深度神经网络模型：对长期和短期时间模式

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

Guokun Lai
Carnegie Mellon University
guokun@cs.cmu.edu

Wei-Cheng Chang
Carnegie Mellon University
wchang2@andrew.cmu.edu

Yiming Yang
Carnegie Mellon University
yiming@cs.cmu.edu

Hanxiao Liu
Carnegie Mellon University
hanxiaol@cs.cmu.edu

The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval

Introduction

多变量时间序列预测往往面临着一个重要的研究挑战，即如何捕捉和利用多变量之间的动态依赖关系。

现实世界中的时序数据通常存在时间上的依赖，即存在短期和长期的重复模式，如高速公路上的车流量以天和周为重复周期。

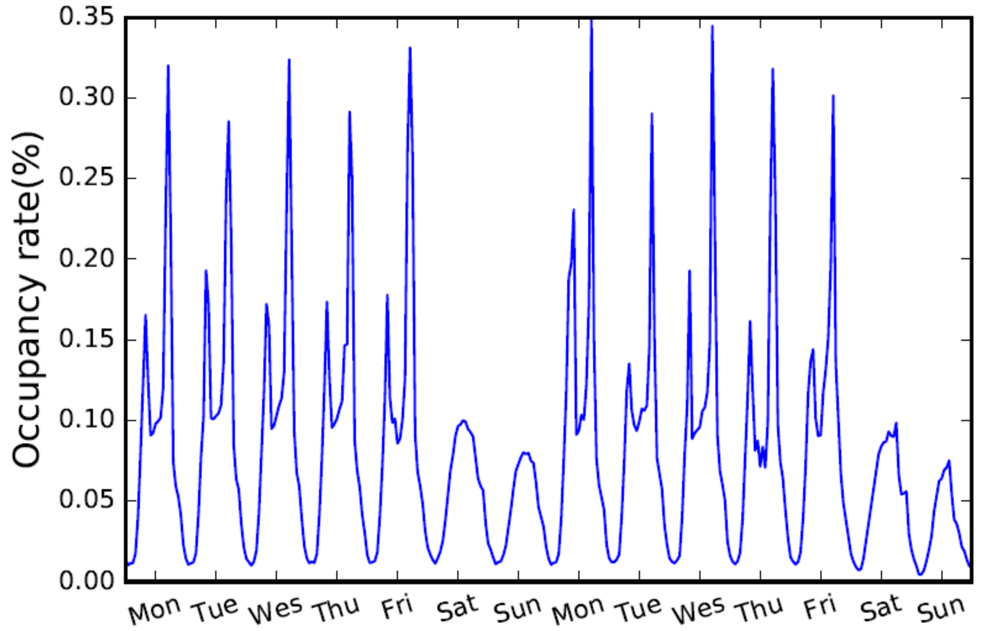


Figure 1: The hourly occupancy rate of a road in the bay area for 2 weeks

Introduction

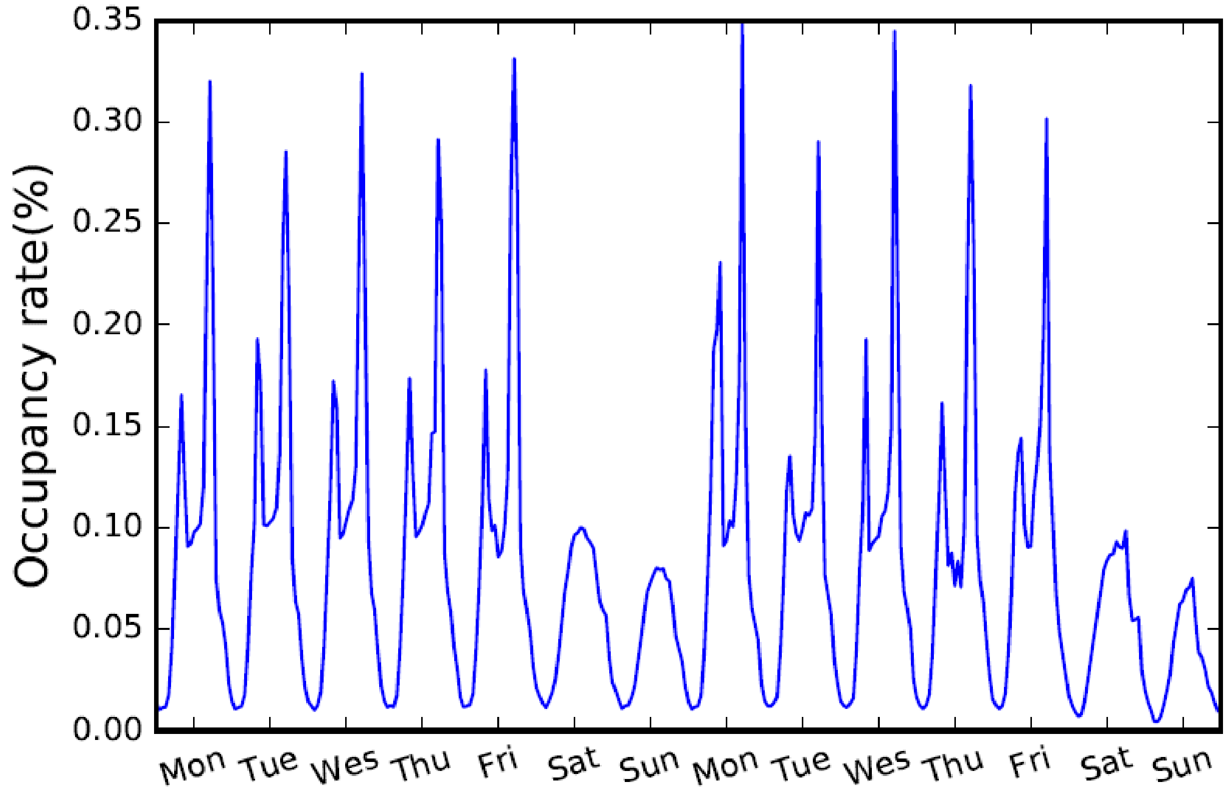


Figure 1: The hourly occupancy rate of a road in the bay area for 2 weeks

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

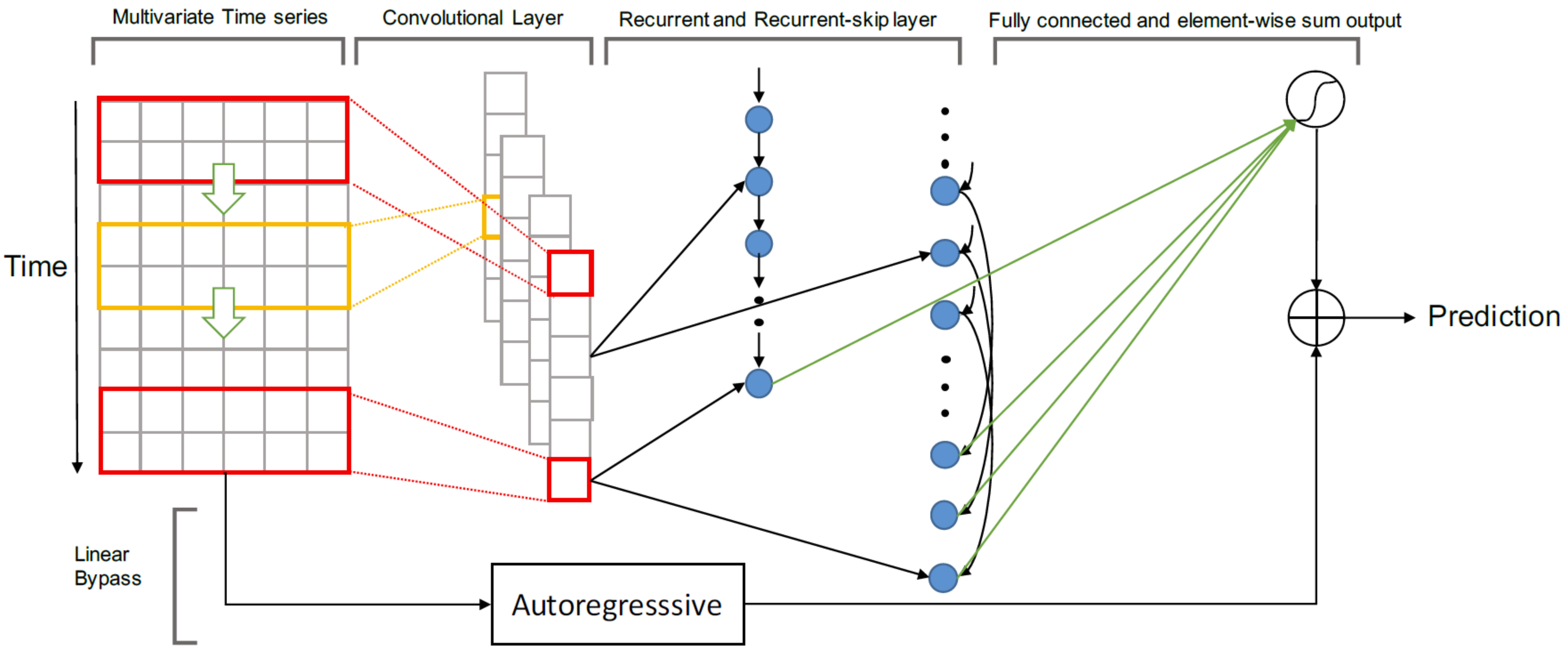
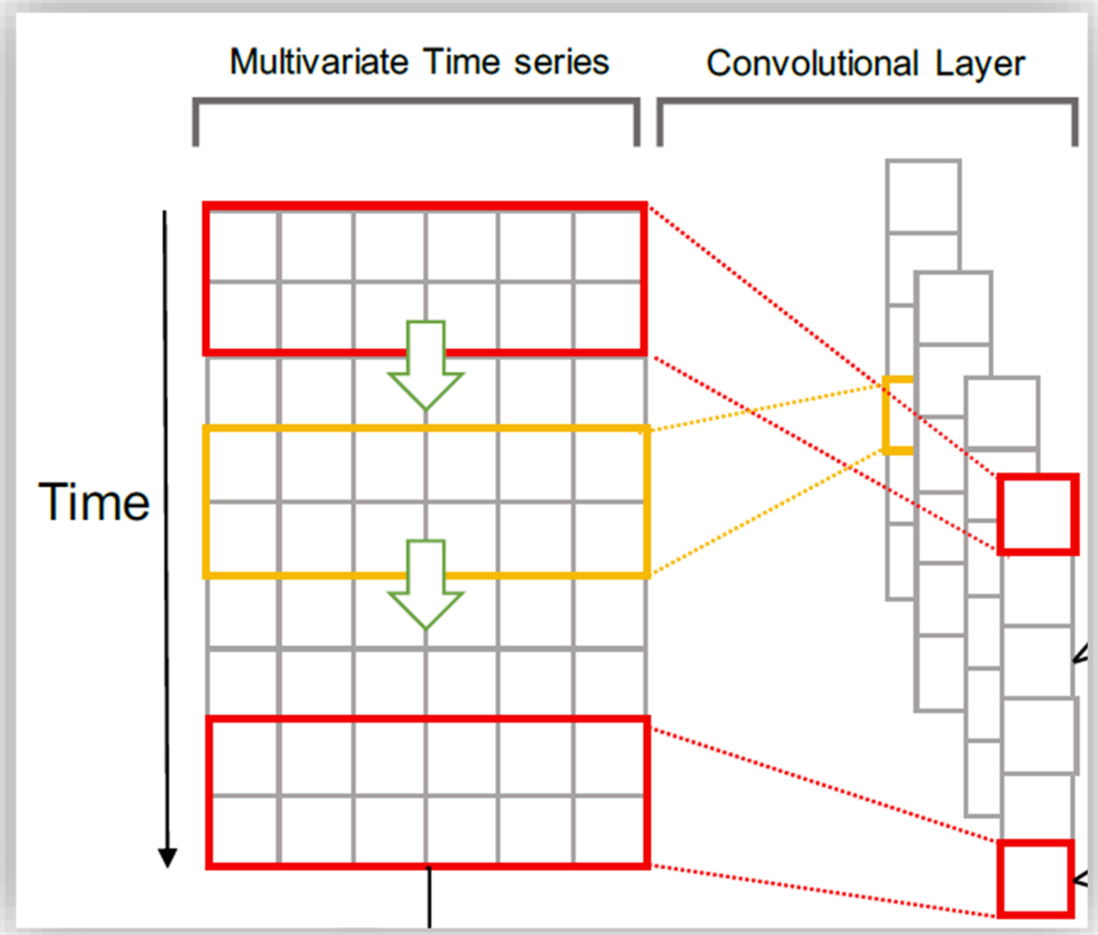


Figure 2: An overview of the Long- and Short-term Time-series network (LSTNet)

