Rolling Forecasting Forward by Boosting Heterogeneous Kernels

Di Zhang[1(✉)], Yunquan Zhang[1,2], Qiang Niu[3], and Xingbao Qiu[4]

[1] School of Computer Science, Communication University of China, Beijing 100024, People's Republic of China
di.zhang@cuc.edu.cn

[2] State Key Lab of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, People's Republic of China
zyq@ict.ac.cn

[3] Department of Mathematical Sciences, Xi'an Jiaotong-Liverpool University, Suzhou 215123, People's Republic of China
qiang.niu@xjtlu.edu.cn

[4] China Mobile Communications Corporation, Beijing 100032, People's Republic of China
qiuxingbao@sn.chinamobile.com

Abstract. The problem discussed in this paper stems from a project of cellular network traffic prediction, the primary step of network planning striving to serve the continuously soaring network traffic best with limited resource. The traffic prediction emphasizes two aspects: (1) how to exploit the potential value of physical and electronic properties for tens of thousands of wireless stations, which may partly determine the allocation of traffic load in some intricate way; (2) the lack of sufficient and high-quality historical records, for the appropriate training of long-term predictions, further aggravated by frequent reconfigurations in daily operation. To solve this problem, we define a general framework to accommodate several variants of multi-step forecasting, via decomposing the problem into a series of single-step vector-output regression tasks. They can further be augmented by miscellaneous attributive information, in the form of boosted multiple kernels. Experiments on multiple telecom datasets show that the solution outperforms conventional time series methods on accuracy, especially for long horizons. Those attributes

The work is partially supported by National Key R&D Program of China under Grant No. 2016YFB0200803, 2017YFB0202302, 2017YFB0202001, 2017YFB0202502, 2017YFB0202105; the National Natural Science Foundation of China under Grant No.61432018, No. 61521092, No. 61272136, No. 61402441, No. 61502450; the National High Technology Research and Development Program of China under Grant No. 2015AA011505; Key Technology Research and Development Programs of Guangdong Province under Grant No. 2015B010108006; the CAS Interdisciplinary Innovation Team of Efficient Space Weather Forecast Models; NSF of China under no. 11301420; NSF of Jiangsu province under no. BK20150373 and no. BK20171237; Suzhou science and technology program under no. SZS201613 and the XJTLU Key Programme Special Fund (KSF) under no. KSF-A-01.

D. Phung et al. (Eds.): PAKDD 2018, LNAI 10937, pp. 248–260, 2018.
https://doi.org/10.1007/978-3-319-93034-3_20

describing the macroscopic factors, such as the network type, topology, locations, are significantly helpful for longer horizons, whereas the immediate values in the near future are mainly determined by their recent records.

Keywords: Multi-dimensional time series · Multi-horizon prediction
Multi-kernel learning · Network traffic prediction

1 Introduction

Nowadays, mobile communication has become a pivot ingredient for everyone's life and delivers the connectivity and infrastructure powering new digital economies and unleashing novel applications. The telecom operators must conduct network planning and optimization with limited budget at least one year ahead, to support the surging data traffic growing 18-fold for every five years in global [1]. The first step of network construction is to predict how traffic will evolve in a long-term view for tens of thousands of cell towers in the broad scope of a region. The underlying power of pushing the traffic blow up or fluctuate arises in two aspects: one is objective, including natural growth in consumption, the movement of population, and seasonal oscillation. The other is subjective, which means the reaction of user behavior towards the network change. Usually, given the higher bandwidth of a network, the users will intend to spend more time on enjoying the mobile apps, and vice versa.

The problem of traffic forecasting can be initially recognized as the geographical time series prediction, which has been extensively studied for decades, as summarized in [2]. These models include linear model, like ARIMA (Autoregressive Integrated Moving Average), VAR (Vector Autoregression), which have the advantage of simplicity and robust, and non-linear model, such as neural network and its deep learning variants [6], which sometimes can provide higher accuracy, but also need more training data and more computing resource.

