EnhanceNet: Plugin Neural Networks for Enhancing Correlated Time Series Forecasting

International Conference on Data Engineering 2021

January 17, 2021

Razvan-Gabriel Cirstea[1], Tung Kieu[1], Chenjuan Guo[1], Bin Yang[1] and Sinno Jialin Pan[2]

{razvan, tungkvt, cguo, byang}@cs.aau.dk, sinnopan@ntu.edu.sg

[1]Department of Computer Science
Aalborg University
Denmark

[2]School of Computer Science and Engineering
Nanyang Technological University
Singapore

AALBORG UNIVERSITY
DENMARK

NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

Agenda

- Introduction
- Temporal Correlations
  - RNN and TCN based models
  - Limitations of the current methods
  - Dynamic Filter Generation
- Spatial Correlations
  - Graph Convolution
  - Limitations of the current approaches
  - Dynamic Adjacency Matrix Generation
- Experimental setup
- Experimental Results
- Conclusion

Introduction

- With continued digitization, various cyber physical systems (CPSs) are deployed
- A CPS contains multiple sensors producing a collection of time series which might exhibit strong spatial and temporal correlations
- Accurate forecasting of time series helps:
  - revealing holistic system
  - identifying trends
  - predicting future behavior
  - detecting outlier

Introduction

Accurate time series forecasting relies on models that are able to capture temporal dynamics and correlations among different entities

- Temporal dynamics is important because the attributes of entities, e.g., travel speeds and traffic flows of roads, change over time and historical attributes influence future attributes
- Entities often interact with each other, such as traffic at different roads influencing each other

Our Goal

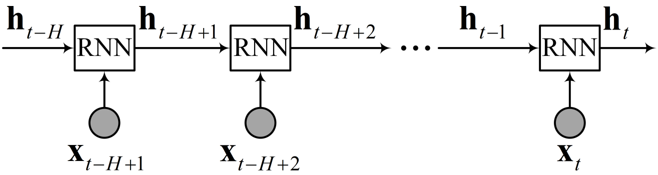
Our goal is to capture **distinct temporal dynamics for different entities** and **dynamic entity correlations across time**, so that forecasting accuracy is improved while model parameters to be learned are reduced.

Capturing Temporal Correlations

- When dealing with sequential data it is important to capture its temporal dynamics
- Most of the the proposed models falls into 2 families
  - Recurrent Neural Networks (RNN)
  - Temporal Convolutional Networks (TCN)

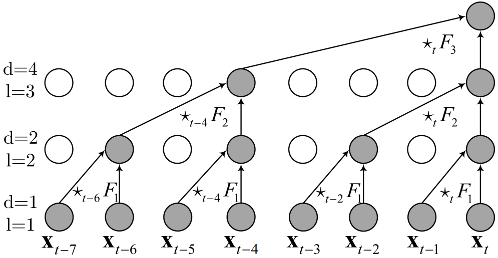
Recurrent Neural Networks

- RNNs are called recurrent because they perform the same task for every element of a sequence, with the output being depended on the previous computations
- The idea behind RNNs is to make use of sequential information



Temporal Convolutional Networks

- A TCN is a hierarchical model that consists of multiple layers of dilated causal convolutions
- It is more computational efficient when compared with RNN

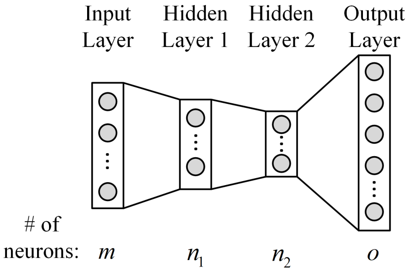


Limitations of the current methods

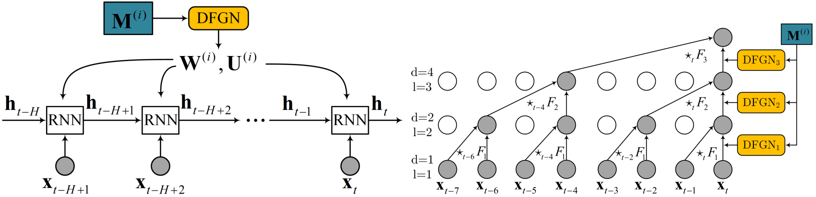
- The proposed methods can not capture distinct temporal dynamics, instead they use the same filters to model all the entities which might lead to overgeneralization
- One naive way of solving the problem is to have a separate model for each node individually
  - High number of parameters will lead to memory problems
  - It will be computationally expensive and might overfit

Dynamic Filter Generation Network (DFGN)