Convolutional Component



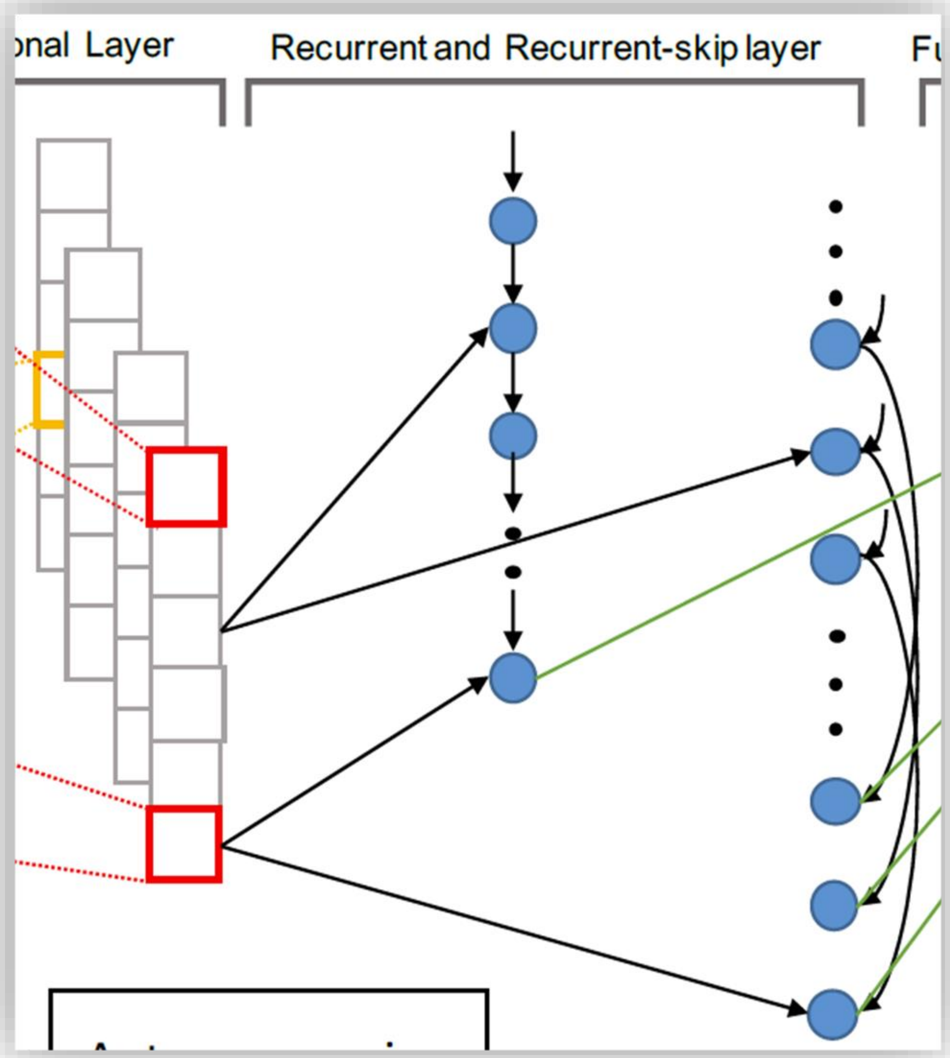
LSTNet 的第一层是一个没有池化的卷积网络，旨在提取时间维度上的短期模式以及变量之间的局部依赖关系。

第 k 个滤波器扫过输入矩阵 X 并产生 h_k

$$h_k = RELU(W_k * X + b_k)$$

其中 $*$ 表示卷积运算，输出向量h_k

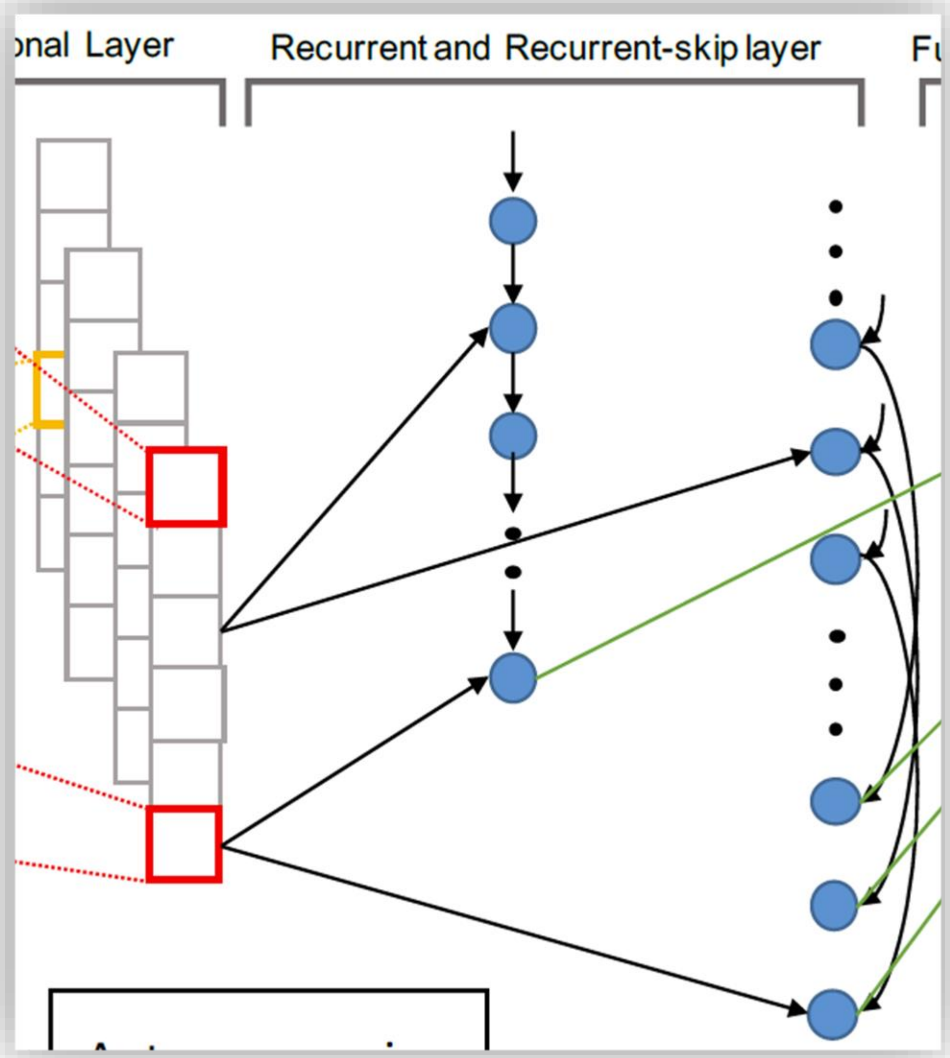
Recurrent Component



循环分量是门控递归单元 (GRU) 的递归层，使用 RELU 函数作为隐式更新激活函数。t 时刻递归单元的隐藏状态计算为：

$$
\begin{aligned}
r_t &= \sigma(x_t W_{xr} + h_{t-1} W_{hr} + b_r) \\
u_t &= \sigma(x_t W_{xu} + h_{t-1} W_{hu} + b_u) \\
c_t &= RELU(x_t W_{xc} + r_t \odot (h_{t-1} W_{hc}) + b_c) \\
h_t &= (1 - u_t) \odot h_{t-1} + u_t \odot c_t
\end{aligned}
$$