We have adopted and tested these popular solutions to our problem, and pay attention to its specific difficulties: (1) long-term prediction of noisy data. The interested target in practice is the peak load of every day, not the summation or average, which results in the forecasting be much noisier. As the predicting horizon moves forward, the inherent noise of time series will accumulate with the increasing variation, and the possible bias will be amplified, which leads to the rapid deterioration of accuracy. (2) high dimensionality and scarceness of data. The number of cell towers serving as access points in a metropolis can reach up to more than ten thousand, which make the classic VAR vulnerable by the curse of dimensionality. Meanwhile, the traffic volumes are also determined by the layout and configuration of the network itself, and the model may benefit from these properties at inferring the underlying manifolds where series evolves, though some of them may be redundant and irrelevant. Though we can give an explicit explanation for every attribute as they are defined in a humanmade system, it is impossible to figure out, which part of them and how they influence

the complex dynamics of user behavior and traffic characteristics. It seems that we must resort to some variable selection methods to find out a useful subset of attributes.

In the previous work of wireless network prediction, various popular technologies have been tested on this problem [2]. Two jobs worth to be noticed particularly. The work [9] arranges the historical measurements and values to be predicted (as zeros) into a matrix, which will be factorized based on a compressed sensing approach with spatial constraints. The work [5] makes use of the sensors' location based on CNN (Convolutional Neural Networks) in deep learning, via converting the traffic snapshot into images describing the spatial-temporal relation of traffic flows, and thus automatic feature extraction becomes viable. Although the advantage of these spatial-temporal models is proven, none of above works considers how to make use of miscellaneous properties other than locations, and inspect their potential benefit as forecasting horizons vary.

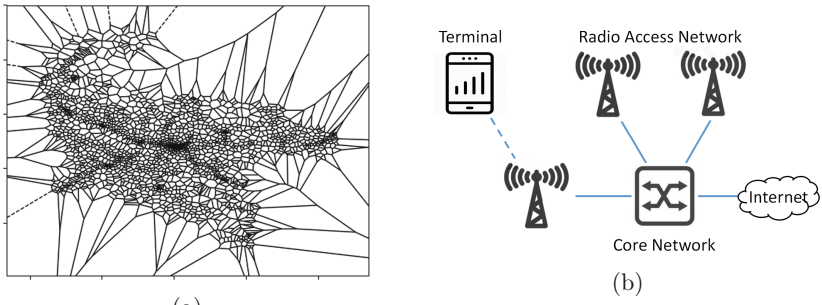
To explore this issue, we clarify several variations of long-term prediction and define a general framework to entangle one-step tasks in a cascading fashion. The properties of each entity are divided into small groups based on the business knowledge, and they are encoded by various kernels; all of these make the function space larger than an original linear regression could reach. The importance of attributes given by model can be used as an important reference for data collectors and system admins. The contributions of this paper include: (1) a customizable solution to transform, select and fuse properties containing context information, and apply them sequentially into a multi-step forecasting; (2) some practical skills are given, including a set of commonly used kernels and how they are combined; (3) effectiveness on multiple telecom sites, compared to commonly-used methods, are validated, and each kernel's contribution to different horizons are analyzed.

The left of paper is organized as follows: in Sects. 2 and 3, we briefly introduce the necessary background knowledge to understand the problem and summarize the works in related domains. Next, the solution is presented in detail, with its formalized model and kernel design. In Sect. 5, a set of experiments are executed and demonstrated with results. Finally, we conclude the whole article.

2 Background: Network Traffic and Device Configuration

The mobile network is a communication network where the last link is wireless. In a range of territory, the base stations are scattered with proper intervals to carry the network packets issued from a specific block of an area, shown in Fig. 1a. It can be further split into several sectors, each of which served by an individual cell transceiver installed on the same station tower, but with an independent antenna pointing towards a unique direction. In Fig. 1b, the high-level network architecture of such network is comprised of three main components: the user equipment, the radio access network composing of a bunch of base stations, and the core network offering routing and management services. Usually, the traffic meters are deployed at the interface between the base stations and

the core network, and their readings are reported for every a predefined period. In practice, only the max of aggregated traffic for every clock hour, namely $\max_{t\in\{0,\ldots,23\}} \sum_i \mathbb{I}(\text{hour}(i) = t)\text{len}(p_i)$ where $\text{len}(p_i)$ is the length of the ith packet per day, will be studied for later engineering propose. This transformation makes the data much noisier and even harder to predict. For the cell transceivers,



