



Nhóm 9

# Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks

Lớp : IS403.Q11  
GVHD: Dương Phi Long

# Danh sách thành viên

▶	Phạm Bá Bằng	23520090
▶	Nguyễn Võ Ngọc Bảo	23520131
▶	Nguyễn Gia Bảo	23520121
▶	Võ Đức Tài	23521381
▶	Thái Văn Vũ	23521816



# Overview

►	Bối cảnh & Bài toán	01
►	Các nghiên cứu liên quan	02
►	Phương pháp	03
►	Kết quả	04
►	Kết luận	05





# Bối cảnh & Bài toán

**Bài toán:** dự báo chuỗi thời gian đa biến  
(multivariate time series forecasting)

**Thách thức:** dữ liệu thực tế thường

- ▶ Phụ thuộc ngắn hạn (short-term local dependency)
- ▶ Chu kỳ lặp lại dài hạn (Long-term complex repetitive patterns)
- ▶ Sự thay đổi về quy mô dữ liệu (scale changing)

# Các nghiên cứu liên quan

## Phương pháp thống kê (ARIMA, VAR)

- ▶ Tốt khi dự đoán dữ liệu đơn biến tuyến tính
- ▶ Kém khi dự đoán các dữ liệu phi tuyến tính phức tạp
- ▶ Chi phí cao

## RNN/LSTM truyền thống

- ▶ Bị Vanishing Gradient khi chuỗi quá dài
- ▶ Khó bắt được chu kỳ (periodicity) xa

# LSTNet

(Long- and Short-term Time-series Network)

- ▶ Là mô hình Deep Learning kết hợp **CNN**, **RNN (RNN-Skip)** và **AR**
- ▶ Có khả năng giải quyết đồng thời
  - Phụ thuộc ngắn hạn
  - Chu kỳ dài hạn
  - Thành phần tuyến tính
- ▶ **Thành phần chính**
  - **CNN**: trích xuất đặc trưng cục bộ và mối quan hệ giữa các biến
  - **RNN**: học chu kỳ dài hạn bằng cách nhảy theo chu kỳ
  - **AR**: mô hình hóa xu hướng tuyến tính và scale

# Phương pháp thực hiện

## Tổng quan 4 bộ dữ liệu

**electricity.txt:** Dữ liệu đo lường điện tiêu thụ tính theo kWh, thu thập mỗi 15 phút từ 2011 tới 2014

**Time steps:** 26303

**Variables:** 321 (tương đương 321 khách hàng)

# Phương pháp thực hiện

## Tổng quan 4 bộ dữ liệu

**traffic.txt:** Dữ liệu giao thông thu thập trong vòng 48 tháng (2015-2016) theo từng giờ của bang California. Dữ liệu đo mức độ “đầy” của giao thông (0-1)

**Time steps:** 17543

**Variables:** 862 (tương đương 862 cảm biến)



# Phương pháp thực hiện

## Tổng quan 4 bộ dữ liệu

**solar\_AL.txt:** Dữ liệu sản xuất điện mặt trời thu thập mỗi 10 phút trong năm 2006

**Time steps:** 52559

**Variables:** 137 (tương đương 137 nhà máy điện)

# Phương pháp thực hiện

## Tổng quan 4 bộ dữ liệu

**exchange\_rate.txt:** Dữ liệu tỷ số đổi hoái hằng ngày của 8 quốc gia đo từ 1990 tới 2016

**Time steps:** 7587

**Variables:** 8 (tương đương 8 quốc gia)

# Phương pháp thực hiện

Tổng quan 4 bộ dữ liệu

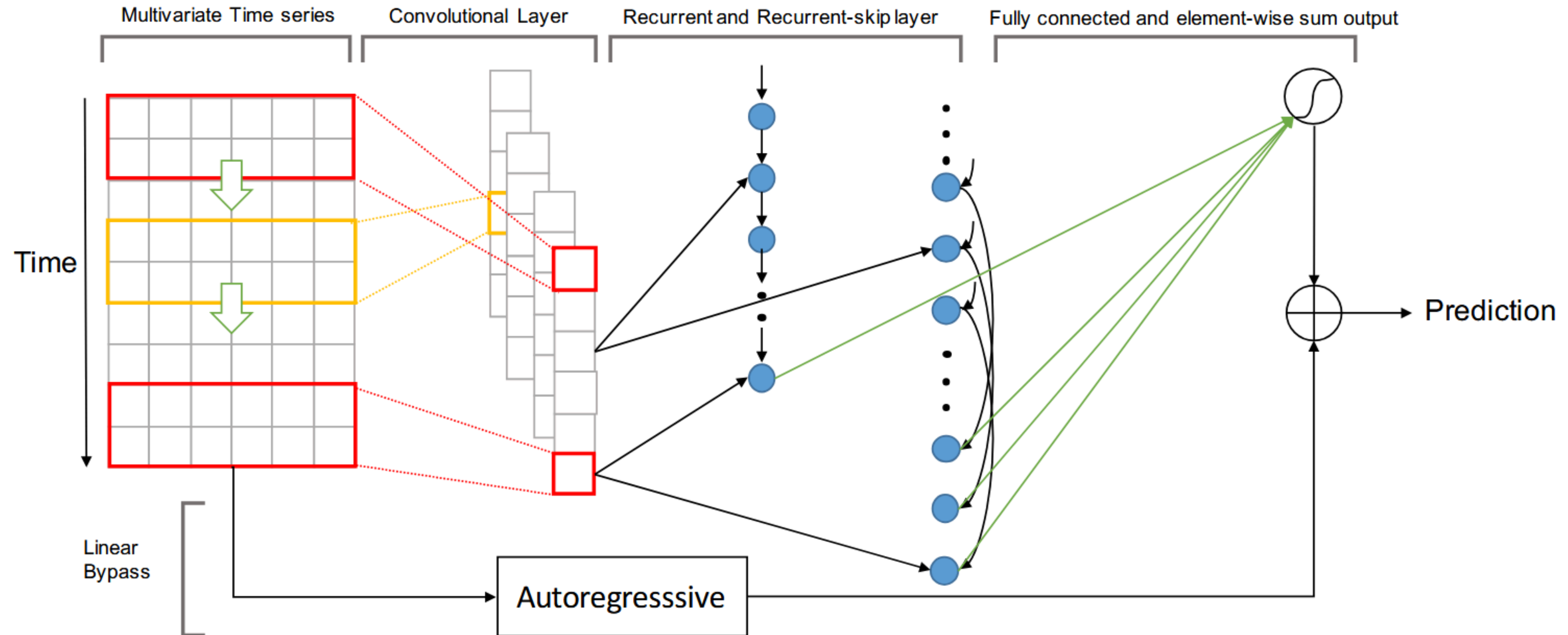
**Window:**  $24 * 7$

**Chuẩn hóa (Normalize):** Chuẩn hóa từng time series

**Chia dữ liệu:** 6-2-2

# Phương pháp thực hiện

## Kiến trúc tổng quát



# Phương pháp thực hiện

## Kiến trúc tổng quát

$$Y_{\text{pred}} = Y_{\text{Neural}} + Y_{\text{AR}}$$

## Trong đó

$Y_{\text{Neural}}$

- ▶ **Convolutional Layer**
- ▶ **Recurrent Layer & Recurrent-skip**

$Y_{\text{AR}}$

- ▶ **Autoregressive Layer**



# Phương pháp thực hiện

## Convolutional Component

### Vai trò:

- Trích xuất các đặc trưng ngắn hạn
- Nhắm bắt các mối quan hệ cục bộ giữa các biến

Ví dụ: Xe ở đường A đông thì xe ở đường B bên cạnh cũng đông

### Cơ chế hoạt động

- Không dùng pooling
- Trượt các filters qua trục thời gian. Chiều cao filter bằng số lượng biến, chiều rộng là  $w$

**Công thức:**  $h_k = \text{ReLU}(W_k * X + b_k)$

# Phương pháp thực hiện

## Recurrent Component

### Vai trò:

- Nắm bắt sự phụ thuộc theo trình tự thời gian dài hơn so với CNN

### Cơ chế hoạt động

- Sử dụng cổng (gate) để quyết định giữ lại hay quên thông tin từ các bước trước đó

### Công thức:

$$\mathbf{h}_t^R = \text{GRU}(\mathbf{h}_{t-1}^R, \mathbf{c}_t)$$

# Phương pháp thực hiện

## Recurrent-skip Component

**Vấn đề:** Các dữ liệu chuỗi thời gian thường có tính chu kỳ (periodicity/seasonality)

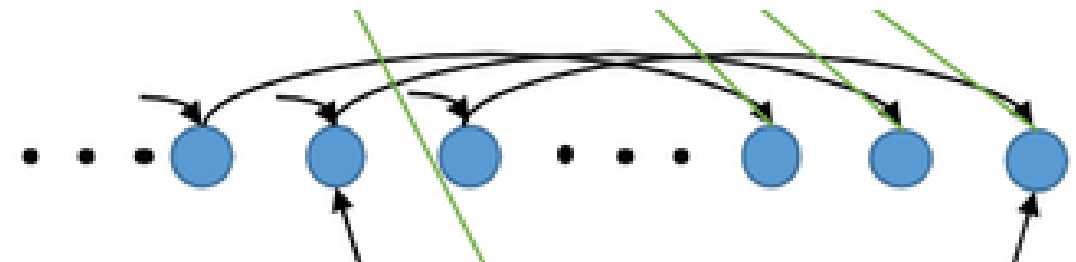
- Ví dụ: Giao thông giờ cao điểm ngày hôm nay sẽ giống giờ cao điểm hôm qua (~24 giờ trước).

RNN thường phải đi qua 24 bước mới nhớ lại được

=> **Thông tin bị phai nhạt**

## Giải pháp (Recurrent-skip):

- Tạo ra các kết nối bước nhảy giữa các hidden state hiện tại và hidden state ở quá khứ **cách p bước**
- Với tham số p là số bước nhảy



**Công thức:** 
$$\mathbf{h}_t^R = \text{GRU}(\mathbf{h}_{t-p}^R, \mathbf{c}_t)$$

# Phương pháp thực hiện

## Autoregressive Component

**Vấn đề:** các mạng neuron rất giỏi trong việc dự đoán phi tuyến tính nhưng lại dở khi dự đoán về thay đổi quy mô

- Ví dụ: Giá cổ phiếu tăng từ 100 lên 2000 trong 1 giờ

=> Các mạng neuron thường dự đoán sai biên độ này

## Giải pháp (AR):

- Sử dụng mô hình tuyến tính để hỗ trợ

## Cơ chế:

- Nhìn vào input gate và thực hiện phép nhân trọng số đơn giản

**Công thức:** 
$$\mathbf{h}_{t,i}^{AR} = \sum_{k=0}^{q-1} \mathbf{W}_{k,i}^{AR} \mathbf{y}_{t-k,i} + \mathbf{b}_i^{AR}$$

# Phương pháp thực hiện

**Tham số cài đặt**

**Môi trường:**



Python 3.6.8



Tensorflow 1.11.0



Keras 2.1.6-tf



# Phương pháp thực hiện

## Siêu tham số

Window	24*7	
CNNFilters	100	Số lượng filter ở lớp CNN
CNNKernel	6	Kích cỡ filter
GRUUnits	100	Số lượng hidden state
SkipGRUUnits	5	Số lượng hidden state ở skip-GRU
Skip	24	Bước nhảy
Dropout	2	
Batchsize	128	
Epoch	20 (100)	
Initializer	glorot_uniform	
Data split	6-2-2	
Loss_Function	mean_absolute_error	
Learning_Rate	1	
Optimizer	Adam	

# Đánh giá kết quả

- Root Relative Squared Error (RSE):

$$RSE = \frac{\sqrt{\sum_{(i,t) \in \Omega_{Test}} (Y_{it} - \hat{Y}_{it})^2}}{\sqrt{\sum_{(i,t) \in \Omega_{Test}} (Y_{it} - \text{mean}(Y))^2}}$$

- Empirical Correlation Coefficient (CORR)

$$CORR = \frac{1}{n} \sum_{i=1}^n \frac{\sum_t (Y_{it} - \text{mean}(Y_i)) (\hat{Y}_{it} - \text{mean}(\hat{Y}_i))}{\sqrt{\sum_t (Y_{it} - \text{mean}(Y_i))^2 (\hat{Y}_{it} - \text{mean}(\hat{Y}_i))^2}}$$

# Đánh giá kết quả

## LSTNet

	Solar				Traffic			
RSE	0.1981	0.2526	0.3447	0.3049	0.4321	0.6134	0.4537	0.5059
CORR	0.9808	0.9689	0.9423	0.9547	0.8827	0.7611	0.8684	0.8391
RSE	0.1843	0.2559	0.3254	0.4643	0.4777	0.4893	0.495	0.4973
CORR	0.9843	0.969	0.9467	0.887	0.8721	0.869	0.8614	0.8588

## Nhóm 9

# Đánh giá kết quả

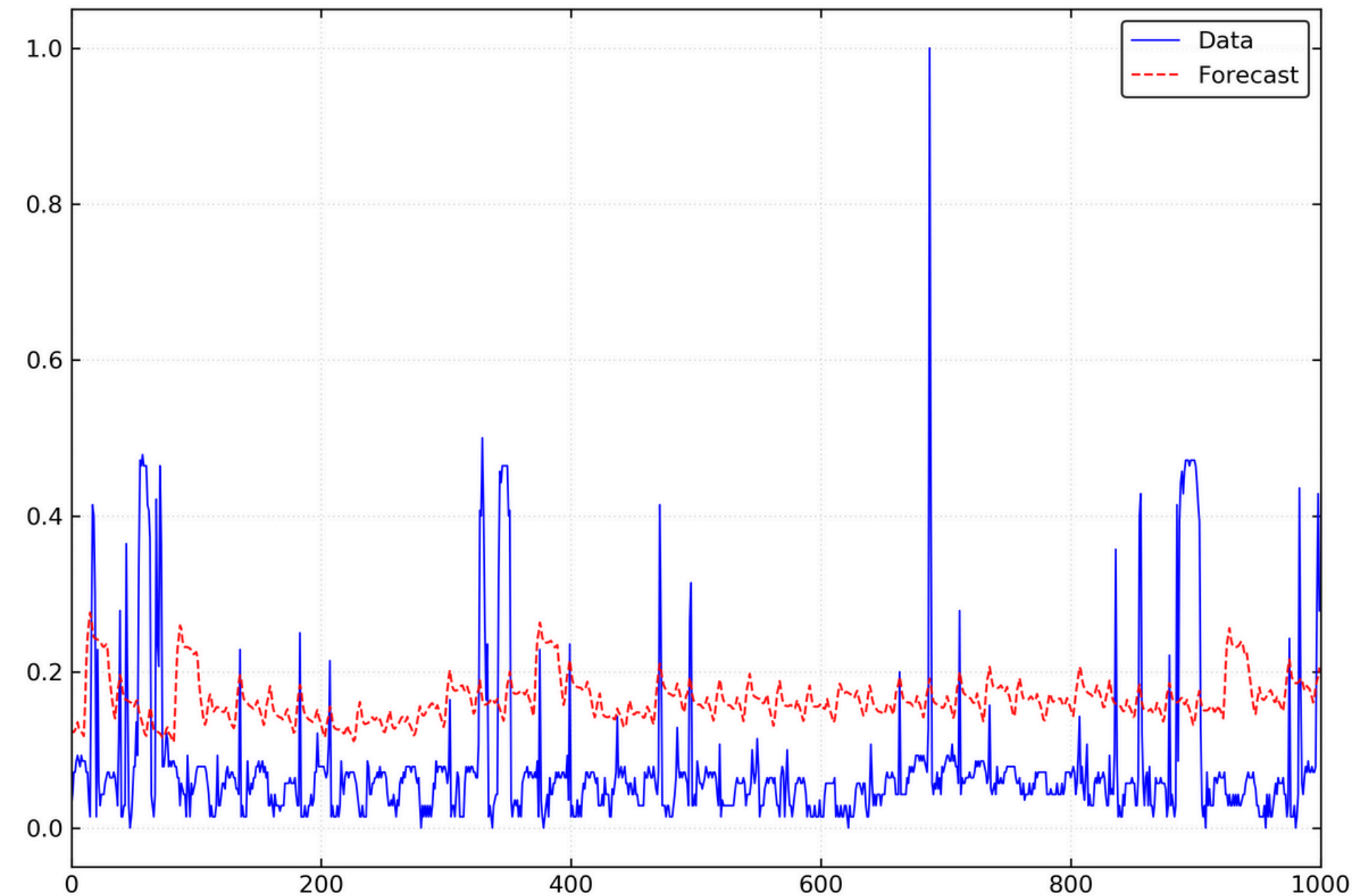
	Electricity				Exchange			
<b>RSE</b>	0.2661	0.2715	0.3757	0.3745	0.1334	0.1082	0.1562	0.1744
<b>CORR</b>	0.9375	0.9347	0.8855	0.8855	0.9807	0.9865	0.9722	0.966
<b>RSE</b>	0.0864	0.0931	0.1007	0.1007	0.0226	0.028	0.0356	0.0449
<b>CORR</b>	0.9283	0.9135	0.9077	0.9119	0.9735	0.9658	0.9511	0.9354

CORR đã tiệm cận kết quả của bài báo chứng tỏ mô hình bắt đúng xu hướng nhưng RSE vẫn cách xa

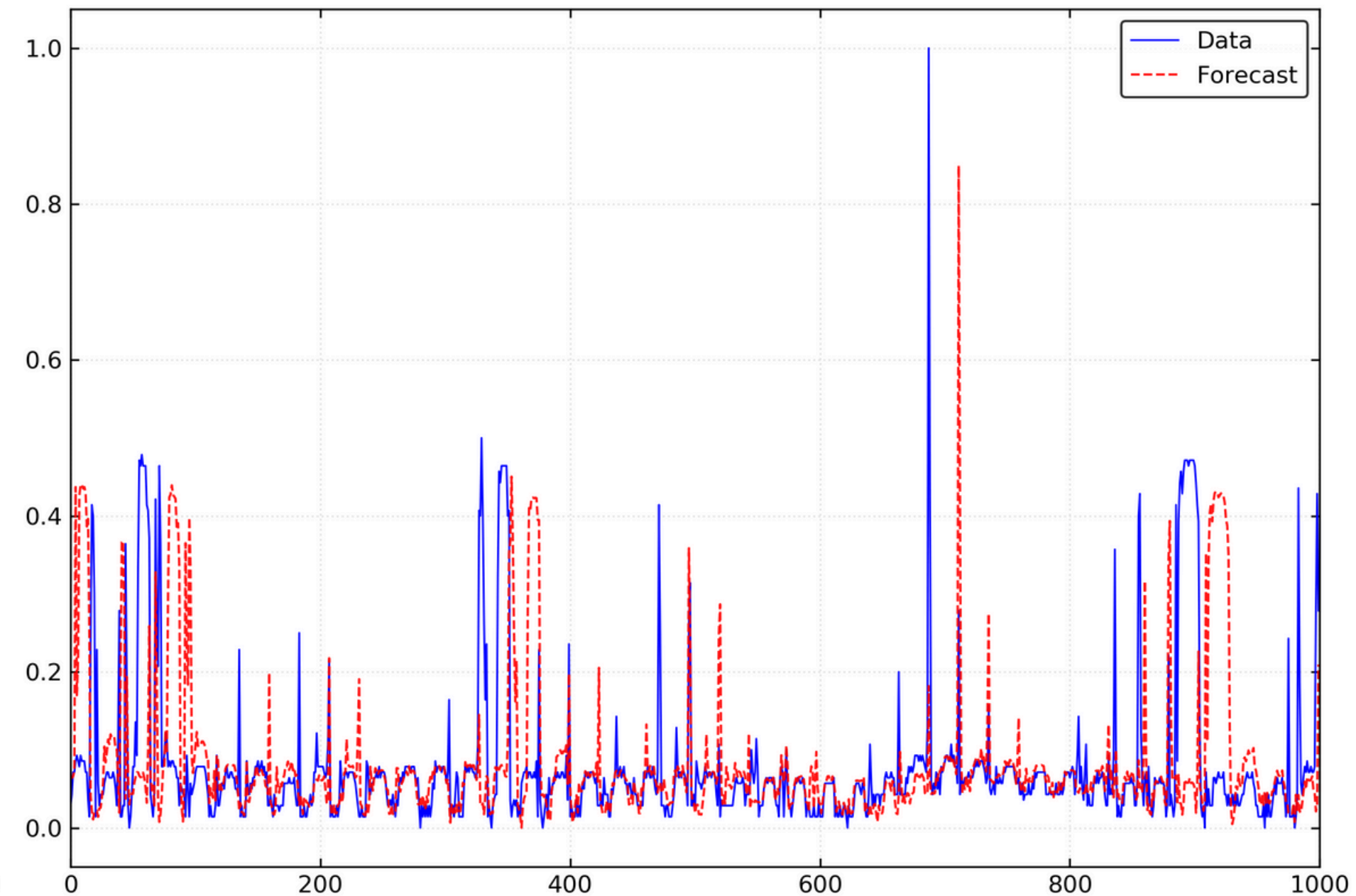
Điều này cho thấy mô hình ưu tiên học đặc trưng hình dạng (temporal pattern) trước khi tối ưu hóa sai số tuyệt đối