Recurrent-skip Component

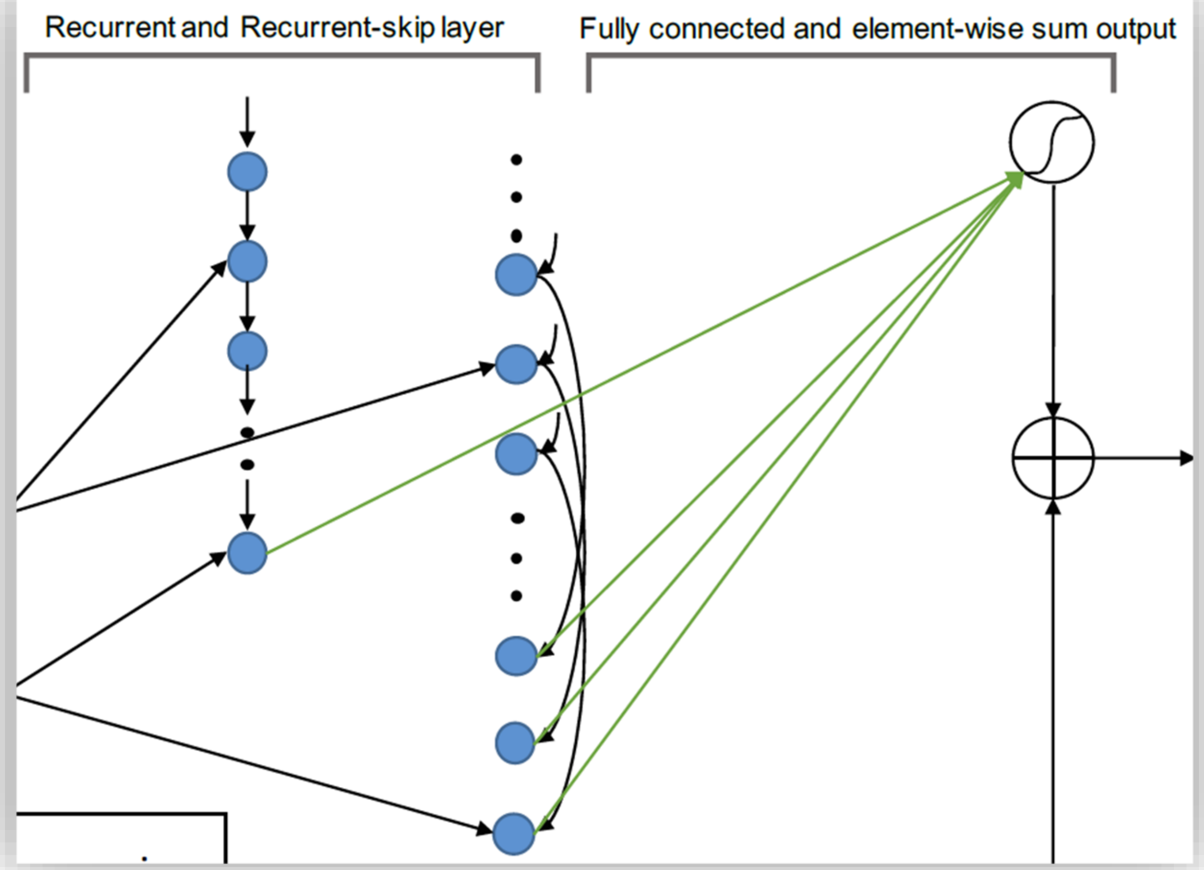


由于梯度消失问题，GRU和LSTM在实际应用中往往不能捕捉到非常长期(very long-term)的相关性。

于是开发了一个具有时间跳跃连接的递归结构，具体地说，在当前隐藏单元与相邻周期中处于同一阶段的隐藏单元之间添加 *skip-links*。更新过程可以表示为:

$$
\begin{aligned}
r_t &= \sigma(x_t W_{xr} + h_{t-p} W_{hr} + b_r) \\
u_t &= \sigma(x_t W_{xu} + h_{t-p} W_{hu} + b_u) \\
c_t &= RELU(x_t W_{xc} + r_t \odot (h_{t-p} W_{hc}) + b_c) \\
h_t &= (1 - u_t) \odot h_{t-p} + u_t \odot c_t
\end{aligned}
$$

Combine the Outputs



使用一个 Dense 层来组合 Recurrent 层和 Recurrent-skip 层的输出：

$$h_t^D = W^R h_t^R + \sum_{i=0}^{p-1} W_i^S h_{t-i}^S + b$$

Temporal Attention Layer

上述的 *Recurrent-skip Component* 中有一个超参，即周期 p 。有些情况下周期未知或周期在动态变化，预测是不方便的。为了解决这一问题，本文又提出了一种 *attention* 机制。

$$\boldsymbol{\alpha}_t = AttnScore(H_t^R, h_{t-1}^R)$$

$$H_t^R = [h_{t-q}^R, \ldots, h_{t-1}^R]$$

计算 $t-q$ 到 $t-1$ 时刻所有隐向量的权重，得到带权的上下文向量 c_t ，和 $t-1$ 时刻的隐向量拼接后做线性投影得到 t 时刻的隐向量。

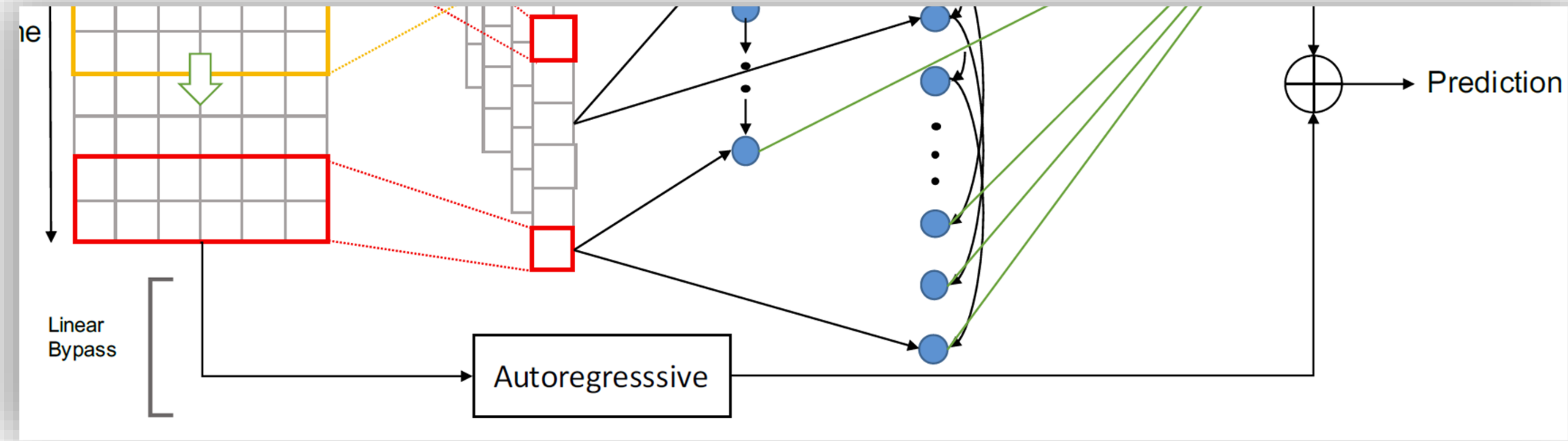
$$\mathbf{c}_t = H_t \boldsymbol{\alpha}_t \qquad h_t^D = W[\mathbf{c}_t; h_{t-1}^R] + b$$