(a) (b)

Fig. 1. (a) The base station layout of dataset D1 in Table 2. Each polygon roughly represents the land area a base station ought to cover, in the form of Voronoi diagram. (b) The concise architecture of the mobile network. The core network acts as the bridge of base stations and the Internet.

Table 1. The primary attributes in the Engineering Parameter table. Only the parameters relevant to our problem are kept, either based on domain knowledge or empirical tests.

Group	Attr.	Type	Desc.
Location	Longitude	Real	Acquired from GPS (Global Positioning System)
	Latitude	Real	
Topology	Cell id	Nominal	A hierarchical structure, where tens of nodes at a lower level are connected to one node at a higher level
	Station id	Nominal	
	District id	Nominal	
	Site id	Nominal	
Scene	Outdoor	Boolean	Indicate if the antenna is installed outdoor or indoor
Antenna	Azimuth	Real	The direction on horizontal plane
	Downtilt	Real	The direction on vertical plane
	Antenna type	Nominal	Manufacturers and versions
	Power	Real	The transmission power of electromagnetic signal
Extra	Converge	Real	A derived property, suggested by domain experts, approximately indicates the area size of a cell estimated by Voronoi method (Fig. 1(a))

there are many (suppose K) engineering parameters, to describe various configurations, whose names, data types, and business meaning are briefly given in Table 1.

3 Related Work

There are two purposes for traffic prediction: one is for planning&optimization, corresponding to month-level prediction; the other is for day-2-day maintenance or device controlling at minute-level. We focus on the former one. Many popular models have been tested on this problem, including ARIMA and RNN (Recurrent Neural Networks) [2]. If we take the purely multi-dimensional approach, there are already works based on VAR, sometimes considering with geographic information. Even deep learning methods are tested on this problem [6]. Models from other related domains, such as transportation traffic [7] and geostatistics [3], can also be immigrated onto this problem.

As for the long-term time series prediction, the paper [8] summarize the main strategies of forecasting, including direct, recursive, and hybrid. The most counter-intuitive result is that the recursive and direct are not necessarily equivalent especially under the nonlinear situation, derived from both theoretical and empirical results. The main reason is that the repetitive applying of the same non-linear generative function, even it is the ground truth for one-step prediction, will possibly result in asymptotical bias and cannot be eliminated during recursion [8]. Influenced by many factors, such as ground truth, data size, and optimization process, the question of which multi-step strategy best is an empirical one.

A guide for using multiple kernels can be found in [4]. Here we use the boosting methods [10], for its advantage in easy-to-implement and low demand on the computing resource.

4 Solution

4.1 Overall Process

The multi-horizon prediction can be formalized as:

$$\mathbf{Y}_{t+1:t+H} = \{\mathbf{y}_{t+1}, \ldots, \mathbf{y}_{t+H}\} = \mathbf{F}(\mathbf{Y}_{1:t}), \tag{1}$$

where the value of the series $\mathbf{Y}$, having N consecutive and D dimensional observations, at the future H steps, are determined by an unknown stochastic function $\mathbf{F}$, taking the $\mathbf{Y}_{1:t}$ as input. The auxiliary attributes $\{\mathbf{A}_{i,j} : i \in 1..D, j \in 1..K\}$ are not digested by $\mathbf{F}$ directly, but used for designing candidate kernels later in Sect. 4.2. The $\mathbf{F}$ can be designed to be the composition of a set of single horizon models $\mathbf{f}$, according to the four schemes in Fig. 2:

