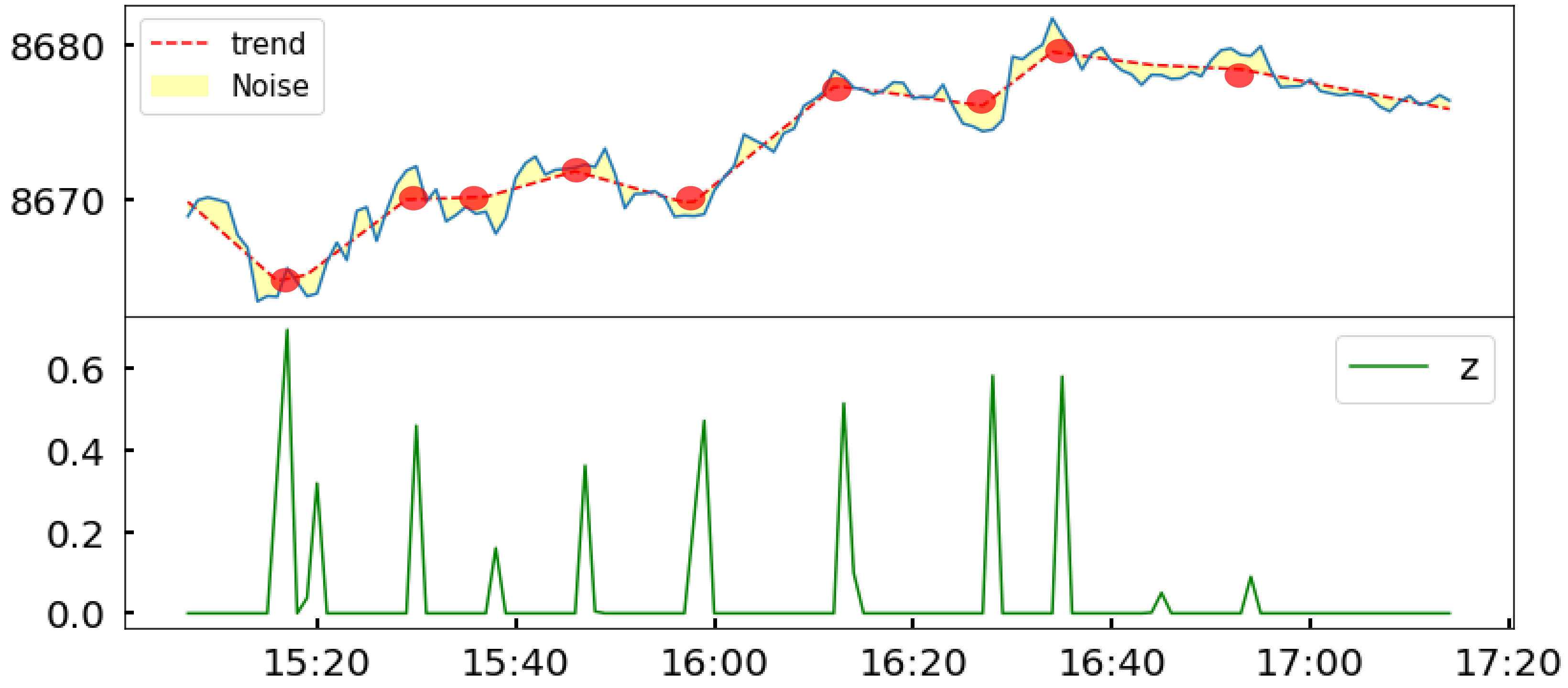
- Objective function

$$\min_{y_{1:n}} \frac{1}{2}\sum_{t=1}^{n}(x_t - y_t)^2 + \lambda \sum_{t=2}^{n-1} \underbrace{|y_{t-1} - 2y_t + y_{t+1}|}_{z}, \qquad t = 1, \dots, n.\ \lambda \geq 0.$$

x_t : univariate time series

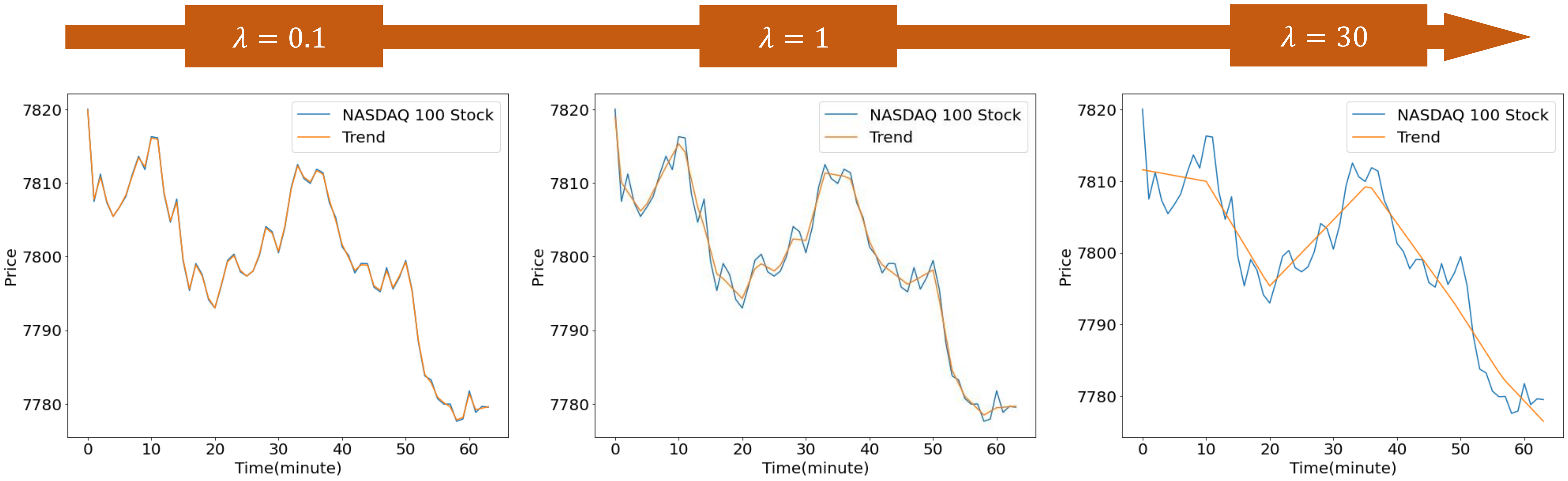
y_t : trend component in x_t

λ : regularization parameter



L1 trend filtering(λ)

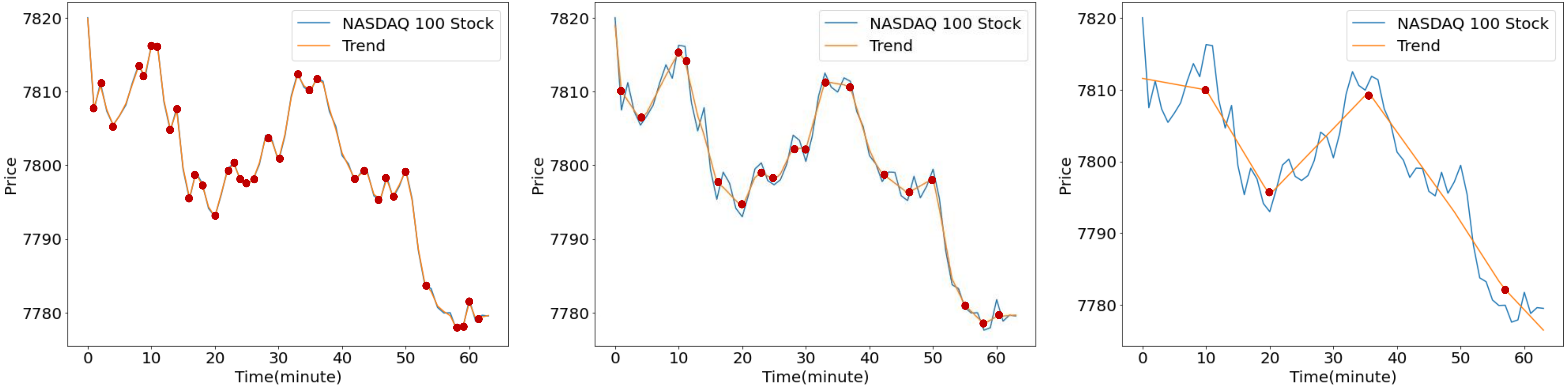
According to λ, knot is determined, and optimal trend is approximated.



L1 trend filtering(λ)

According to λ, knot is determined, and optimal trend is approximated.