Autoregressive Component

神经网络模型的一个缺点是输出尺度对输入尺度的变化不敏感，而在具体的实际数据集中，输入信号的尺度不断发生非周期的变化，显著降低了神经网络模型的预测精度。

自回归模型是一个很好的线性模型来建模局部的尺度变化问题。

将LSTNet的最终预测分解为线性部分(主要关注局部尺度问题)和包含重复模式的非线性部分。



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

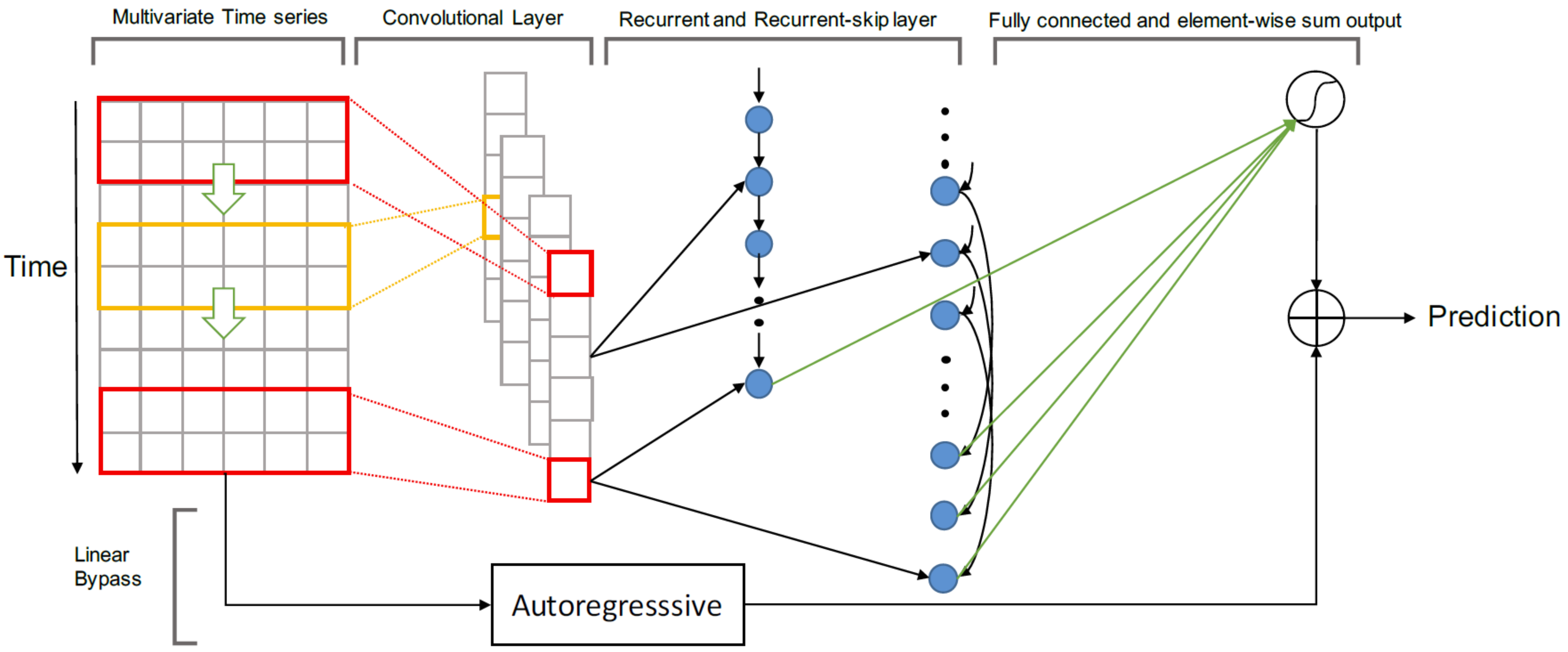
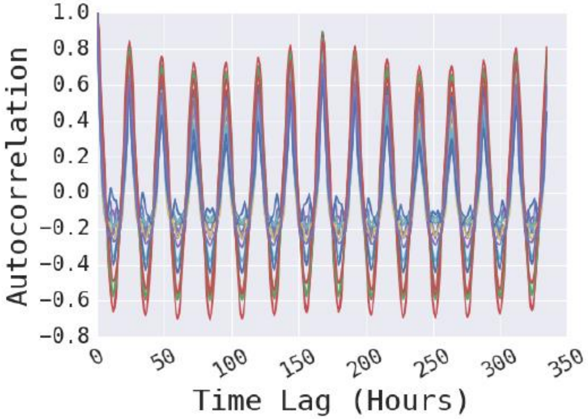
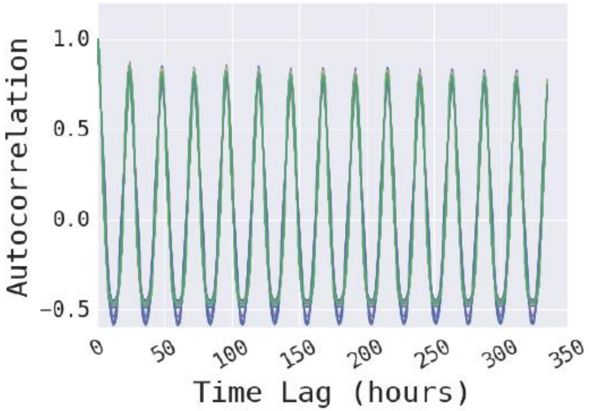


Figure 2: An overview of the Long- and Short-term Time-series network (LSTNet)