(1) Recursive, shown in Fig. 2a, only an one-step-forward model is trained and applied repeatedly to future steps:

$$\mathbf{y}_{t+h} = \mathbf{f}_1^{(h)}(\mathbf{X_t}) + \varepsilon; \tag{2}$$

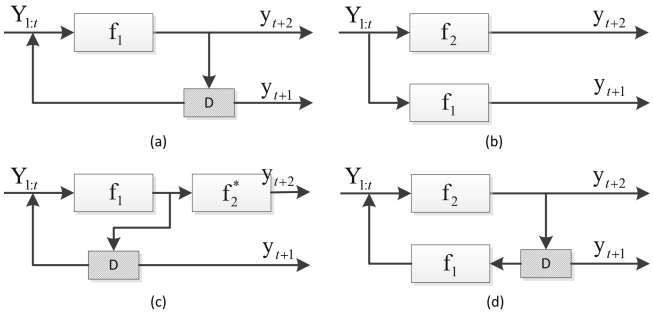


Fig. 2. Four approaches for composing a multiple-horizon model from single ones. Here we give the organization when $H = 2$. The D block means delay one step, i.e., passing through the input at next period.

(2) Direct, shown in Fig. 2b, one model for each step:

$$\mathbf{y}_{t+h} = \mathbf{f}_h(\mathbf{X_t}) + \varepsilon; \tag{3}$$

(3) Adjustable, shown in Fig. 2c, adding a rectification for each step before output based on (1), specially:

$$\mathbf{y}_{t+h} = \mathbf{f}_1^{(h)}(\mathbf{X_t}) + \mathbf{f}_h^*(\mathbf{X_{t+h}}) + \varepsilon, \text{ specially } \mathbf{f}_1^* \equiv \mathbf{0}; \tag{4}$$

(4) Multi-recursive, shown in Fig. 2d, overlay existing models at every step with one new recursive model:

$$\mathbf{y}_{t+h} = \sum_{i=1}^{h} \mathbf{f}_i^{(h)}(\mathbf{X_{t+i}}) + \varepsilon, \tag{5}$$

where $\mathbf{f}^{(h)}$ means applying $\mathbf{f}$ repeated along the time axis for h times, $\mathbf{X_{t+i}}$ denotes the concatenation of true measurements $\mathbf{Y}_{1:t}$ and estimated futures $\hat{\mathbf{Y}}_{t+1:t+i}(\mathbf{X_{t+i}} \triangleq [\mathbf{Y}_{1:t}, \hat{\mathbf{Y}}_{t+1:t+i}])$, and $\varepsilon \backsim \mathcal{N}(0, \sigma^2)$. After the entire model $\mathbf{F}$ has converted to a sequence of single-step tasks, at each step we are seeking a function $\mathbf{f}$ to solve the following optimization problem:

$$\hat{\mathbf{f}} = \min_{\mathbf{f}} L\left[\mathbf{y}_{t+h}, \mathbf{f}(\mathbf{X})\right], \tag{6}$$

where L is the squared error loss on $N - h$ samples, $\mathbf{X}$ is a unified denotation of all variable length of inputs from (1) to (4). The vector-output $\mathbf{f}$ can be solved under a gradient descent approach:

$$\mathbf{f}_m = \mathbf{f}_{m-1} - \rho \mathbf{g}_m, \tag{7}$$

where ρ is the step length, and $\mathbf{g}_m$ is the gradient residual at each data point. We use a weak learner $\mathbf{h}$, here is the multi-output ridge regression equipped with a matrix-valued kernel, to approximate the negative gradient signal, then the

problem is transformed to seek a best-effort kernel representing the underlying correlation between every two observations:

$$\hat{\mathbf{K}}_m = \text{argmin}_{\mathbf{K}_m} \sum_{i=1}^{N-h} \left[-\mathbf{g}_{im} - \mathbf{h}(\mathbf{X}^{(i)}; \mathbf{K}_m) \right]^2, \tag{8}$$

$$\mathbf{h}(\mathbf{X}; \mathbf{K}_m) = \sum_{i=1}^{N-h} \boldsymbol{\alpha}_{im} \mathbf{K}_m(\mathbf{X}, \mathbf{X}^{(i)}), \tag{9}$$