$\lambda = 0.1$ $\lambda = 1$ $\lambda = 30$



knot = 34 knot = 18 knot = 4

Models

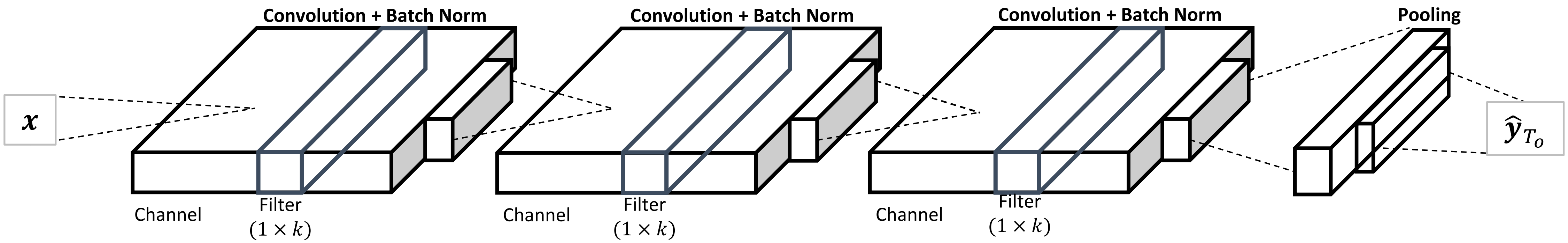
Fully Convolutional Network (FCN)*

Dual Self-Attention Network (DSANet)†

Dual-stage Attention-based Recurrent Neural Network (DA-RNN)††

* Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.
† Huang, Siteng, et al. "DSANet: Dual Self-Attention Network for Multivariate Time Series Forecasting." Proceedings of the 28th ACM International Conference on Information and Knowledge Management. 2019.
† † Qin, Yao, et al. "A dual-stage attention-based recurrent neural network for time series prediction." arXiv preprint arXiv:1704.02971 (2017).

Fully Convolutional Network (FCN)



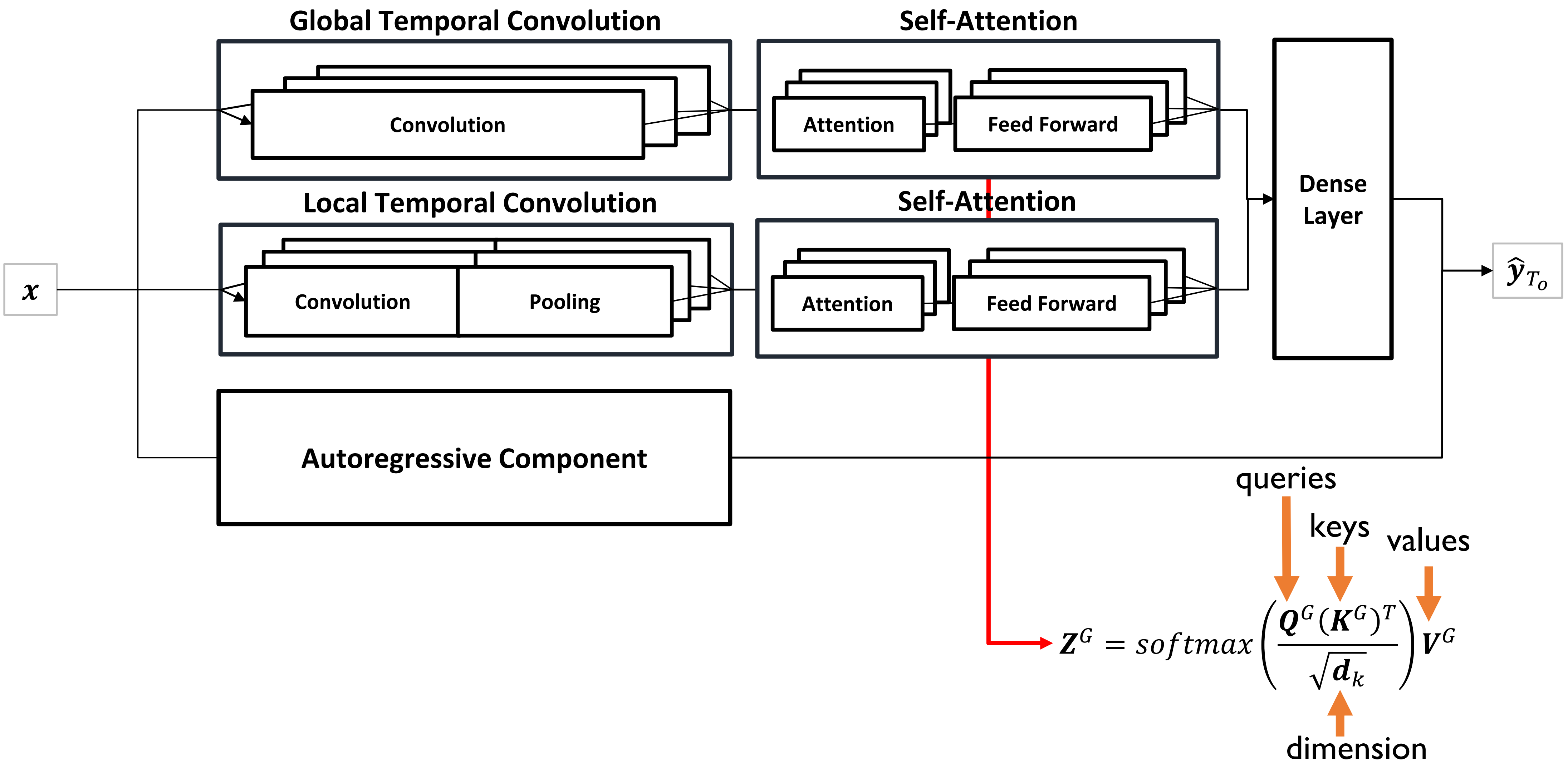
stride

$$\boldsymbol{y}_t = f_{ks}(\{\boldsymbol{x}_{st+\delta}\}_{0 \le \delta \le k})$$

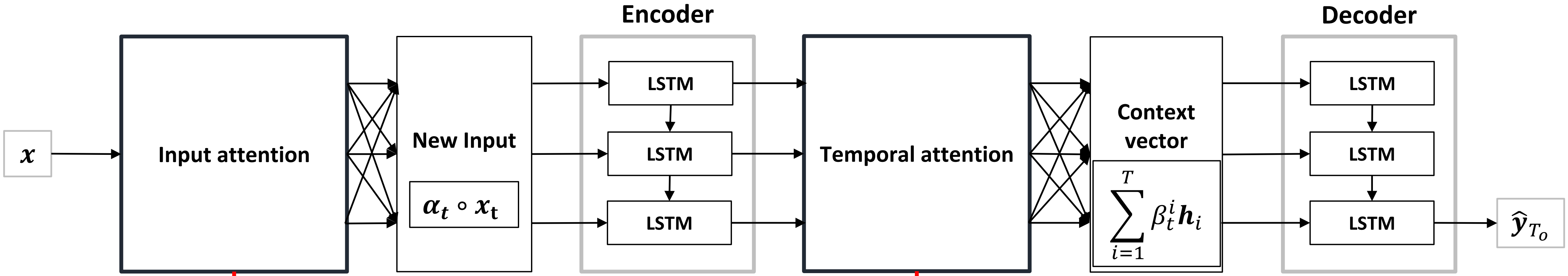
layer type

kernel size

Dual Self-Attention Network (DSANet)



Dual-stage Attention-based RNN (DA-RNN)



input

encoder hidden, cell state

$$e_t^k = \boldsymbol{v}_e^\top \tanh\left(W_e[\boldsymbol{h}_{t-1}; \boldsymbol{s}_{t-1}] + U_e \boldsymbol{x}^k\right)$$