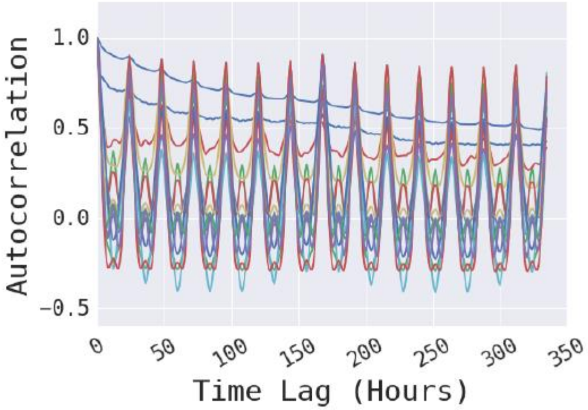
Evaluation



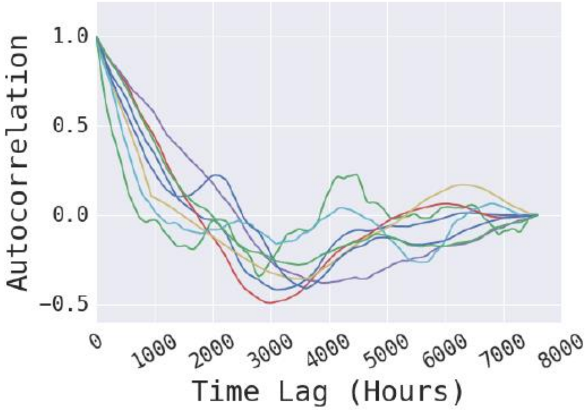
(a) Traffic **dataset**



(b) Solar-Energy **dataset**



(c) Electricity **dataset**



(d) Exchange-Rate **dataset**

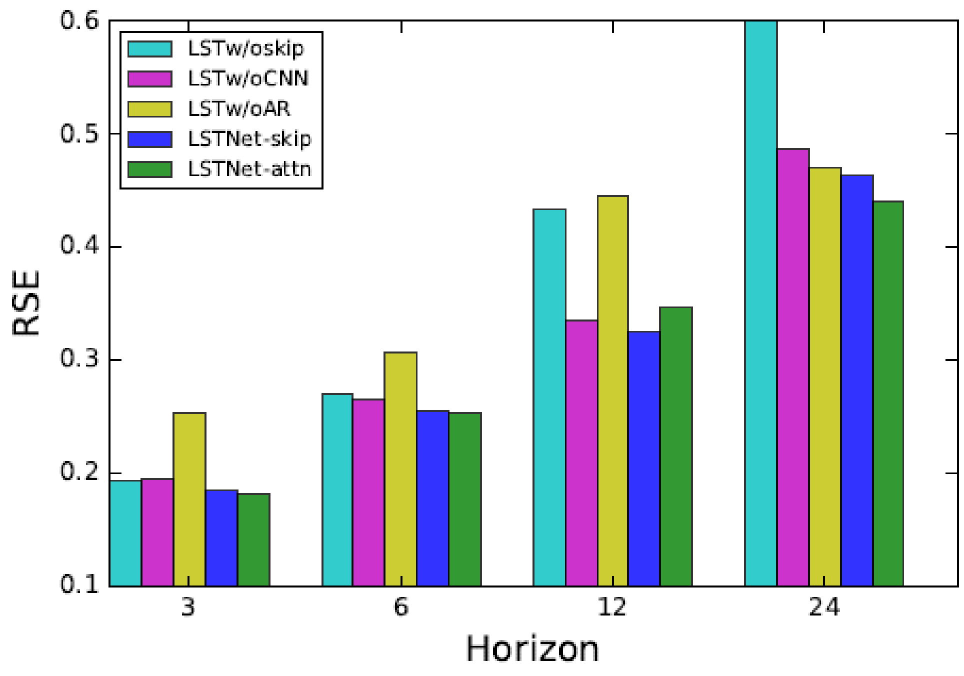
Evaluation

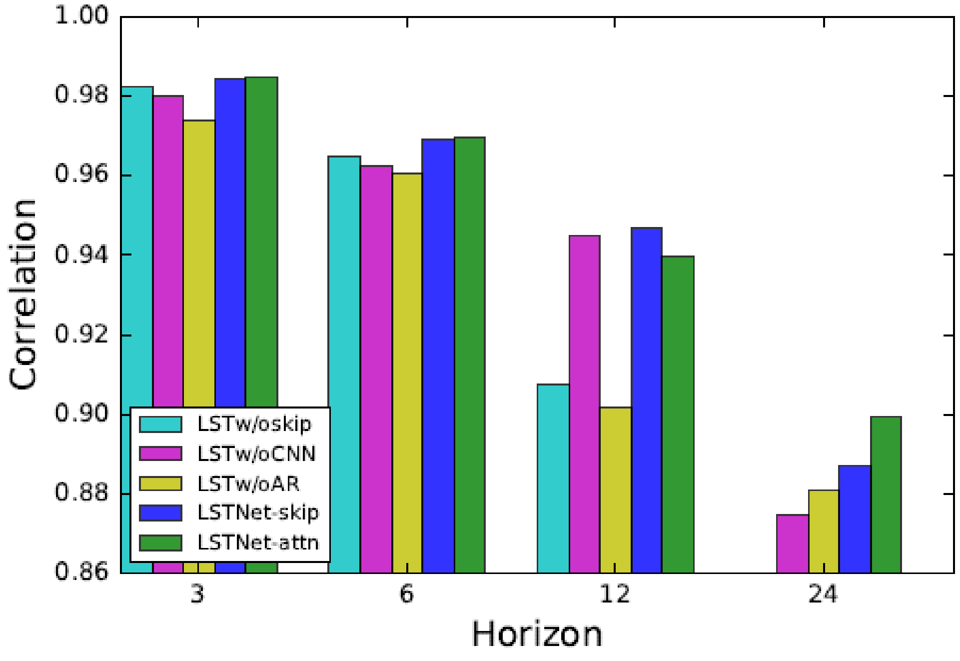
Dataset		Solar-Energy				Traffic				Electricity				Exchange-Rate			
		Horizon				Horizon				Horizon				Horizon			
Methods	Metrics	3	6	12	24	3	6	12	24	3	6	12	24	3	6	12	24
AR	RSE	0.2435	0.3790	0.5911	0.8699	0.5991	0.6218	0.6252	0.6293	0.0995	0.1035	0.1050	0.1054	0.0228	0.0279	**0.0353**	**0.0445**
(3)	CORR	0.9710	0.9263	0.8107	0.5314	0.7752	0.7568	0.7544	0.7519	0.8845	0.8632	0.8591	0.8595	0.9734	0.9656	0.9526	**0.9357**
LRidge	RSE	0.2019	0.2954	0.4832	0.7287	0.5833	0.5920	0.6148	0.6025	0.1467	0.1419	0.2129	0.1280	**0.0184**	0.0274	0.0419	0.0675
(3)	CORR	0.9807	0.9568	0.8765	0.6803	0.8038	0.8051	0.7879	0.7862	0.8890	0.8594	0.8003	0.8806	**0.9788**	**0.9722**	0.9543	0.9305
LSVR	RSE	0.2021	0.2999	0.4846	0.7300	0.5740	0.6580	0.7714	0.5909	0.1523	0.1372	0.1333	0.1180	0.0189	0.0284	0.0425	0.0662
(1)	CORR	0.9807	0.9562	0.8764	0.6789	0.7993	0.7267	0.6711	0.7850	0.8888	0.8861	0.8961	0.8891	0.9782	0.9697	**0.9546**	0.9370
TRMF	RSE	0.2473	0.3470	0.5597	0.9005	0.6708	0.6261	0.5956	0.6442	0.1802	0.2039	0.2186	0.3656	0.0351	0.0875	0.0494	0.0563
(0)	CORR	0.9703	0.9418	0.8475	0.5598	0.6964	0.7430	0.7748	0.7278	0.8538	0.8424	0.8304	0.7471	0.9142	0.8123	0.8993	0.8678
GP	RSE	0.2259	0.3286	0.5200	0.7973	0.6082	0.6772	0.6406	0.5995	0.1500	0.1907	0.1621	0.1273	0.0239	**0.0272**	0.0394	0.0580
(1)	CORR	0.9751	0.9448	0.8518	0.5971	0.7831	0.7406	0.7671	0.7909	0.8670	0.8334	0.8394	0.8818	0.8713	0.8193	0.8484	0.8278
VARMLP	RSE	0.1922	0.2679	0.4244	0.6841	0.5582	0.6579	0.6023	0.6146	0.1393	0.1620	0.1557	0.1274	0.0265	0.0304	0.0407	0.0578
(0)	CORR	0.9829	0.9655	0.9058	0.7149	0.8245	0.7695	0.7929	0.7891	0.8708	0.8389	0.8192	0.8679	0.8609	0.8725	0.8280	0.7675
RNN-GRU	RSE	0.1932	0.2628	0.4163	0.4852	0.5358	0.5522	0.5562	0.5633	0.1102	0.1144	0.1183	0.1295	0.0192	0.0264	0.0408	0.0626
(0)	CORR	0.9823	0.9675	0.9150	0.8823	0.8511	0.8405	0.8345	0.8300	0.8597	0.8623	0.8472	0.8651	0.9786	0.9712	0.9531	0.9223
LST-Skip	RSE	0.1843	0.2559	**0.3254**	0.4643	**0.4777**	**0.4893**	**0.4950**	**0.4973**	**0.0864**	**0.0931**	0.1007	**0.1007**	0.0226	0.0280	0.0356	0.0449
(17)	CORR	0.9843	0.9690	**0.9467**	0.8870	**0.8721**	**0.8690**	**0.8614**	**0.8588**	**0.9283**	**0.9135**	**0.9077**	**0.9119**	0.9735	0.9658	0.9511	0.9354
LST-Attn	RSE	**0.1816**	**0.2538**	0.3466	**0.4403**	0.4897	0.4973	0.5173	0.5300	0.0868	0.0953	**0.0984**	0.1059	0.0276	0.0321	0.0448	0.0590
(7)	CORR	**0.9848**	**0.9696**	0.9397	**0.8995**	0.8704	0.8669	0.8540	0.8429	0.9243	0.9095	0.9030	0.9025	0.9717	0.9656	0.9499	0.9339