$$\boldsymbol{\alpha}_{im} \triangleq (\mathbf{K}_m + \lambda \mathbf{I}_N)^{-1} \mathbf{g}_{im}, \tag{10}$$

where λ in the ridge regression $\mathbf{h}$ is set to avoid overfitting. Mentioned in Sect. 3, matrix-valued kernel $\mathbf{K}$ are usually learned from a linear combination of basic kernels. To make it more applicable, $\mathbf{K}$ is further assumed to be separated into the product of one composite kernel on the input-space, and the other composite kernel representing the correlations among the outputs. Moreover, we add the Dirac Delta kernel, or identity matrix in discrete form, to the pool of candidates for the outputs, which allows the evolvement of devices can be independent if possible. In all, we get the following form of combinations:

$$(\mathbf{K}(\mathbf{X}, \mathbf{X}'))_{i,j} = \left[\sum_{i=1}^{Q_1} \beta_i^{(I)} \kappa_i^{(I)}(\mathbf{X}_{:,i}, \mathbf{X}'_{:,j}) \right] \left[\beta_0^{(O)} \delta_{i,j} + \sum_{i=1}^{Q_2} \beta_i^{(O)} \kappa_i^{(O)}(\mathbf{A}_{:,i}, \mathbf{A}'_{:,j}) \right], \tag{11}$$

where $\kappa^{(I)}$ and $\kappa^{(O)}$ are scalar-value kernels defined in the input and output spaces, and δ is the Dirac Delta kernel; Q_1 and Q_2 are the number of their candidates, and $\boldsymbol{\beta}$ is the coefficients for each kernel. Here we have $\sum_1^{Q_1} \beta_i^{(I)} = 1$ and $\sum_0^{Q_2} \beta_i^{(O)} = 1$. To simplify the computation and make the model at each iteration more sparse, the algorithm selects only one basic kernel, either for inputs or outputs, namely, the $\boldsymbol{\beta}_m \triangleq [\boldsymbol{\beta}_{1:Q_1}^{(I)}, \boldsymbol{\beta}_{1:Q_2}^{(O)}] = \hat{\beta}_m \mathbf{e}_j$ has only one non-zero entry at position p. At the start point of gradient descent, the algorithm firstly seeks the best position p_m at the range of $[1 : Q_1]$, and sets the Dirac Delta kernel as the initial kernel for outputs $[Q_1 + 1 : Q_1 + Q_2 + 1]$, which will be probably enriched by more non-zero values during later iterations.

4.2 Kernel Design

The kernels come from two groups of sources: historical records and device property. The similarity on time series can be further classified into two categories: inter-series and trans-series; the former includes AR, MA, and seasonality, inspired from the seasonal ARIMA model, and the latter is the DTW (Dynamic Time Warping) for capturing the nonlinear correlation, which is not easy to accomplish by VAR in the high dimensional setting. A practical method of converting the distance to a kernel is to encapsulate it in the form of RBF (Radial Basis Function):

$$\kappa(\boldsymbol{x}, \boldsymbol{x}'; d) = e^{\epsilon \cdot d(\boldsymbol{x}, \boldsymbol{x}')}. \tag{12}$$

One common heuristic for choosing ϵ is $1/D$, and the d can replace by the various definitions, such as the 1st–4th rows of Table 2. The usage of DTW is straightforward by substitution in Eq. 12, whereas the distance of two sub-series inside one series is determined by the L2 distance of their next step prediction values, based on AR, MA, or seasonal AR(1).

For devices, there are three kinds of data types: nominal, numerical, and nodes on a linked tree. The numerical and nominal can be simply handled by the RBF and Dirac Delta, while nodes on network topology need to compare their number of common ancestors:

$$\kappa_{Tree}(\mathbf{s}, \mathbf{s}') = \sum_{i=0}^{L} 2^{L-i} \delta_{\mathbf{s}(i),\mathbf{s}'(i)}, \tag{13}$$

where $\mathbf{s}$ is the nominal vector of a node's id along with the top-down path of a L-height topology tree. In all, we have five kinds of attributive kernels, listed in 5th–9th rows of Table 2.

Table 2. The kernels used in our solution. The names in the source column are kept consistent with the group column Table 1

ID	Space	Source	Kernel	Desc.
1	Input	Traffic Record	RBF of AR(2)	Short term correlation
2			RBF of MA(6)	Long term correlation
3			RBF of Seasonal AR(12, 1)	Year-on-year comparison
4	Output		RBF of DTW, window length = int(10% * N)	Nonlinear correlation
5		Location	Matérn	Distance between two points
6		Topology	Lowest common ancestor	#ancestors two nodes share
7		Scene	Dirac	Indoor or outdoor
8		Coverage	RBF	Land area a site serve
9		Antenna	RBF for numeric, Dirac for nominal	Parameters of hardware

5 Experiments

5.1 Data

We have collected three representative datasets from different types of sites, shown in Table 3. They are located in a western province of China and serves more than 250 million users with nearly 80k base stations. The dataset covers different standards of networks, ranging from 2G, 3G, and latest LTE (Long Term Evolution) networks, with different size of nodes. The earlier a network established, the longer records we have. The D2 contains a high rate of missing values, because of the owner's continuously large-scale reconstruction, when nearly 1/3 of stations were newly constructed or removed.

Table 3. Description of network traffic datasets.

Name	Unit	Type	Area	#unit	#month	#record	Missing rate	Mean	STD
D1	Base station	2G	Rural	5191	46	19915	8.34%	1.60E+02	1.94E+02
D2	Cell	3G	Metropolis	17097	35	151394	25.30%	1.01E+03	2.66E+03
D3	Base station	LTE	City	9946	27	36280	13.51%	3.87E+03	3.45E+04

There are some details on data preprocessing. The month level aggregation are taken from the average of top 3 days of every month. Before training, all data points are transformed by $\text{diff}_i(\log(\mathbf{Y}_{i,j}+1))$, to make it linearly predictable and stationary, and an inverse transform is needed before the accuracy evaluation. Miss values are filled by linear interpolation for each series.

5.2 Setup

The experiment composes of two parts: (1) comparative study of different models, including popular time series models as baselines and different strategies of multi-term prediction; (2) the contribution of properties or kernels at different horizons.

The ability of model needs to be evaluated with a proper train/test sets construction on genuine data. With a sliding window of length, $H = 12$ moving from right to left along the time axis, the section lying inside the window is used as test part, and the sub-series before the window is left as train part, also required to not shorter than $H = 12$. We're interested in the MAPE (Mean Absolute Percentage Error), which is calculated firstly by taking the mean of all trails' scores, and take their median for all entities at each horizon, namely $\text{median}_j(\text{mean}_i(\text{MAPE}_{i,j})), i = H+1..N-H, j = 1..D$. The reason for using median is that some sites may be activated or deactivated over a given period, which makes the mean unreliable.

The model is implemented by writing a boosting framework and modifying the KernelRidge in the open source scikit-learn library (to support weights on samples). The ARIMA is implemented by auto.arima in R's forecast package, and the VAR is from the MTS package. The RNN is 3-layered and is built in the style of seq-2-seq, whose number of nodes in the middle layer is determined by grid search, and implemented by Keras with L1 regularization with other default parameters unchanged. Other default settings in experiments include, number of boosting step = 100, shrinkage rate = 0.1, ridge regularization strength = 1.0.

5.3 Comparison Results

Table 4 gives the experimental results of 8 models, which can be compared by 3 aspects: model, horizon, and dataset. For the model, we compare two kinds of models: the 1st–4th are popular models for benchmarks, including a naive way

(as the 1st) using the current observation directly for output; the 5th–8th are the realizations of the 4 strategies illustrated in Sect. 4.1. The classical models relying on the recursive strategy, such as ARIMA, can achieve comparable results at short-term prediction, but deteriorate rapidly, especially for VAR, as steps go forward. The seq-2-seq RNN can do much better than others, but usually worse than our solution without the help of context information. In the 4 strategies, the direct and mixing can keep the errors growing much slower. The mixing strategies, including S3 and S4, are usually better than the direct strategy S1, while S4 is slightly better than S3 in 2 datasets.

Table 4. The comparison of MAPE for 8 models on 3 datasets. The 4 horizons are designed to observe the value change of short, mid, and long-term prediction.

Dataset	Model	Horizon			
		1 m	3 m	6 m	12 m
D1	Naive	0.285	0.511	0.869	1.367
	ARIMA	0.159	0.247	0.378	0.539
	VAR	0.131	0.185	0.261	0.341
	RNN	**0.108**	0.144	0.197	0.258
	S1-dir	0.111	0.126	0.157	0.187
	S2-rec	0.111	0.154	0.219	0.301
	S3-adj	0.111	0.125	0.152	0.170
	S4-mrec	0.111	**0.124**	**0.149**	**0.166**
D2	Naive	0.319	0.589	1.020	1.631
	ARIMA	**0.080**	0.127	0.204	0.315
	VAR	0.124	0.188	0.283	0.399
	RNN	0.082	0.125	0.192	0.281
	S1-dir	0.081	0.114	0.165	0.231
	S2-rec	0.081	0.128	0.204	0.313
	S3-adj	0.081	0.111	0.159	0.220
	S4-mrec	0.081	**0.107**	**0.148**	**0.196**
D3	Naive	0.425	0.781	1.349	2.154
	ARIMA	0.145	0.201	0.28	0.356
	VAR	0.278	0.382	0.538	0.732
	RNN	**0.137**	0.184	0.246	0.294
	S1-dir	0.150	0.176	0.225	0.267
	S2-rec	0.150	0.199	0.267	0.330
	S3-adj	0.150	**0.161**	**0.197**	**0.221**
	S4-mrec	0.150	0.165	0.201	0.225

In all situations, the long-term tasks are always harder than short ones. The single variable models (ARIMA, RNN), usually can achieve well enough results for the short-term prediction, even without the assistance of properties, which means it mainly depends on each series' own recent history. They are surpassed by the more sophisticated methods when the horizon goes into mid-term above. The recursive strategy is obviously not suitable for long terms.

The inherent intensity of noise and the missing rate of datasets result in the fundamental difference, even in applying the naive model. The length of data and the consistency of distribution limit the potential for accuracy improvement. The more aged network, the more stable their trends intend to be, and thus easier to predict. Anyway, the choice of strategy is still an empirical problem in practice.

5.4 Contribution of Kernels

We're interested in the effects of attributes when horizons and strategies change. At each horizon, we take the normalized weights of each basic kernel as their contributions and display them in the heatmap as Fig. 3. It can be discovered that the prediction for next month is mainly related to kernels from AR, MA, and seasonality, and meanwhile Dirac Delta kernel dominates the correlation of outputs, which implies these series can hardly correlate each other given a short period. When we switch to the mid-term, the weights start to shift to the DTW and topology, which means the model starts to refer the factors in a wider