$$\alpha_t^k = \frac{\exp\left(e_t^k\right)}{\sum_{i=1}^{n} \exp(e_t^i)}$$

encoder hidden state

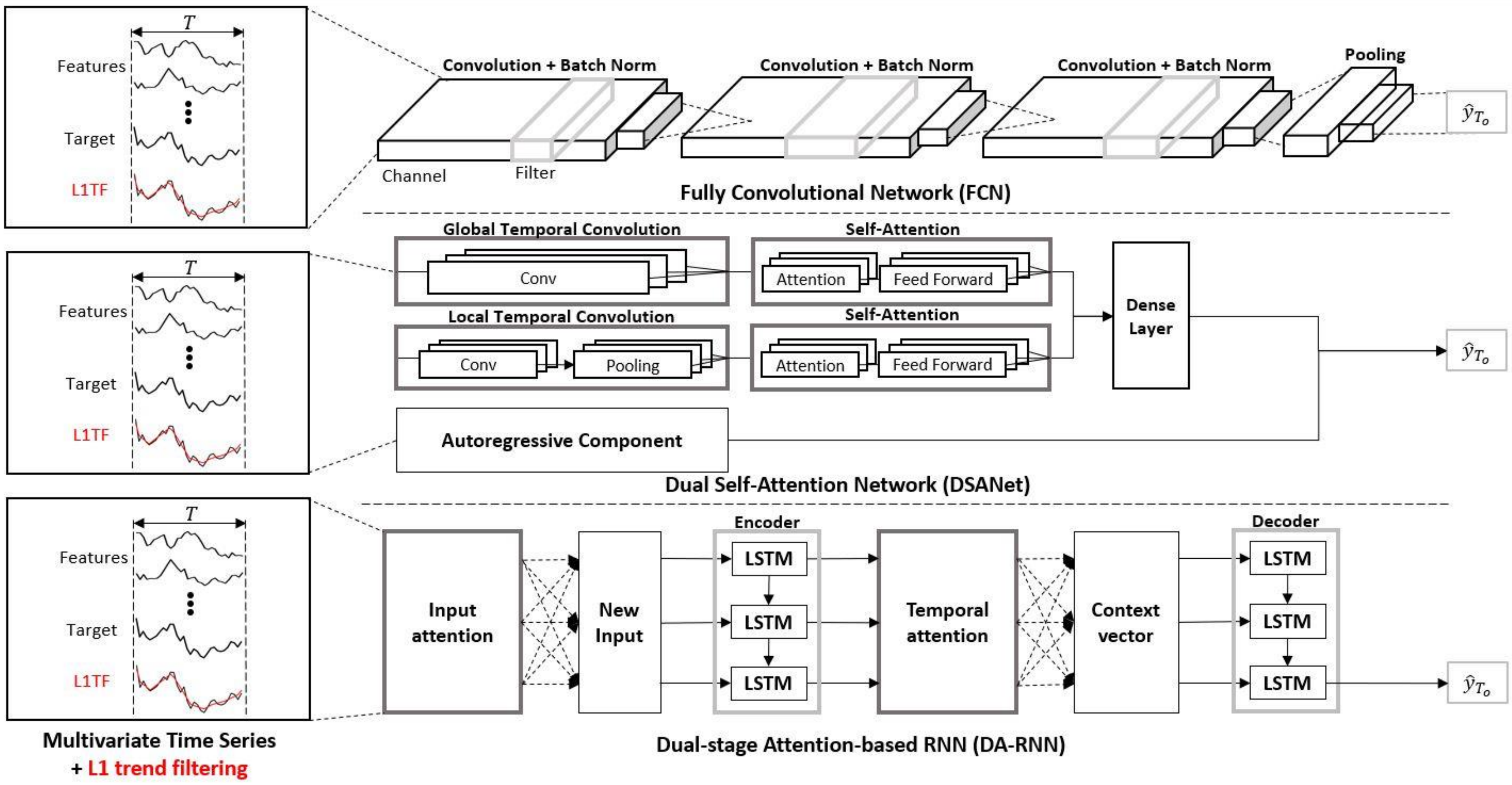
decoder hidden, cell state

$$l_t^i = \boldsymbol{v}_d^\top \tanh(W_d[\boldsymbol{d}_{t-1}; \boldsymbol{s'}_{t-1}] + U_d \boldsymbol{h}_i)$$

$$\beta_t^i = \frac{\exp\left(l_t^i\right)}{\sum_{j=1}^{T} \exp(l_t^j)}$$

Our Approach

additional feature with L1 Trend Filtering (L1TF)* of original input data.



* Kim, Seung-Jean, et al. “L1 trend filtering." SIAM review 51.2 (2009): 339-360.

Experimental Results

Five real-world index stock price datasets:

- NASDAQ 100 index
- EURO STOXX 50 Index
- Dow Jones Industrial Average
- FTSE 100 Index
- TSX 60 Index

Dataset

Dataset	Exchange	Size	
		Train	Test
NASDAQ 100 Stock	NASDAQ	45.6K (19-07-01~19-12-20)	5.0K (19-12-20~20-01-10)
Dow Jones Industrial Average(DIJA)	NYSE[1], NASDAQ	48.7K (19-07-02~19-12-23)	5.4K (19-12-23~20-01-13)
EURO STOXX 50 Stock	Eurozone[2]	65.3K (19-07-22~20-01-14)	7.2K (20-01-14~20-01-31)
FTSE 100 Stock	LSE[3]	61.9K (19-07-22~20-01-14)	6.8K (20-01-14~20-01-31)
TSX 60 Stock	TSE[4]	49.4K (19-07-22~20-01-14)	5.4K (20-01-14~20-01-31)

[1] : New York Stock Exchange (NYSE)
[2] : 12 countries in Europe (e.g Austria, France, Germany, etc)
[3] : London Stock Exchange (LSE)
[4] : Toronto Stock Exchange (TSE)

Parameter settings

	Notation	Hyperparameter	Setting
Data window size	T_i	Input window size	64
	T_o	Output window size	5
ARIMA*	p, q	Period to lag, lag of the error (The order of AR, MA)	1, 0
	d	Differencing	0
FCN	d	Filter	32
	k	Kernel size	7, 5, 3
DSANet	n_{head}	The number of multi-head	8
DARNN	m, p	Encoder, decoder hidden dimension	$\in \{64, 128\}$
L1 trend filtering	λ	Lambda	0.005

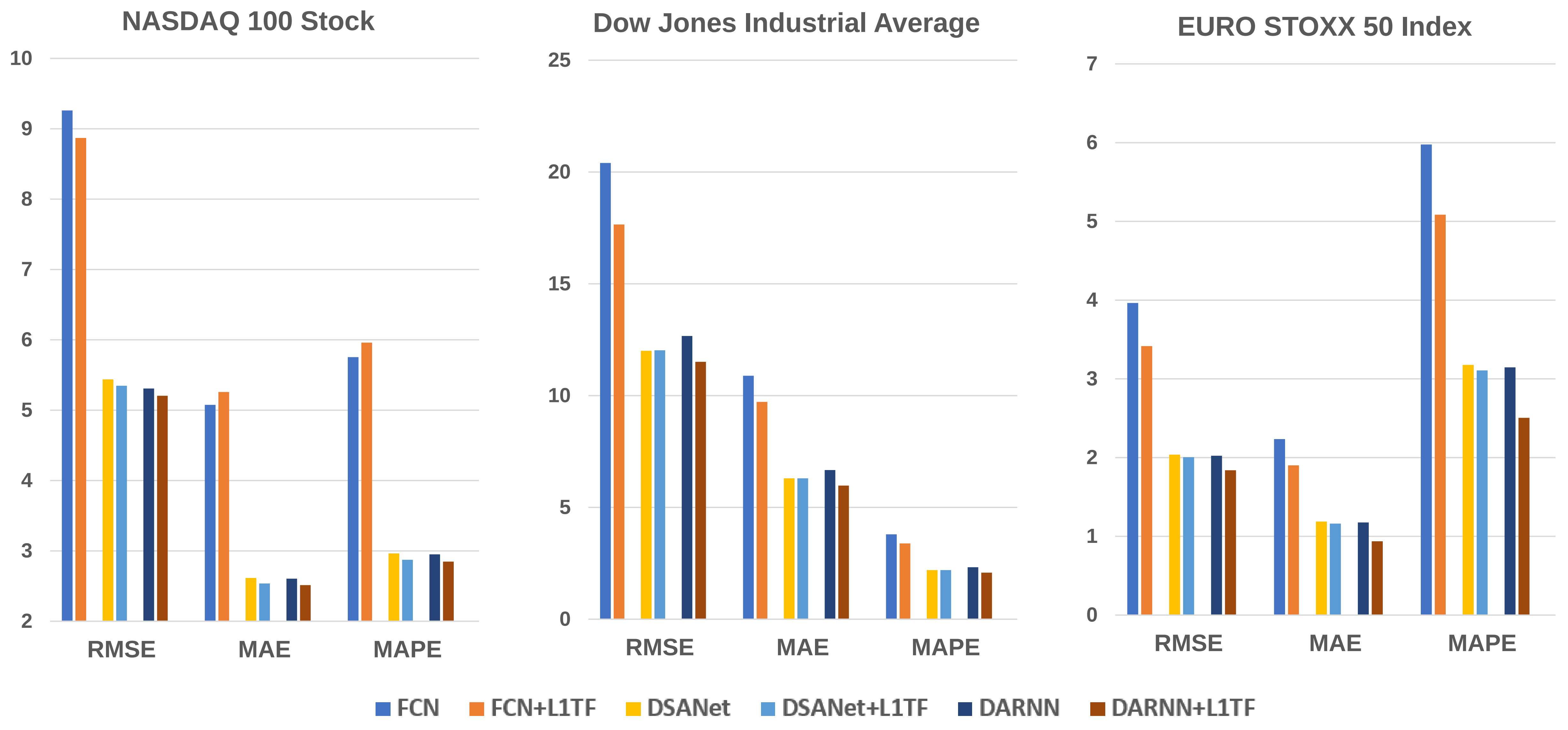
* Hillmer, Steven Craig, and George C. Tiao. "An ARIMA-model-based approach to seasonal adjustment." Journal of the American Statistical Association 77.377 (1982): 63-70.

Quantitative Results

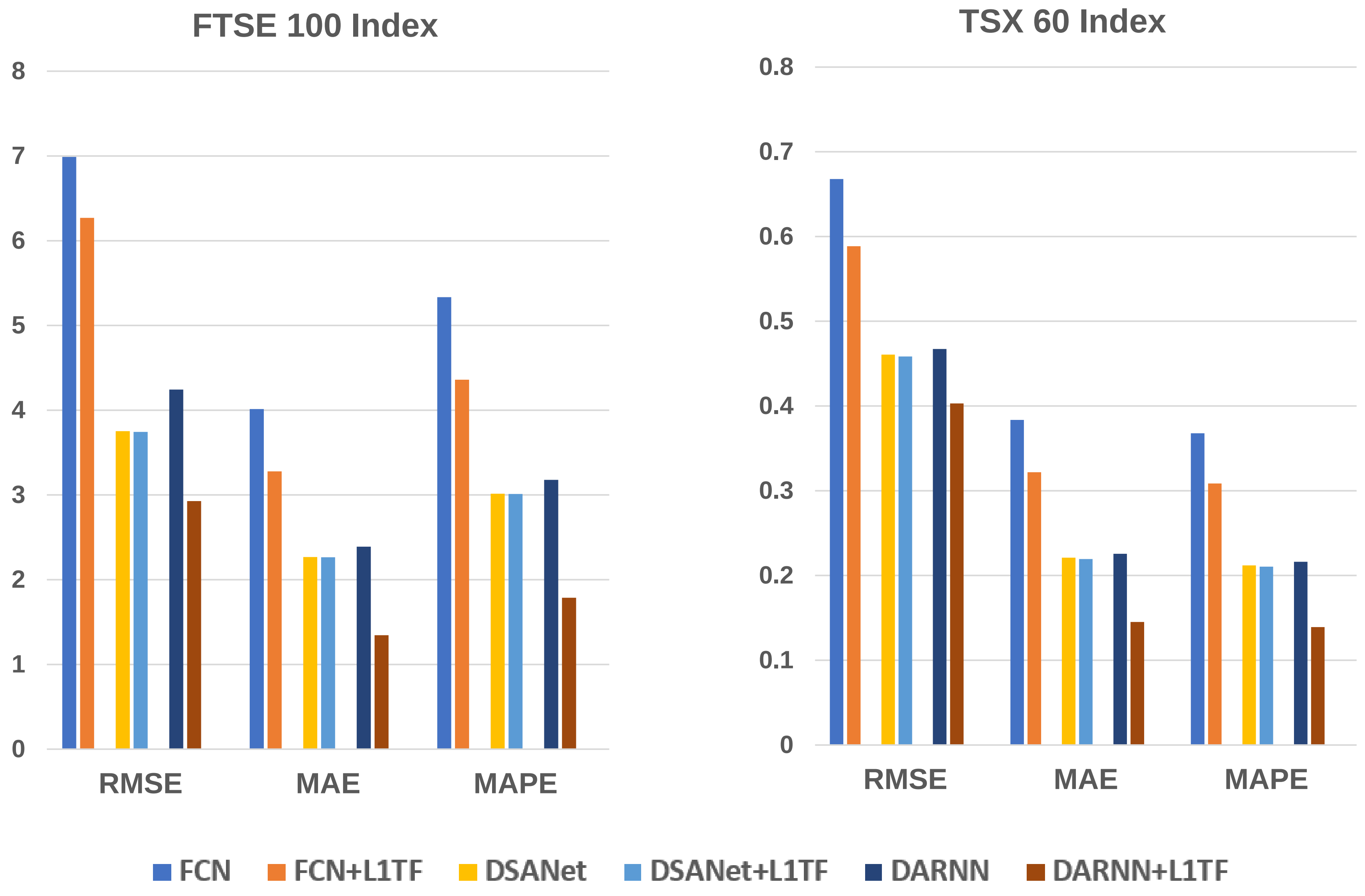
Model	NASDAQ 100 Index (Growth rate : 3.26%)				Dow Jones Industrial Average (Growth rate : 1.16%)				EURO STOXX 50 Index (Growth rate : -3.47%)				FTSE 100 Index (Growth rate : -4.26%)				TSX 60 Index (Growth rate : 0.16%)			
	RMSE	MAE	MAPE ($\times 10^4$)	Rate of Return (%)	RMSE	MAE	MAPE ($\times 10^4$)	Rate of Return (%)	RMSE	MAE	MAPE ($\times 10^4$)	Rate of Return (%)	RMSE	MAE	MAPE ($\times 10^4$)	Rate of Return (%)	RMSE	MAE	MAPE ($\times 10^4$)	Rate of Return (%)
LA*	252.65	10.2805	11.52	0	85.681	3.5038	9.5152	0	787.98	29.0741	10.0784	0	176.34	7.1058	9.6254	0	27.976	1.0155	9.8054	0
ARIMA	8.5656	4.4327	5.02287	1.01	19.7453	10.4543	3.6458	-1.21	3.8829	2.1286	5.6934	0.82	6.9347	3.8055	5.0598	-0.55	0.6146	0.3508	3.3636	0.001
FCN	9.2603	5.0765	5.7540	7.02	20.4051	10.8882	3.7968	9.83	3.9643	2.2349	5.9765	1.08	6.9883	4.0133	5.3341	-0.43	0.6678	0.3836	3.6775	-0.003
FCN +LITF	8.8692	5.2586	5.9606	7.02	17.6532	9.7217	3.3899	9.83	3.4165	1.9012	5.0851	1.04	6.2695	3.2788	4.3594	-0.30	0.5886	0.3219	3.0861	-0.003
DSANet	5.4383	2.6163	2.9638	6.99	12.0051	6.3011	2.1974	9.69	2.0377	1.1880	3.1777	-1.85	3.7534	2.2669	3.0142	-0.25	0.4606	0.2210	2.1185	0.65
DSANet +LITF	5.3484	2.5368	2.874	**7.05**	12.0281	6.3009	2.1973	**12.41**	2.0049	1.1619	3.1077	-0.18	3.7444	2.2648	3.0115	0.33	0.4584	0.2195	2.1044	**0.67**
DARNN	5.3079	2.6057	2.9514	-3.93	12.6665	6.6695	2.3258	-0.75	2.0232	1.1764	3.1469	-0.14	4.244	2.3906	3.1788	-0.24	0.4673	0.2256	2.1627	0.09
DARNN +LITF	**5.2053**	**2.5146**	**2.8486**	-0.71	**11.5129**	**5.9758**	**2.0844**	8.08	**1.8392**	**0.9368**	**2.5057**	**7.39**	**2.9276**	**1.3448**	**1.7874**	**5.75**	**0.4031**	**0.1451**	**1.3917**	0.3

*Lookahead : the metric is calculated between the points of the last prediction steps and current value

