TRAFFIC FLOW FORECASTING ON HIGHWAYS EVALUATION AND COMPARISON OF COMMON APPROACHES

Hao Peng

Research Questions and Challenges

- Data collection
  - *Storage*
  - *Missing value imputation*
  - *What types of data to collect*
- Data analytics
  - *Historical data*
  - ***Predicative analytics***
  - *Real-time forecasting*
  - *Video/image analytics*

Benefits of Research

- Intelligent Transportation Systems
- Road expansion
- Traffic Apps developers
- Everyday commuters/travelers

Contributions

- To evaluate the effectiveness of commonly used statistical and machine learning models on univariate traffic flow forecasting using large amounts of temporal data
- To study the impacts of incorporating spatially dependent data into multivariate forecasting models
- To examine the performance of multi-step forecasts and the impacts of varying data resolutions
- To explore the various trade-offs/pros and cons of the models in terms of accuracy, stability, computational cost, and ease of use

Progress in Traffic Forecasting

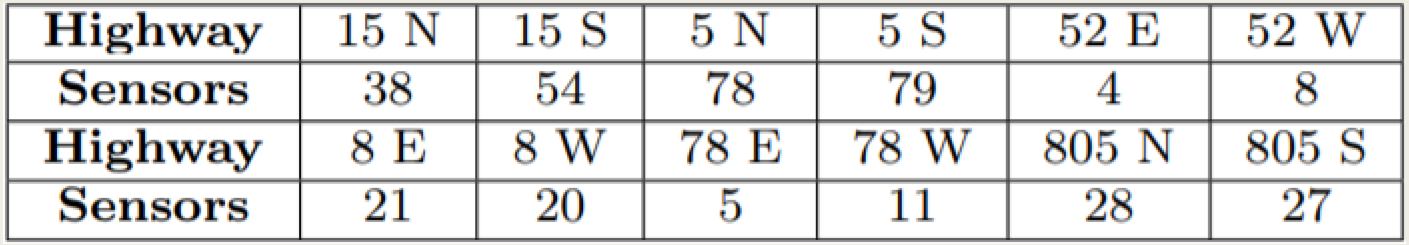
- Univariate -> Multivariate
- Single step ahead -> Multiple steps ahead
- Few sensors -> Many sensors
- Simple models/Few parameters -> Complex models/Deep learning

Big Data

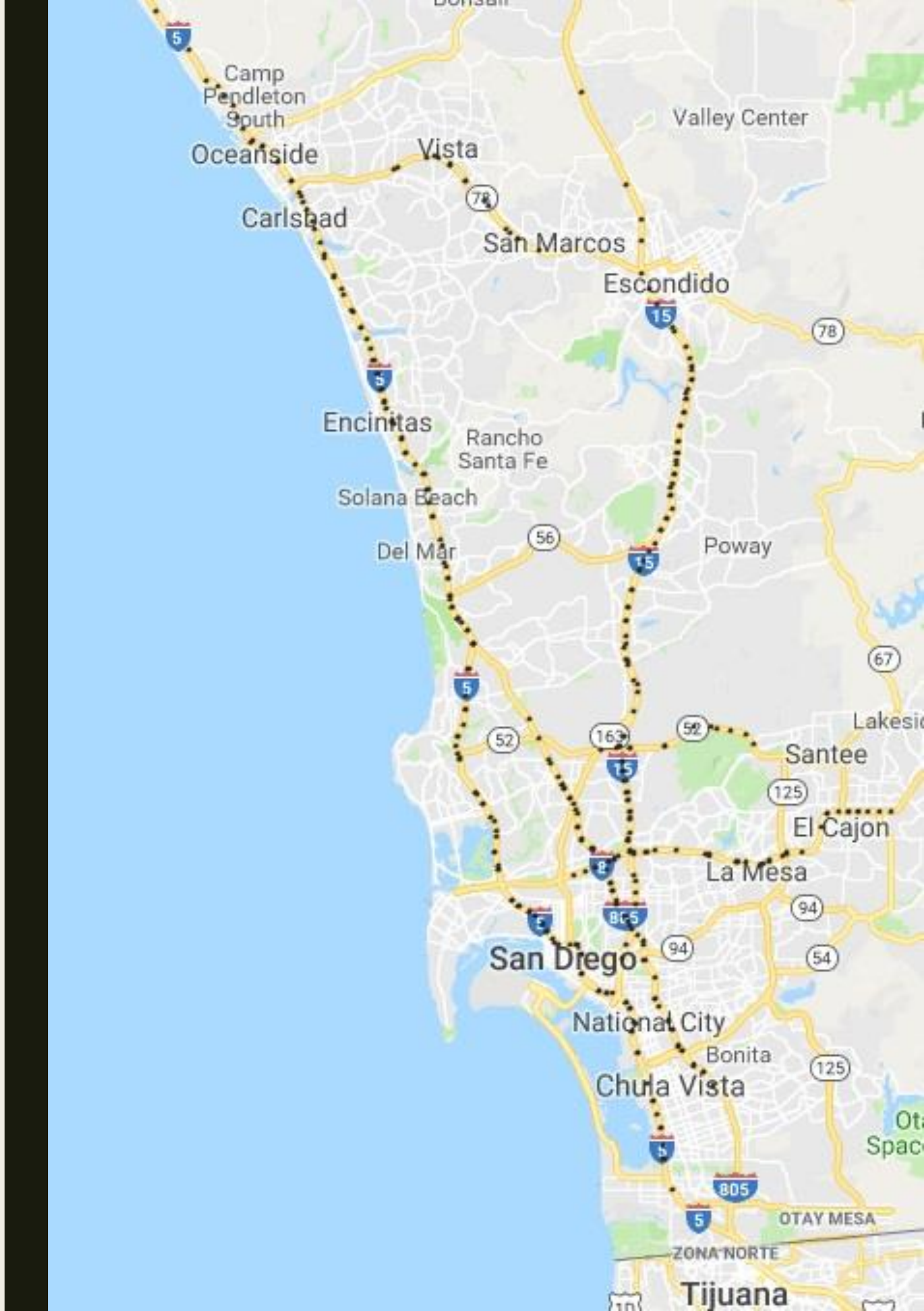
- Large amounts of high quality, high-resolution data
- Caltrans Performance Measurement System (PeMS)
  - *More than 39,000 sensors deployed in major urban areas and highways*
  - *5-minute resolution traffic flow, speed, occupancy of lanes, etc., available in real-time*

Our Focus

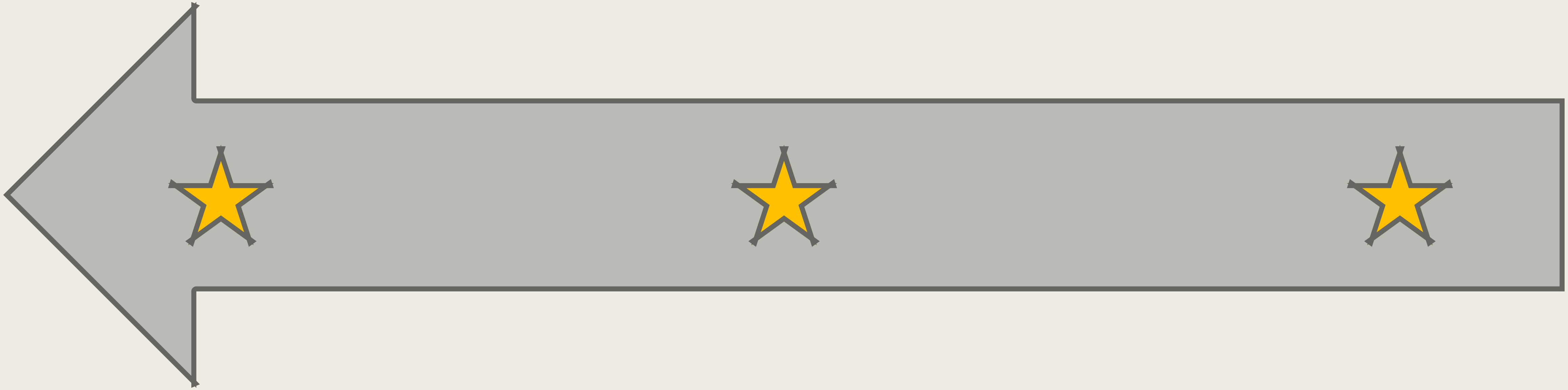
- Southern California, San Diego area
- 373 sensors, ~1.5 Gigabytes.
- Two data resolutions
  - *5-min (original)*
  - *15-min (aggregated)*
- Forecasting horizon
  - *12 steps*
  - *Equivalent of up to 1 hour and 3 hours ahead, for 5-min and 15-min data, respectively*



Highway	15 N	15 S	5 N	5 S	52 E	52 W
Sensors	38	54	78	79	4	8
Highway	8 E	8 W	78 E	78 W	805 N	805 S
Sensors	21	20	5	11	28	27



Spatial Dependency



Downstream Sensor

Provides information on possible congestion down the road

Sensor of Focus

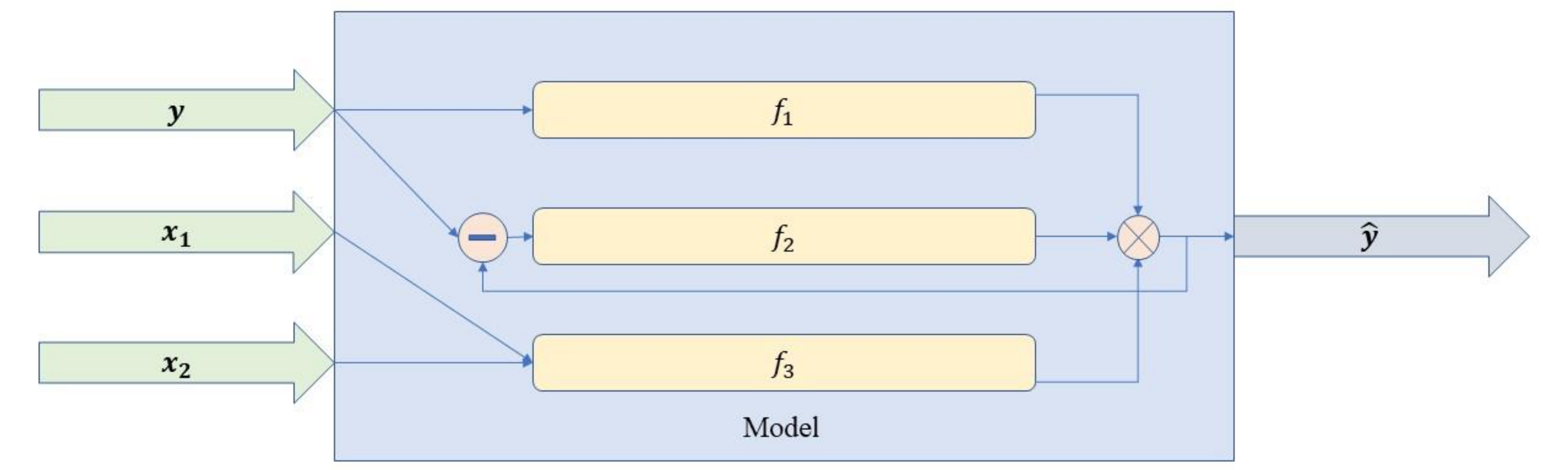
Upstream Sensor

Traffic flowing from this location, may include any highway exits

Model

$$y_t = f_1(\mathbf{y}_{tl_1}; \boldsymbol{\alpha}) + f_2(\boldsymbol{\epsilon}_{tl_2}; \boldsymbol{\theta}) + f_3(X_{tl_3}; \boldsymbol{\beta}) + \epsilon_t$$

$$z_t = \nabla^d y_t$$



Our models

- Seasonal Autoregressive Integrated Moving Average
- Seasonal Vector Autoregressive Integrated Moving Average
- Regression
- QuadRegression
- QuadXRegression
- CubicRegressionNI
- Support Vector Regression
- Extreme Learning Machines
- Feedforward Neural Networks (keras + tensorflow)
- Long Short-Term Memory Neural Networks (keras + tensorflow)

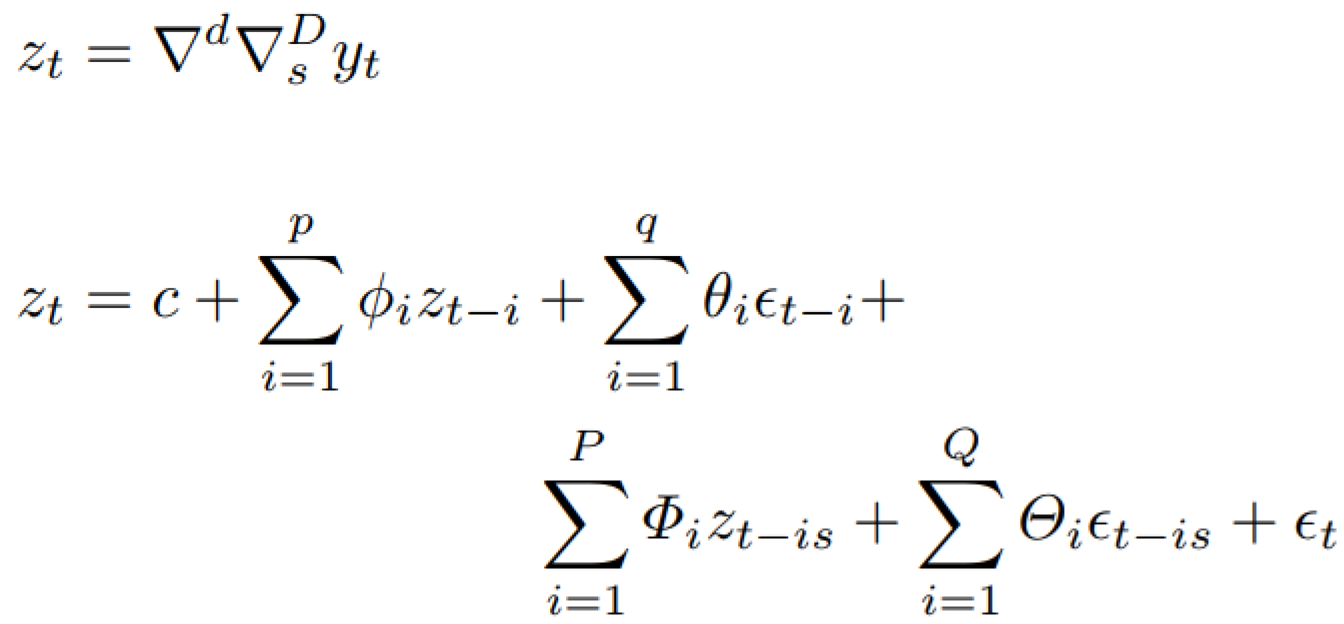
Input and Output

- Time Series models (e.g., SARIMA)
  - *Directly use the time series of interests as input*
- Other models
  - *Organize data from the time series into training instances x features*
  - *Input matrix V and output matrix W containing row vectors:*

$$\mathbf{v}_t = [\bar{y}_t, \ldots, \bar{y}_{t+h-1}, y_{t-s}, \ldots, y_{t+h-1-s},\\ y_{t-h}, \ldots, y_{t-1}, hr_t]$$

$$\mathbf{w}_t = [y_t, \ldots, y_{t+h-1}]$$

SARIMA

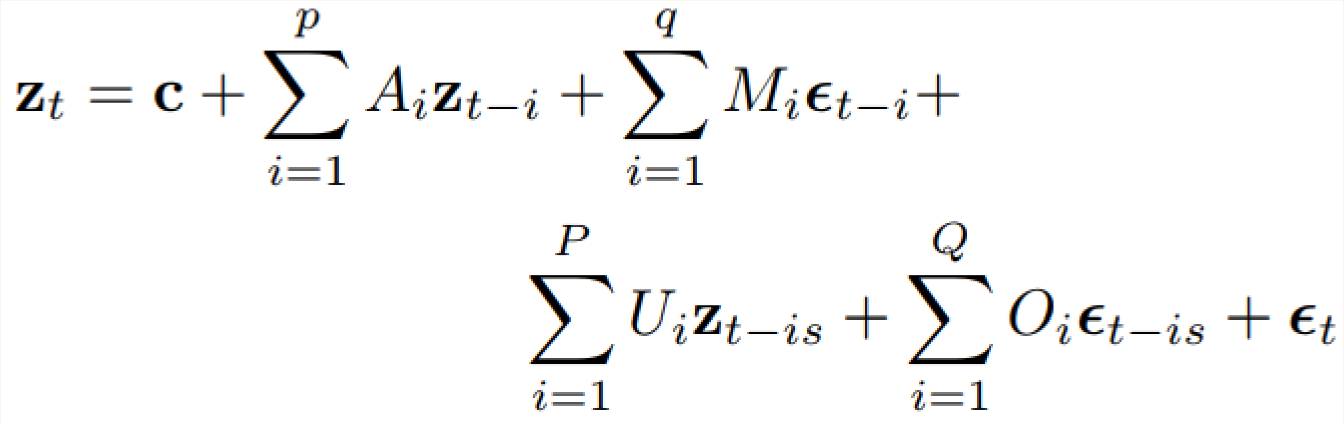


$$z_t = \nabla^d \nabla_s^D y_t$$

$$z_t = c + \sum_{i=1}^{p} \phi_i z_{t-i} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \sum_{i=1}^{P} \Phi_i z_{t-is} + \sum_{i=1}^{Q} \Theta_i \epsilon_{t-is} + \epsilon_t$$