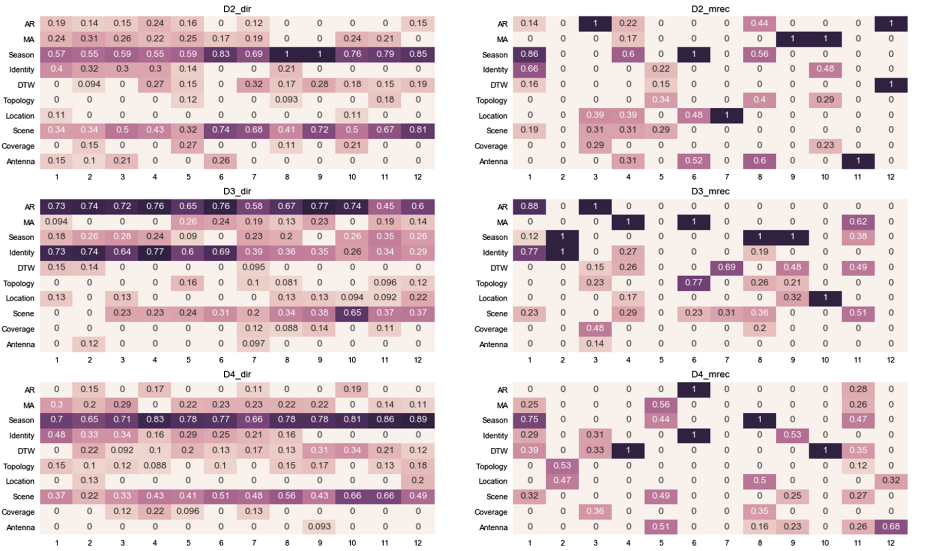


Fig. 3. The contribution of 9 kernels in boosting for the stages 1–12. Every row of sub-figures denotes a dataset, from D1 to D3; the left column denotes the S1 strategy, and the right column is the S4 strategy. In every sub-figure, the above 3 rows mean the kernel for inputs, while the others below are for outputs.

range. For long-term, the indoor/outdoor factor nearly dominates the decision, which is the most macroscopic factor we can find. The seasonality takes heavier proportion on D1, and D3 collected at higher station level. The weights of S4 are much sparser than S1, since the kernels found at previous stages will be applied repeatedly for later ones, and have much less possibility to appear again.

6 Conclusion

In this paper, we incorporated the context information into the multi-horizon network traffic prediction, and verify its effectiveness on long-term requests for the first time. The introduction of multi-kernel and multi-recursive is helpful to this problem, though with more development cost. This discovery builds a connection between the scales of temporal requests and the ranges of geographical terrain or the grains of device properties. This solution can further suggest the business operators find more macroscopic factors to support even longer forecasting, such as more traffic records from nearby provinces, or macroeconomic statistics of the whole country. The model may also be simplified or promoted by a solid theoretical analysis on the relationship between temporal scales and contextual information in future work.

References

1. Cisco Visual Networking Index: Global mobile data traffic forecast update, 2016–2021 white paper. https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html. Accessed Mar 2017
2. Bui, N., Cesana, M., Hosseini, S.A., Liao, Q., Malanchini, I., Widmer, J.: A survey of anticipatory mobile networking: context-based classification, prediction methodologies, and optimization techniques. IEEE Commun. Surv. Tutor. (2017)
3. Das, A.K., Pathak, P.H., Chuah, C.-N., Mohapatra, P.: Contextual localization through network traffic analysis. In: Proceedings of INFOCOM 2014, pp. 925–933. IEEE (2014)
4. Gönen, M., Alpaydın, E.: Multiple kernel learning algorithms. J. Mach. Learn. Res. **12**, 2211–2268 (2011)
5. Ma, X., Dai, Z., He, Z., Ma, J., Wang, Y., Wang, Y.: Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. Sensors **17**(4), 818 (2017)
6. Oliveira, T.P., Barbar, J.S., Soares, A.S.: Computer network traffic prediction: a comparison between traditional and deep learning neural networks. Int. J. Big Data Intell. **3**(1), 28–37 (2016)
7. Park, J., Raza, S.M., Thorat, P., Kim, D.S., Choo, H.: Network traffic prediction model based on training data. In: Gervasi, O., Murgante, B., Misra, S., Gavrilova, M.L., Rocha, A.M.A.C., Torre, C., Taniar, D., Apduhan, B.O. (eds.) ICCSA 2015. LNCS, vol. 9158, pp. 117–127. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21410-8_9

8. Taieb, S.B.: Machine learning strategies for multi-step-ahead time series forecasting. Ph.D. Thesis (2014)
9. Wen, Q., Zhao, Z., Li, R., Zhang, H.: Spatial-temporal compressed sensing based traffic prediction in cellular networks. In: 1st IEEE International Conference on Communications in China Workshops (ICCC), pp. 119–124. IEEE (2012)
10. Xia, H., Hoi, S.C.H.: MKBoost: a framework of multiple kernel boosting. IEEE Trans. Knowl. Data Eng. **25**(7), 1574–1586 (2013)