- We propose assigning each node a **trainable memory**
- Afterwards we can use a neural network to generate **node specific** filters for RNN/TCN given its memory



Dynamic Filter Generation Integration



$$\mathbf{W}^{(i)}, \mathbf{U}^{(i)} = DFGN(\mathbf{M}^{(i)})$$

$$\mathbf{h}_t^{(i)} = GRU(\mathbf{x}_t^{(i)}, \mathbf{h}_{t-1}^{(i)} | \mathbf{W}^{(i)}, \mathbf{U}^{(i)})$$

$$\mathbf{F}_l^{(i)} = DFGN_l(\mathbf{M}^{(i)})$$

$$H = TCN(\mathbf{x}_t^{(i)} | \mathbb{F}_1^{(i)}, \mathbb{F}_2^{(i)}, \ldots, \mathbb{F}_l^{(i)})$$

Capturing Spatial Correlations

- Capturing correlations among time series of different entities is essential to ensure forecasting accuracy
- Existing studies model relationships among the entities as a graph, where a vertex represents an entity and an edge represents some relationship between two entities
- For example, two entities are connected by an edge if their distance is smaller than a threshold

Graph Convolution

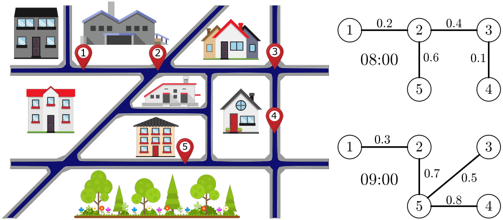
- Graph convolution (GC) is a powerful operation where the model learns the features by inspecting neighboring nodes
- Different variations of graph convolution exist, such as graph convolution, diffusion convolution, and spectral graph convolution but they all follow the same principle:

$$\mathbf{Z} = \mathbf{S} \star_{\mathcal{G}} \mathbf{x}_t = \mathbf{A}\mathbf{x}_t\mathbf{S} \tag{1}$$

Limitations of the current methods

The adjacency matrix derived from distances is sub-optimal because:

- it utilize a static adjacency matrix failing to capture time-varying correlations
- it is data-dependent, and thus incapable to capture, for example, correlations between entities that are far away but still show correlations

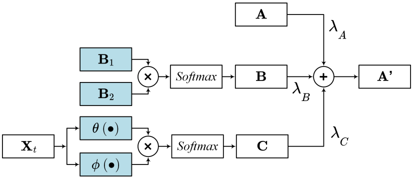


Dynamic Adjacency Matrix Generation Network

To address the previous limitations, we propose Dynamic Adjacency Matrix Generation Network (DAMGN) to generate dynamic and adaptive adjacency matrices

- it is dynamic since unique adjacency matrices at different timestamps are generated
- it is adaptive since the adjacency matrices are derived from the input time series, which may potential capture any correlations, but not just based on, e.g., distances

Dynamic Adjacency Matrix Generation Network



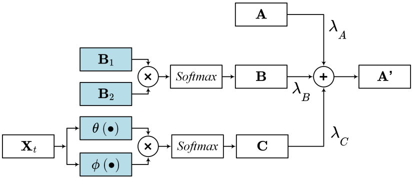
A is the distance based matrix

Generating B:

- Is intended to capture hidden correlations among different entities, which cannot be captured by, e.g., distance based adjacency matrix

$$\mathbf{B} = \mathit{Softmax}(\mathit{ReLU}(\mathbf{B}_1 \mathbf{B}_2^\top)) \quad (2)$$

Dynamic Adjacency Matrix Generation Network



Generating C:

- C represents a time-specific adjacency matrix, which aims at capturing correlations among entities at a specific timestamp.

$$\mathbf{C}[i,j] = f(\mathbf{x}_t^{(i)}, \mathbf{x}_t^{(j)}) = \frac{e^{\theta(\mathbf{x}_t^{(i)})^\top \phi(\mathbf{x}_t^{(j)})}}{\sum_{j=1}^{N} e^{\theta(\mathbf{x}_t^{(i)})^\top \phi(\mathbf{x}_t^{(j)})}}. \qquad (3)$$

Dynamic Adjacency Matrix Generation Network

When combining the three adjacency matrices we used a weighted sum, where each adjacency matrix has its own **parameter** λ that controls its contribution to the final adjacency matrix.

$$\mathbf{Z} = \mathbf{S} \star_{\mathcal{G}} \mathbf{x}_t = (\lambda_A \mathbf{A} + \lambda_B \mathbf{B} + \lambda_C \mathbf{C}) \mathbf{x}_t \mathbf{S} \tag{4}$$