- Classical time series model that uses lagged and correlated values of the time series and errors to make forecasts
- Differencing may be necessary to make the time series stationary

Seasonal VARIMA



$$\mathbf{z}_t = \mathbf{c} + \sum_{i=1}^{p} A_i \mathbf{z}_{t-i} + \sum_{i=1}^{q} M_i \boldsymbol{\epsilon}_{t-i} + \sum_{i=1}^{P} U_i \mathbf{z}_{t-is} + \sum_{i=1}^{Q} O_i \boldsymbol{\epsilon}_{t-is} + \boldsymbol{\epsilon}_t$$

- Multivariate generalization of the SARIMA model

Regression Family

$$y_t = \mathbf{b} \cdot [1, \mathbf{v}_t] + \epsilon_t$$

- Global minimization of SSE/MSE
- Polynomial regression introduces non-linearity into the model by including additional input features such as the powers of the original input features
- Interaction/cross terms, which are products of pairs of features, may also be included
- Recent interest in polynomial regression as viable alternatives to feedforward neural networks [Cheng et al., 2018].

Support Vector Regression

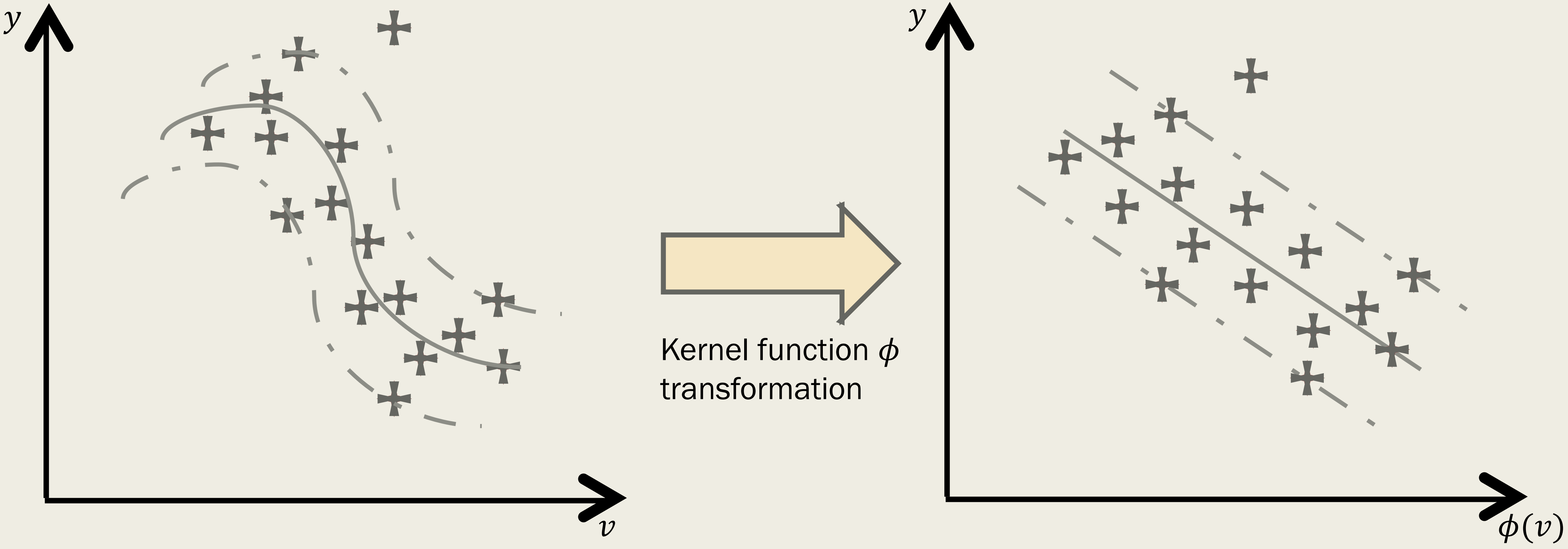
$$y_t = c + \mathbf{b} \cdot \mathbf{v}_t + \epsilon_t$$

minimize $\frac{1}{2}||\mathbf{b}||^2$

subject to $|y_t - (c + \mathbf{b} \cdot \mathbf{v}_t)| \leq \rho$

- A kernel function may be used to map the training instances into higher dimensional spaces to find the hyperplane
- Slack variables may be added to the objective function and constraint to allow some points to lie outside the boundary

Support Vector Regression



Kernel function ϕ transformation

Feedforward Neural Networks

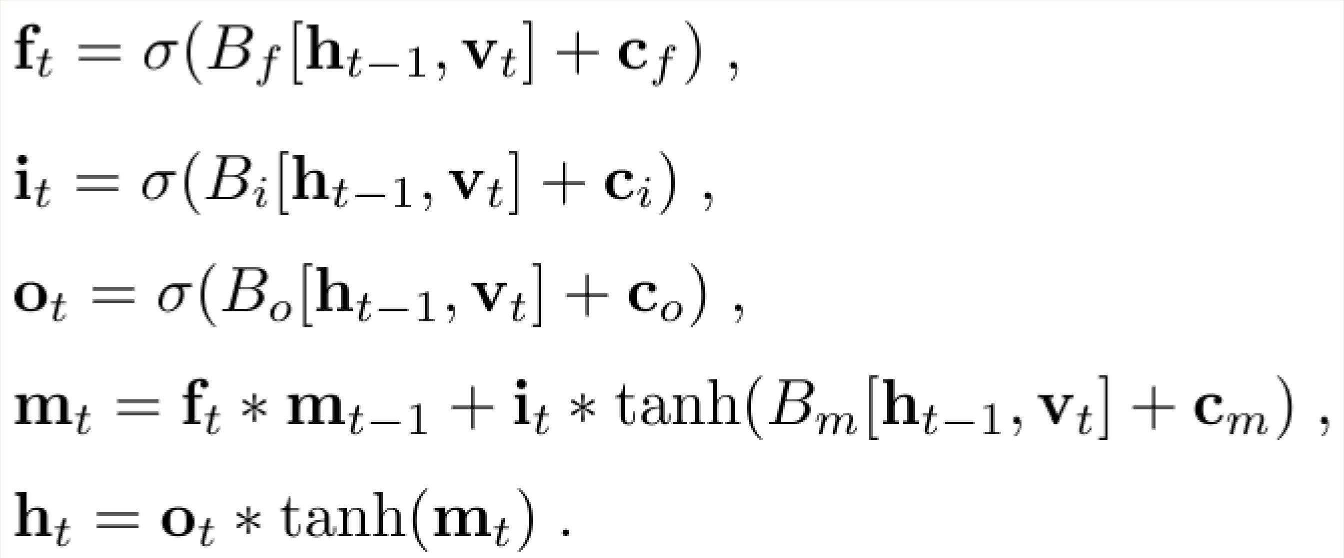
$$\mathbf{y}_t = f_o^a(B_o^T f_h^a(B_h^T \mathbf{v}_t + \mathbf{c}_h) + \mathbf{c}_o) + \boldsymbol{\epsilon}_t$$

- Vanilla, 3-layer, feedforward neural network
- Foundation for other deep neural network structures

Extreme Learning Machine

- A special type of feedforward neural network, as follows
  - *Input-to-Hidden: apply activation function to inputs with fixed, random weights. This introduces non-linearity into the data.*
  - *Hidden-to-Output: use linear regression to learn the outputs from the hidden features.*
- Compared with a standard feedforward neural network, ELM requires minimal parameter tuning and much lower computational costs.

Long Short-Term Memory Neural Networks



$$\mathbf{f}_t = \sigma(B_f[\mathbf{h}_{t-1}, \mathbf{v}_t] + \mathbf{c}_f)\ ,$$

$$\mathbf{i}_t = \sigma(B_i[\mathbf{h}_{t-1}, \mathbf{v}_t] + \mathbf{c}_i)\ ,$$

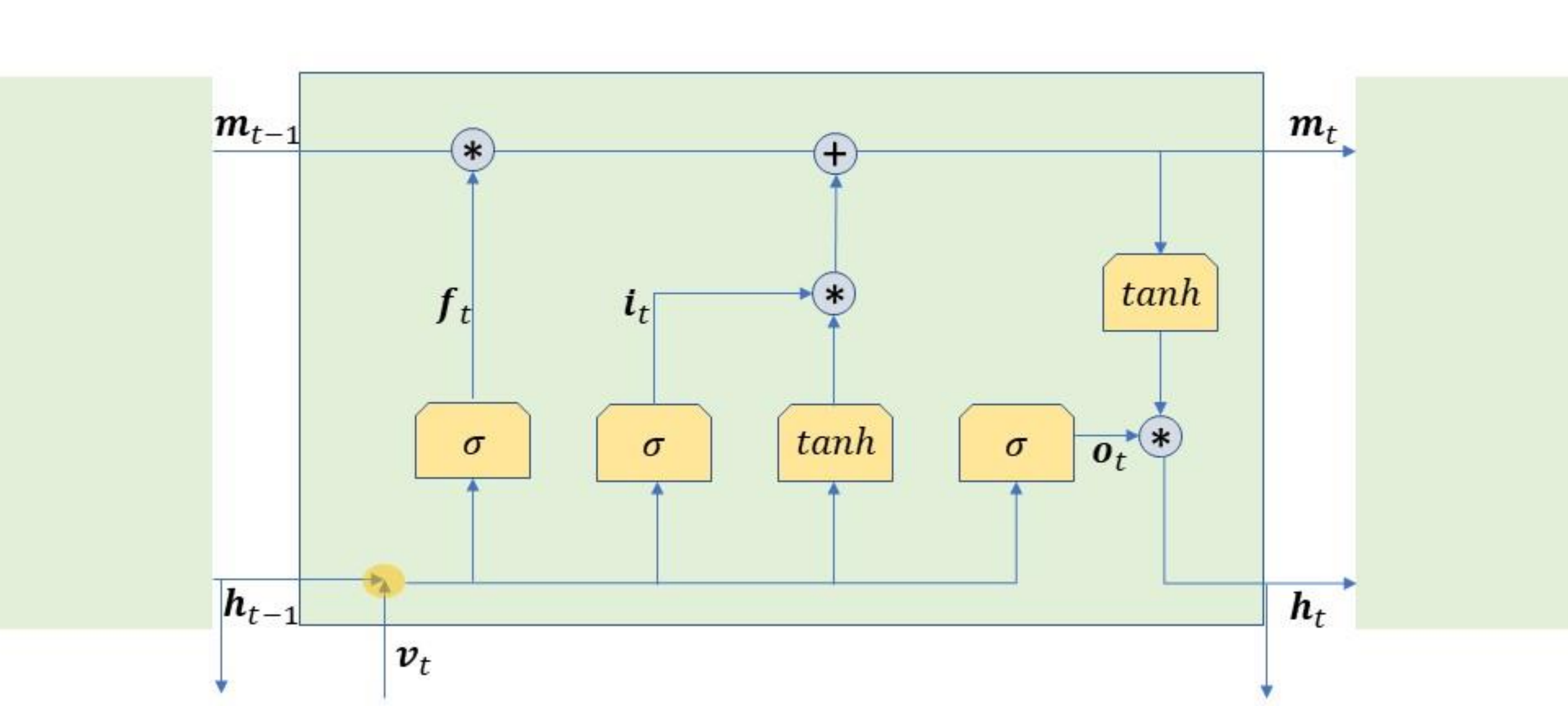
$$\mathbf{o}_t = \sigma(B_o[\mathbf{h}_{t-1}, \mathbf{v}_t] + \mathbf{c}_o)\ ,$$

$$\mathbf{m}_t = \mathbf{f}_t * \mathbf{m}_{t-1} + \mathbf{i}_t * \tanh(B_m[\mathbf{h}_{t-1}, \mathbf{v}_t] + \mathbf{c}_m)\ ,$$

$$\mathbf{h}_t = \mathbf{o}_t * \tanh(\mathbf{m}_t)\ .$$

- Cell state memory is able to retain useful information from many time steps ago
- Regained popularity for achievements in speech recognition, language translation, etc.

Long Short-Term Memory Neural Network



Other Deep Learning Models

- Stacked Autoencoders
  - *Dimensionality reduction when give large amounts of traffic data*
  - *Filtering out noise in input*
- Convolutional Neural Network
  - *Models spatial dependencies using shared weights in a specific region*
  - *May be used to model spatially dependent traffic sensors*
- Gated Recurrent Unit
  - *A variant of LSTM*
  - *Combined forget and input gates*
  - *Merged cell state and hidden state*

Related Work using PeMS datasets

- [Lv et al., 2015]: forecast up to one hour ahead, 15-min resolution, first 3 months of 2013, SAE deep neural networks with 2-4 hidden layers, 200-500 neurons per hidden layer, 6%-7% MAPE
  - *Compared to our MAPE values, we have slightly lower MAPE values for the first 2 steps and slightly higher MAPE values for the next 2 steps.*
- [Shao and Soong, 2016]: small scaled study using a single sensor in Irvine, CA, Encoder-Decoder LSTM NN, one-step ahead, 5.4% MAPE
  - *Difficult to conclude anything in comparing with our work*
- [Wu et al., 2018]: 33 sensors on I-405, deep neural networks built with convolutional neural network and gated recurrent unit, training data available from 4/1/2014 to 6/20/2015, tested on the remaining 10 days in June 2015, concurrent forecasts for all sensors, 9-steps (45 minutes) ahead, MAPE from 7% to 9%.
  - *The MAPE values of our best models are 6% to 8%.*
- [Yang et al., 2019]: 50 sensors from San Diego/Imperial area, LSTM trained on March-April 2017 data, tested on May's, one step-ahead MAPE in 15-min resolution was 6.54%
  - *Our best MAPE was 5.5%.*

Evaluation Metrics and Platform

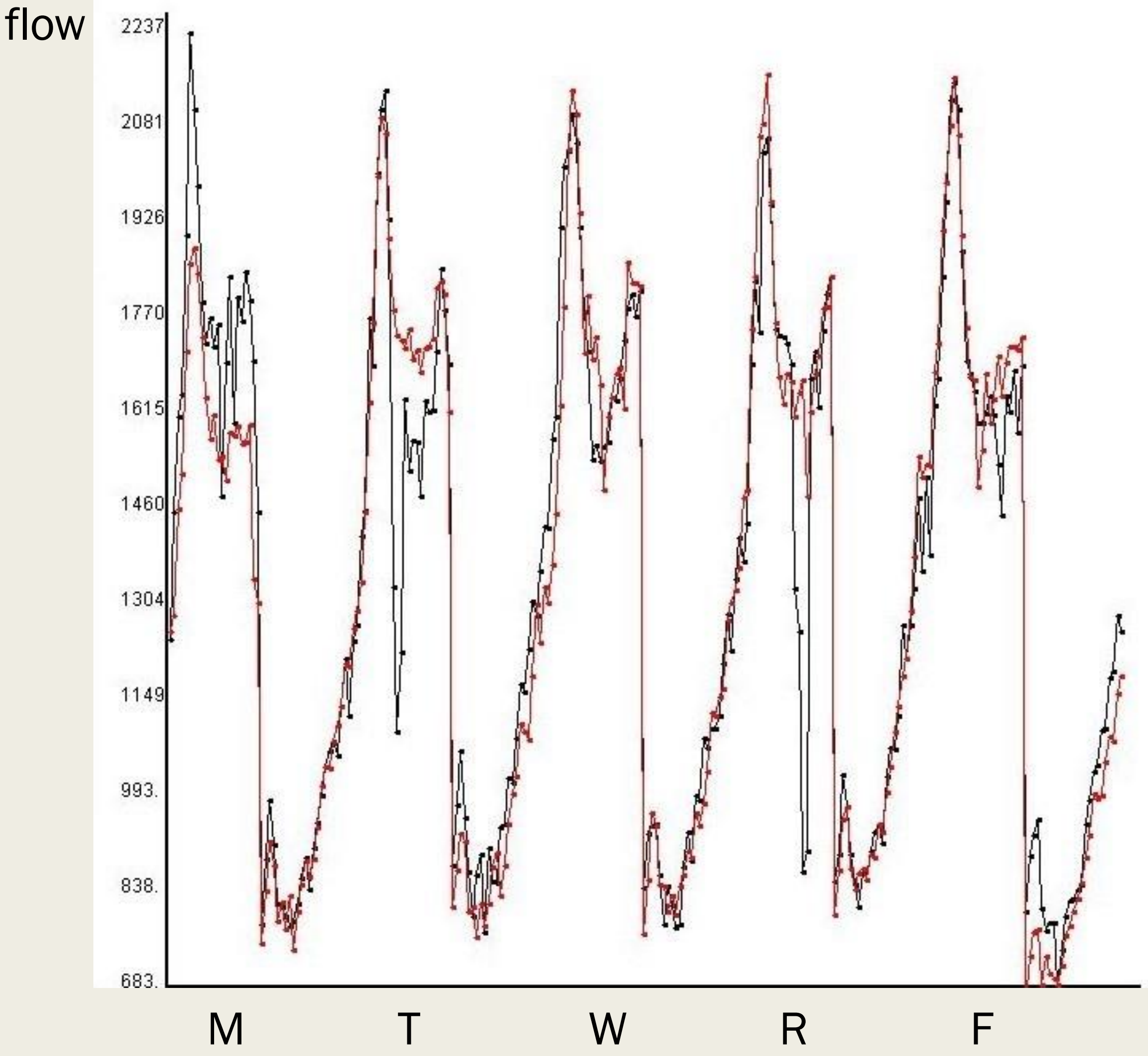
- Metrics
  - *MAPE*
  - *NRMSE*
  - R^2
- Platform
  - *Sapelo2 cluster of Georgia Advanced Computing Resource Center*
  - *32-core Intel Xeon Skylake processor and 187GB of RAM*

Problem Analysis and Modeling

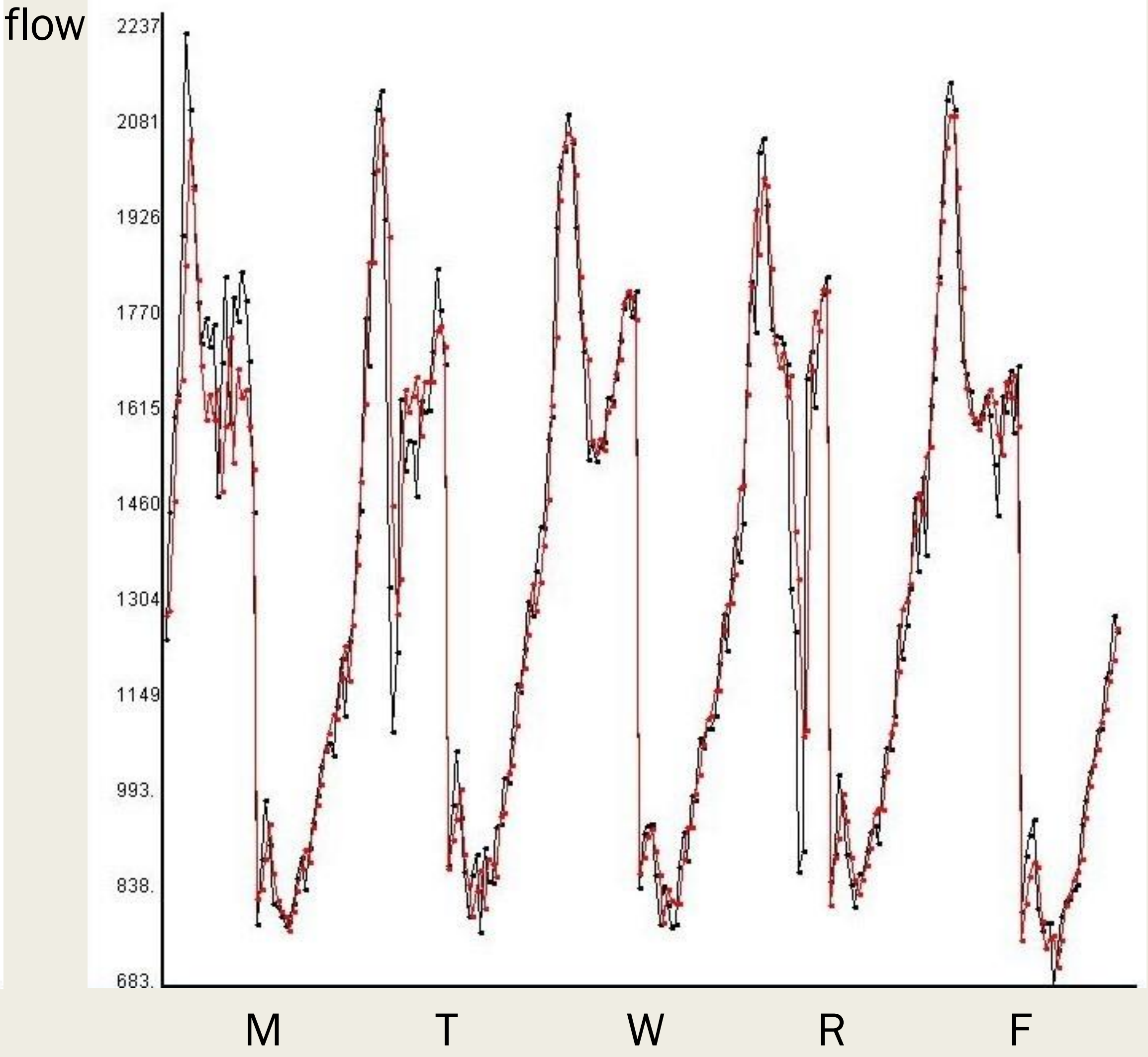
- Training data: Jan – Aug 2018
- Testing data: Sep – Dec 2018
- Only weekdays are considered
- Evaluation includes 7:00 AM to 7:00 PM daytime traffic only
- Weekly historical average, for 4 weeks, as baseline

Baseline

Actual (Black) vs. Baseline (Red)



Actual (Black) vs. Regression (Red)



Univariate Experiments

- SARIMA (1, 0, 1)×(0, 1, 1), a simple traffic flow forecasting model found in [Williams and Hoel, 2003], [Shekhar and Williams, 2008], [Lippi et al., 2013], [Peng et al., 2018].
- SARIMA2, BIC based search
  - *SARIMA (12, 1, 4)×(1, 1, 1)$_{1440}$, 5-min resolution data*
  - *SARIMA (5, 0, 3)×(1, 1, 1)$_{480}$, 15-min resolution data*
- Regression family
  - *Parameters are solved using QR factorization*
- SVR, based on LIBSVM implementation
  - *ν-SVR works better than ϵ-SVR*
  - *RBF kernel*

Univariate Experiments

- Extreme Learning Machine
  - *tanh activation function*
  - *Hidden layer size tuned to be 8 times the size of input layer*
- Feedforward NN
  - *4 layer structure, can extract features better than a shallow NN [Schmidhuber, 2015]*
  - *Two leaky ReLU activation functions and the identity/linear activation function in the output layer*

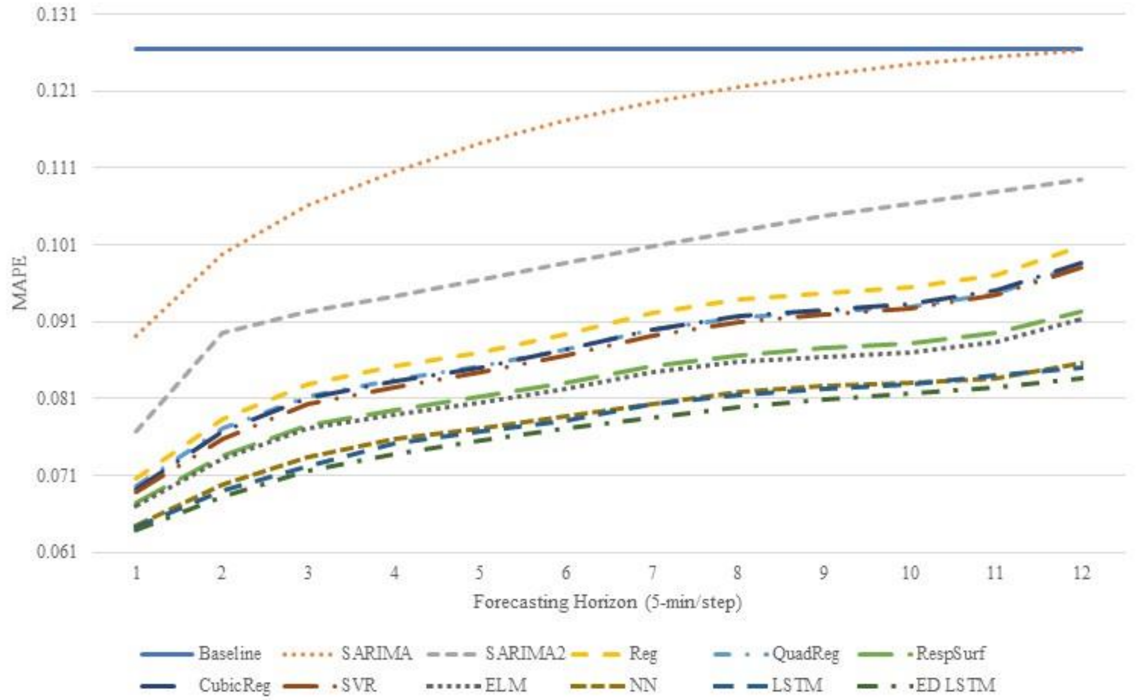
Univariate Experiments

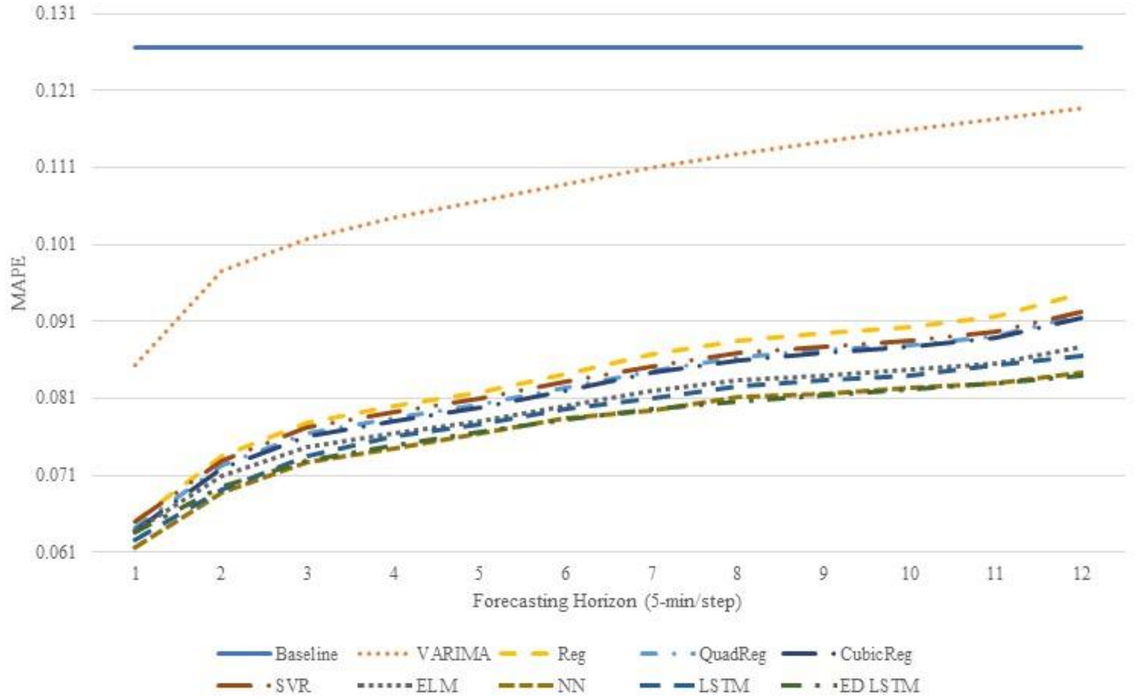
- LSTM
  - *Similar to a vanilla LSTM, but with one additional leaky ReLU layer before the output layer*
  - *Requires 3-dimensional input, instances x time steps x features*
  - *To preserve the features in V, the temporal evolution of the feature values for 4 weeks are used to generate the additional time steps dimension.*
- Encoder-Decoder LSTM
  - *The feature dimension is redefined to represent a time series*
  - *Temporal lagged values are moved to the time steps dimension*
  - *12 most recent obs + 12 obs from last week => 24 temporal layers*

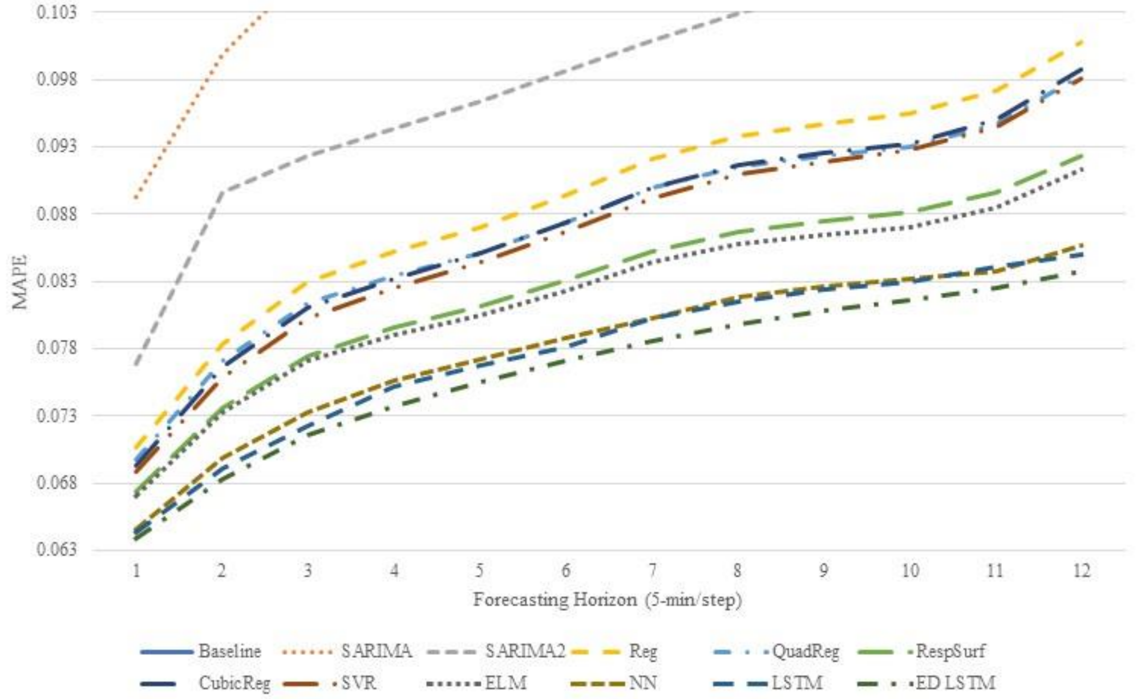
Multivariate Experiments

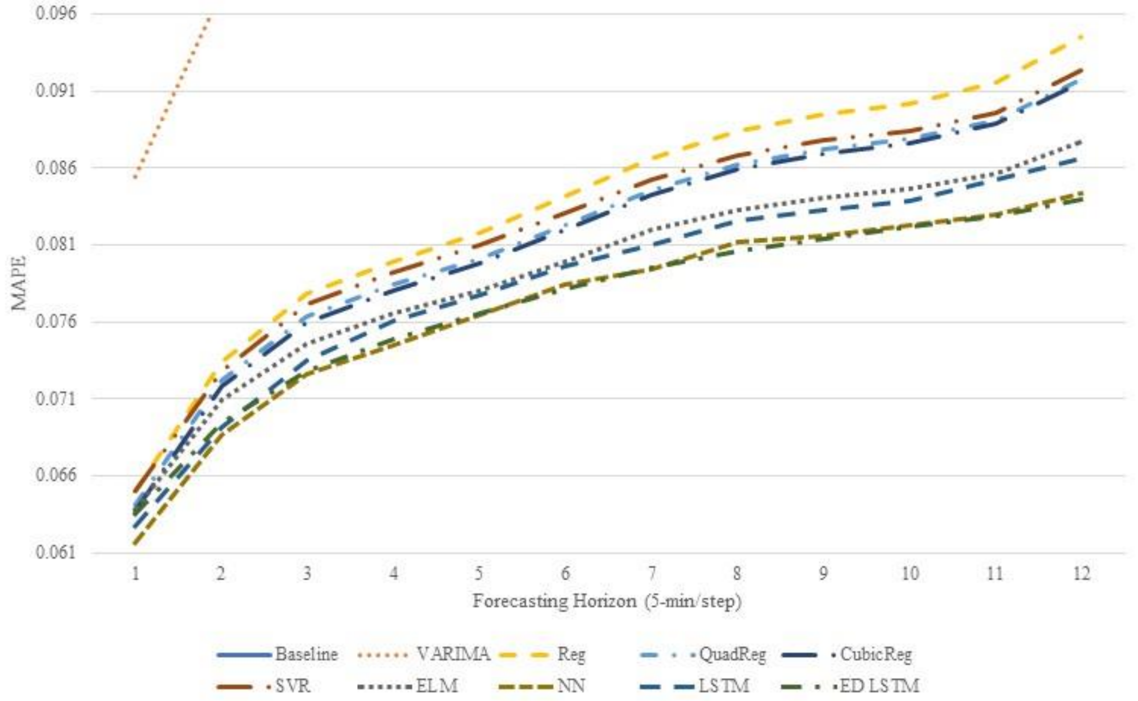
- Expand input features with data from one upstream and one downstream sensor, each at least 5 miles apart
- VARIMA2 (high order) omitted due to high computational cost and relatively low performance of its univariate counterpart comparing with most other models
- Response Surface omitted due to excessive interaction/cross terms that significantly increased computational cost and decreased accuracy
- Feature selection to extract 72 (one-third) most useful features based on the adjusted R^2 metric training linear regression models to forecast step 12 (the most difficult to forecast)

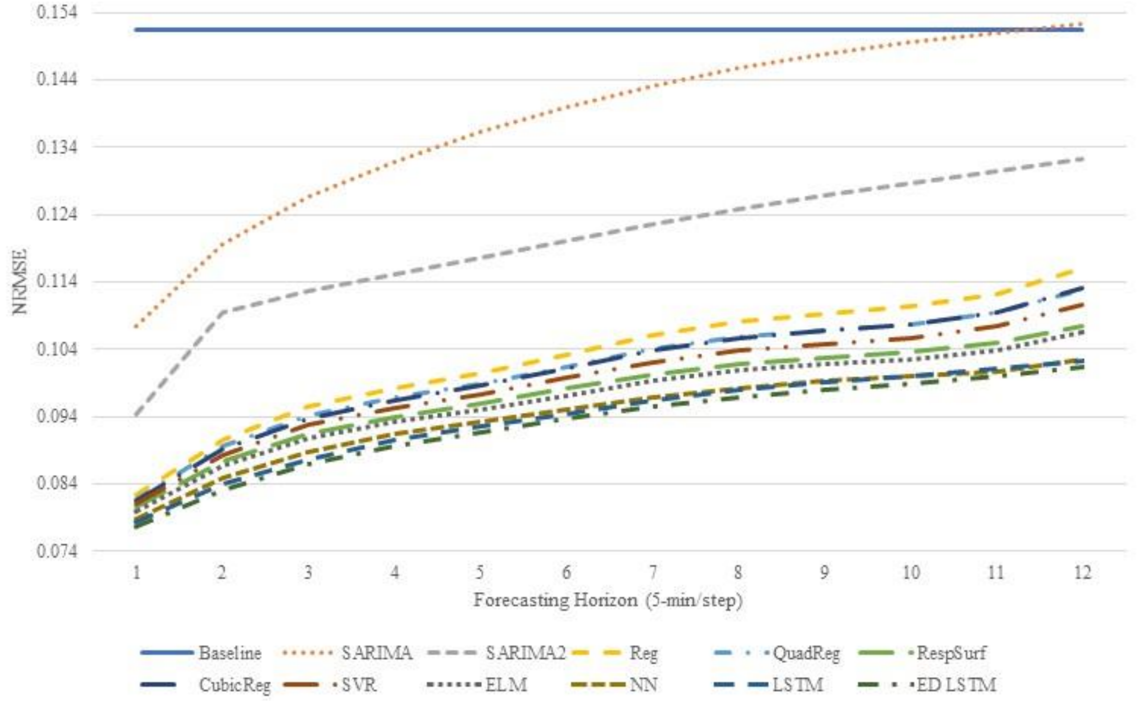
5-min resolution data
univariate (left) vs. multivariate (right)
(bottom image zooms in on the top performers)

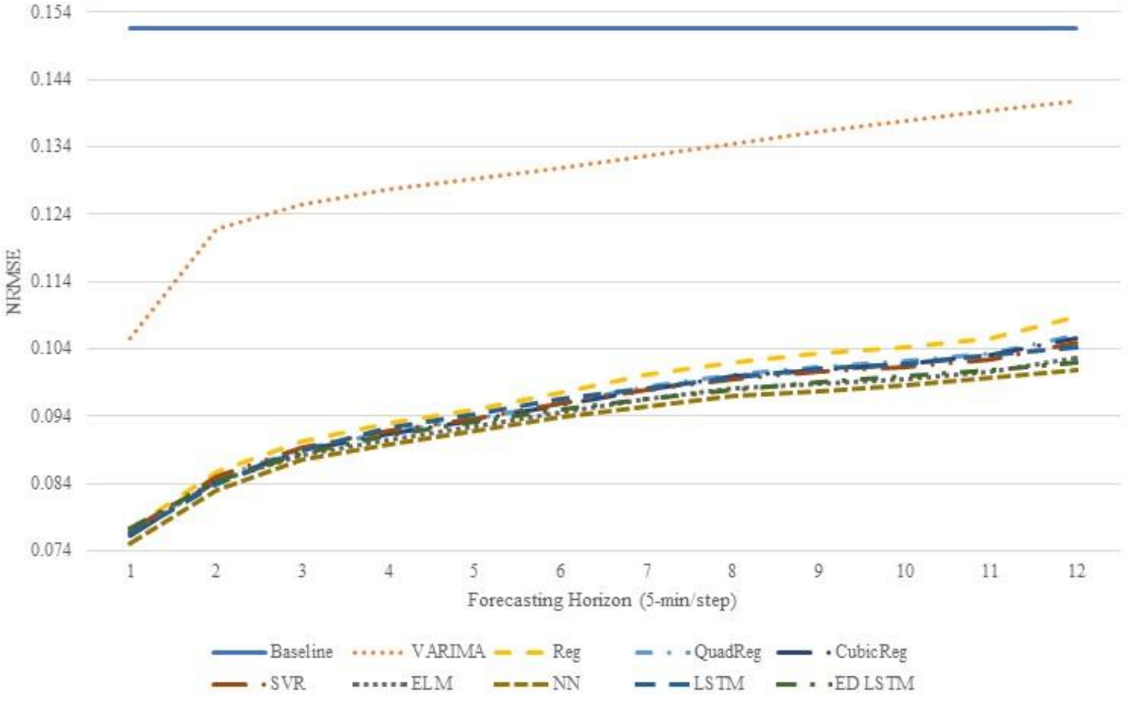

MAPE
0.131
0.121
0.111
0.101
0.091
0.081
0.071
0.061
1
2
3
4
5
6
7
8
9
10
11
12
Forecasting Horizon (5-min/step)
Baseline
SARIMA
SARIMA2
Reg
QuadReg
RespSurf
CubicReg
SVR
ELM
NN
LSTM
ED LSTM

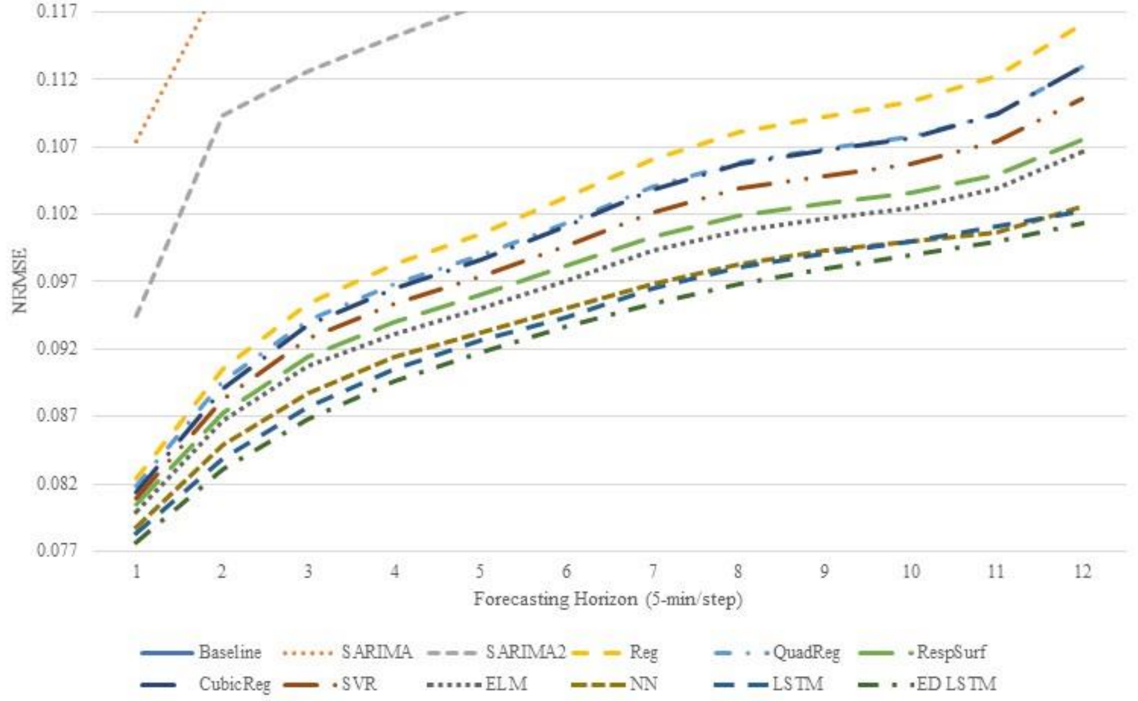

MAPE
0.131
0.121
0.111
0.101
0.091
0.081
0.071
0.061
1
2
3
4
5
6
7
8
9
10
11
12
Forecasting Horizon (5-min/step)
Baseline
VARIMA
Reg
QuadReg
CubicReg
SVR
ELM
NN
LSTM
ED LSTM

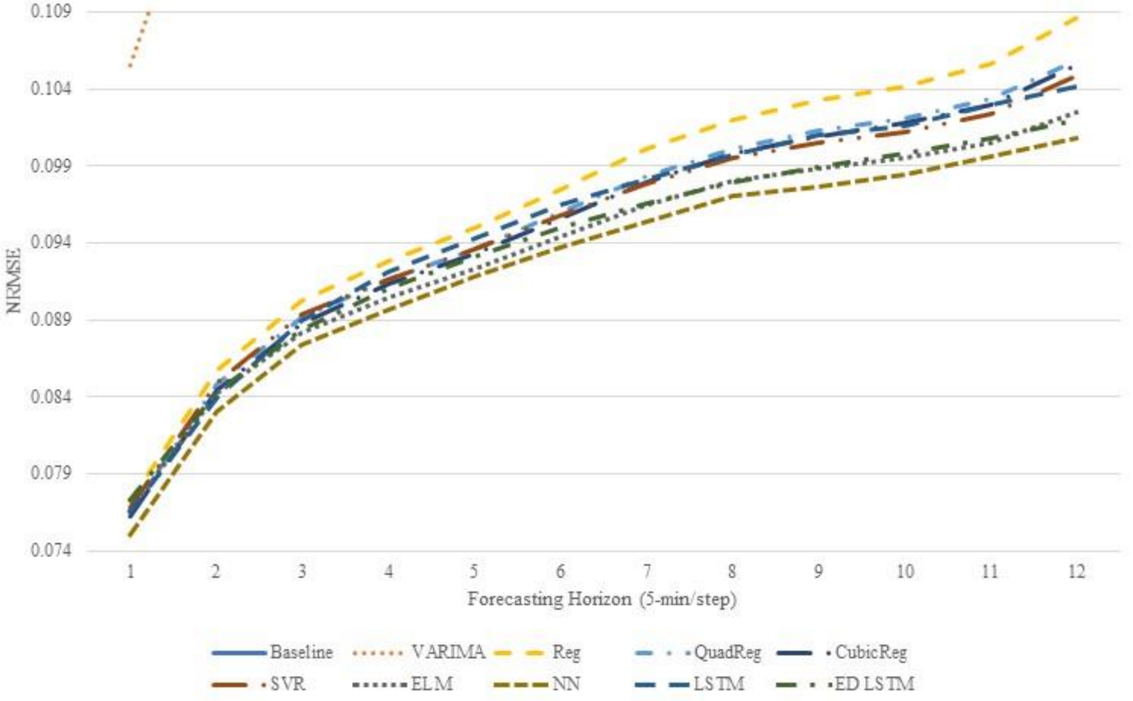

MAPE
0.103
0.098
0.093
0.088
0.083
0.078
0.073
0.068
0.063
1
2
3
4
5
6
7
8
9
10
11
12
Forecasting Horizon (5-min/step)
Baseline
SARIMA
SARIMA2
Reg
QuadReg
RespSurf
CubicReg
SVR
ELM
NN
LSTM
ED LSTM

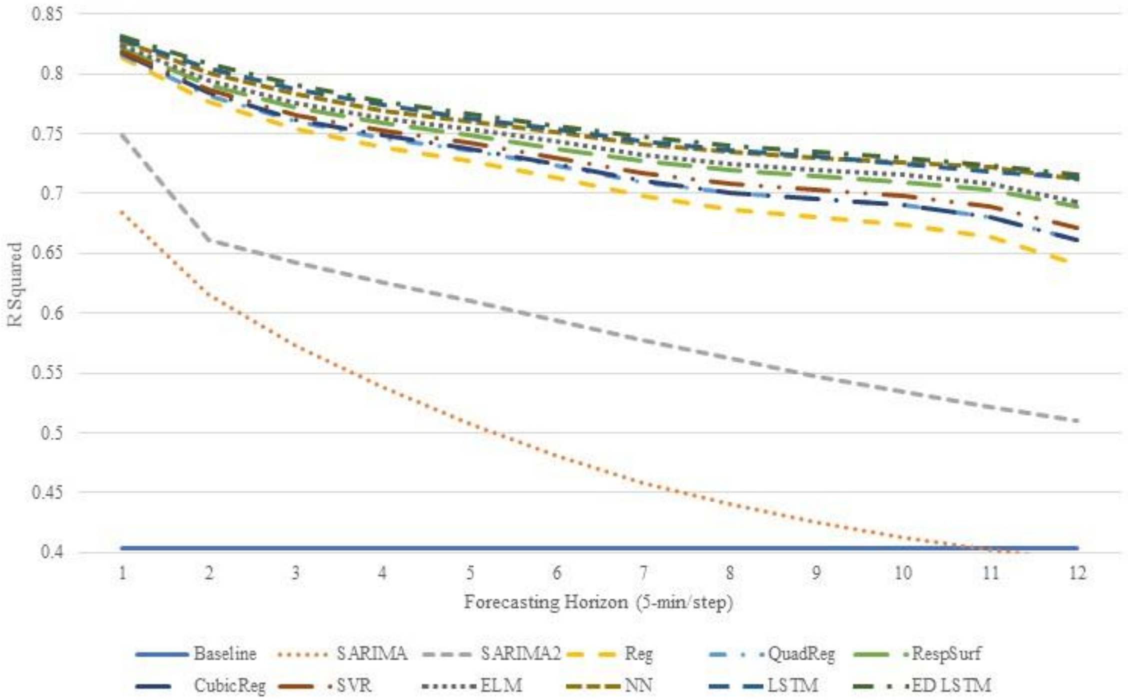

MAPE
0.096
0.091
0.086
0.081
0.076
0.071
0.066
0.061
1
2
3
4
5
6
7
8
9
10
11
12
Forecasting Horizon (5-min/step)
Baseline
VARIMA
Reg
QuadReg
CubicReg
SVR
ELM
NN
LSTM
ED LSTM

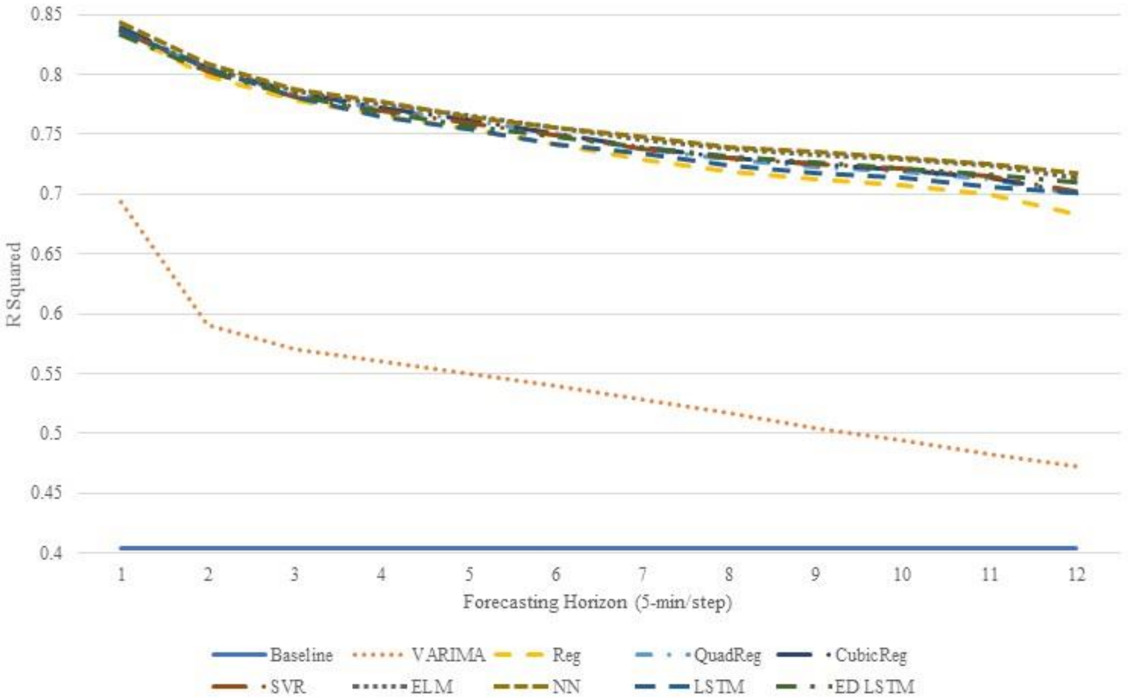

0.154
0.144
0.134
0.124
0.114
0.104
0.094
0.084
0.074
NRMSE
1
2
3
4
5
6
7
8
9
10
11
12
Forecasting Horizon (5-min/step)
Baseline
SARIMA
SARIMA2
Reg
QuadReg
RespSurf
CubicReg
SVR
ELM
NN
LSTM
ED LSTM

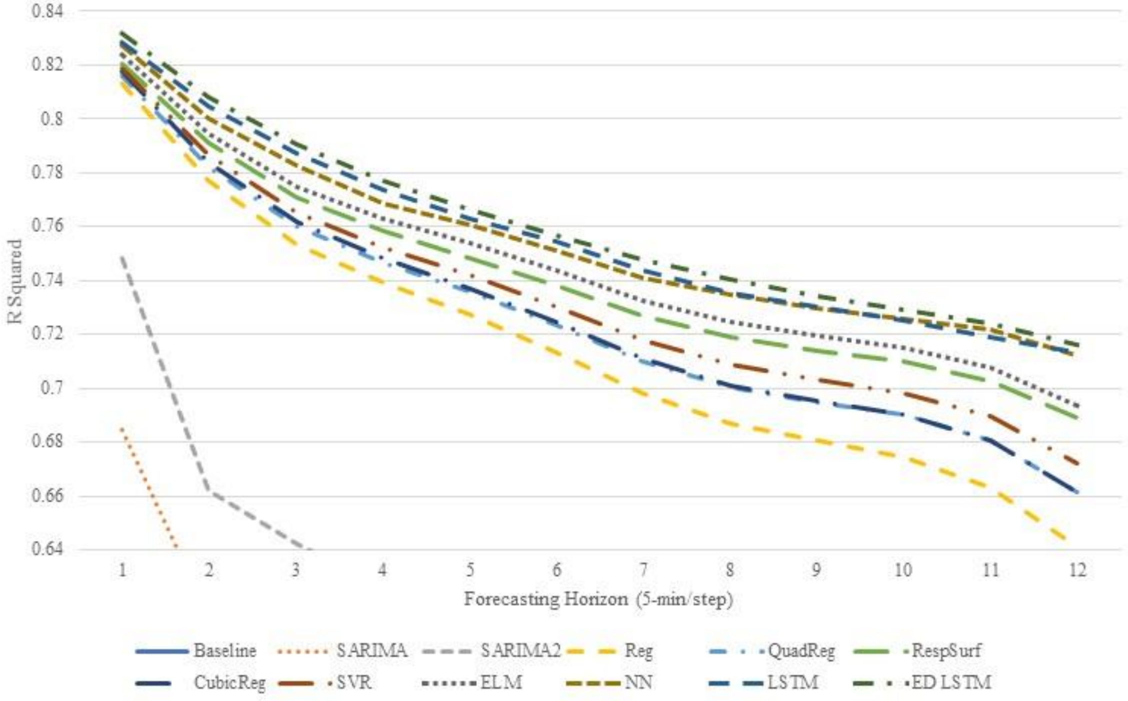

0.154
0.144
0.134
0.124
0.114
0.104
0.094
0.084
0.074
NRMSE
1
2
3
4
5
6
7
8
9
10
11
12
Forecasting Horizon (5-min/step)
Baseline
VARIMA
Reg
QuadReg
CubicReg
SVR
ELM
NN
LSTM
ED LSTM

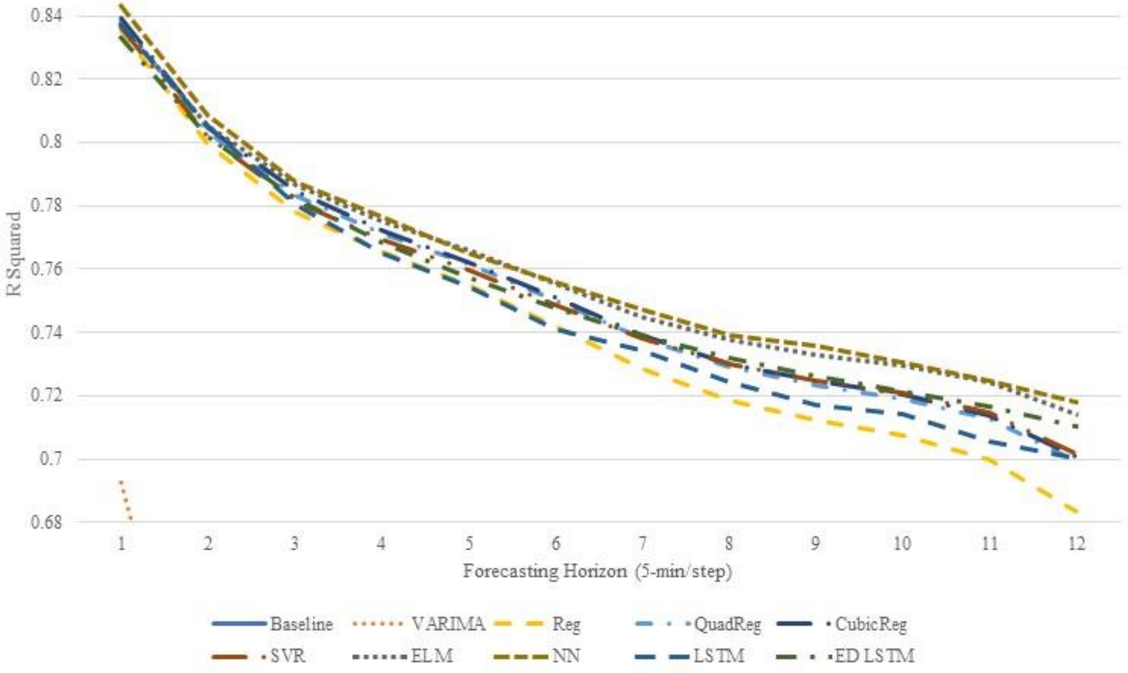

0.117
0.112
0.107
0.102
0.097
0.092
0.087
0.082
0.077
NRMSE
1
2
3
4
5
6
7
8
9
10
11
12
Forecasting Horizon (5-min/step)
Baseline
SARIMA
SARIMA2
Reg
QuadReg
RespSurf
CubicReg
SVR
ELM
NN
LSTM
ED LSTM


0.109
0.104
0.099
0.094
0.089
0.084
0.079
0.074
NRMSE
1
2
3
4
5
6
7
8
9
10
11
12
Forecasting Horizon (5-min/step)
Baseline
VARIMA
Reg
QuadReg
CubicReg
SVR
ELM
NN
LSTM
ED LSTM

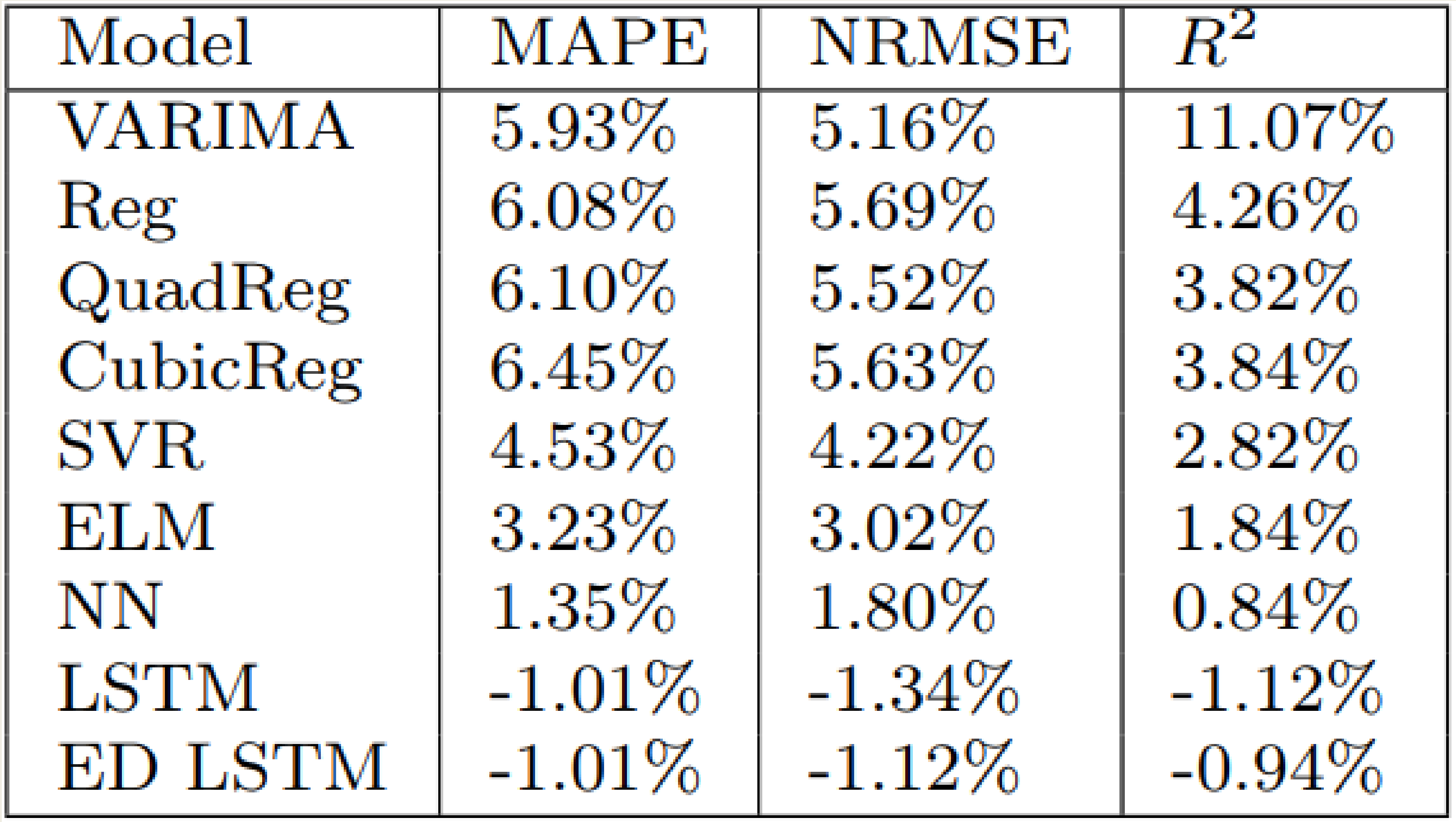

R Squared
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0.45
0.4
1 2 3 4 5 6 7 8 9 10 11 12
Forecasting Horizon (5-min/step)
Baseline
SARIMA
SARIMA2
Reg
QuadReg
RespSurf
CubicReg
SVR
ELM
NN
LSTM
ED LSTM


R Squared
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0.45
0.4
1 2 3 4 5 6 7 8 9 10 11 12
Forecasting Horizon (5-min/step)
Baseline
VARIMA
Reg
QuadReg
CubicReg
SVR
ELM
NN
LSTM
ED LSTM


R Squared
0.84
0.82
0.8
0.78
0.76
0.74
0.72
0.7
0.68
0.66
0.64
1 2 3 4 5 6 7 8 9 10 11 12
Forecasting Horizon (5-min/step)
Baseline
SARIMA
SARIMA2
Reg
QuadReg
RespSurf
CubicReg
SVR
ELM
NN
LSTM
ED LSTM

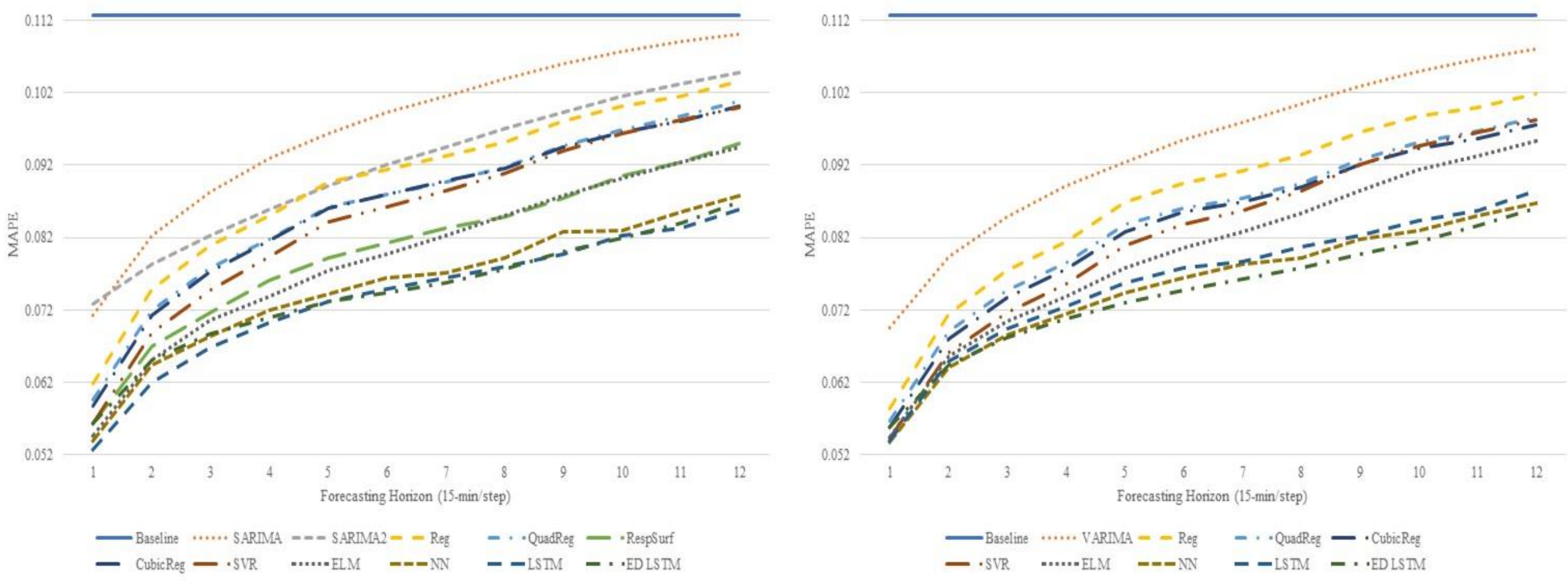

R Squared
0.84
0.82
0.8
0.78
0.76
0.74
0.72
0.7
0.68
1 2 3 4 5 6 7 8 9 10 11 12
Forecasting Horizon (5-min/step)
Baseline
VARIMA
Reg
QuadReg
CubicReg
SVR
ELM
NN
LSTM
ED LSTM

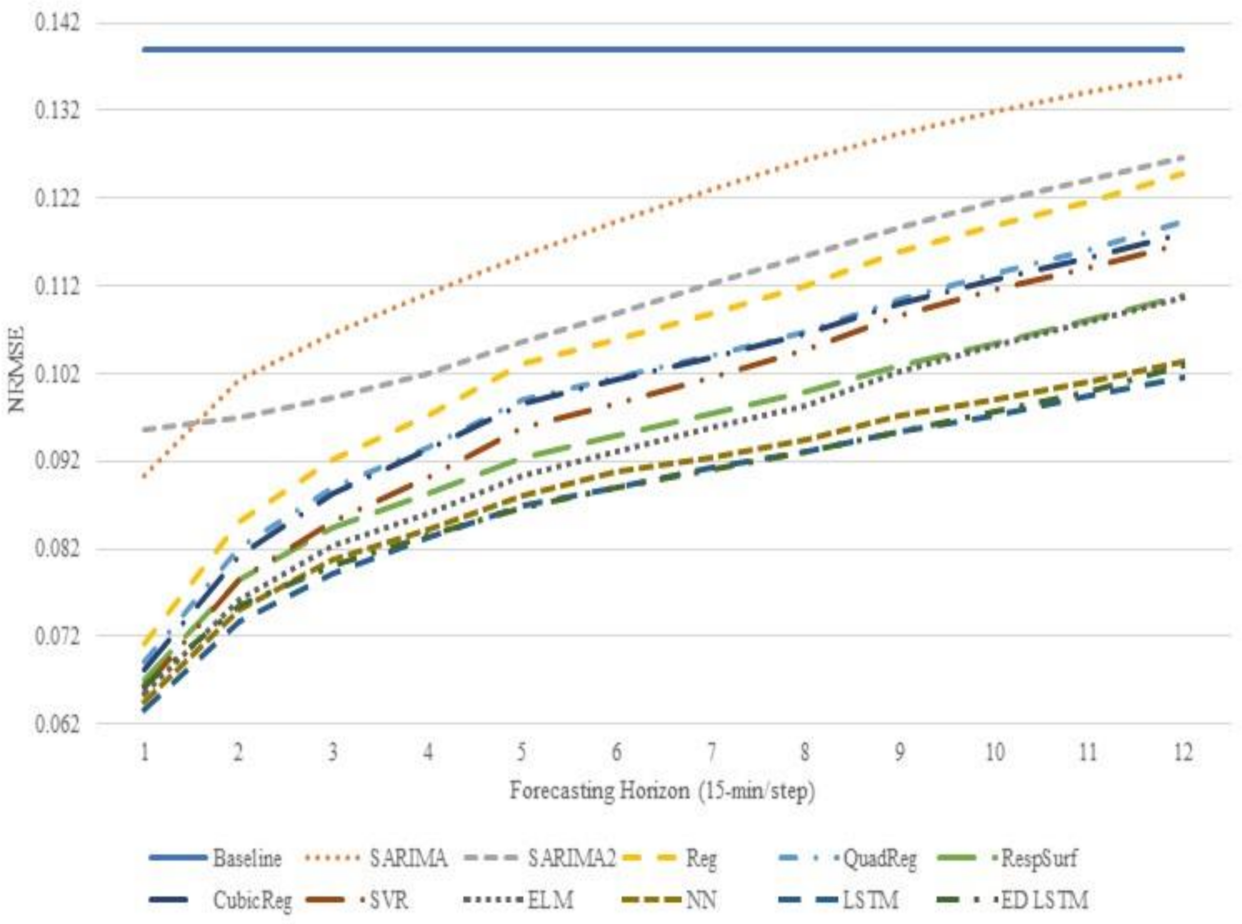
Average improvements of multivariate models over their univariate counterparts 5-min resolution

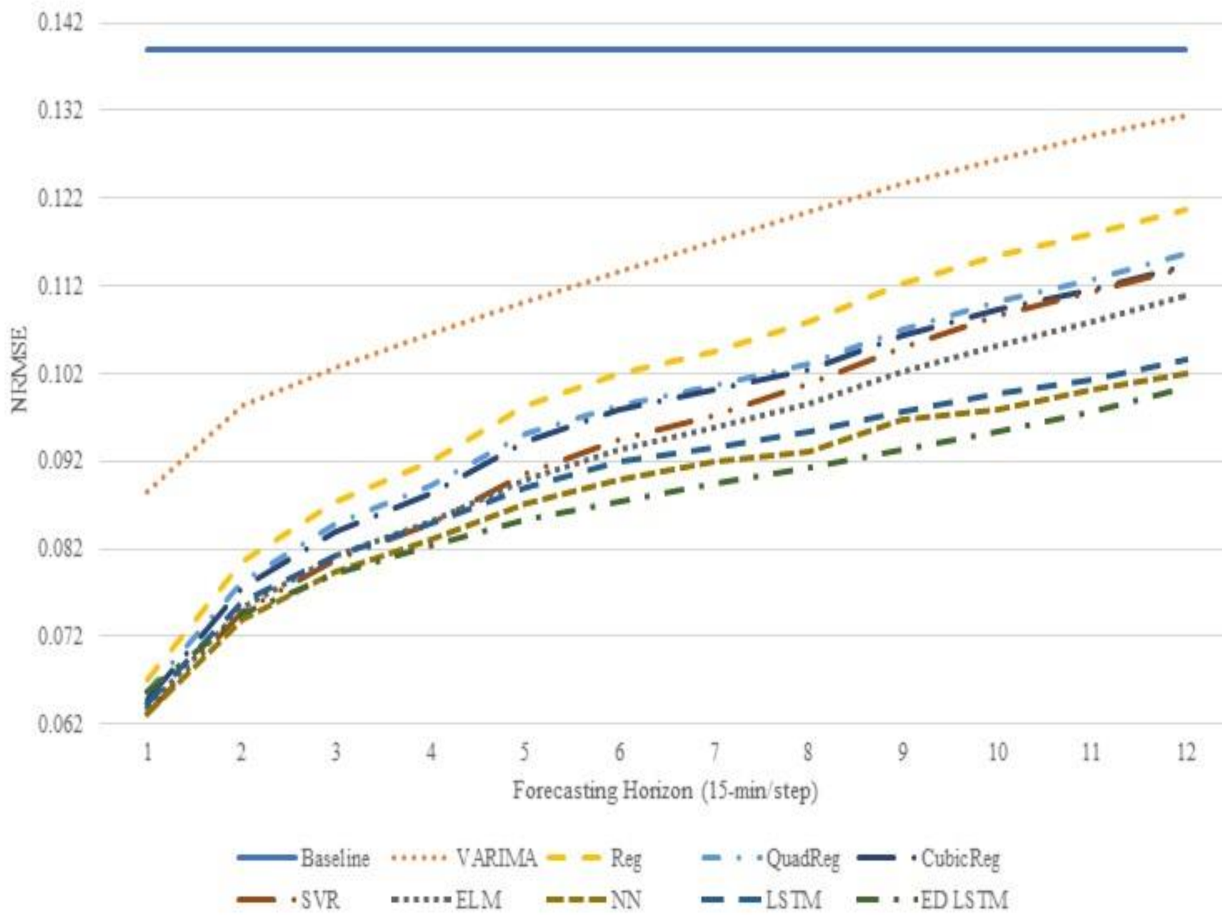


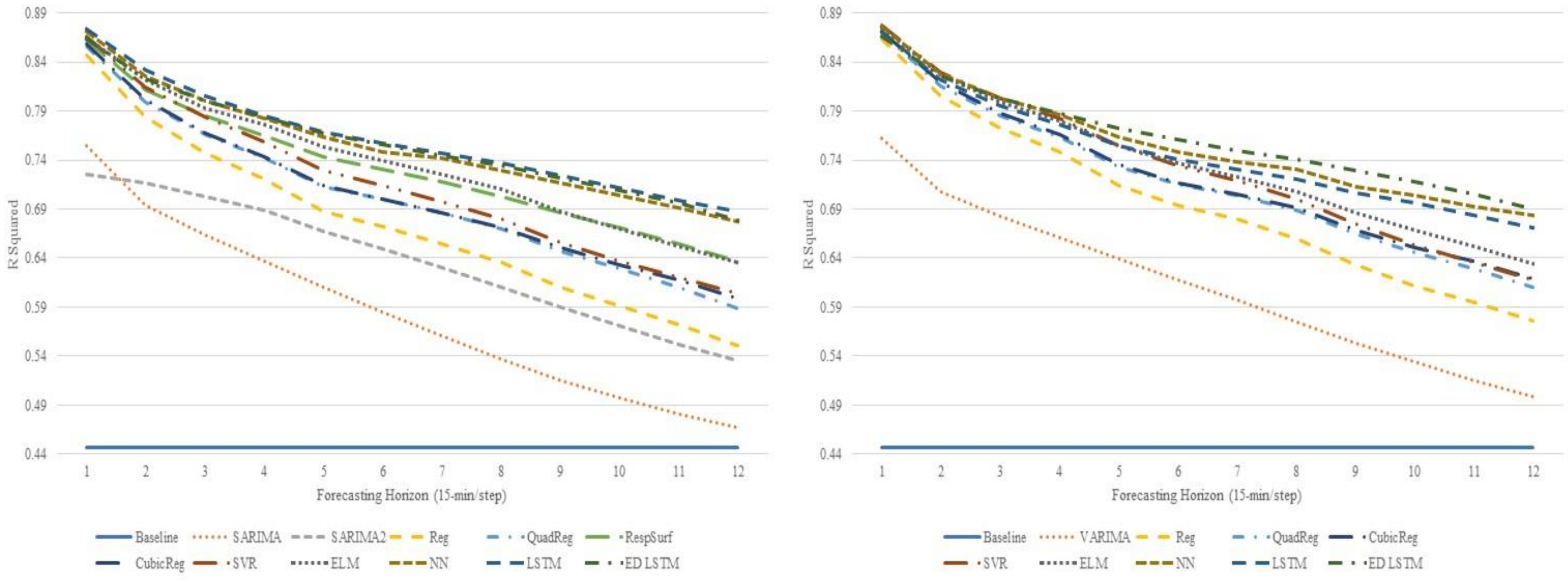
Model	MAPE	NRMSE	R^2
VARIMA	5.93%	5.16%	11.07%
Reg	6.08%	5.69%	4.26%
QuadReg	6.10%	5.52%	3.82%
CubicReg	6.45%	5.63%	3.84%
SVR	4.53%	4.22%	2.82%
ELM	3.23%	3.02%	1.84%
NN	1.35%	1.80%	0.84%
LSTM	-1.01%	-1.34%	-1.12%
ED LSTM	-1.01%	-1.12%	-0.94%

15-min resolution data
univariate (left) vs. multivariate (right)

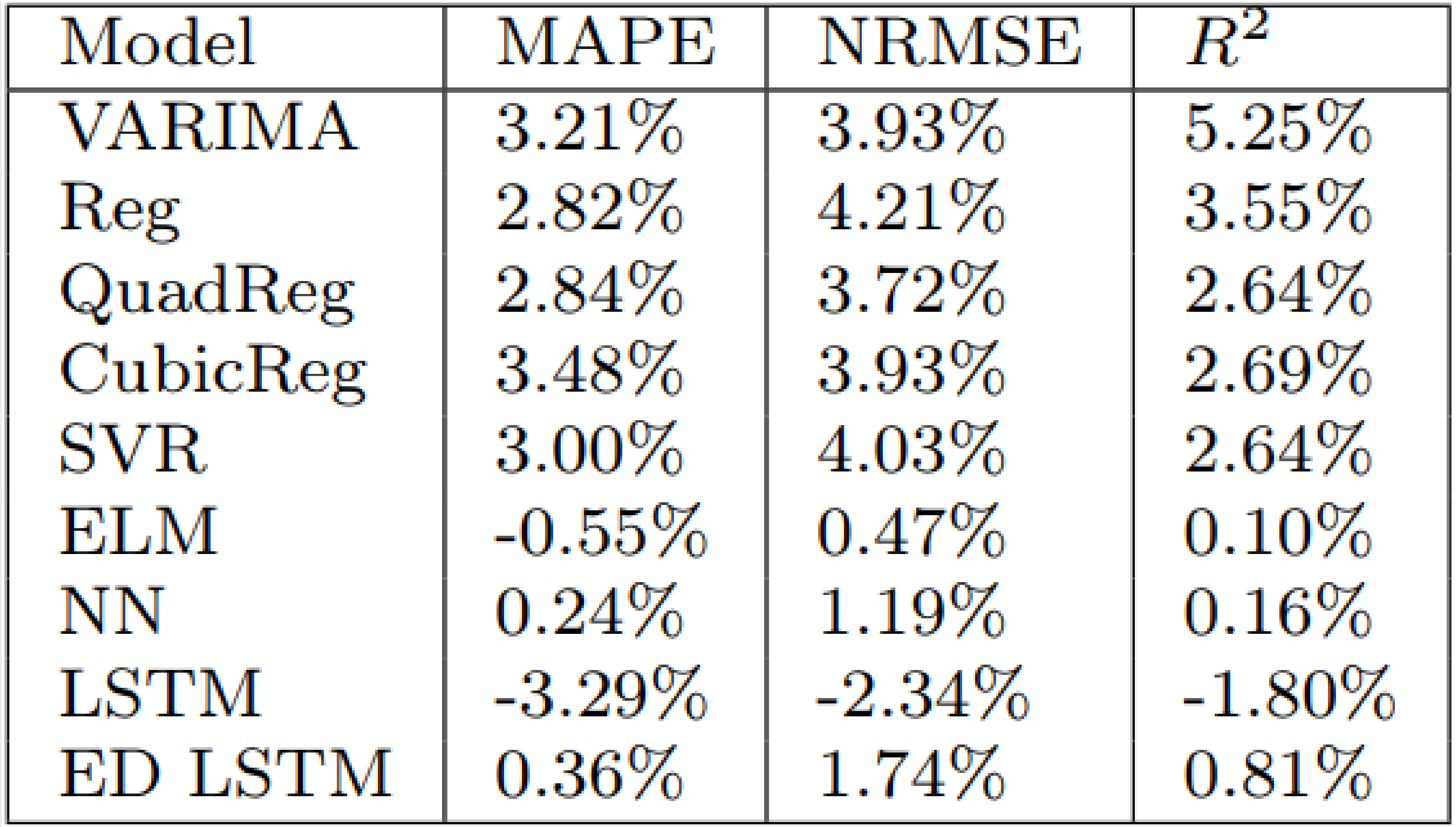

0.112
0.102
0.092
0.082
0.072
0.062
0.052
MAPE
1
2
3
4
5
6
7
8
9
10
11
12
Forecasting Horizon (15-min/step)
Baseline
SARIMA
SARIMA2
Reg
QuadReg
RespSurf
CubicReg
SVR
ELM
NN
LSTM
ED LSTM
0.112
0.102
0.092
0.082
0.072
0.062
0.052
MAPE
1
2
3
4
5
6
7
8
9
10
11
12
Forecasting Horizon (15-min/step)
Baseline
VARIMA
Reg
QuadReg
CubicReg
SVR
ELM
NN
LSTM
ED LSTM


0.142
0.132
0.122
0.112
0.102
0.092
0.082
0.072
0.062
NRMSE
1
2
3
4
5
6
7
8
9
10
11
12
Forecasting Horizon (15-min/step)
Baseline
SARIMA
SARIMA2
Reg
QuadReg
RespSurf
CubicReg
SVR
ELM
NN
LSTM
ED LSTM


0.142
0.132
0.122
0.112
0.102
0.092
0.082
0.072
0.062
NRMSE
1
2
3
4
5
6
7
8
9
10
11
12
Forecasting Horizon (15-min/step)
Baseline
VARIMA
Reg
QuadReg
CubicReg
SVR
ELM
NN
LSTM
ED LSTM

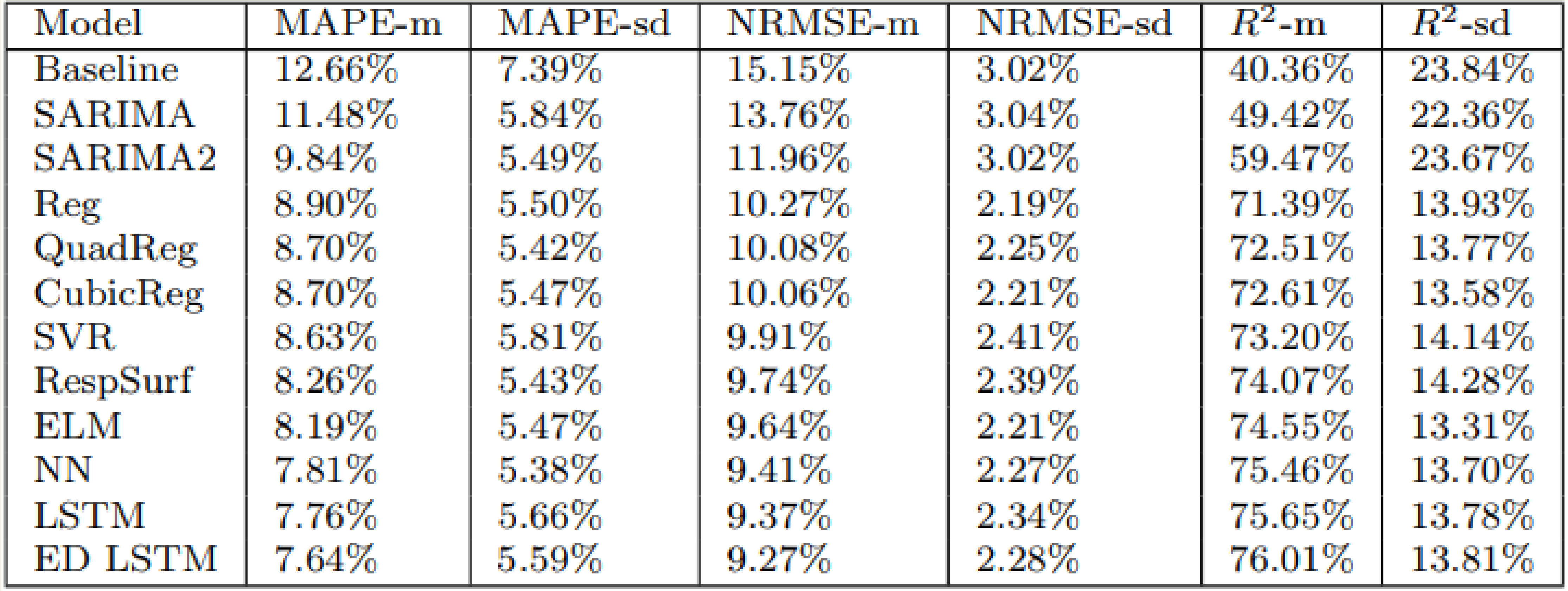

R Squared
0.89
0.84
0.79
0.74
0.69
0.64
0.59
0.54
0.49
0.44
1
2
3
4
5
6
7
8
9
10
11
12
Forecasting Horizon (15-min/step)
Baseline
SARIMA
SARIMA2
Reg
QuadReg
RespSurf
CubicReg
SVR
ELM
NN
LSTM
ED LSTM
R Squared
0.89
0.84
0.79
0.74
0.69
0.64
0.59
0.54
0.49
0.44
1
2
3
4
5
6
7
8
9
10
11
12
Forecasting Horizon (15-min/step)
Baseline
VARIMA
Reg
QuadReg
CubicReg
SVR
ELM
NN
LSTM
ED LSTM

Average improvements of multivariate models over their univariate counterparts 15-min resolution



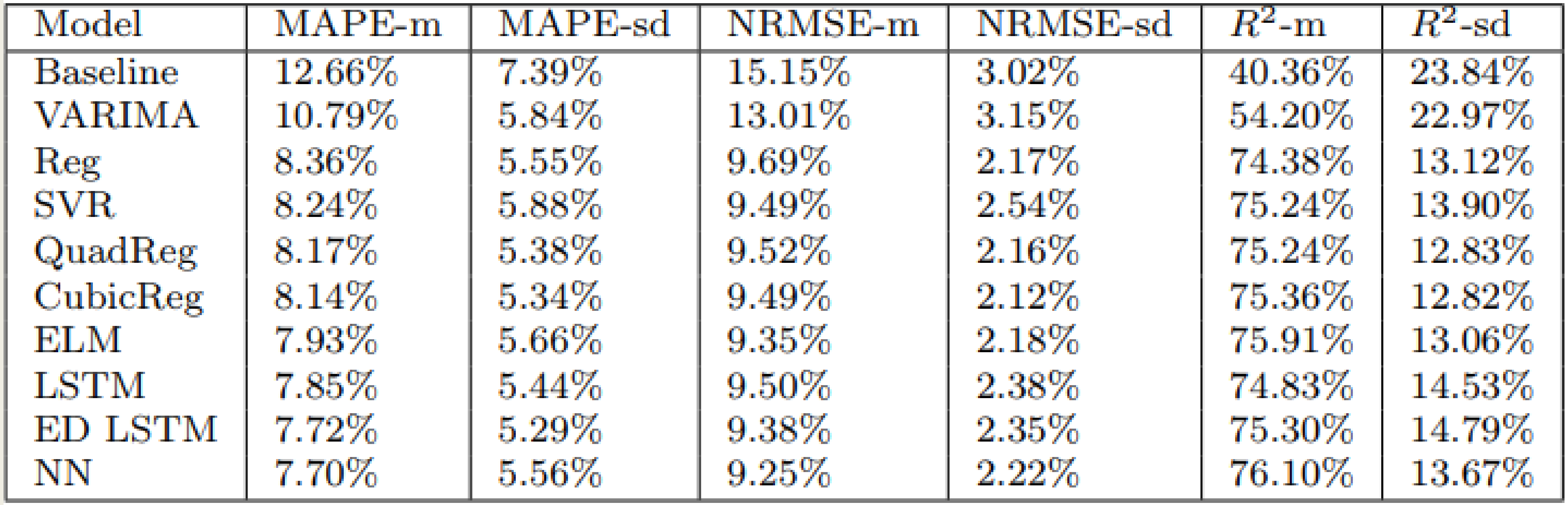
Model	MAPE	NRMSE	R^2
VARIMA	3.21%	3.93%	5.25%
Reg	2.82%	4.21%	3.55%
QuadReg	2.84%	3.72%	2.64%
CubicReg	3.48%	3.93%	2.69%
SVR	3.00%	4.03%	2.64%
ELM	-0.55%	0.47%	0.10%
NN	0.24%	1.19%	0.16%
LSTM	-3.29%	-2.34%	-1.80%
ED LSTM	0.36%	1.74%	0.81%

Average means and standard deviations (across all sensors and forecasting steps)
Ranked by mean MAPE, univariate 5-min resolution



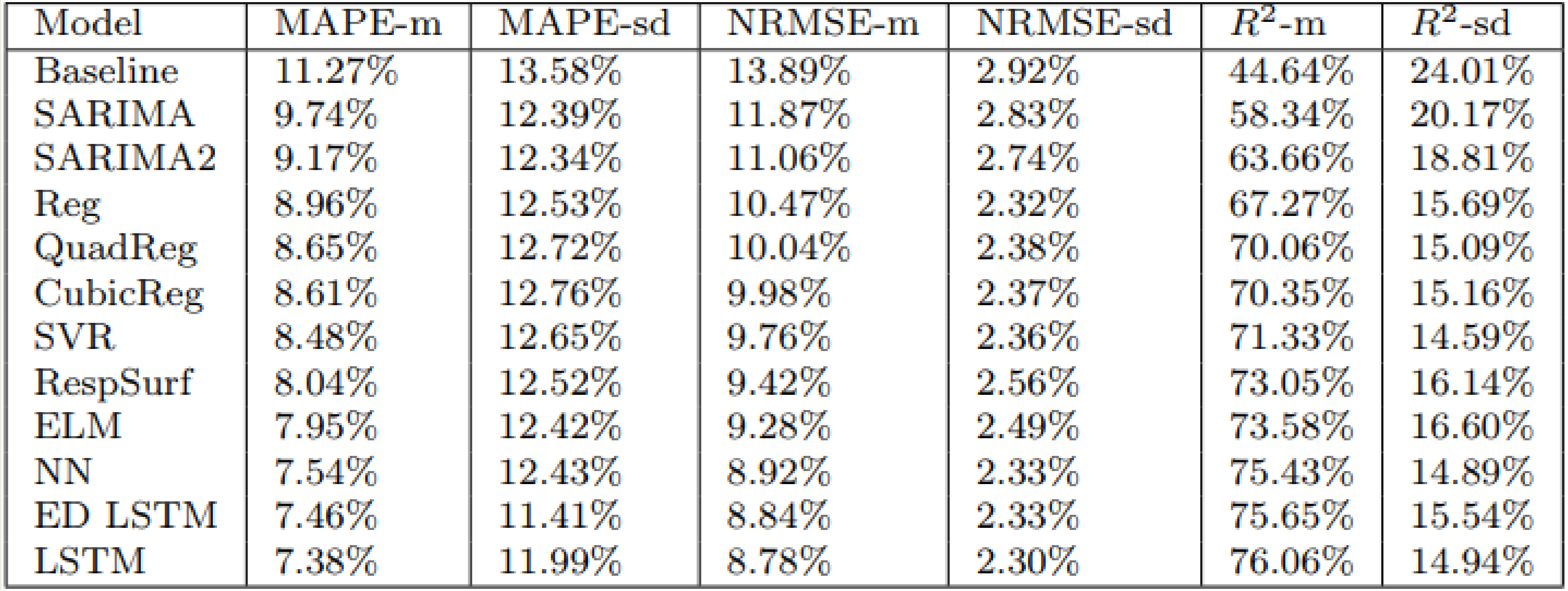
Model	MAPE-m	MAPE-sd	NRMSE-m	NRMSE-sd	R^2-m	R^2-sd
Baseline	12.66%	7.39%	15.15%	3.02%	40.36%	23.84%
SARIMA	11.48%	5.84%	13.76%	3.04%	49.42%	22.36%
SARIMA2	9.84%	5.49%	11.96%	3.02%	59.47%	23.67%
Reg	8.90%	5.50%	10.27%	2.19%	71.39%	13.93%
QuadReg	8.70%	5.42%	10.08%	2.25%	72.51%	13.77%
CubicReg	8.70%	5.47%	10.06%	2.21%	72.61%	13.58%
SVR	8.63%	5.81%	9.91%	2.41%	73.20%	14.14%
RespSurf	8.26%	5.43%	9.74%	2.39%	74.07%	14.28%
ELM	8.19%	5.47%	9.64%	2.21%	74.55%	13.31%
NN	7.81%	5.38%	9.41%	2.27%	75.46%	13.70%
LSTM	7.76%	5.66%	9.37%	2.34%	75.65%	13.78%
ED LSTM	7.64%	5.59%	9.27%	2.28%	76.01%	13.81%

Average means and standard deviations (across all sensors and forecasting steps) Ranked by mean MAPE, multivariate 5-min resolution



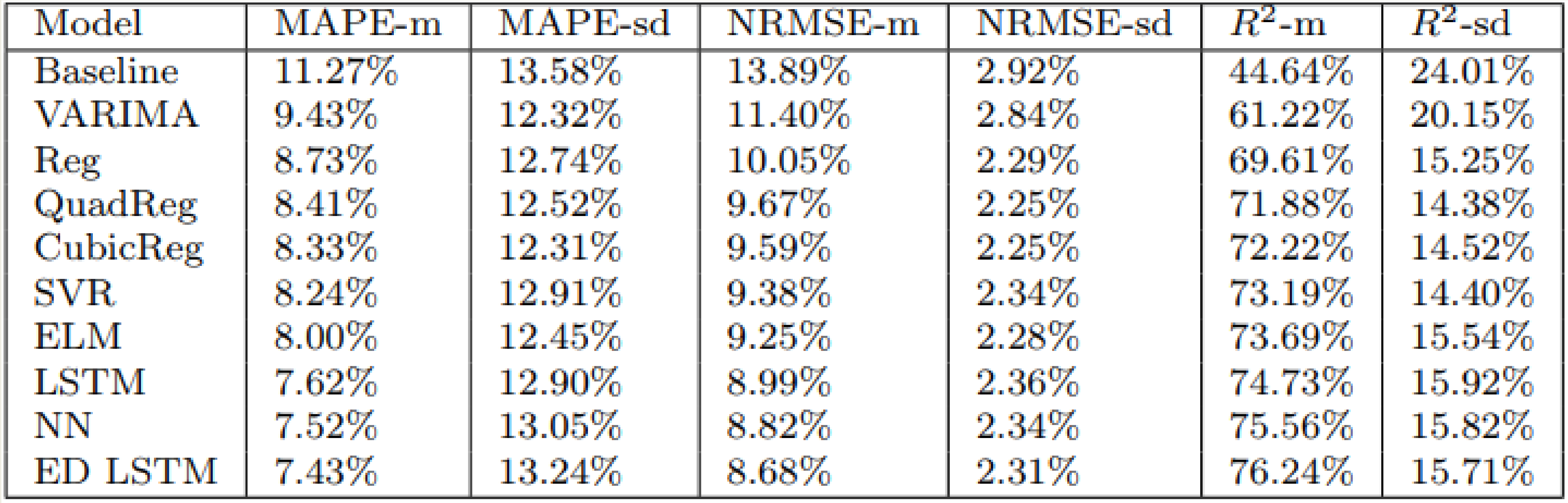
Model	MAPE-m	MAPE-sd	NRMSE-m	NRMSE-sd	R^2-m	R^2-sd
Baseline	12.66%	7.39%	15.15%	3.02%	40.36%	23.84%
VARIMA	10.79%	5.84%	13.01%	3.15%	54.20%	22.97%
Reg	8.36%	5.55%	9.69%	2.17%	74.38%	13.12%
SVR	8.24%	5.88%	9.49%	2.54%	75.24%	13.90%
QuadReg	8.17%	5.38%	9.52%	2.16%	75.24%	12.83%
CubicReg	8.14%	5.34%	9.49%	2.12%	75.36%	12.82%
ELM	7.93%	5.66%	9.35%	2.18%	75.91%	13.06%
LSTM	7.85%	5.44%	9.50%	2.38%	74.83%	14.53%
ED LSTM	7.72%	5.29%	9.38%	2.35%	75.30%	14.79%
NN	7.70%	5.56%	9.25%	2.22%	76.10%	13.67%

Average means and standard deviations (across all sensors and forecasting steps) Ranked by mean MAPE, univariate 15-min resolution



Model	MAPE-m	MAPE-sd	NRMSE-m	NRMSE-sd	R^2-m	R^2-sd
Baseline	11.27%	13.58%	13.89%	2.92%	44.64%	24.01%
SARIMA	9.74%	12.39%	11.87%	2.83%	58.34%	20.17%
SARIMA2	9.17%	12.34%	11.06%	2.74%	63.66%	18.81%
Reg	8.96%	12.53%	10.47%	2.32%	67.27%	15.69%
QuadReg	8.65%	12.72%	10.04%	2.38%	70.06%	15.09%
CubicReg	8.61%	12.76%	9.98%	2.37%	70.35%	15.16%
SVR	8.48%	12.65%	9.76%	2.36%	71.33%	14.59%
RespSurf	8.04%	12.52%	9.42%	2.56%	73.05%	16.14%
ELM	7.95%	12.42%	9.28%	2.49%	73.58%	16.60%
NN	7.54%	12.43%	8.92%	2.33%	75.43%	14.89%
ED LSTM	7.46%	11.41%	8.84%	2.33%	75.65%	15.54%
LSTM	7.38%	11.99%	8.78%	2.30%	76.06%	14.94%

Average means and standard deviations (across all sensors and forecasting steps)
Ranked by mean MAPE, multivariate 15-min resolution



Model	MAPE-m	MAPE-sd	NRMSE-m	NRMSE-sd	R^2-m	R^2-sd
Baseline	11.27%	13.58%	13.89%	2.92%	44.64%	24.01%
VARIMA	9.43%	12.32%	11.40%	2.84%	61.22%	20.15%
Reg	8.73%	12.74%	10.05%	2.29%	69.61%	15.25%
QuadReg	8.41%	12.52%	9.67%	2.25%	71.88%	14.38%
CubicReg	8.33%	12.31%	9.59%	2.25%	72.22%	14.52%
SVR	8.24%	12.91%	9.38%	2.34%	73.19%	14.40%
ELM	8.00%	12.45%	9.25%	2.28%	73.69%	15.54%
LSTM	7.62%	12.90%	8.99%	2.36%	74.73%	15.92%
NN	7.52%	13.05%	8.82%	2.34%	75.56%	15.82%
ED LSTM	7.43%	13.24%	8.68%	2.31%	76.24%	15.71%

Effects of Data Resolution

- Aggregating time series into 15-minute resolution has an inherent smoothing effects
- One-step ahead MAPE in 15-min resolution: ~5%
- Three-step ahead MAPE in 5-min resolution: ~6-7%
- Time series models like SARIMA can be sensitive to sudden fluctuations

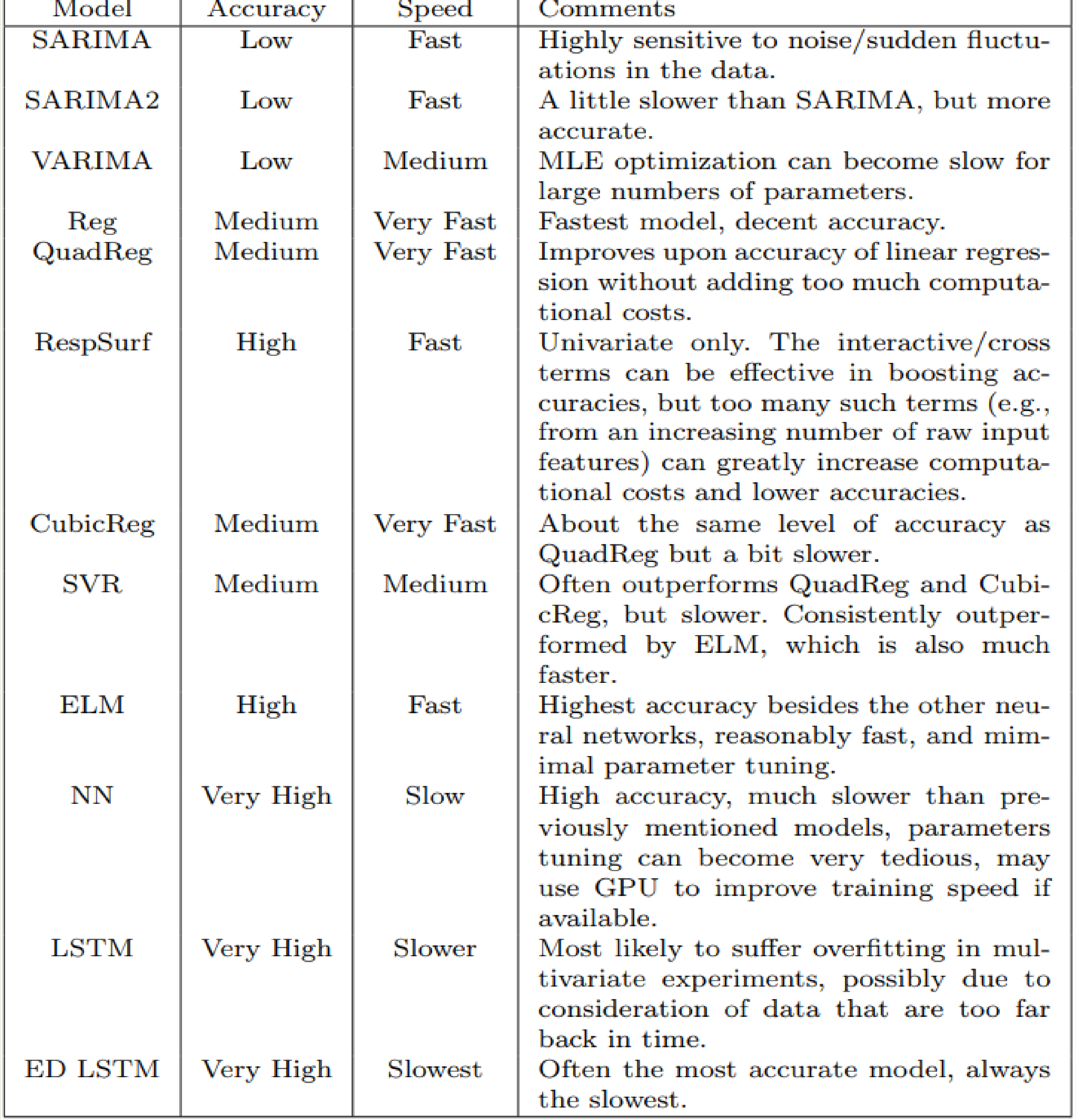
Parameter tuning

- Models that require effort in tuning parameters
  - *SVR*
  - *NN*
  - *LSTM*
- Models that require little effort in tuning parameters
  - *Time series models*
  - *Regression family*
  - *ELM*

Overfitting

- NN: slight improvements in multivariate experiment
- LSTM: minor decrease in accuracy in multivariate experiment, seems to suggest that data for the past 4 weeks may be a bit “excessive”, and causing models to unnecessarily overfitted to the traffic patterns that may not be current/useful for the immediate short terms.
- ED LSTM: minor decrease in 5-min multivariate experiment, minor improvements in 15-minute multivariate experiment. It’s possible that it is easier for ED LSTM to fit the smoother 15-minute resolution data, especially since it has 24 temporal layers. The 5-minute resolution data are subject to more sudden fluctuations, and causing the model to be more distracted by such fluctuations.
- All 3 keras+tensorflow neural networks use random 20% validation sets during training

SUMMARY OF MODELS



Model	Accuracy	Speed	Comments
SARIMA	Low	Fast	Highly sensitive to noise/sudden fluctuations in the data.
SARIMA2	Low	Fast	A little slower than SARIMA, but more accurate.
VARIMA	Low	Medium	MLE optimization can become slow for large numbers of parameters.
Reg	Medium	Very Fast	Fastest model, decent accuracy.
QuadReg	Medium	Very Fast	Improves upon accuracy of linear regression without adding too much computational costs.
RespSurf	High	Fast	Univariate only. The interactive/cross terms can be effective in boosting accuracies, but too many such terms (e.g., from an increasing number of raw input features) can greatly increase computational costs and lower accuracies.
CubicReg	Medium	Very Fast	About the same level of accuracy as QuadReg but a bit slower.
SVR	Medium	Medium	Often outperforms QuadReg and CubicReg, but slower. Consistently outperformed by ELM, which is also much faster.
ELM	High	Fast	Highest accuracy besides the other neural networks, reasonably fast, and mimimal parameter tuning.
NN	Very High	Slow	High accuracy, much slower than previously mentioned models, parameters tuning can become very tedious, may use GPU to improve training speed if available.
LSTM	Very High	Slower	Most likely to suffer overfitting in multivariate experiments, possibly due to consideration of data that are too far back in time.
ED LSTM	Very High	Slowest	Often the most accurate model, always the slowest.

Conclusion

- Incorporation of spatially dependent data generally help
- Various neural networks have the highest accuracy, but are also the most computationally expensive
- ELM is fairly fast, requires little parameter tuning, and has the best accuracy besides the other neural networks.
- The cell state in ED LSTM helps forecasting accuracy at higher steps (more distant future), while the stateless NN may be able to do better in the first couple of steps but its accuracy depreciates more quickly than ED LSTM.
- In comparing with similar traffic forecasting studies that used huge/deep neural networks, their performance does not seem to be significantly different from ours.

Next Steps

- Use Markov Logic Network to model traffic conditions and First Order Logic to reason
  - *Determination of upstream sensors where traffic is likely to flow from. If a upstream sensor location is congested, choose closer sites.*
  - *Determination of sensor sites similarity to apply transfer learning, which will significantly reduce training time of neural networks.*
- Real-time forecasting by repeatedly re-training on the most recent data. Preliminary work shows improved accuracy.

Questions?