Ablation Study

为了证明我们的框架设计的效率，进行了仔细的消融研究。

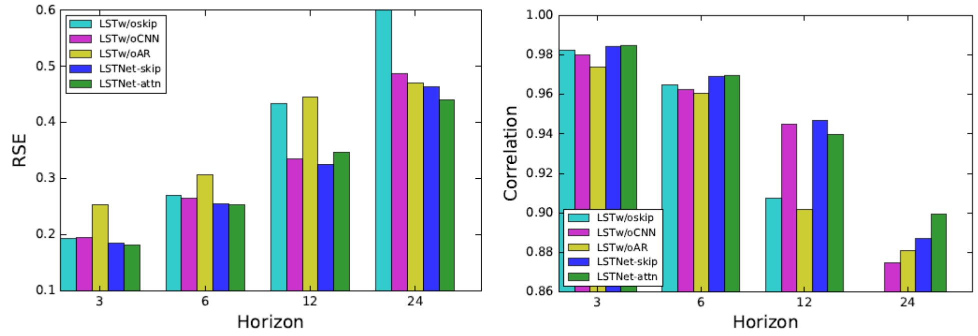
具体来说，在 LSTNet 框架中一次删除一个组件，对比不同情况下的结果。



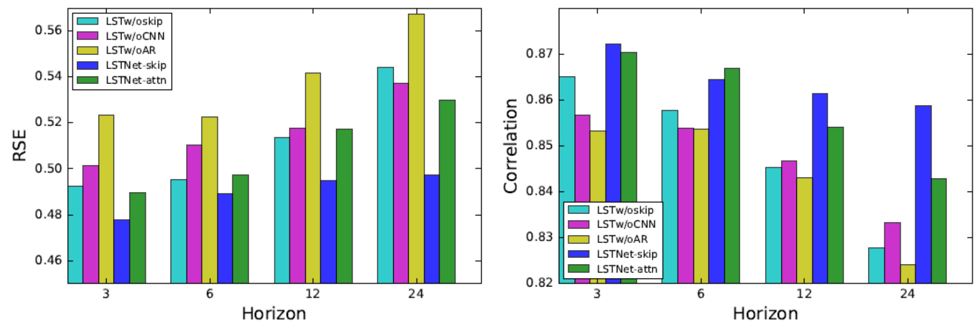


(a) Solar-Energy dataset

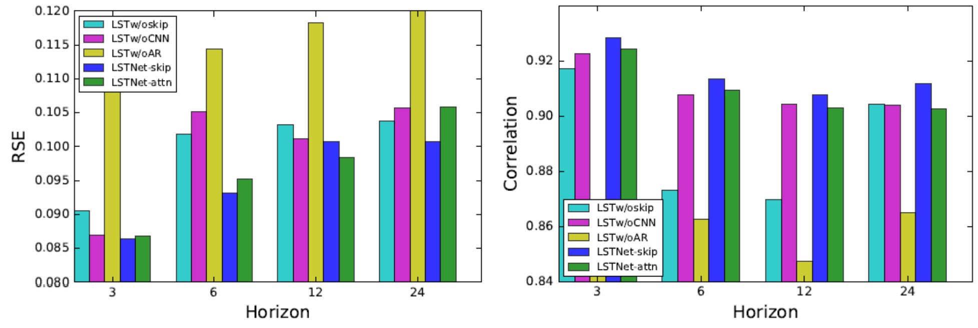
Ablation Study



(a) Solar-Energy dataset



(b) Traffic dataset



(c) Electricity dataset

- 每个数据集的最佳结果都是 LST-Skip 或 LST-Attn 中的一个；
- 从完整模型中删除自回归部分，导致大多数数据集的性能下降最显着，这显示了AR组件的关键作用。
- 删除 Skip 或 CNN 组件会导致某些数据集的性能下降，但不是全部。