Dynamic Adjacency Matrix Generation Network Integration

Integration with GRU

$$
\begin{aligned}
\mathbf{r}_t &= \sigma(\mathbf{W}_r \star_{\mathcal{G}} \mathbf{x}_t + \mathbf{U}_r \star_{\mathcal{G}} \mathbf{h}_{t-1}) \\
\mathbf{u}_t &= \sigma(\mathbf{W}_u \star_{\mathcal{G}} \mathbf{x}_t + \mathbf{U}_u \star_{\mathcal{G}} \mathbf{h}_{t-1}) \\
\mathbf{h}_t &= (\mathbf{W}_h \star_{\mathcal{G}} \mathbf{x}_t + \mathbf{U}_h \star_{\mathcal{G}} (\mathbf{r}_t \odot \mathbf{h}_{t-1})).
\end{aligned}
$$

Integration with TCN

$$
\mathbf{h}_t = \mathbf{W} \star_{\mathcal{G}} \mathbf{TCN}(x) \tag{5}
$$

Datasets

We utilize 3 data sets from two different domains:

- EB includes 182 traffic sensors in the East Bay area. It covers 3 months and every 5 minutes we have an average speed reading from each sensor.
- LA includes 207 traffic sensors across Los Angeles County highways. It covers 4 months and every 5 minutes we get 2 attributes: an average speed and information on time and date.
- US includes 36 meteorological stations in the United States, covering 5 years. Every hour we get 6 attributes: temperature, humidity, pressure, wind direction, wind speed, and weather description.

Experimental setup

- The three datasets are split chronologically into 3 partitions— 70% for training, 10% for validation, and 20% for testing.
- We consider a commonly used setting where we use recent 12 timestamps as input to predict the next 12 timestamps.
- We used a Gaussian kernel when constructing the adjacency matrix: $\mathbf{A}_{ij} = \exp -\frac{dist(v_i, v_j)^2}{\sigma^2}$

Experimental setup

We design 3 types of experiments:

- We assess the efficiency of the DFGN by comparing the base models vs the enhanced ones (D)
- We assess the efficiency of the DAMG by comparing the base models vs the enhanced ones (DA)
- We assess the efficiency of both components (D-DA) when compared with other baselines such as: ARIMA, STGCN, DCRNN and Graph WaveNet

Results: Accuracy enhancement on RNN and TCN

Data	Models	3^{rd} *timestamp*			6^{th} *timestamp*			12^{th} *timestamp*			# Para
		MAE	*MAPE*	*RMSE*	*MAE*	*MAPE*	*RMSE*	*MAE*	*MAPE*	*RMSE*	
EB	RNN	3.34	6.67	6.32	4.03	8.48	7.90	5.17	11.41	10.07	74k
	D-RNN	**3.28**	6.52	**6.18**	3.90	8.20	**7.62**	4.83	10.76	**9.44**	49k
	TCN	3.37	6.65	8.08	4.05	8.51	9.32	5.12	11.43	11.10	247k
	D-TCN	3.30	**6.45**	7.78	**3.89**	**8.05**	8.91	**4.76**	**10.41**	10.43	100k
LA	RNN	3.03	8.09	5.98	3.69	10.46	7.48	4.69	14.13	9.35	75k
	D-RNN	2.89	7.79	5.73	3.36	9.70	6.94	3.90	12.03	8.24	49k
	TCN	2.98	7.91	5.89	3.58	10.18	7.27	4.44	13.61	8.95	247k
	D-TCN	**2.85**	**7.59**	**5.67**	**3.29**	**9.37**	**6.82**	**3.79**	**11.41**	**8.02**	103k
US	RNN	1.25	0.43	1.83	1.91	0.66	2.65	2.11	0.73	2.89	75k
	D-RNN	1.25	0.43	1.83	1.90	0.65	2.65	2.14	0.74	2.95	32k
	TCN	1.15	0.40	1.67	1.78	0.62	2.47	2.07	0.72	2.84	247k
	D-TCN	**1.12**	**0.39**	**1.63**	**1.70**	**0.59**	**2.37**	**1.99**	**0.69**	**2.75**	104k

TABLE I: Effect of *DFGN* on capturing distinct temporal dynamics.