Epoch 25 => thành phần AR chưa kịp hội tụ hoàn toàn để cân chỉnh lại scale của dữ liệu

# Đánh giá kết quả



**(a) LSTNet w/o AR**



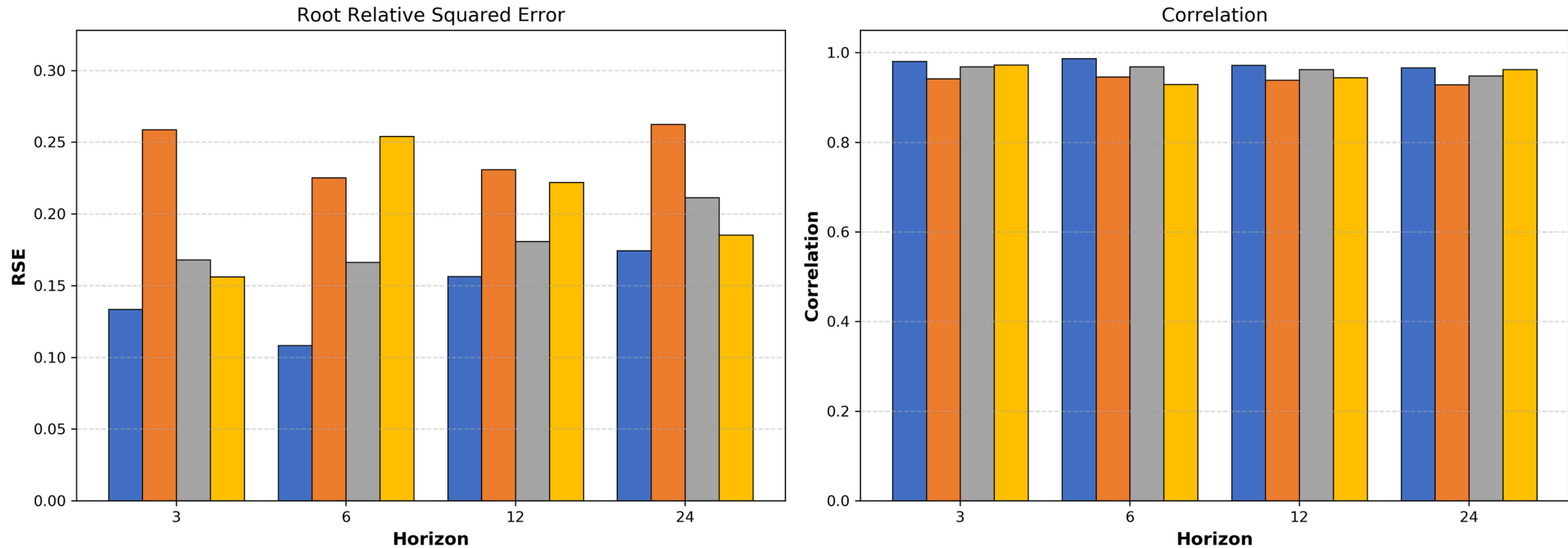
**(b) LSTNet Full**

Có thể thấy thành phần AR đặc biệt quan trọng



# exchange\_rate.txt

LSTNet LST-AR LST-Skip LST-CNN



Với dữ liệu chuỗi thời gian có sự biến thiên lớn về quy mô (scale), các mô hình Deep Learning phức tạp vẫn cần sự hỗ trợ của các phương pháp thống kê cổ điển (như AR) để đạt được hiệu suất dự báo cao nhất và ổn định nhất.



Nhóm 9

**THANK YOU  
FOR LISTENING**