LSTNet的所有部分共同作用使得我们的方法在所有数据集上有强大性能。

Ablation Study

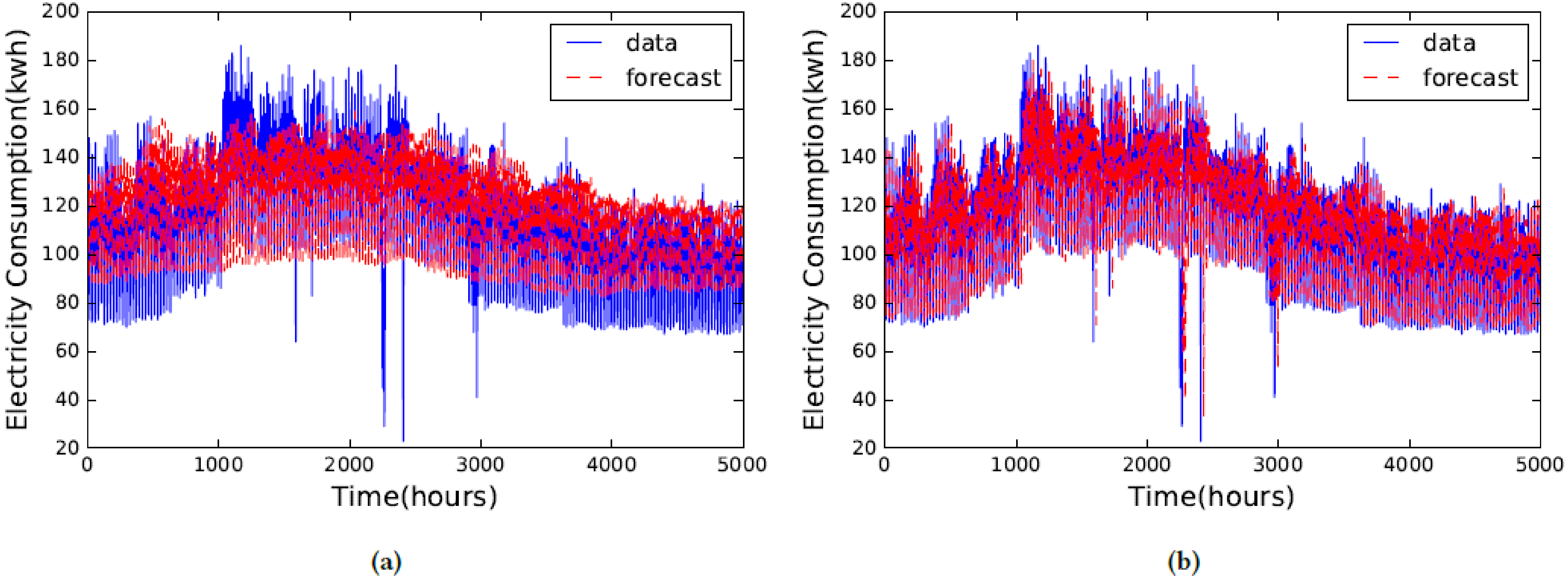


Figure 6: The predicted time series (red) by LSTw/oAR (a) and by LST-Skip (b) vs. the true data (blue) on Electricity dataset with $horizon = 24$

Mixture of long- and short-term patterns

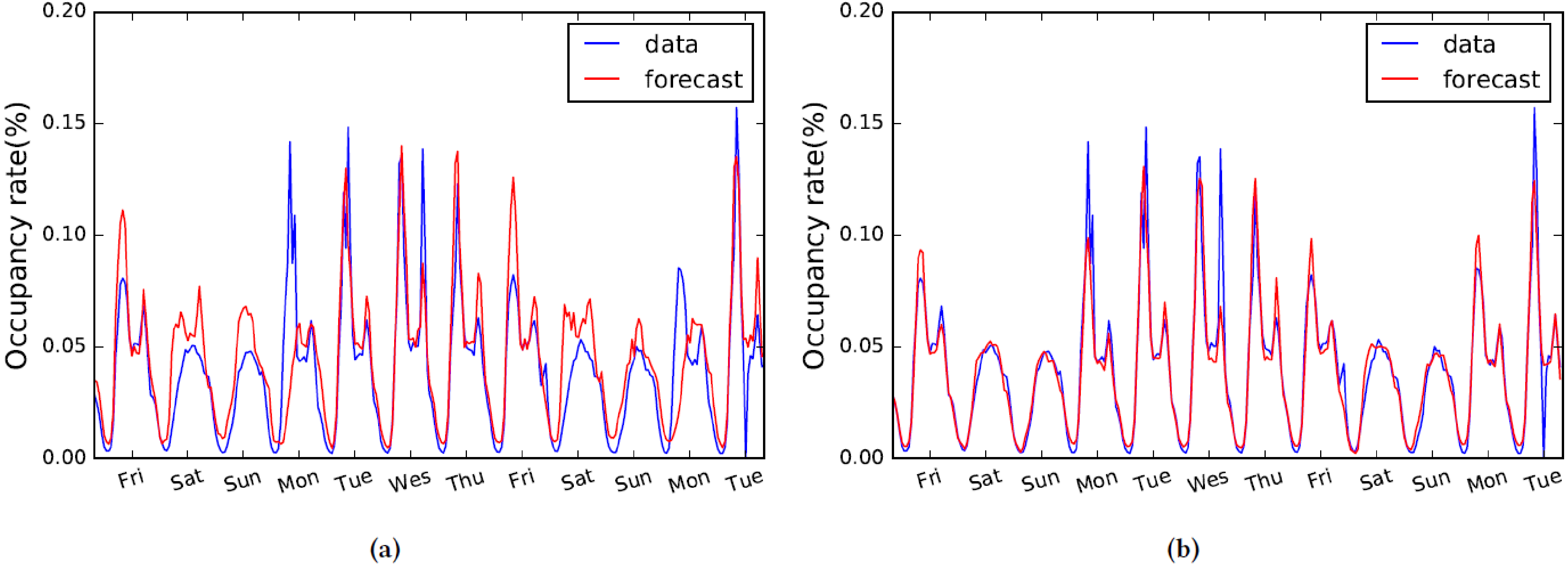


Figure 7: The true time series (blue) and the predicted ones (red) by VAR (a) and by LSTNet (b) for one variable in the Traffic **occupation dataset. The X axis indicates the week days and the forecasting** *horizon* = 24**. VAR inadequately predicts similar patterns for Fridays and Saturdays, and ones for Sundays and Mondays, while LSTNet successfully captures both the daily and weekly repeating patterns.**

TO-DO

Temporal Pattern Attention for Multivariate Time Series Forecasting

**Shun-Yao Shih* · Fan-Keng Sun* ·
Hung-yi Lee**

Received: date / Accepted: date

Abstract Forecasting of multivariate time series data, for instance the prediction of electricity consumption, solar power production, and polyphonic piano pieces, has numerous valuable applications. However, complex and non-linear interdependencies between time steps and series complicate this task. To obtain accurate prediction, it is crucial to model long-term dependency in time series data, which can be achieved by recurrent neural networks (RNNs) with an attention mechanism. The typical attention mechanism reviews the information at each previous time step and selects relevant information to help generate the outputs; however, it fails to capture temporal patterns across multiple time steps. In this paper, we propose using a set of filters to extract time-invariant temporal patterns, similar to transforming time series data into its "frequency domain". Then we propose a novel attention mechanism to select relevant time series, and use its frequency domain information for multivariate forecasting. We apply the proposed model on several real-world tasks and achieve state-of-the-art performance in all of these with a single exception. Our source code is available at https://github.com/gantheory/TPA-LSTM.

THANKS