Results: Accuracy enhancement on GRNN and GTCN

Data	Models	3^{rd} timestamp			6^{th} timestamp			12^{th} timestamp			# Para
		MAE	*MAPE*	*RMSE*	*MAE*	*MAPE*	*RMSE*	*MAE*	*MAPE*	*RMSE*	
EB	GRNN	3.20	6.41	5.96	3.68	7.75	7.11	4.35	9.42	8.52	371k
	D-GRNN	3.17	6.34	5.88	3.59	7.57	6.94	4.15	8.96	8.15	167k
	DA-GRNN	3.07	6.10	5.70	3.46	7.14	6.71	3.95	8.25	7.77	378k
	D-DA-GRNN	**3.05**	**6.09**	**5.67**	**3.40**	**7.12**	**6.57**	**3.83**	**8.17**	**7.53**	172k
	GTCN	3.15	6.18	7.46	3.60	7.44	8.33	4.21	8.86	9.39	288k
	D-GTCN	3.17	6.22	7.47	3.58	7.37	8.25	4.15	8.77	9.24	252k
	DA-GTCN	3.10	**6.07**	7.38	3.50	7.16	8.13	3.96	8.35	8.96	296k
	D-DA-GTCN	3.11	6.14	7.43	3.45	7.15	8.09	3.87	8.09	8.82	195k
LA	GRNN	2.77	7.30	5.38	3.15	8.80	6.45	3.60	10.50	7.60	372k
	D-GRNN	2.66	6.81	5.11	3.04	8.24	6.17	3.47	9.93	7.32	172k
	DA-GRNN	2.67	6.86	5.17	3.06	8.27	6.25	3.49	9.86	7.38	381k
	D-DA-GRNN	**2.64**	**6.75**	**5.07**	**3.02**	**8.19**	**6.12**	**3.45**	**9.85**	7.27	180k
	GTCN	2.72	6.94	5.21	3.12	8.46	6.27	3.59	10.20	7.43	288k
	D-GTCN	2.71	6.85	5.23	3.08	8.28	6.27	3.55	9.96	7.43	113k
	DA-GTCN	2.71	7.00	5.12	3.08	8.44	6.20	3.53	10.06	7.25	297k
	D-DA-GTCN	2.69	6.93	5.14	3.06	8.29	6.19	3.49	9.96	**7.23**	261k
US	GRNN	0.89	**0.30**	1.32	1.28	0.44	1.84	1.56	0.54	2.22	226k
	D-GRNN	0.92	0.31	1.36	1.30	0.45	1.89	1.57	0.54	2.23	96k
	DA-GRNN	0.88	**0.30**	1.30	1.24	**0.42**	1.78	1.52	**0.52**	2.15	227k
	D-DA-GRNN	0.88	**0.30**	1.29	1.23	**0.42**	1.75	**1.50**	**0.52**	**2.13**	97k
	GTCN	0.90	0.31	1.33	1.30	0.45	1.86	1.58	0.55	2.24	272k
	D-GTCN	0.91	0.31	1.35	1.29	0.45	1.85	1.59	0.55	2.24	111k
	DA-GTCN	0.88	**0.30**	1.30	1.26	0.44	1.80	1.55	0.54	2.20	273k
	D-DA-GTCN	**0.86**	**0.30**	**1.27**	**1.22**	**0.42**	**1.74**	1.53	0.53	2.15	156k

TABLE II: Effect of *DFGN* and *DAMGN* on capturing distinct temporal dynamics and entity correlations.