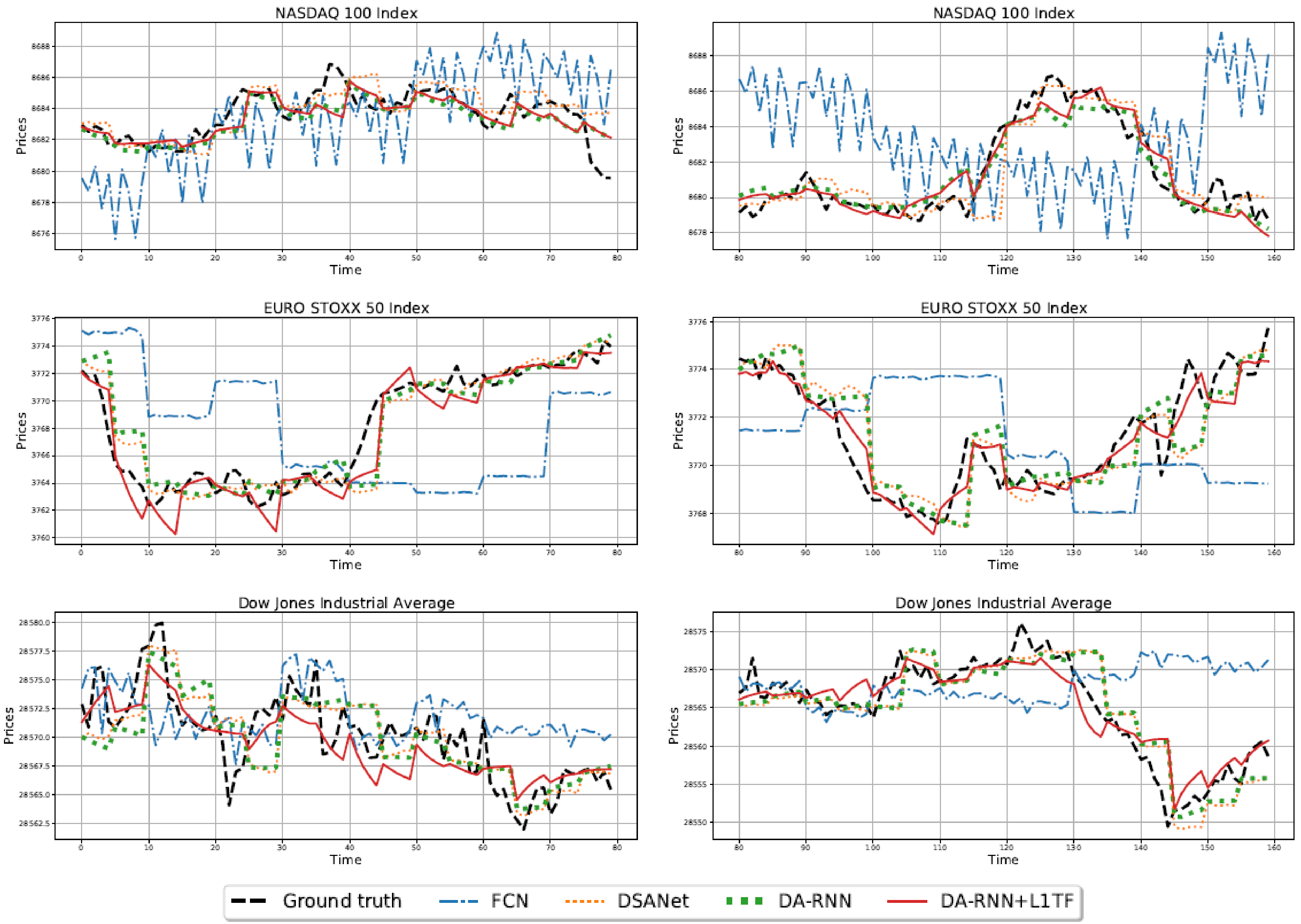
Quantitative Results



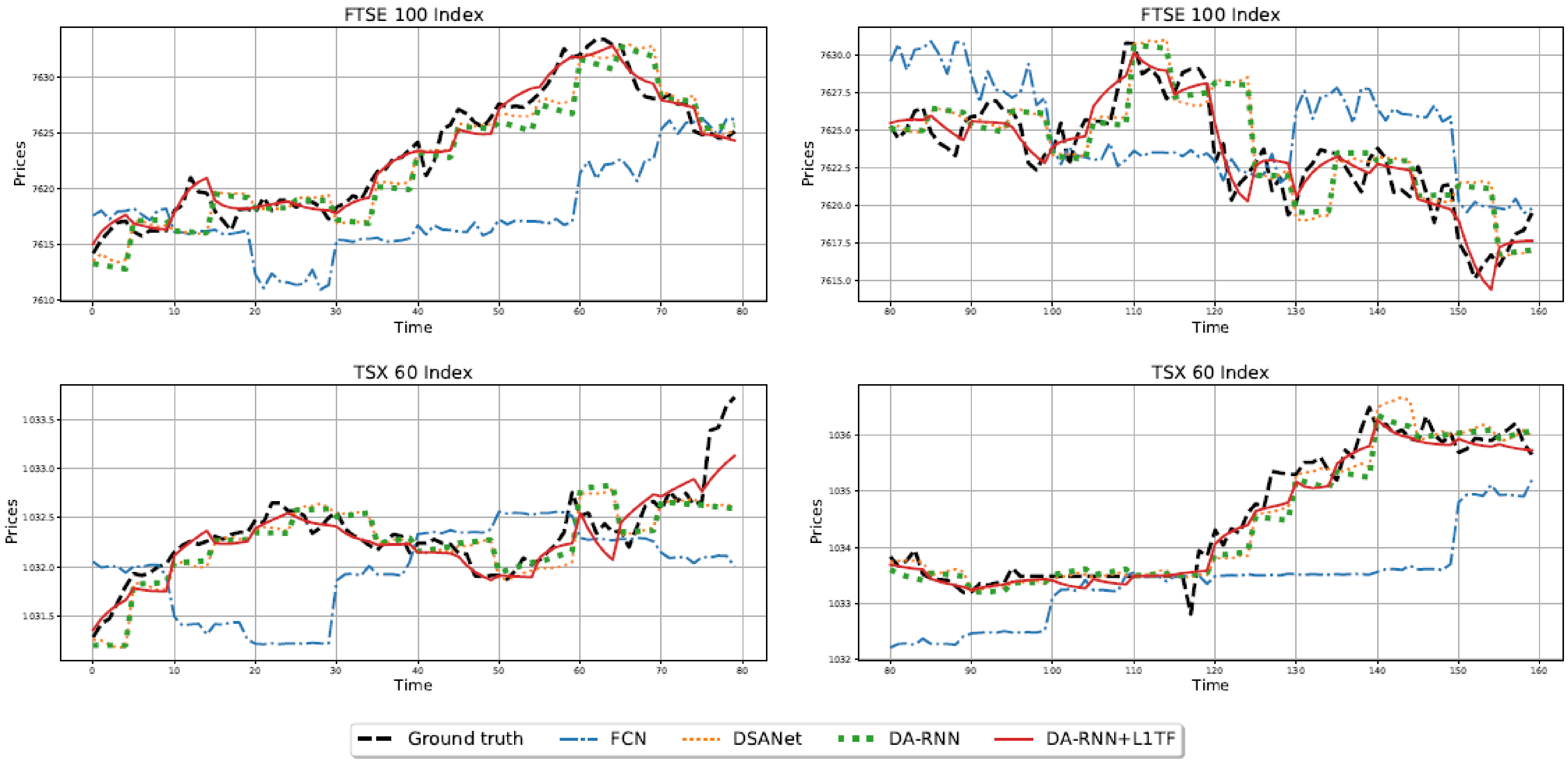
Quantitative Results



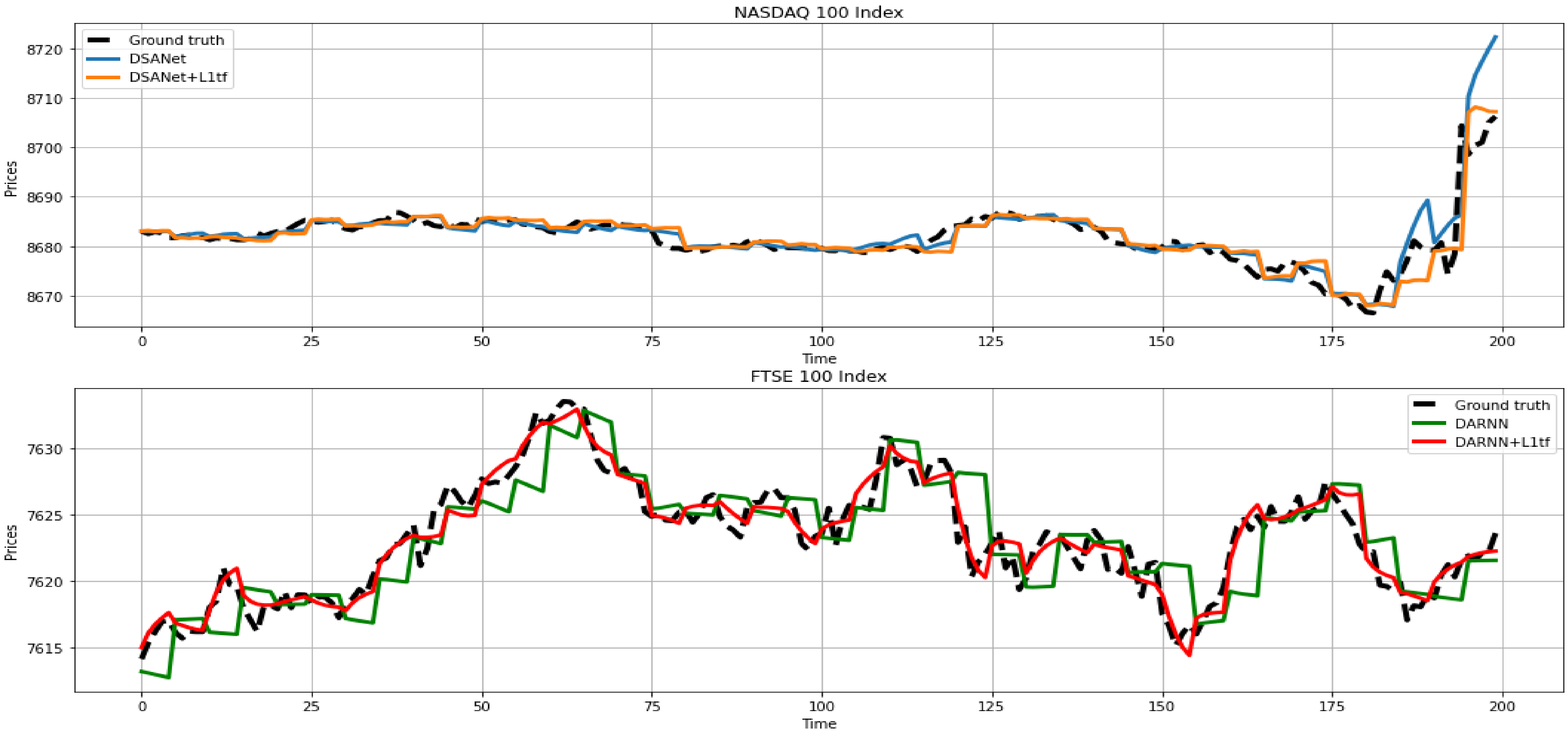
Prediction results (FCN, DSANet, DA-RNN, DA-RNN+L1TF)



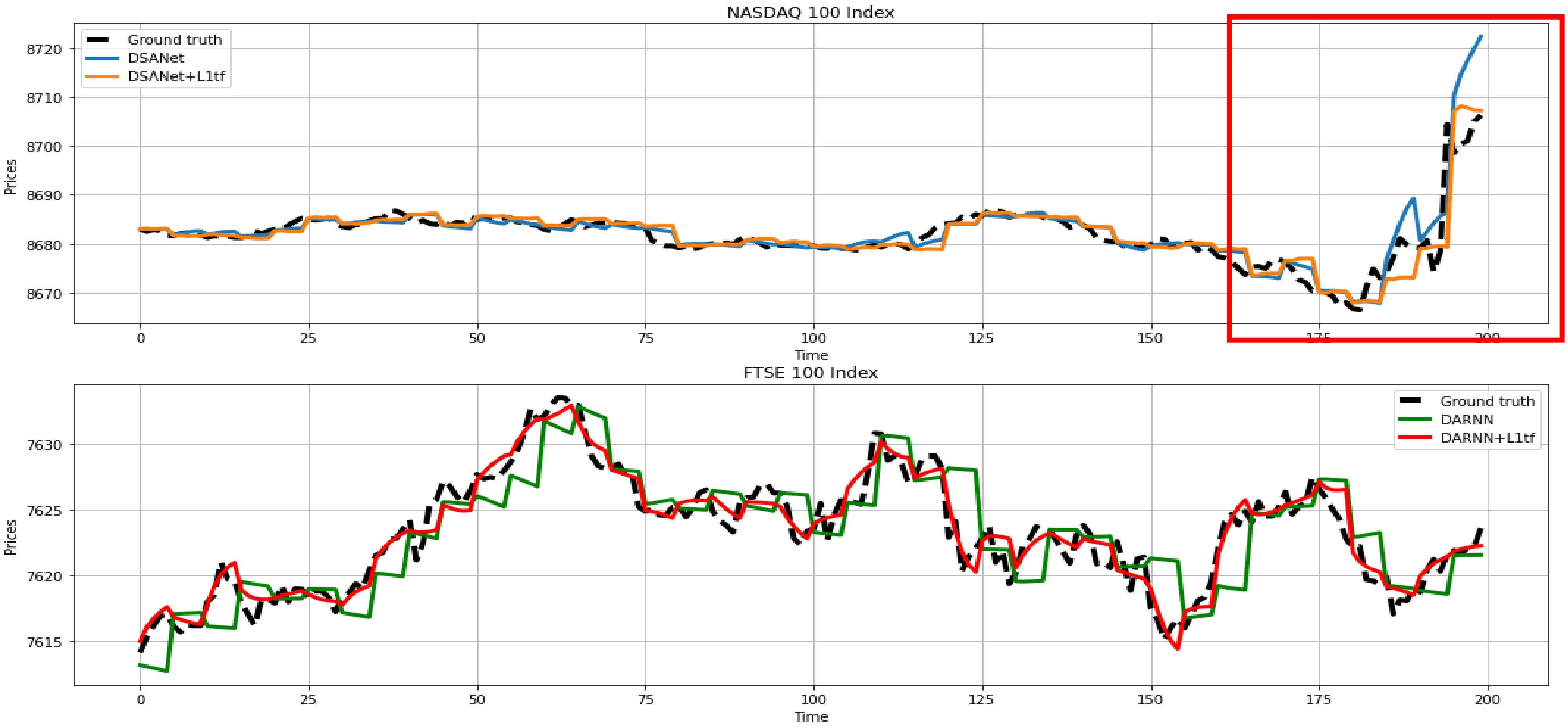
Prediction results (FCN, DSANet, DA-RNN, DA-RNN+L1TF)



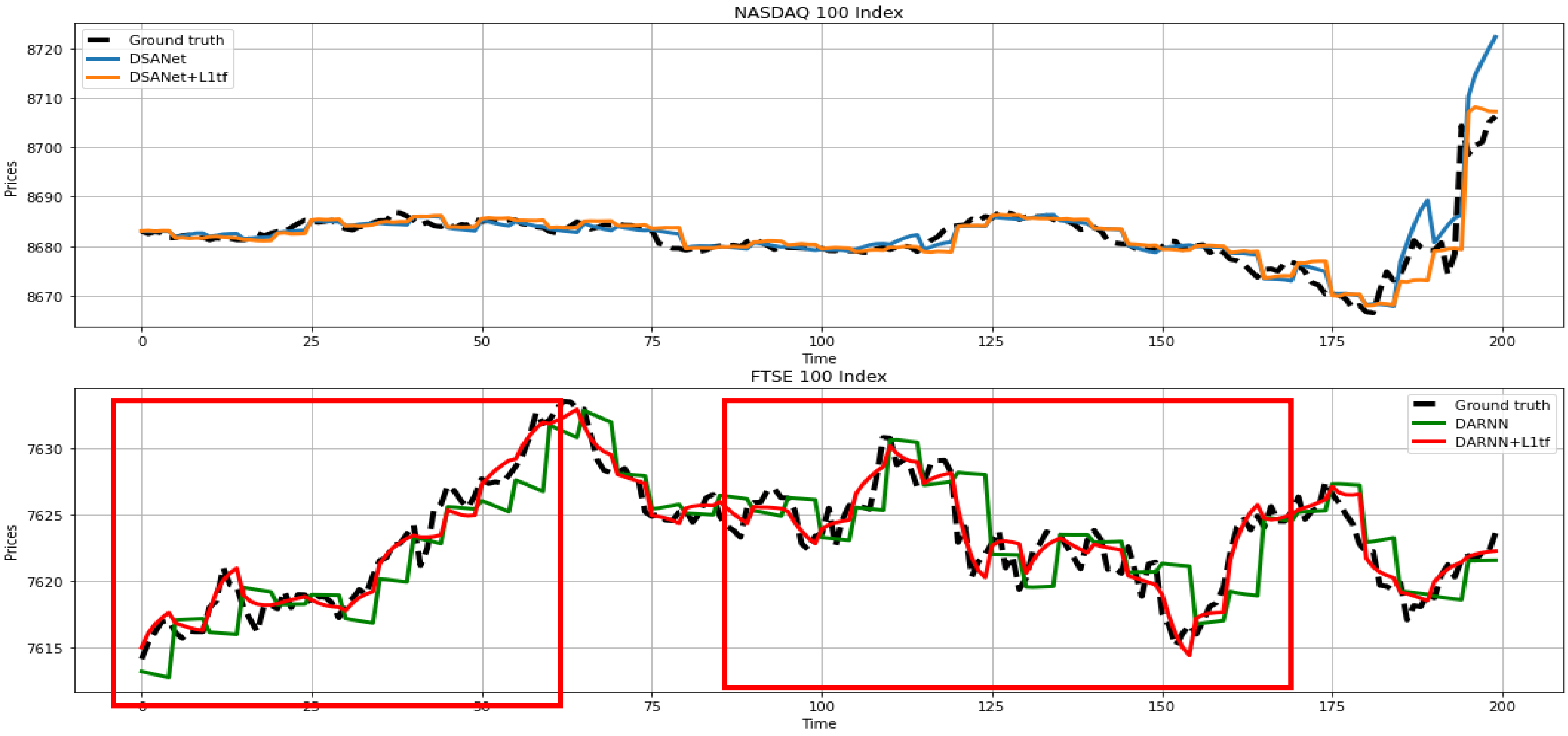
Prediction comparison in NASDAQ 100 Index, FTSE 100 Index



Prediction comparison in NASDAQ 100 Index, FTSE 100 Index



Prediction comparison in NASDAQ 100 Index, FTSE 100 Index



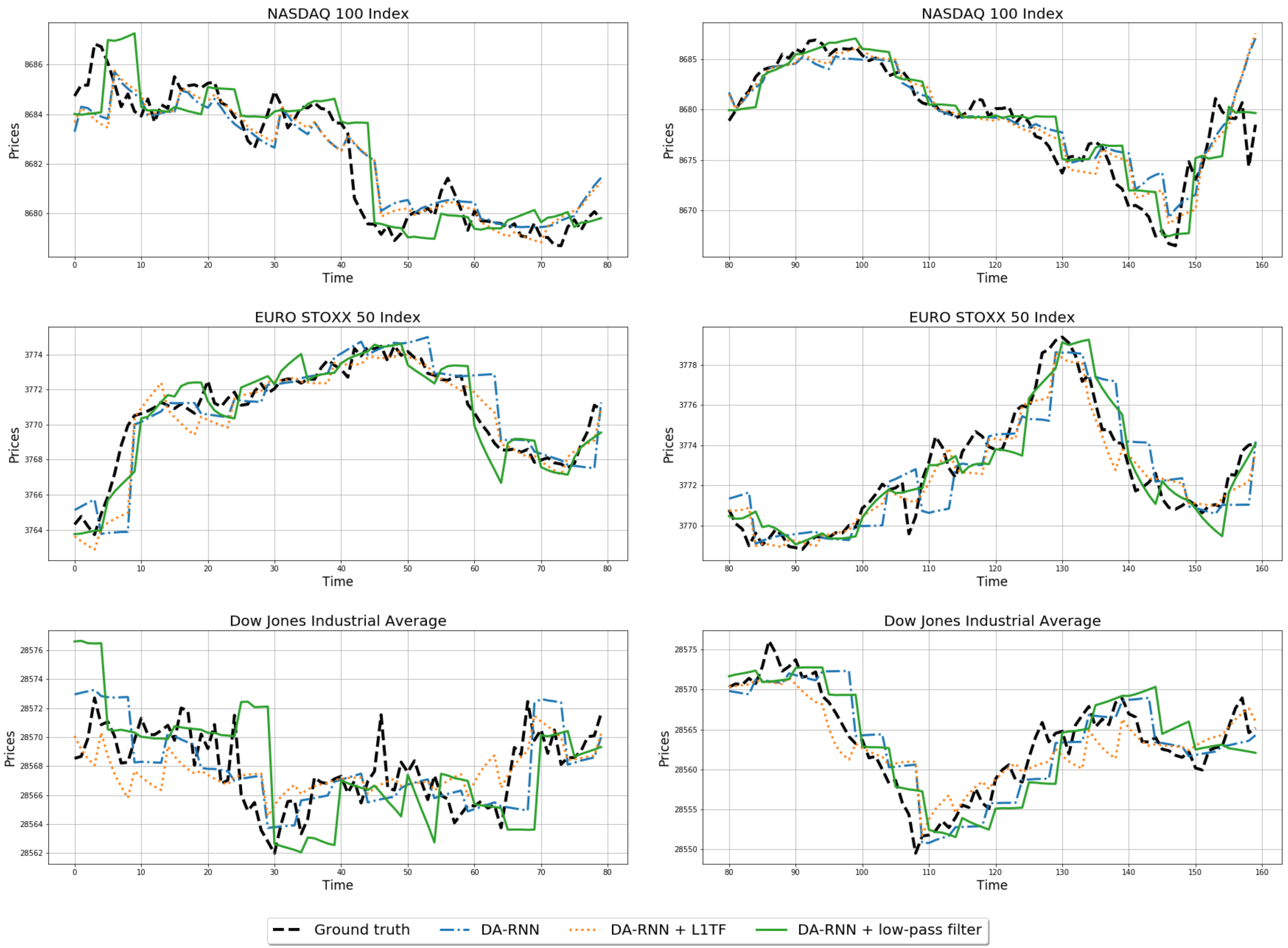
T-test with (w/) L1TF and without (w/o) L1TF

Dataset	Model	w/ L1TF	w/o L1TF	p-value
NASDAQ 100 Index	FCN	8.8692	9.2603	**0.007**
	DSANet	5.3484	5.4383	0.998
	DARNN	5.2053	5.3079	**0.027**
EURO STOXX 50 Index	FCN	3.4165	3.9643	**0.000**
	DSANet	2.0049	2.0377	**0.000**
	DARNN	1.8392	2.0232	**0.000**
Dow Jones Industrial Average	FCN	17.6532	20.4051	**0.000**
	DSANet	12.0281	12.0051	0.466
	DARNN	11.5129	12.6665	**0.000**
FTSE 100 Index	FCN	6.2695	6.9883	**0.000**
	DSANet	3.7444	3.7534	0.380
	DARNN	2.9276	4.244	**0.000**
TSX 60 Index	FCN	0.5886	0.6678	**0.000**
	DSANet	0.4584	0.4606	**0.013**
	DARNN	0.4031	0.4673	**0.000**

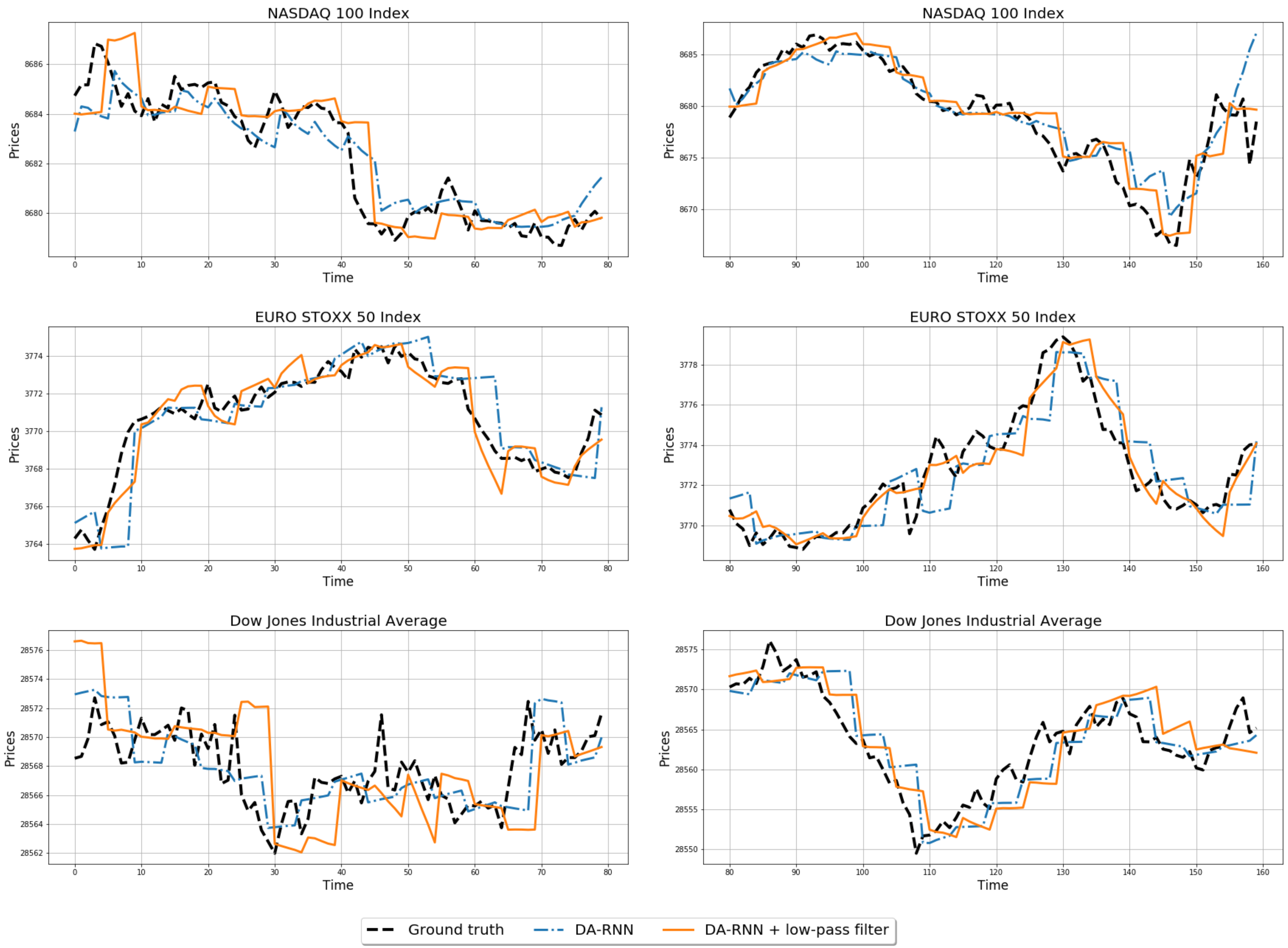
Conclusion

- Our research topic is about "Prediction with noisy multivariate time series".
- We proposed a novel method that includes the L1 trend filtering feature which is helpful for the task of multivariate time series forecasting.
- We applied this method to deep temporal neural networks that can detect certain important signals more easily and the filtering simplifies the prediction of noise time series.
- The paired t-test of two methods shows the statistical significance of the fact that proposed method achieves better performances.

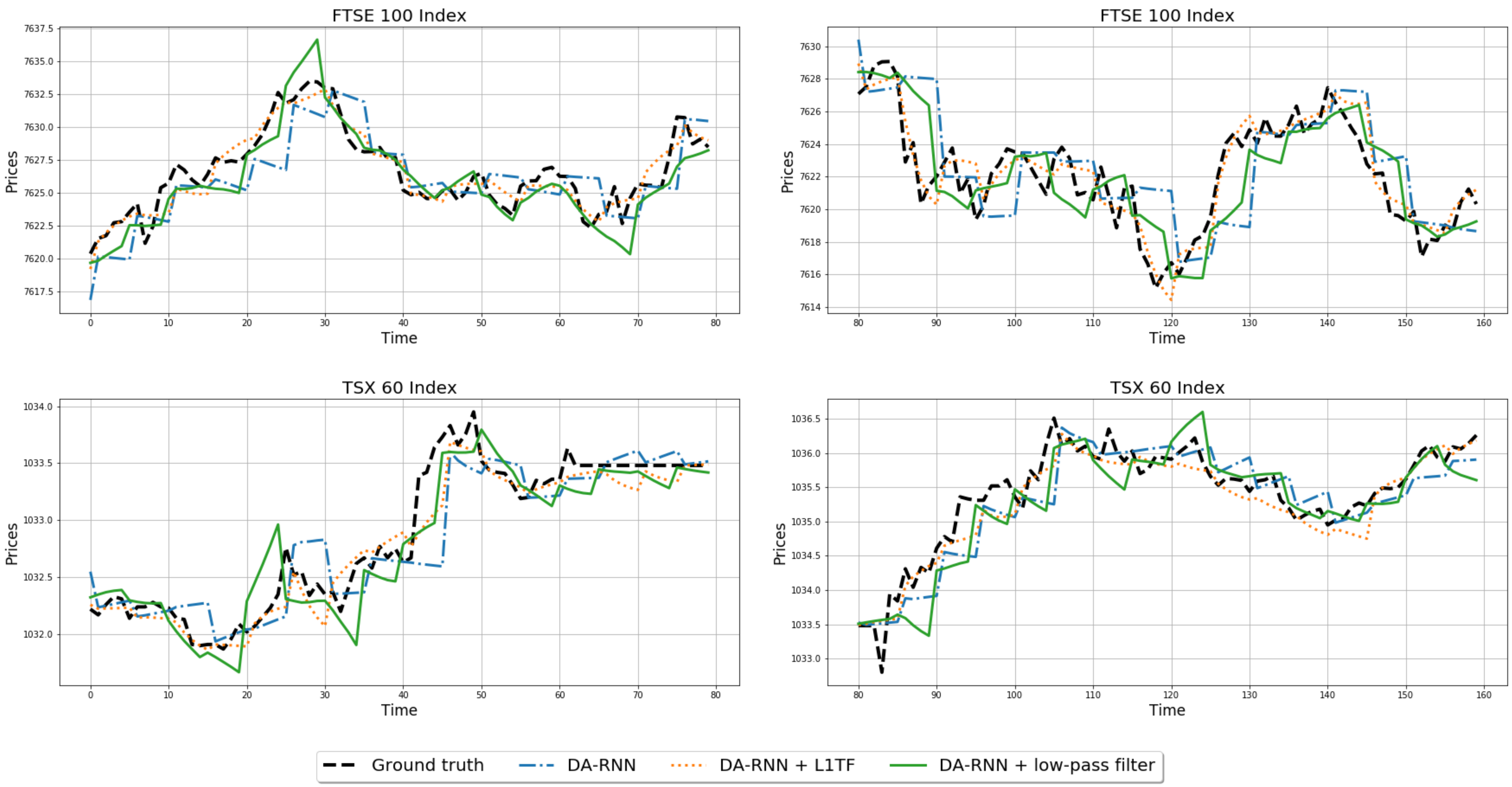
Prediction comparison between L1TF and Low-pass filter



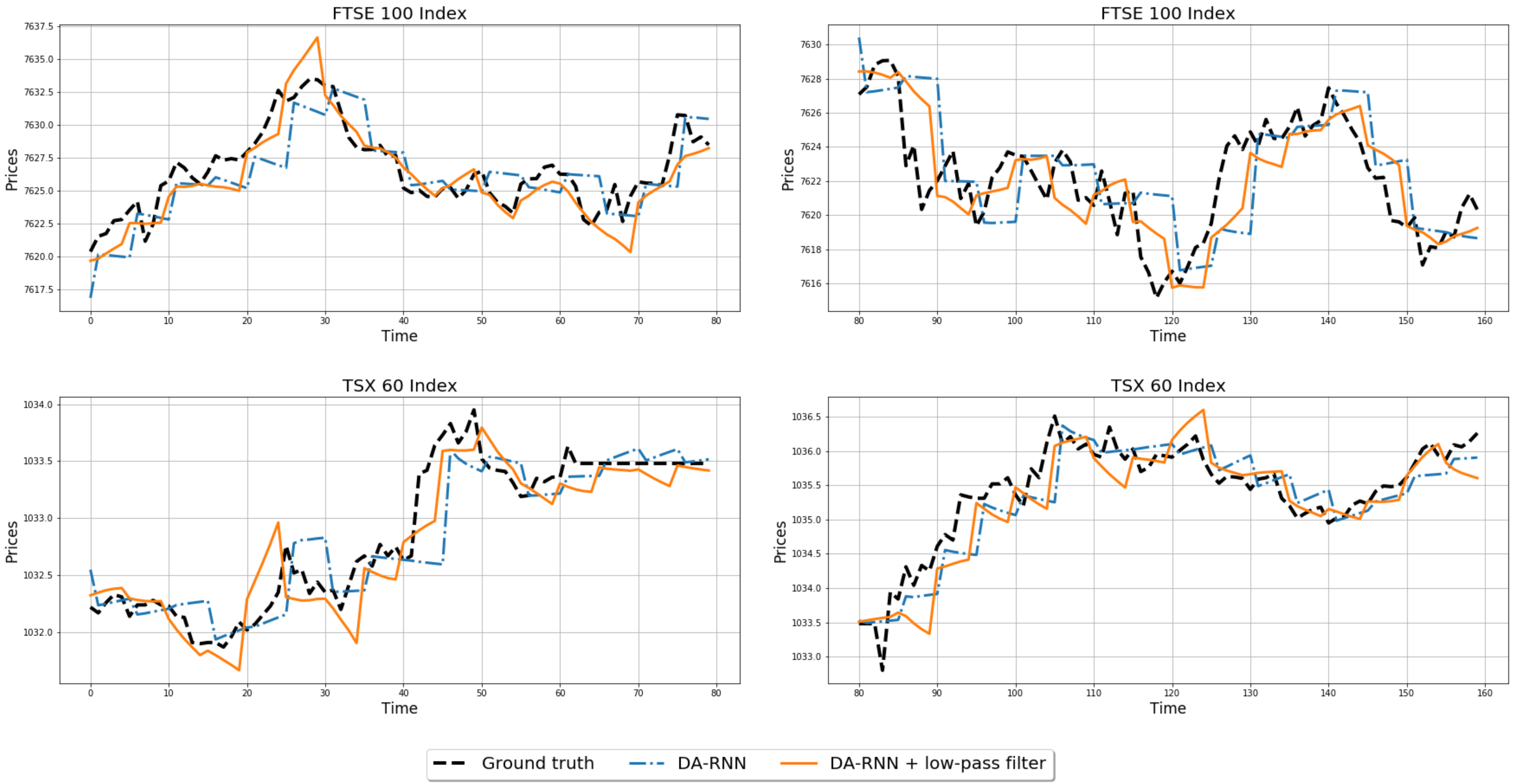
Prediction comparison between L1TF and Low-pass filter



Prediction comparison between L1TF and Low-pass filter



Prediction comparison between L1TF and Low-pass filter



Prediction comparison between L1TF and Low-pass filter

Model	NASDAQ 100 Index (Growth rate : 3.26%)			Dow Jones Industrial Average (Growth rate : 1.16%)			EURO STOXX 50 Index (Growth rate : -3.47%)			FTSE 100 Index (Growth rate : -4.26%)			TSX 60 Index (Growth rate : 0.16%)		
	RMSE	MAE	MAPE ($\times 10^4$)	RMSE	MAE	MAPE ($\times 10^4$)	RMSE	MAE	MAPE ($\times 10^4$)	RMSE	MAE	MAPE ($\times 10^4$)	RMSE	MAE	MAPE ($\times 10^4$)
DARNN	5.3079	2.6057	2.9514	12.6665	6.6695	2.3258	2.0232	1.1764	3.1469	4.244	2.3906	3.1788	0.4673	0.2256	2.1627
DARNN + L1TF	5.2053	**2.5146**	**2.8486**	**11.5129**	**5.9758**	**2.0844**	1.8392	**0.9368**	**2.5057**	**2.9276**	**1.3448**	**1.7874**	0.4031	**0.1451**	**1.3917**
DARNN + low-pass filter	**4.7811**	2.5989	2.9449	12.2035	6.5590	2.2874	**1.7443**	1.0740	2.8732	5.6033	2.3179	3.0831	**0.3366**	0.1946	1.8647

Thank you!

This work is was supported by IITP grant funded by the Korea government (MSIT)
No.2017-0-01779, Explainable Artificial Intelligence and
No.2019-0-00075, Artificial Intelligence Graduate School Program (KAIST))

Appendix A. Evaluation Metrics

Metric		Equation
RMSE	Root Mean Squared Error	$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_t^i - \hat{y}_t^i\right)^2}$
MAE	Mean Absolute Error	$MAE = \frac{1}{N}\sum_{i=1}^{N}\lvert y_t^i - \hat{y}_t^i\rvert$
MAPE	Mean Absolute Percentage Error	$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{\lvert y_t^i - \hat{y}_t^i\rvert}{\lvert y_t^i\rvert}$
Rate of Return (%)	the Rate of Return	$Rate\ of\ Return\ (\%) = \frac{B_{Last} - B_{Initial} + Stocks \times P_{close}}{B_{Initial}} \times 100,$ * $Stocks$: the number of stocks, $B_{Last}, B_{Initial}$: the last / initial balance