Final Model Results

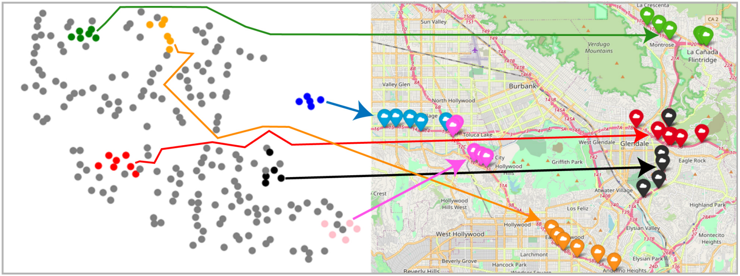
Data	Models	3rd timestamp			6th timestamp			12th timestamp		
		MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE
EB	ARIMA	6.08	11.65	10.75	6.48	13.22	11.55	6.61	13.81	10.87
	FC-LSTM	4.17	8.67	7.83	4.31	9.08	8.16	4.56	9.90	8.70
	WaveNet	3.37	6.65	8.08	4.05	8.51	9.32	5.12	11.43	11.10
	STGCN	4.37	8.45	10.83	4.81	9.67	11.48	5.56	11.82	12.53
	Graph WaveNet	3.10	6.05	7.36	3.47	7.07	8.07	3.93	8.18	8.86
	DCRNN	3.20	6.41	5.96	3.68	7.75	7.11	4.35	9.42	8.52
	D-DA-GTCN	3.11	6.14	7.43	3.45	7.15	8.09	3.87	**8.09**	8.82
	D-DA-GRNN	**3.05**	**6.04**	**5.67**	**3.40**	**7.05**	**6.57**	**3.83**	8.17	**7.53**
LA	ARIMA	3.99	9.60	8.21	5.15	12.70	10.45	6.90	17.40	13.23
	FC-LSTM	3.44	9.60	6.30	3.77	10.90	7.23	4.37	13.20	8.69
	WaveNet	2.99	8.04	5.89	3.59	10.25	7.28	4.45	13.62	8.93
	STGCN	2.88	7.62	5.74	3.47	9.57	7.24	4.59	12.70	9.40
	Graph WaveNet	2.69	6.90	5.15	3.07	8.37	6.22	3.53	10.01	7.37
	DCRNN	2.77	7.30	5.38	3.15	8.80	6.45	3.60	10.50	7.59
	D-DA-GTCN	2.69	6.93	5.14	3.06	8.29	6.19	3.49	9.96	**7.23**
	D-DA-GRNN	**2.64**	**6.75**	**5.07**	**3.02**	**8.19**	**6.12**	**3.45**	**9.85**	**7.23**
US	ARIMA	2.02	0.73	2.69	3.78	1.37	4.54	5.61	2.05	6.95
	FC-LSTM	1.68	0.58	2.30	1.84	0.63	2.54	2.02	0.70	2.77
	WaveNet	1.15	0.40	1.67	1.78	0.62	2.47	2.07	0.72	2.84
	STGCN	1.74	0.60	2.31	1.94	0.68	2.58	2.03	0.71	2.69
	Graph WaveNet	0.87	0.31	1.28	1.23	0.43	1.76	1.54	0.55	2.19
	DCRNN	0.89	**0.30**	1.32	1.28	0.44	1.84	1.56	0.54	2.22
	D-DA-GTCN	**0.86**	**0.30**	**1.27**	**1.22**	**0.42**	**1.74**	1.53	0.53	2.15
	D-DA-GRNN	**0.88**	**0.30**	1.29	1.23	**0.42**	1.75	**1.50**	**0.52**	**2.13**

TABLE III: Comparison with baselines and state-of-the-art methods.

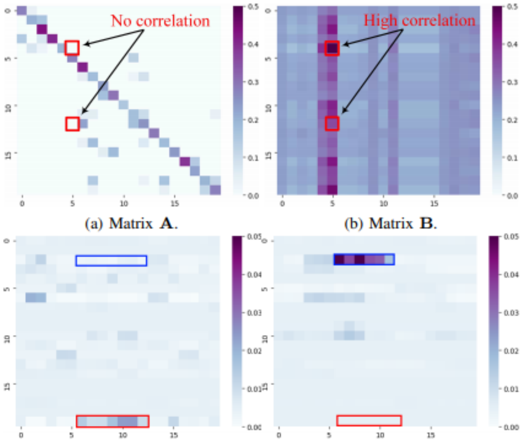
Runtime

Models	EB		LA		US	
	T (s)	P (ms)	T (s)	P (ms)	T (s)	P (ms)
RNN	17.4	0.31	25.5	0.35	25.9	0.29
D-RNN	24.8	0.31	36.1	0.33	34.6	0.29
TCN	21.8	0.33	33.6	0.16	14.4	0.28
D-TCN	76.7	0.23	108.7	0.27	46.1	0.23
GRNN	54.4	1.31	88.1	1.59	50.2	0.60
D-GRNN	74.4	0.95	115.7	1.18	59.1	0.57
DA-GRNN	66.0	1.35	106.9	1.51	50.8	0.60
D-DA-GRNN	82.6	0.90	129.5	1.10	60.5	0.52
GTCN	29.8	0.25	45.9	0.29	17.6	0,18
D-GTCN	77.7	0.27	131.3	0.33	50.3	0,19
DA-GTCN	34.7	0.27	53.6	0.32	19.6	0.19
D-DA-GTCN	85.0	0.29	129.6	0.32	51.7	0.20

Memories Learned



Learned Adjacency Matrices



(a) Matrix **A**. (b) Matrix **B**.

(c) Matrix **C** at two different timestamps.

Conclusion

- We present a framework with two plugin networks Distinct Filter Generation Network and Dynamic Adjacency Matrix Generation Network, which can be employed to enhance the accuracy of many time series forecasting models.
- The two plugin networks are able to capture distinct temporal dynamics among entities and dynamic entity correlations, which existing models fail to explore.
- The two plugin networks are also able to reduce the total number of parameters.

Thank you for your attention!

AALBORG UNIVERSITY
DENMARK

NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE