The Learning and Prediction of Application-level Traffic Data in Cellular Networks

Rongpeng Li, Zhifeng Zhao, Jianchao Zheng, Yan Chen, Chengli Mei, Yueming Cai, and Honggang Zhang

***Abstract*—Traffic learning and prediction is at the heart of the evaluation of the performance of telecommunications networks and attracts a lot of attention in wired broadband networks. Now, benefiting from the big data in cellular networks, it becomes possible to make the analyses one step further into the application level. In this paper, we firstly collect a significant amount of application-level traffic data from cellular network operators. Afterwards, with the aid of the traffic "big data", we make a comprehensive study over the modeling and prediction framework of cellular network traffic. Our results solidly demonstrate that there universally exist some traffic statistical modeling characteristics, including α-stable modeled property in the temporal domain and the sparsity in the spatial domain. Meanwhile, the results also demonstrate the distinctions originated from the uniqueness of different service types of applications. Furthermore, we propose a new traffic prediction framework to encompass and explore these aforementioned characteristics and then develop a dictionary learning-based alternating direction method to solve it. Besides, we validate the prediction accuracy improvement and the robustness of the proposed framework through extensive simulation results.**

***Index Terms*—Big data, cellular networks, traffic prediction, α-stable models, dictionary learning, alternative direction method, sparse signal recovery.**

I. Introduction

Traffic learning prediction in cellular networks, which is a classical yet still appealing field, yields a significant number of meaningful results. From a macroscopic perspective, it provides the commonly believed result that mobile Internet will witness a 1000-folded traffic growth in the next 10 years [1], which is acting as a crucial anchor for the design of next-generation cellular network architecture and embedded algorithms. On the other hand, the fine traffic prediction on a daily, hourly or even minutely basis could contribute to the optimization and management of cellular networks like energy savings [2], opportunistic scheduling [3], and network anomaly detection [4] . In other words, a precisely predicted future traffic load knowledge, which contributes to improving the network energy efficiency by dynamically adjusting the cell working status [5], [6], plays an important role in designing greener traffic-aware cellular networks. For example, it is inevitable to precisely forecast the traffic in one single cell, so as to turn some base stations (BSs) into sleeping mode when the traffic demand becomes low. Moreover, there comes a trend to enhance the edge functionalities of cellular networks, by deploying more resources in edge nodes (e.g., BSs). Hence, it is also meaningful to learn the traffic characteristics in one single cell.

Our previous research [7] has demonstrated the microscopic traffic predictability in cellular networks for circuit switching's voice and short message service and packet switching's data service. However, compared to the more accurate prediction performance for voice and text service in circuit switching domain, the state-of-the-art research in packet switching's data service is still not satisfactory enough. Therefore, a learning and prediction study over application-level data traffic might contribute to understanding data service's traffic characteristics, thus potentially leading to better prediction performance. Consequently, in this paper, we focus on the analyses for application-level traffic generated by three popular service types (i.e., instantaneous message (IM), web browsing, video). In order to obtain general results, we firstly collect a significant amount of practical traffic records from China Mobile[1]. By taking advantage of the traffic "big data", we then confirm the preciseness of fitting α-stable models to these typical service types of traffic and demonstrate α-stable models' universal existence in cellular network traffic. We later show that α-stable models can be used to leverage the temporally long range dependence, and guide linear algorithms to conduct traffic prediction. Besides, we find that spatial sparsity is also applicable for the application-level traffic, and propose that the predicted traffic should be able to be mapped to some sparse signals. In this regard, benefiting from the latest progress in compressive sensing [8]–[11], we could calibrate the traffic prediction results on the condition of not knowing the transform matrix a priori. Finally, in order to forecast the traffic with the aforementioned characteristics, we formulate the prediction problem by a new framework and then develop a dictionary learning based alternating direction method (ADM) [11] to solve it.

R. Li and Y. Chen are with Huawei Technologies Co. Ltd., Shanghai 210256, China (email: {lirongpeng, bigbird.chenyan }@huawei.com).

Z. Zhao, and H. Zhang are with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China (email:{zhaozf, honggangzhang}@zju.edu.cn).

J. Zheng, and Y. Cai are with the College of Communications Engineering, PLA University of Science and Technology, Nanjing 210007, China (email: longxingren.zjc.s@163.com, caiym@vip.sina.com).

C. Mei is with the China Telecom Technology Innovation Center, Beijing, China (email: meichl@ctbri.com.cn).

This paper is supported by the National Basic Research Program of China (973Green, No. 2012CB316000), the Key (Key grant) Project of Chinese Ministry of Education (No. 313053), the Key Technologies R&D Program of China (No. 2012BAH75F01).

[1]It is worthwhile to note here that we also collect another dataset from China Telecom to further verify the effectiveness of the thoughts inside in this paper. Due to the space limitation, we put the results related to China Telecom dataset in a separate file available at http://www.rongpeng.info/files/sup_file_twc.pdf.

A. Related Works

Due to its apparent significance, there have already existed two research streams toward the fine traffic prediction issue in wired broadband networks and cellular networks [7]. One is based on fitting models (e.g., ON-OFF model [12], ARIMA model [13], FARIMA model [14] , mobility model [15], [16], network traffic model [16], and α-stable model [17], [18]) to explore the traffic characteristics, such as spatial and temporal relevancies [19] or self-similarity [20], [21], and obtain the future traffic by appropriate prediction methods. The other is based on modern signal processing techniques (e.g., principal components analysis method [22], [23], Kalman filtering method [23], [24] or compressive sensing method [2], [11], [22], [25]) to capture the evolution of traffic. Meanwhile, it is useful to firstly model large-scale traffic vectors as sparse linear combinations of basis elements. Therefore, some dictionary learning method [26] is necessary to learn and construct the basis sets or dictionaries.

However, the existing traffic prediction methods in this microscopic case still lag behind the diverse requirements of various application scenarios. Firstly, most of them still focus on the traffic of all data services [27], and seldom shed light on a specific type of services (e.g., video, web browsing, IM, etc). Secondly, the existing prediction methods usually follow the analysis results in wired broadband networks like the α-stable models[2] [28], [29] or the often accompanied self-similarity [20] to forecast future traffic values [14], [18], [21]. However, since cellular networks have more stringent constraints on radio resources [30], relatively expensive billing polices and different user behaviors due to mobility [31] and thus exhibit distinct traffic characteristics, the corresponding results need to be validated before being directly applied to cellular networks [7].

B. Contribution

Compared to the previous works, this paper, belonging to one of the pioneering works toward application-level traffic analyses, depends on a large amount of practical records (as summarized in Table I and Table II) from China Mobile, and provides the following key insights:

- Firstly, this paper revisits α-stable models, and confirms their accuracy to model the application-level cellular network traffic for all three service types (i.e., IM, web browsing, video). Moreover, this paper shows the application-level traffic obeys the sparse property, and demonstrates the distinct characteristics among different service types. Therefore, the paper contributes to a general understanding of the cellular network traffic.
- Secondly, in order to encompass and explore these aforementioned characteristics, this paper provides a traffic prediction framework in Fig. 1. Specifically, the proposed framework consists of an "α-Stable Model & Prediction" module to generate coarse prediction results, a "Sparsity & Dictionary Learning" module to impose a sparse constraint and refine the prediction results, and an "Alternating Direction Method" module to provide the algorithmic details and obtain the final results. Meanwhile, this paper further validates the prediction performance by extensive simulation results .

[2]In this paper, the term "α-stable models" is interchangeable with α-stable distributions.

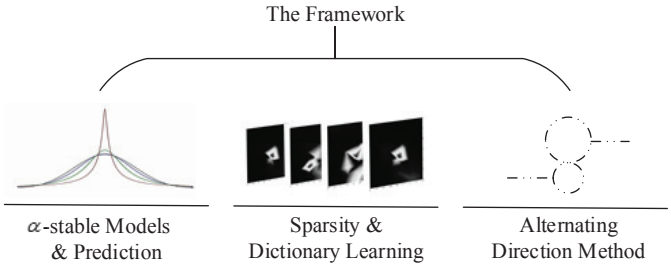

Fig. 1. The illustration of application-level traffic prediction framework .

The remainder of the paper is organized as follows. In Section II, we firstly present some necessary background of required mathematical tools. In Section III, we introduce the dataset for traffic prediction analyses, and later talk about the characteristics (i.e., α-stable models, and spatial sparsity) of the application-level dataset. In Section IV, we propose a new traffic prediction framework and its corresponding solution. Section V evaluates the proposed schemes and presents the validity and effectiveness. Finally, we conclude this paper in Section VI.

Notation: In the sequel, bold lowercase and uppercase letters (e.g., $\boldsymbol{x}$ and $\boldsymbol{X}$) denote a vector and a matrix, respectively. $(\cdot)^T$ denotes a transpose operation of a matrix or vector. $\|\boldsymbol{x}\|_0$ is an l_0-norm, counting the number of non-zero entries in $\boldsymbol{x}$, while an l_p-norm $\|\boldsymbol{x}\|_p$, $p \geq 1$ of a $1 \times n$ vector $\boldsymbol{x} = (x_1, \cdots, x_n)$ is defined by $\sqrt[p]{\sum_i^n |x_i|^p}$. The operation $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ denotes the summation operation of element-wise multiplication in $\boldsymbol{x}$ and $\boldsymbol{y}$ with the same size. $\text{sgn}(x)$ with respect to $x \in \mathcal{R}$ is defined as $\text{sgn}(x) = x/|x|$ when $x \neq 0$; and $\text{sgn}(x) = 0$ when $x = 0$.

II. Mathematical Background

A. α-Stable Models

Following the generalized central limit theorem, α-stable models manifest themselves in the capability to approximate the distribution of normalized sums of a relatively large number of independent identically distributed random variables [32] and lead to the accumulative property. Besides, α-stable models produce strong bursty results with properties of heavy tailed distributions and long range dependence. Therefore, they arise in a natural way to characterize the traffic in wired broadband networks [33], [34] and have been exploited in resource management analyses [35], [36].

α-stable models, with few exceptions, lack a closed-form expression of the probability density function (PDF), and are generally specified by their characteristic functions.

Definition 1. A random variable T is said to obey α-stable models if there are parameters $0 < \alpha \leq 2$, $\sigma \geq 0$, $-1 \leq \beta \leq$

1, and $\mu \in \mathcal{R}$ such that its characteristic function is of the following form:

$$\Phi(\omega) = E(\exp j\omega T) = \begin{cases} \exp\left\{-\sigma^\alpha|\omega|^\alpha\left(1 - j\beta(\text{sgn}(\omega))\tan\frac{\pi\alpha}{2}\right) + j\mu\omega\right\}, \\ \qquad\qquad\qquad\qquad\qquad\qquad \alpha \neq 1; \\ \exp\left\{-\sigma|\omega|\left(1 + j\frac{2\beta}{\pi}(\text{sgn}(\omega))\ln|\omega|\right) + j\mu\omega\right\}, \alpha = 1. \end{cases} \tag{1}$$

Here, the function $E(\cdot)$ represents the expectation operation with respect to a random variable. α is called the characteristic exponent and indicates the index of stability, while β is identified as the skewness parameter. α and β together determine the shape of the models. Moreover, σ and μ are called scale and shift parameters, respectively. In particular, if $\alpha = 2$, α-Stable models reduce to Gaussian distributions.

Furthermore, for an α-stable modeled random variable T, there exists a linear relationship between the parameter α and the function $\Psi(\omega) = \ln\{-\text{Re}[\ln(\Phi(\omega))]\}$ as

$$\Psi(\omega) = \ln\{-\text{Re}[\ln(\Phi(\omega))]\} = \alpha\ln(\omega) + \alpha\ln(\sigma), \tag{2}$$

where the function $\text{Re}(\cdot)$ calculates the real part of the input variable.

Usually, it's challenging to prove whether a dataset follows a specific distribution, especially for α-stable models without a closed-form expression for the PDF. Therefore, when a dataset is said to satisfy α-stable models, it usually means the dataset is consistent with the hypothetical distribution and the corresponding properties. In other words, the validation needs to firstly estimate parameters of α-stable models from the given dataset, and then compare the real distribution of the dataset with the estimated α-stable model [34]. Specifically, the corresponding parameters in α-stable models can be determined by maximum likelihood methods, quantile methods, or sample characteristic function methods [33], [34].

B. Sparse Representation and Dictionary Learning

In recent years, sparsity methods or the related compressive sensing (CS) methods have been significantly investigated [8]–[11]. Mathematically, sparsity methods aim to tackle this sparse signal recovery problem in the form of

$$\begin{aligned} &\min \|\boldsymbol{s}\|_0, \\ &s.t.\ \boldsymbol{y} = \boldsymbol{D}\boldsymbol{s}, \end{aligned} \tag{3}$$

or

$$\begin{aligned} &\min \|\boldsymbol{s}\|_0, \\ &s.t.\ \|\boldsymbol{y} - \boldsymbol{D}\boldsymbol{s}\| \leq \epsilon. \end{aligned} \tag{4}$$

Here, $\boldsymbol{s}$ denotes a sparse signal vector while $\boldsymbol{y}$ denotes a measurement vector based on a transform matrix or dictionary $\boldsymbol{D}$. Besides, ϵ is a predefined integer indicating the sparsity. By leveraging the embedded sparsity in the signals, sparsity methods could successfully recover the sparse signal with a high probability, depending on a small number of measurements fewer than that required in Nyquist sampling theorem. Basis pursuit (BP) [37], one of typical sparsity methods, solves the problem in terms of maximizing a posterior (MAP) criterion by relaxing the l_0-norm to an l_1-norm. On the other hand, orthogonal matching pursuit (OMP) [38] greedily achieves the final outcome in a sequential manner, by computing inner products between the signal and dictionary columns, and possibly solving them using the least square criterion.

For sparsity methods above, there usually exists an assumption that the transform matrix or dictionary $\boldsymbol{D}$ is already known or fixed. However, in spite of their computation simplicity, such pre-specified transform matrices like Fourier transforms and overcomplete wavelets might not be suitable to lead to a sparse signal [39]. Consequently, some researchers proposed to design $\boldsymbol{D}$ based on learning [26], [39]. In other words, during the sparse signal recovery procedure, machine learning and statistics are leveraged to compute the vectors in $\boldsymbol{D}$ from the measurement vector $\boldsymbol{y}$, so as to grant more flexibility to get a sparse representation $\boldsymbol{s}$ from $\boldsymbol{y}$. Mathematically, dictionary learning methods would yield a final transform matrix by alternating between a sparse computation process based on the current dictionary and a dictionary update process to approach the measurement vector.

TABLE I
DATASET 1 UNDER STUDY

	IM (Weixin)	Web Browsing (HTTP)	Video (QQLive)
Traffic Resolution (Collection Interval)	5 min	5 min	5 min
Duration	1 day	1 day	1 day
No. of Active Cells	2292	4507	4472
Location Info. (Latitude & Longitude)	Yes	Yes	Yes

TABLE II
DATASET 2 UNDER STUDY

		IM (QQ)	Web Browsing (HTTP)	Video (QQLive)
Traffic Resolution (Collection Interval)	Δt	30 min	30 min	30 min
Duration		2 week	2 week	2 week
No. of Active Cells		5868	5984	5906
Location Info. (Latitude & Longitude)		Yes	Yes	Yes

III. APPLICATION-LEVEL TRAFFIC DATASET AND ITS CHARACTERISTICS

A. Traffic Dataset Description

In this paper, our datasets are based on a significant number of practical traffic records from China Mobile in Hangzhou, an eastern provincial capital in China via the Gb interface of 2G/3G cellular networks or S1 interface of 4G cellular networks [40]. Specifically, these records are concerned with the packets from the Layer 7 (i.e., Application Layer) of OSI model. Specifically, the datasets encompass

nearly 6000 cells' location records[3] with more than 7 million subscribers involved. The datasets also contain the information like timestamp, corresponding cell ID, and traffic-related application name, by taking advantage of matching traffic records with specific protocols. In particular, we can determine web browsing service and video service by the applied HTTP protocol and streaming protocol respectively, while we assume traffic records to belong to the IM service, after fitting them to learning results of regular IM packet pattern. Moreover, the traffic volume could be calculated after aggregating packets to each influx base station. Notably, the paper aims to predict the traffic volume in each BS instead of the entire cellular network.

According to the traffic resolution (e.g., the traffic collection interval, namely 5 minutes and 30 minutes), the collected data can be sorted into two categories. Table I summarizes the information of per 5-minute traffic records collected on September 9th, 2014 with Weixin/Wechat[4], HTTP Web Browsing, and QQLive Video[5] selected as the representatives of these three service types. Here, the term "no. of active cells" refers to the number of cells where a specific type of service happened. Similarly, Table II lists the corresponding details of per 30-minute traffic records from July 14th, 2014 to July 27th, 2014 with QQ[6], HTTP Web Browsing, and QQLive Video as the representatives, respectively.

Based on the datasets in Table I and Table II, Fig. 2 illustrates the traffic variations generated by these applications in the randomly selected cells. Indeed, the phenomena in Fig. 2 universally exist in other individual cells and lead to the following insight.

Remark 1. *Different services exhibit distinct traffic characteristics. IM and HTTP web browsing services frequently produce traffic loads; while distinct from them, video service with more sporadic activities may generate more significant traffic loads.*

For simplicity of representation, we introduce a traffic vector $\boldsymbol{x}$, whose entries archives the volume of traffic in one given cell at different moments. Furthermore, by augmenting the traffic vectors for different cells, we refer to a traffic matrix $\boldsymbol{X}$ to denote the traffic records in an area of interest. Then, every row vector of traffic matrix indicates traffic loads at one specific cell with respect to the time while every column vector reflects volumes of traffic of several adjacent cells at one specific moment. Specifically, for a traffic resolution Δt, $X(i,t)$ in a traffic matrix $\boldsymbol{X}$ denotes traffic loads of cell i from t to $t+\Delta t$.

[3]Indeed, at one specific location, there might exist several cells operating on different frequencies or modes. For simplicity of representation, in the following analyses, we merge the information for different cells at the same location into one.

[4]Weixin/Wechat provides a Whatsapp-alike instant messaging service developed by Tencent Inc., and is one of the most popular mobile social applications in China with more than 400 million active users.

[5]QQLive Video is a popular live streaming video platform in China.

[6]QQ is another instant messaging service developed by Tencent Inc. with more than 800 million active users. Due to some practical reasons, per 30-minute Weixin traffic records are unavailable. Therefore, Table II includes QQ's traffic records.

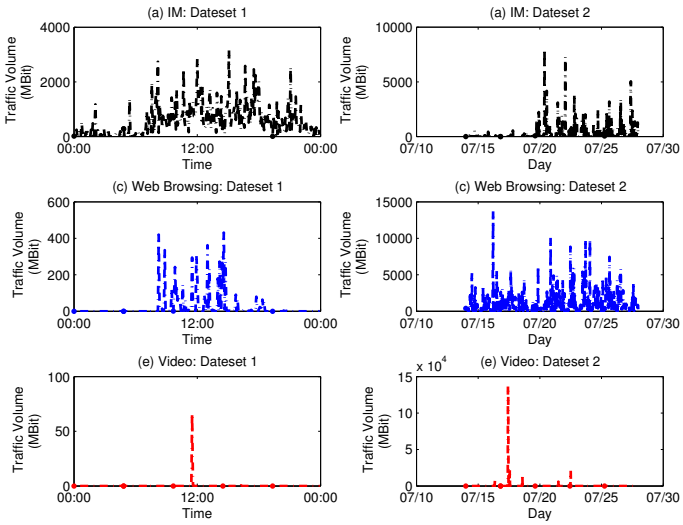

Fig. 2. The traffic variations of applications in different service types in the randomly selected (single) cells.

TABLE III
THE PARAMETER FITTING RESULTS IN THE α-STABLE MODELS BASED ON DATASET 1

App Name	Parameters			
	α (Stability)	β (Skewness)	σ (Scale)	μ (Shift)
IM	1.61	1	188.67	221.83
Web Browsing	1.60	1	32.33	42.75
Video	0.51	1	1×10^{-10}	0

Remark 2. *Traffic prediction can be regarded as the procedure to obtain a column vector $\hat{\boldsymbol{x}}_p = \hat{\boldsymbol{X}}(:,t)^T$ at a future moment t, based on the already known traffic records. Each entry $\hat{x}_p(\cdot)$ in $\hat{\boldsymbol{x}}_p$ corresponds to the future traffic in one cell.*

B. The α-Stable Modeled and Sparse Properties

In this section, we examine the results of fitting the application-level dataset to α-stable models. Firstly, in Table III and Table IV , we list the parameter fitting results using quantile methods [41], when we take into consideration the traffic records in three randomly selected cells (each for one service type) of Table I and Table II.

Afterwards, we use the α-stable models, produced by the aforementioned estimated parameters, to generate some random variable, and compare the induced cumulative distribution function (CDF) with the exact (empirical) one. Fig. 3 presents the corresponding comparison between the simulated results

TABLE IV
THE PARAMETER FITTING RESULTS IN THE α-STABLE MODELS BASED ON DATASET 2

App Name	Parameters			
	α (Stability)	β (Skewness)	σ (Scale)	μ (Shift)
IM	0.70	1	26.32	-100.69
Web Browsing	2	1	2.03×10^{3}	2.01×10^{3}
Video	0.51	1	136.52	-341.15

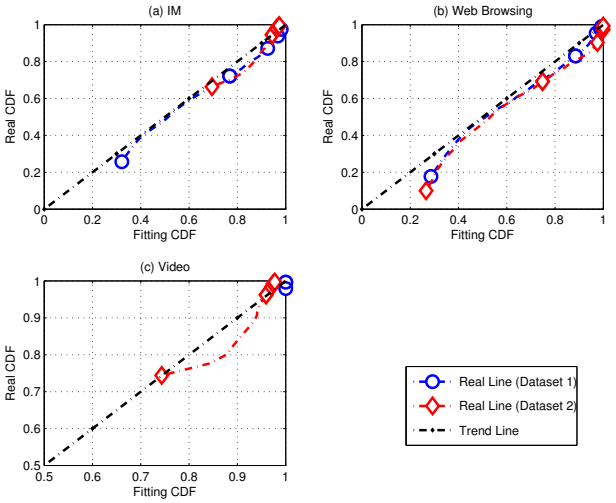

Fig. 3. For different service types, α-stable model fitting results versus the real (empirical) ones in terms of the cumulative distribution function (CDF).

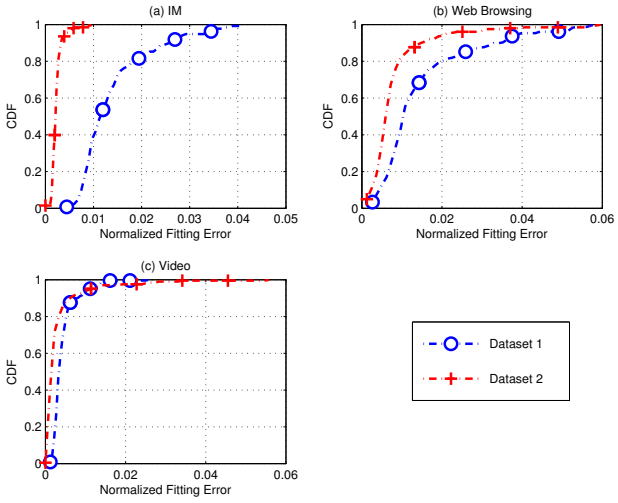

Fig. 4. The preciseness error CDF for all the cells after fitting $\Psi(\omega)$ with respect to $\ln(\omega)$ to a linear function.

and the real ones. Recalling the statements in Section II-A, if the simulated dataset has the same or approximately same distribution as the real one, the empirical dataset could be deemed as α-stable modeled. Therefore, Fig. 3 indicates the traffic records in these selected areas could be simulated by α-stable models.

On the other hand, recalling the statements in Section II, for an α-stable modeled random variable X, there exists a linear relationship between the parameter α and the function $\Psi(\omega) = \ln\{-\mathrm{Re}[\ln(\Phi(\omega))]\}$. Thus, we fit the estimated parameter α with the computing function $\Psi(\omega)$, and provide the preciseness error CDF for all the cells in Fig. 4. According to Fig. 4, the normalized fitting errors for 80% cells in both datasets are less than 0.02. Therefore, the practical application-level traffic records follow the property of α-stable models (in Eq. (2)), and further enhance the validation results by Fig. 3. Meanwhile, different application-level traffic exhibits different fitting accuracy. In that regard, the video traffic in Fig. 4(c) has the minimal fitting error, while the fitting error of the web browsing traffic in Fig. 4(b) is largest. Moreover, since a larger traffic resolution means a confluence of more application-level traffic packets and could better demonstrate the accumulative property of α-stable models, the fitting error quickly decreases along with the increase in traffic resolution.

Remark 3. *Due to their generality, α-stable models are suitable to characterize the application-level traffic loads in cellular networks, even though it might not be the most accurate one.*

Indeed, the universal existence of α-stable models also implies the self-similarity of application-level traffic [20]. Hence, in the following sections, it is sufficient to only present and discuss the results from Dataset 1 in Table I. On the other hand, the phenomena that application-level traffic universally obeys α-stable models can be explained as follows. Our previous study [42] unveiled that the message length of one individual IM activity follows a power-law distribution. Meanwhile, the traffic distribution within one cell can be regarded as the accumulation of lots of IM activities. Moreover, according to the generalized central limit theorem [43], the sum of a number of random variables with power-law distributions decreasing as $|x|^{-\alpha-1}$ where $0 < \alpha < 2$ (and therefore having infinite variance) will tend to an α-stable model as the number of summands grows. Hence, the application-level traffic within one cell follows α-stable models.

Additionally, data traffic in wired broadband networks [22] and voice and text traffic in circuit switching domain of cellular networks [2] prove to possess the spatio-temporal sparsity characteristic. Indeed, the application-level traffic spatially possesses this sparse property as well. Fig. 5 depicts the traffic density in 10AM and 4PM in randomly selected dense urban areas. As Fig. 5 shows, there appear a limited number of traffic hotspots. But these hotspots are closed and correlated with each other. Comparatively, there exists a larger area with less traffic. This spatially clustering property is also consistent with the findings in [19], and proves the traffic's spatial sparsity. Moreover, the number of hotspots for video service is smallest, which indicates QQLive Video traffic has the strongest sparsity.

Remark 4. *The application-level traffic dataset further validates that the traffic for different service types of applications follows a spatially sparse property. Besides, compared to IM and web browsing service, video service exhibits the strongest sparsity.*

IV. Application-level Traffic Prediction Framework

Section III unveils that the application-level cellular network traffic could be characterized by α-stable models and obey sparse property. In this section, we aim to fully take advantage of these results, and propose a new framework in Fig. 1 to predict the traffic. The proposed framework consists of three modules. Among them, the "α-Stable Model & Prediction" module would take advantage of the already known traffic knowledge to learn and distill the parameters in α-stable

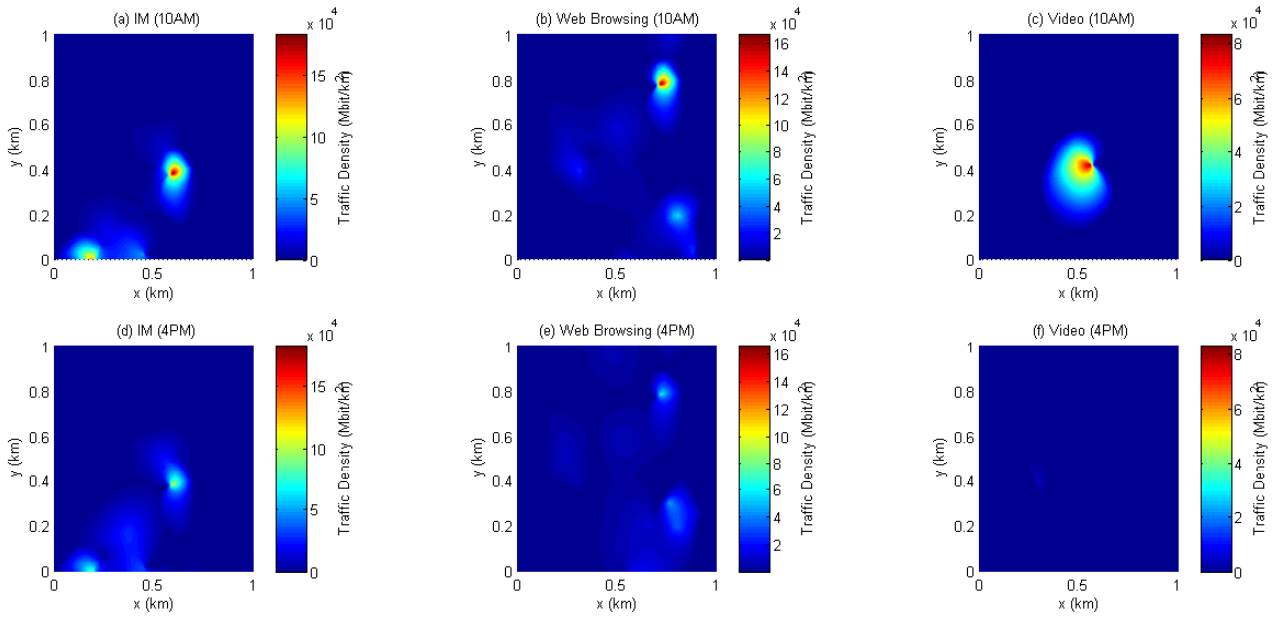

Fig. 5. The application-level cellular network traffic density in 10AM and 4PM in randomly selected dense urban areas for three service types of applications. The area for IM, Web Browsing, and Video contains 23, 39, 35 active cells, respectively.

models and provide a coarse prediction result. Meanwhile, the "Sparsity & Dictionary Learning" module imposes constraints to make the final prediction results satisfy the spatial sparsity. But, these two modules inevitably add multiple parameters without known a priori, and thus need specific mathematical operations to obtain a solution. Hence, the proposed framework also contain a "Alternating Direction Method" module to iteratively process the other modules and yield the final result.

A. Problem Formulation

Previous sections unearth several important characteristics in application-level traffic in cellular networks, including spatial sparsity, and temporally modeling by α-stable processes. All these factors could be leveraged for forecasting the future traffic vector $\hat{\boldsymbol{x}}_p$.

- Temporal modeling component. As Section III-B states, the application-level traffic loads follow α-stable models. Therefore, benefiting from the substantial body of works towards α-stable model based linear prediction [28], [34], coarse prediction results can be achieved by computing linear prediction coefficients in terms of the least mean square error criterion, the minimum dispersion criterion, or the covariation orthogonal criterion [17]. Due to its simplicity and comparatively low variability, the covariation orthogonal criterion [17], [18] is chosen in this paper to demonstrate the α-stable based linear prediction performance.
  Without loss of generality, assume that there exist N cells in the area of interest. For a cell $i \in N$ with a known n-length traffic vector $\boldsymbol{x}^{(i)} = (x^{(i)}(1), \cdots, x^{(i)}(n))$, $\hat{x}_\alpha(i)$ in α-stable models-based predicted traffic vector $\hat{\boldsymbol{x}}_\alpha = (\hat{x}_\alpha(1), \cdots)$ is approximated by

$$\tilde{x}_\alpha(i) = \sum_{j=1}^{m} a^{(i)}(j) x^{(i)}(n+1-j), \tag{5}$$

with $1 < m \leq n$, where $\boldsymbol{a}^{(i)} = (a^{(i)}(1), \cdots, a^{(i)}(m))$ denotes the prediction coefficients by α-stable models-based linear prediction algorithms. For example, in order to make the 1-step-ahead linear prediction $\tilde{x}_\alpha(i)$ covariation orthogonal to $x^{(i)}(t), \forall t \in \{1, \cdots, n\}$, coefficient $a_i, \forall i \in \{1, \cdots, m\}$ should be given as [34]

$$a_i = \sum_{l=1}^{m} \left[\left(\sum_{j=\max(i,l)}^{n} x_{j-(l-1)} x_{j-(i-1)}^{<\alpha-1>} \right) \times \left(\sum_{j=l+k}^{n} x_j x_{j-k-(l-1)}^{<\alpha-1>} \right) \right] \tag{6}$$

Here, the signed-power $\nu^{<\alpha-1>} = |\nu|^{(\alpha-1)}\mathrm{sgn}(\nu)$. For simplicity of representation, the terminology "$(n = 36, m = 10, k = 1)$-linear prediction" is used to denote a prediction method, which firstly utilizes $n = 36$ consecutive traffic records in one randomly selected cell, then calculates $m = 10$ prediction coefficients, and finally predicts the traffic value at the next (i.e., $k = 1$) moment.
- Noise component. For any prediction algorithm, there dooms to exist some prediction error. Therefore, final traffic prediction vector $\hat{\boldsymbol{x}}_p$ is approximated by $\hat{\boldsymbol{x}}_\alpha$ plus

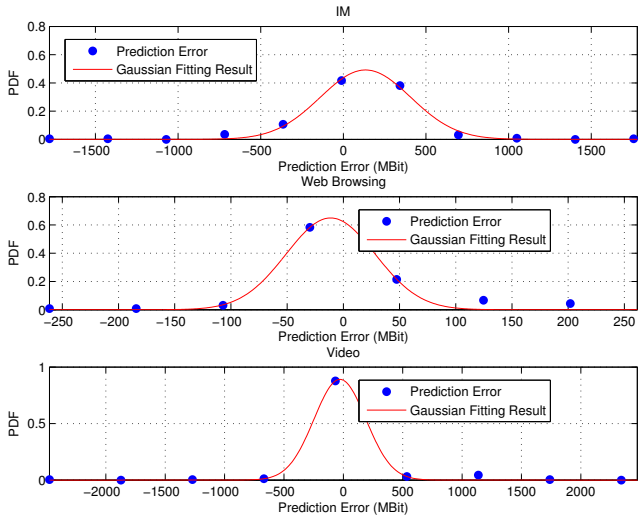

Fig. 6. The result by fitting the prediction error to a Gaussian distribution, after an α-stable model-based $(36, 10, 1)$-linear prediction method.

Gaussian noise $\boldsymbol{z}$ [7]. Combining the temporal modeling and noise components, $\hat{\boldsymbol{x}}_p$ could be achieved by

$$\begin{aligned} \min_{\hat{\boldsymbol{x}}_p, \hat{\boldsymbol{x}}_\alpha, \boldsymbol{z}} \quad & \|\hat{\boldsymbol{x}}_\alpha - \tilde{\boldsymbol{x}}_\alpha\|_2^2 + \lambda_1 \|\boldsymbol{z}\|_2^2, \\ s.t. \quad & \hat{\boldsymbol{x}}_p = \hat{\boldsymbol{x}}_\alpha + \boldsymbol{z}, \qquad (7) \\ & \tilde{\boldsymbol{x}}_\alpha = (\tilde{x}_\alpha(1), \cdots, \tilde{x}_\alpha(N)), \qquad (8) \\ & \tilde{x}_\alpha(i) = \sum_{j=1}^{m} a^{(i)}(j) x^{(i)}(n+1-j), \qquad (9) \\ & \forall i \in \{1, \cdots, N\}. \end{aligned}$$

For simplicity of representation, we omit constraints in Eq. (8) and Eq. (9) in the following statements.

- Spatial sparse component. In Section III-B, application-level traffic is shown to exhibit the spatial sparsity. Therefore, $\hat{\boldsymbol{x}}_p$ could be further refined by minimizing the gap between $\hat{\boldsymbol{x}}_p$ and a sparse linear combination (i.e., $\boldsymbol{s} \in \mathcal{R}^{K\times 1}$) of a dictionary $\boldsymbol{D} \in \mathcal{R}^{N\times K}$, namely

$$\begin{aligned} \min_{\hat{\boldsymbol{x}}_p, \boldsymbol{D}, \boldsymbol{s}} \quad & \|\hat{\boldsymbol{x}}_p - \boldsymbol{D}\boldsymbol{s}\|_2^2, \qquad (10) \\ s.t. \quad & \|\boldsymbol{s}\|_0 \le \epsilon. \end{aligned}$$

Notably, in Fig. 5, we observe sparse application-level cellular network traffic density. In other words, there merely exist few traffic spots with significantly large traffic volume. On the other hand, in the area of sparse representation, a l_0-norm, which counts the number of nonzero elements in the vector, is often used to characterize the sparse property. Therefore, in Eq. (10), we use a l_0-norm to add the sparse constraint to the final optimization problem. Moreover, the exact representation of the dictionary, which the previous sparsity analyses do not mention, remains a problem and would be solved later.

Therefore, it is natural to consider the original dataset as a mixture of these effects and propose a new framework to combine these two components together to get a superior forecasting performance.

In order to capture the temporal α-stable modeled variations while keeping the spatial sparsity, a new framework is proposed as follows:

$$\begin{aligned} \min_{\hat{\boldsymbol{x}}_p, \hat{\boldsymbol{x}}_\alpha, \boldsymbol{z}, \boldsymbol{D}, \boldsymbol{s}} \quad & \|\hat{\boldsymbol{x}}_\alpha - \tilde{\boldsymbol{x}}_\alpha\|_2^2 + \lambda_1 \|\boldsymbol{z}\|_2^2 + \lambda_2 \|\hat{\boldsymbol{x}}_p - \boldsymbol{D}\boldsymbol{s}\|_2^2, \\ s.t. \quad & \hat{\boldsymbol{x}}_p = \hat{\boldsymbol{x}}_\alpha + \boldsymbol{z}, \qquad (11) \\ & \|\boldsymbol{s}\|_0 \le \epsilon. \end{aligned}$$

Due to the nonconvexity of l_0-norm, the constraints in Eq. (11) are not directly tractable. Thanks to the sparsity methods discussed in Section II-B, an l_1-norm relaxation is employed to make the problem convex while still preserving the sparsity property [44]. Therefore, Eq. (11) can be reformulated as

$$\begin{aligned} \min_{\hat{\boldsymbol{x}}_p, \hat{\boldsymbol{x}}_\alpha, \boldsymbol{z}, \boldsymbol{D}, \boldsymbol{s}} \quad & \|\hat{\boldsymbol{x}}_\alpha - \tilde{\boldsymbol{x}}_\alpha\|_2^2 + \lambda_1 \|\boldsymbol{z}\|_2^2 + \lambda_2 \|\hat{\boldsymbol{x}}_p - \boldsymbol{D}\boldsymbol{s}\|_2^2, \\ s.t. \quad & \hat{\boldsymbol{x}}_p = \hat{\boldsymbol{x}}_\alpha + \boldsymbol{z}, \qquad (12) \\ & \|\boldsymbol{s}\|_1 \le \varepsilon. \end{aligned}$$

where ε is a predefined constraint, similar to ϵ.

Remark 5. *This proposed framework integrates the temporal modeling and spatial correlation together. Moreover, by adjusting λ_1 and λ_2 to some extreme values, it's easy to show that the framework in Eq.* (12) *is closely tied to some typical methods in other references.*

- *If λ_1 and λ_2 are extremely small, the framework is simplified to a simple α-stable linear prediction method [17], [18].*
- *If λ_2 is extremely large, the spatial sparsity factor dominates in the framework [22].*

B. Optimization Algorithm

In order to optimize the generalized framework in Eq. (12), we first reformulate Eq. (12) by taking advantage of the augmented Lagrangian function [45] and then develop an alternating direction method (ADM) [11] to solve it. Specifically, the corresponding augmented Lagrangian function can be formulated as

$$\begin{aligned} & \mathcal{L}(\hat{\boldsymbol{x}}_p, \hat{\boldsymbol{x}}_\alpha, \boldsymbol{z}, \boldsymbol{D}, \boldsymbol{s}, \boldsymbol{m}, \gamma, \eta) \\ \triangleq \; & \|\hat{\boldsymbol{x}}_\alpha - \tilde{\boldsymbol{x}}_\alpha\|_2^2 + \lambda_1 \|\boldsymbol{z}\|_2^2 + \lambda_2 \|\hat{\boldsymbol{x}}_p - \boldsymbol{D}\boldsymbol{s}\|_2^2 \\ & + \langle \boldsymbol{m}, \hat{\boldsymbol{x}}_p - \hat{\boldsymbol{x}}_\alpha - \boldsymbol{z} \rangle \qquad (13) \\ & + \gamma \cdot \|\boldsymbol{s}\|_1 \qquad (14) \\ & + \eta \cdot \|\hat{\boldsymbol{x}}_p - \hat{\boldsymbol{x}}_\alpha - \boldsymbol{z}\|_2^2, \qquad (15) \end{aligned}$$

where the operation $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ denotes the summation operation of element-wise multiplication in $\boldsymbol{x}$ and $\boldsymbol{y}$ with the same size. Besides, $\boldsymbol{m}$ and γ are the Lagrangian multipliers, while η is a factor for the penalty term. Essentially, the augmented Lagrangian function includes the original objective,

[7] There are two reasons leading to the assumption that noise is Gaussian distributed. Firstly, Gaussian distributed noise is widely used to characterize the fitting error between models and practical data. Secondly, we conducted a experiment to examine the prediction performance of a simple $(n = 36, m = 10, k = 1)$-linear prediction procedure, and found that the prediction procedure could well predict the traffic trend. However, there would exist some gap between the real traffic trace and the predicted one. Fortunately, as shown in Fig. 6, the prediction error can be approximated by the Gaussian distribution.

two Lagrange multiplier terms (i.e., Eq. (13) and Eq. (14)), and one penalty term converted from the equality constraint (i.e., Eq. (15)). Specifically, introducing Lagrange multipliers conveniently converts an optimization problem with equality constraints into an unconstrained one. Moreover, for any optimal solution that minimizes the (augmented) Lagrangian function, the partial derivatives with respect to the Lagrange multipliers must be zero [46]. Additionally, the penalty terms enforce the original equality constraints. Consequently, the original equality constraints are satisfied. Besides, by including Lagrange multiplier terms as well as the penalty terms, it's not necessary to iteratively increase η to ∞ to solve the original constrained problem, thereby avoiding ill-conditioning [45].

The ADM algorithm progresses in an iterative manner. During each iteration, we alternate among the optimization of the augmented function by varying each one of $(\hat{\boldsymbol{x}}_p, \hat{\boldsymbol{x}}_\alpha, \boldsymbol{z}, \boldsymbol{D}, \boldsymbol{s}, \boldsymbol{m}, \gamma, \eta)$ while fixing the other variables. Correspondingly, we present the reasons for each iteration of the ADM algorithm (i.e., Algorithm 2). Indeed, the ADM algorithm involves the following steps:

1) Find $\hat{\boldsymbol{x}}_\alpha$ to minimize the augmented Lagrangian function $\mathcal{L}(\hat{\boldsymbol{x}}_p, \hat{\boldsymbol{x}}_\alpha, \boldsymbol{z}, \boldsymbol{D}, \boldsymbol{s}, \boldsymbol{m}, \gamma, \eta)$ with other variables fixed. Removing the fixed items, the objective turns into

$$\begin{aligned}\arg\min_{\hat{\boldsymbol{x}}_\alpha} &\|\hat{\boldsymbol{x}}_\alpha - \tilde{\boldsymbol{x}}_\alpha\|_2^2 + \langle \boldsymbol{m}, \hat{\boldsymbol{x}}_p - \hat{\boldsymbol{x}}_\alpha - \boldsymbol{z}\rangle \\ &+ \eta \cdot \|\hat{\boldsymbol{x}}_p - \hat{\boldsymbol{x}}_\alpha - \boldsymbol{z}\|_2^2,\end{aligned}$$

which can be further reformulated as

$$\arg\min_{\hat{\boldsymbol{x}}_\alpha} \frac{1}{\eta}\cdot\|\hat{\boldsymbol{x}}_\alpha - \tilde{\boldsymbol{x}}_\alpha\|_2^2 + \|\hat{\boldsymbol{x}}_\alpha - (\hat{\boldsymbol{x}}_p - \boldsymbol{z} + \frac{\boldsymbol{m}}{2\eta})\|_2^2. \quad (16)$$

Letting $\boldsymbol{J}_{\hat{\boldsymbol{x}}_\alpha} = \hat{\boldsymbol{x}}_p - \boldsymbol{z} + \frac{\boldsymbol{m}}{2\eta}$ and setting the gradient of the objective function in Eq. (16) to be zero, it yields

$$\hat{\boldsymbol{x}}_\alpha = \frac{1}{\eta+1}\cdot(\tilde{\boldsymbol{x}}_\alpha + \eta\cdot\boldsymbol{J}_{\hat{\boldsymbol{x}}_\alpha}). \quad (17)$$

2) Find $\boldsymbol{z}$ to minimize the augmented Lagrangian function $\mathcal{L}(\hat{\boldsymbol{x}}_p, \hat{\boldsymbol{x}}_\alpha, \boldsymbol{z}, \boldsymbol{D}, \boldsymbol{s}, \boldsymbol{m}, \gamma, \eta)$ with other variables fixed. The corresponding mathematical formula is

$$\begin{aligned}\arg\min_{\boldsymbol{z}} &\lambda_1\|\boldsymbol{z}\|_2^2 + \langle \boldsymbol{m}, \hat{\boldsymbol{x}}_p - \hat{\boldsymbol{x}}_\alpha - \boldsymbol{z}\rangle \\ &+ \eta\cdot\|\hat{\boldsymbol{x}}_p - \hat{\boldsymbol{x}}_\alpha - \boldsymbol{z}\|_2^2.\end{aligned}$$

Similarly, it can be reformulated as

$$\arg\min_{\hat{\boldsymbol{x}}_\alpha} \frac{\lambda_1}{\eta}\cdot\|\boldsymbol{z}\|_2^2 + \|\boldsymbol{z} - (\hat{\boldsymbol{x}}_p - \hat{\boldsymbol{x}}_\alpha + \frac{\boldsymbol{m}}{2\eta})\|_2^2. \quad (18)$$

Letting $\boldsymbol{J}_{\boldsymbol{z}} = \hat{\boldsymbol{x}}_p - \hat{\boldsymbol{x}}_\alpha + \frac{\boldsymbol{m}}{2\eta}$ and setting the gradient of the objective function in Eq. (18) to be zero, it yields

$$\boldsymbol{z} = \frac{1}{\lambda_1/\eta + 1}\cdot\boldsymbol{J}_{\boldsymbol{z}}. \quad (19)$$

3) Find $\hat{\boldsymbol{x}}_p$ to minimize the augmented Lagrangian function $\mathcal{L}(\hat{\boldsymbol{x}}_p, \hat{\boldsymbol{x}}_\alpha, \boldsymbol{z}, \boldsymbol{D}, \boldsymbol{s}, \boldsymbol{m}, \gamma, \eta)$ with other variables fixed. It gives

$$\begin{aligned}\arg\min_{\hat{\boldsymbol{x}}_p} &\lambda_2\|\hat{\boldsymbol{x}}_p - \boldsymbol{D}\boldsymbol{s}\|_2^2 + \langle \boldsymbol{m}, \hat{\boldsymbol{x}}_p - \hat{\boldsymbol{x}}_\alpha - \boldsymbol{z}\rangle \\ &+ \eta\cdot\|\hat{\boldsymbol{x}}_p - \hat{\boldsymbol{x}}_\alpha - \boldsymbol{z}\|_2^2.\end{aligned}$$

That is

$$\arg\min_{\hat{\boldsymbol{x}}_p} \frac{\lambda_2}{\eta}\cdot\|\hat{\boldsymbol{x}}_p - \boldsymbol{D}\boldsymbol{s}\|_2^2 + \|\hat{\boldsymbol{x}}_p - (\hat{\boldsymbol{x}}_\alpha + \boldsymbol{z} - \frac{\boldsymbol{m}}{2\eta})\|_2^2. \quad (20)$$

Define $\boldsymbol{J}_{\hat{\boldsymbol{x}}_p} = \hat{\boldsymbol{x}}_\alpha + \boldsymbol{z} - \frac{\boldsymbol{m}}{2\eta}$ and set the corresponding gradient in Eq. (20) to be zero. It becomes

$$\hat{\boldsymbol{x}}_p = 1 \Big/ (\frac{\lambda_2}{\eta} + 1)\cdot(\frac{\lambda_2}{\eta}\boldsymbol{D}\boldsymbol{s} + \boldsymbol{J}_{\hat{\boldsymbol{x}}_p}). \quad (21)$$

4) Find $\boldsymbol{D}$ and $\boldsymbol{s}$ to minimize the augmented Lagrangian function $\mathcal{L}(\hat{\boldsymbol{x}}_p, \hat{\boldsymbol{x}}_\alpha, \boldsymbol{z}, \boldsymbol{D}, \boldsymbol{s}, \boldsymbol{m}, \gamma, \eta)$ with other variables fixed. In fact, the objective function turns into

$$\arg\min_{\boldsymbol{D},\boldsymbol{s}} \lambda_2\|\hat{\boldsymbol{x}}_p - \boldsymbol{D}\boldsymbol{s}\|_2^2 + \gamma\cdot\|\boldsymbol{s}\|_1. \quad (22)$$

Obviously, this optimization problem in Eq. (22) is exactly the sparse signal recovery problem without the dictionary a priori in Section II-B. Inspired by the dictionary learning methodology (namely the means to learn the dictionary or basis sets of large-scale data) in [26], the corresponding solution alternatively determines $\boldsymbol{D}$ and $\boldsymbol{s}$ and thus involves two sub-procedures, namely online learning algorithm [26] and LARS-lasso algorithm [47]. Algorithm 1 provides the skeleton of this solution.

Algorithm 1 The Sparse Signal Recovery Algorithm without a Predetermined Dictionary

initialize the dictionary $\boldsymbol{D}$ as an input dictionary $\boldsymbol{D}^{(0)}$ (which could be the dictionary learned in last calling this Algorithm), the number of iterations for learning a dictionary as T, two auxiliary matrices $\boldsymbol{A}^{(0)} \in \mathcal{R}^{K\times K}$ and $\boldsymbol{B}^{(0)} \in \mathcal{R}^{K\times K}$ with all elements therein equaling zero.

1: **for** $t = 1$ to T **do**

2: Sparse coding: computing $\boldsymbol{s}^{(t)}$ using LARS-Lasso algorithm [47] to obtain

$$\boldsymbol{s}^{(t)} = \arg\min_{\boldsymbol{s}} \lambda_2\|\hat{\boldsymbol{x}}_p - \boldsymbol{D}^{(t-1)}\boldsymbol{s}\|_2^2 + \gamma\cdot\|\boldsymbol{s}\|_1. \quad (23)$$

3: Update $\boldsymbol{A}^{(t)}$ according to

$$\boldsymbol{A}^{(t)} \leftarrow \boldsymbol{A}^{(t)} + \boldsymbol{s}^{(t)}(\boldsymbol{s}^{(t)})^T.$$

4: Update $\boldsymbol{B}^{(t)}$ according to

$$\boldsymbol{B}^{(t)} \leftarrow \boldsymbol{B}^{(t)} + \hat{\boldsymbol{x}}_p(\boldsymbol{s}^{(t)})^T.$$

5: Dictionary Update: computing $\boldsymbol{D}^{(t)}$ online learning algorithm [26] to obtain

$$\begin{aligned}\boldsymbol{D}^{(t)} &= \arg\min_{\boldsymbol{D}} \lambda_2\|\hat{\boldsymbol{x}}_p - \boldsymbol{D}\boldsymbol{s}^{(t)}\|_2^2 + \gamma\cdot\|\boldsymbol{s}^{(t)}\|_1 \\ &= \arg\min_{\boldsymbol{D}} \mathrm{Tr}(\boldsymbol{D}^T\boldsymbol{D}\boldsymbol{A}^{(t)}) - 2\mathrm{Tr}(\boldsymbol{D}^T\boldsymbol{B}^{(t)}).\end{aligned} \quad (24)$$

6: **end for**

7: **return** the learned dictionary $\boldsymbol{D}^{(t)}$ and the sparse coding vector $\boldsymbol{s}^{(t)}$.

In order to update the dictionary in Eq. (24), the proposed sparse signal recovery algorithm utilizes the

concept of stochastic approximation, which is firstly introduced and mathematically proved convergent to a stationary point in [26].

On the other hand, based on the learned dictionary, the concerted effort to recover a sparse signal could be exploited. As mentioned above, the well known LARS-lasso algorithm [47], which is a forward stagewise regression algorithm and gradually finds the most suitable solution along a equiangular path among the already known predictors, is used here to solve the problem in Eq. (23). Meanwhile, it is worthwhile to note that other compressive sensing algorithms [38] could also be used here.

5) Update estimate for the Lagrangian multiplier $\boldsymbol{m}$ according to steepest gradient descent method [48], namely $\boldsymbol{m} \leftarrow \boldsymbol{m} + \eta \cdot (\hat{\boldsymbol{x}}_p - \hat{\boldsymbol{x}}_\alpha - \boldsymbol{z})$. Similarly, update estimate γ by $\gamma \leftarrow \gamma + \eta \cdot \|s\|_1$.
6) Update $\eta \leftarrow \eta \cdot \rho$.

In Algorithm 2, we summarize the steps during each iteration. Notably, without loss of generality, consider a known traffic vector $\mathbf{x}(0, \cdots, t)$ of a given cell at different moments $(0, \cdots, t)$. Then, we could estimate the α-stable related parameters according to maximum likelihood methods, quantile methods, or sample characteristic function methods in [33], [34]. Afterwards, we could conduct Algorithm 2 to predict the traffic volume at moment $t+1$. Similarly, we need re-estimate the α-stable related parameters according to methods in [33], [34], in terms of the traffic vector $\mathbf{x}(0, \cdots, t+1)$, and perform Algorithm 2 to predict the traffic volume at moment $t+2$. It can be observed that, compared to Algorithm 1, which is an application of the lines in [26], Algorithm 2 is made up of some additional iterative procedures to procure the parameters without known a priori. Besides, most steps involved in Algorithm 2 are deterministic vector computations and thus computationally efficient. Therefore, the whole framework could effectively yield the traffic forecasting results.

Algorithm 2 The Dictionary Learning-based Alternating Direction Method

initialize $\hat{\boldsymbol{x}}_p$, $\hat{\boldsymbol{x}}_\alpha$, $\boldsymbol{z}$, $\boldsymbol{D}$, $\boldsymbol{s}$, $\boldsymbol{m}$, γ, η according to $\hat{\boldsymbol{x}}_p^{(0)}$, $\hat{\boldsymbol{x}}_\alpha^{(0)}$, $\boldsymbol{z}^{(0)}$, $\boldsymbol{D}^{(0)}$, $\boldsymbol{s}^{(0)}$, $\boldsymbol{m}^{(0)}$, $\gamma^{(0)}$, $\eta^{(0)}$, and the number of iterations T. Compute $\tilde{\boldsymbol{x}}_\alpha$ according to α-stable model based linear prediction algorithms [28], [34].

for $t = 1$ to T **do**
  Update $\hat{\boldsymbol{x}}_\alpha$ according to $\hat{\boldsymbol{x}}_\alpha^{(t)} \leftarrow \frac{1}{\eta^{(t-1)}+1} \cdot \left(\tilde{\boldsymbol{x}}_\alpha + \eta^{(t-1)} \cdot \left(\hat{\boldsymbol{x}}_p^{(t-1)} - \boldsymbol{z}^{(t-1)} + \frac{\boldsymbol{m}^{(t-1)}}{2\eta^{(t-1)}}\right)\right)$.
  Update $\boldsymbol{z}$ according to $\boldsymbol{z}^{(t)} \leftarrow \frac{1}{\lambda_1/\eta^{(t-1)}+1} \cdot \left(\hat{\boldsymbol{x}}_p^{(t-1)} - \hat{\boldsymbol{x}}_\alpha^{(t)} + \frac{\boldsymbol{m}^{(t-1)}}{2\eta^{(t-1)}}\right)$.
  Update $\hat{\boldsymbol{x}}_p$ according to $\hat{\boldsymbol{x}}_p^{(t)} \leftarrow 1 \Big/ (\frac{\lambda_2}{\eta^{(t-1)}} + 1) \cdot \left(\frac{\lambda_2}{\eta^{(t-1)}} \boldsymbol{D}^{(t-1)} \boldsymbol{s}^{(t-1)} + \hat{\boldsymbol{x}}_\alpha^{(t)} + \boldsymbol{z}^{(t)} - \frac{\boldsymbol{m}^{(t-1)}}{2\eta^{(t-1)}}\right)$.
  Update $\boldsymbol{D}$ and $\boldsymbol{s}$ according to sparse signal recovery algorithm (i.e., Algorithm 1). In particular, use two sub-procedures namely online learning algorithm [26] and LARS-lasso algorithm [47] to update $\boldsymbol{D}$ and $\boldsymbol{s}$, respectively.
  Update $\boldsymbol{m}$ according to $\boldsymbol{m}^{(t)} \leftarrow \boldsymbol{m}^{(t-1)} + \eta^{(t-1)} \cdot (\hat{\boldsymbol{x}}_p^{(t)} - \hat{\boldsymbol{x}}_\alpha^{(t)} - \boldsymbol{z}^{(t)})$.
  Update γ by $\gamma^{(t)} \leftarrow \gamma^{(t-1)} + \eta^{(t-1)} \cdot \|s^{(t)}\|_1$.
  Update η by $\eta^{(t)} \leftarrow \eta^{(t-1)} \cdot \rho$, here ρ is an iteration ratio.
end for
return the predicted traffic vector $\hat{\boldsymbol{x}}_p$.

V. Performance Evaluation

We validate the prediction accuracy improvement of our proposed framework in Algorithm 2 relying on the practical traffic dataset. Specifically, we choose the traffic load records of these three service types of applications generated in 113 cells within a randomly selected region from Dataset 1. Moreover, we intentionally divide the traffic dataset into two part. One is used to learn and distill the parameters related to traffic characteristics, and the other part is to conduct the experiments to verify and validate the accuracy of the proposed framework in Algorithm 2. Specifically, we compare our prediction $\hat{\boldsymbol{x}}_p$ with the ground truth $\boldsymbol{x}$ in terms of the normalized mean absolute error (NMAE) [11], which is defined as

$$\text{NMAE} = \frac{\sum_{i=1}^{N} |\hat{x}_p(i) - x(i)|}{\sum_{i=1}^{N} |x(i)|}. \tag{25}$$

Besides, we utilize LARS-lasso algorithm [47] and OMP [38] respectively to perform the sparse recovery process, and choose the α-stable model based (36,10,1)-linear prediction algorithm in Section IV-A as the performance baseline. In other words, we would exploit traffic records in the last three hours to train the parameters of α-stable models and predict traffic loads in the next 5 minutes.

As described in Algorithm 2, most of the parameters could be set easily and tuned dynamically within the framework. Therefore, we can benefit from this advantage and only need to examine the performance impact of few parameters, namely λ_1, λ_2, γ and η, by dynamically adjusting them. By default, we set $\lambda_1 = 10$, $\lambda_2 = 1$, $\gamma = 1$ and $\eta = 10^{-4}$, and the number of iterations[8] in Algorithm 2 and sparse signal recovery algorithm (i.e., Algorithm 1) to be 20 and 3, respectively. We also use (36,10,1)-linear prediction algorithm in Section IV-A to provide the "coarse" prediction results $\tilde{\boldsymbol{x}}_\alpha$, and examine the corresponding performance improvement of the proposed ADM framework with different sparse signal recovery algorithm (i.e., LARS-Lasso algorithm [47] and OMP algorithm [38]). Besides, we impose no prior constraints on $\boldsymbol{D}$, $\boldsymbol{s}$, and $\boldsymbol{z}$, and set them as zero vectors.

Fig. 7 gives a performance comparison in terms of NMAE, when we predict the traffic loads under the default settings. In order to provide a more comprehensive comparison, the simulations run in both busy moments (i.e., 9AM, 12PM, and 4PM) and idle ones (i.e., 6AM and 9PM) of one day. As

[8]Actually, throughout the literature, the stopping criterion could be a predefined sufficiently large number, just as we did in our manuscript. Meanwhile, the iterative process could stop if the results between two consecutive iterations are sufficiently small. In our study, these predefined few iterations could still yield satisfactory good prediction performance.

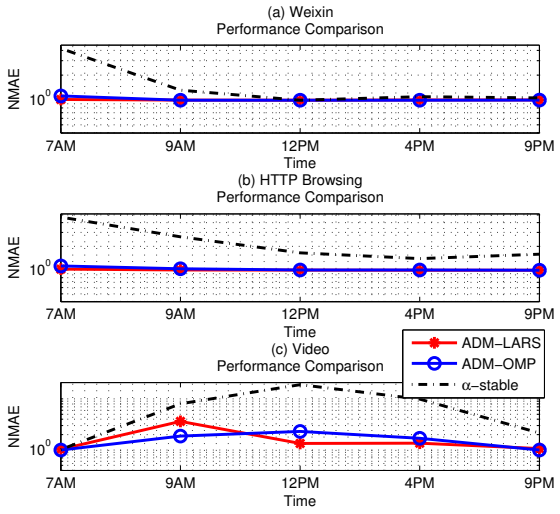

Fig. 7. The performance comparison between the proposed ADM framework with different sparse signal recovery algorithms (i.e., LARS-Lasso and OMP), and the α-stable model based (36,10,1)-linear prediction algorithm.

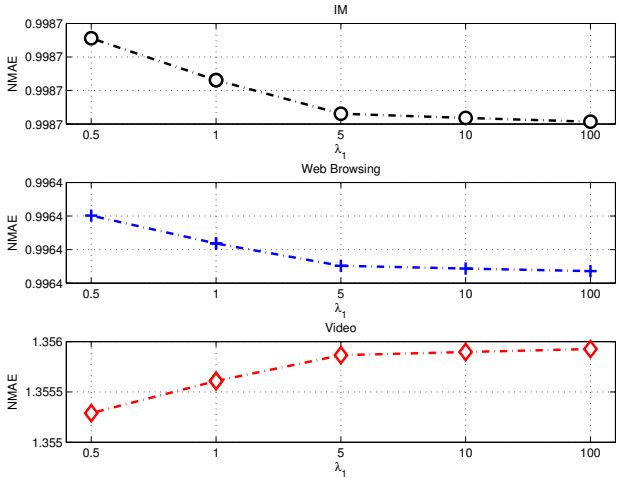

Fig. 8. The performance variation with respect to λ_1 for the proposed ADM framework with LARS-Lasso algorithm.

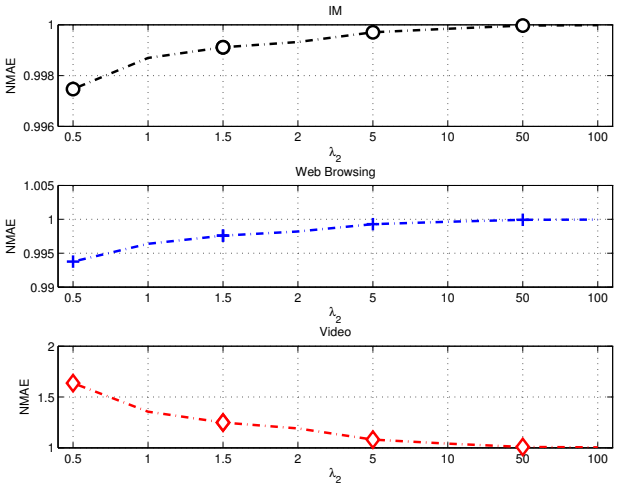

Fig. 9. The performance variation with respect to λ_2 for the proposed ADM framework with LARS-Lasso algorithm.

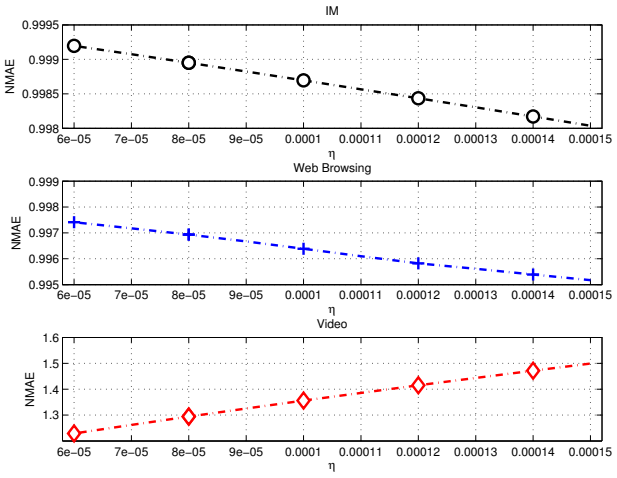

Fig. 10. The performance variation with respect to η for the proposed ADM framework with with LARS-Lasso algorithm.

depicted in Fig. 7, the proposed framework significantly outperforms the α-stable model based (36,10,1)-linear prediction algorithm. In particular, the NMAE of the proposed framework can be as 12% small (e.g., prediction for 12PM video traffic) as that for the classical linear algorithm. This performance improvement can be interpreted as the gain by exploiting the embedded sparsity in traffic and taking account of the originally existing prediction error of linear prediction. Moreover, it can be observed that in most cases, different sparse signal recovery algorithm has little impact on the prediction accuracy. Therefore, the applications of the proposed framework could pay little attention to the involved sparsity methods.

Next, we further evaluate the performance of our proposed ADM framework with LARS-Lasso algorithm and present more detailed sensitivity analyses in Fig. 8, Fig. 9, and Fig. 10, by varying λ_1, λ_2, and η. Fig. 8 shows that the prediction accuracy nearly stays the same irrespective of λ_1. This means that the noise component has limited contribution to the corresponding performance. It also implies that the choice of λ_1 could be flexible when we apply the framework in practice.

According to Fig. 9, the influence of λ_2 is comparatively more obvious and even diverges for different service types. Specifically, a larger λ_2 has a slightly negative impact on predicting the traffic loads for IM and web browsing service, but it contributes to the prediction of video service. Recalling the sparsity analyses in Section III-B, video service demonstrates the strongest sparsity. Meanwhile, a larger λ_2 means to put more emphasis on the importance of sparsity. Hence, compared to the other two service types, it becomes natural to result in a better performance for video service, when λ_2 becomes larger. It's worthwhile to note here that, in Eq. (14), λ_2 and γ are coupled together as well, and should have inverse performance impact. Therefore, due to the space limitation, the performance impact of γ is omitted here.

Fig. 10 depicts the performance variation with respect to η, which is similar to that with respect to λ_2. But, a larger

η has a positive impact on predicting the traffic loads for IM and web browsing service, but it degrades the prediction performance of video service. This phenomenon is potentially originated from the very distinct characteristics of these three services types (e.g., different α-stable models' parameters and different sparsity representation), and needs a further careful investigation. However, it safely comes to the conclusion that the proposed framework provides a superior and robust performance than the classical linear algorithm.

VI. Conclusion

In this paper, we collected the application-level traffic data from one operator in China. With the aid of this practical traffic data, we re-confirmed several important statistical characteristics like temporally α-stable modeled property and spatial sparsity. Afterwards, we proposed a traffic prediction framework, which takes advantage of the already known traffic knowledge to distill the parameters related to aforementioned traffic characteristics and forecasts future traffic results bearing the same characteristics. We also developed a dictionary learning-based alternating direction method to solve the framework, and manifested the effectiveness and robustness of our algorithm through extensive simulation results.

References

[1] Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2012–2017," Feb. 2013. [Online]. Available: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html

[2] R. Li, Z. Zhao, X. Zhou, and H. Zhang, "Energy savings scheme in radio access networks via compressive sensing-based traffic load prediction," *Trans. Emerg. Telecommun. Technol. (ETT)*, vol. 25, no. 4, pp. 468–478, Apr. 2014.

[3] U. Paul, M. Buddhikot, and S. Das, "Opportunistic traffic scheduling in cellular data networks," in *Proc. IEEE DySPAN 2012*, Bellevue, WA, USA, 2012, pp. 339–348.

[4] P. Romirer-Maierhofer, M. Schiavone, and A. DAlconzo, "Device-Specific Traffic Characterization for Root Cause Analysis in Cellular Networks," in *Traffic Monitoring and Analysis*, ser. Lecture Notes in Computer Science, M. Steiner, P. Barlet-Ros, and O. Bonaventure, Eds. Springer International Publishing, Apr. 2015, no. 9053, pp. 64–78. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-319-17172-2_5

[5] Z. Niu, Y. Wu, J. Gong, and Z. Yang, "Cell zooming for cost-efficient green cellular networks," *IEEE Commun. Mag.*, vol. 48, no. 11, pp. 74–79, Nov. 2010.

[6] Z. Niu, "TANGO: traffic-aware network planning and green operation," *IEEE Wireless Commun.*, vol. 18, no. 5, pp. 25 –29, Oct. 2011.

[7] R. Li, Z. Zhao, X. Zhou, J. Palicot, and H. Zhang, "The prediction analysis of cellular radio access network traffic: From entropy theory to networking practice," *IEEE Commun. Mag.*, vol. 52, no. 6, pp. 238–244, Jun. 2014.

[8] R. G. Baraniuk, "Compressive sensing [lecture notes]," *IEEE Signal Process. Mag.*, vol. 24, no. 4, pp. 118–121, Jul. 2007. [Online]. Available: http://omni.isr.ist.utl.pt/~aguiar/CS_notes.pdf

[9] J. Romberg and M. Wakin, "Compressed sensing: A tutorial," in *Proc. IEEE SSP Workshop 2007*, Madison, Wisconsin, Aug. 2007. [Online]. Available: http://people.ee.duke.edu/~willett/SSP//Tutorials/ssp07-cs-tutorial.pdf

[10] D. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 4036–4048, Apr. 2006.

[11] Y.-C. Chen, L. Qiu, Y. Zhang, G. Xue, and Z. Hu, "Robust Network Compressive Sensing," in *Proc. ACM Mobicom 2014*, Maui, Hawaii, USA, Sep. 2014.

[12] I. . B. W. A. W. Group, "IEEE 802.16m Evaluation Methodology Document (EMD)," Jul. 2008. [Online]. Available: http://ieee802.org/16

[13] B. Zhou, D. He, Z. Sun, and W. H. Ng, "Network traffic modeling and prediction with ARIMA/GARCH," in *Proc. HET-NETs Conf.*, Ilkley, UK, Jul. 2005, 00033.

[14] O. Cappe, E. Moulines, J.-C. Pesquet, A. Petropulu, and Y. Xueshi, "Long-range dependence and heavy-tail modeling for teletraffic data," *IEEE Signal Process. Mag.*, vol. 19, no. 3, pp. 14–27, May 2002.

[15] F. Ashtiani, J. Salehi, and M. Aref, "Mobility modeling and analytical solution for spatial traffic distribution in wireless multimedia networks," *IEEE J. Sel. Area. Comm.*, vol. 21, no. 10, pp. 1699 – 1709, Dec. 2003.

[16] K. Tutschku and P. Tran-Gia, "Spatial traffic estimation and characterization for mobile communication network design," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 5, pp. 804–811, Jun. 1998.

[17] L. Xiang, X. Ge, C. Liu, L. Shu, and C. Wang, "A new hybrid network traffic prediction method," in *Proc. IEEE Globecom 2010*, Miami, Florida, USA, Dec. 2010.

[18] X. Ge, S. Yu, W.-S. Yoon, and Y.-D. Kim, "A new prediction method of alpha-stable processes for self-similar traffic," in *Proc. IEEE Globecom 2004*, Dallas, Texas, USA, Nov. 2004.

[19] M. Z. Shafiq, L. Ji, A. X. Liu, J. Pang, and J. Wang, "Geospatial and temporal dynamics of application usage in cellular data networks," *IEEE Trans. Mob. Comput.*, 2014. [Online]. Available: http://myweb.uiowa.edu/mshafiq/files/spatialApp_TMC.pdf

[20] M. Crovella and A. Bestavros, "Self-similarity in World Wide Web traffic: evidence and possible causes," *IEEE/ACM Trans. Netw.*, vol. 5, no. 6, pp. 835–846, Dec. 1997.

[21] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of ethernet traffic," *IEEE/ACM Trans. Netw.*, vol. 2, no. 1, pp. 1–15, Feb. 1994.

[22] Y. Zhang, M. Roughan, W. Willinger, and L. Qiu, "Spatio-temporal compressive sensing and internet traffic matrices," in *Proc. ACM SIGCOMM 2009*, Barcelona, Spain, Aug. 2008.

[23] A. Soule, A. Lakhina, and N. Taft, "Traffic matrices: balancing measurements, inference and modeling," in *Proc. ACM SIGMETRICS 2005*, Banff, Alberta, Canada, Jun. 2005.

[24] M. C. Falvo, M. Gastaldi, A. Nardecchia, and A. Prudenzi, "Kalman filter for short-term load forecasting: an hourly predictor of municipal load," in *Proc. IASTED ASM 2007*, Palma de Mallorca, Spain, Aug. 2007.

[25] R. Li, Z. Zhao, Y. Wei, X. Zhou, and H. Zhang, "GM-PAB: a grid-based energy saving scheme with predicted traffic load guidance for cellular networks," in *Proc. IEEE ICC 2012*, Ottawa, Canada, Jun. 2012.

[26] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online Learning for Matrix Factorization and Sparse Coding," *J. Mach. Learn. Res.*, vol. 11, pp. 19–60, Mar. 2010.

[27] U. Paul, L. Ortiz, S. R. Das, G. Fusco, and M. M. Buddhikot, "Learning probabilistic models of cellular network traffic with applications to resource management," in *Proc. IEEE DySPAN 2014*, McLean, VA, USA, Apr. 2014.

[28] J. B. Hill, "Minimum Dispersion and Unbiasedness: 'Best' Linear Predictors for Stationary ARMA a-Stable Processes," University of Colorado at Boulder, Discussion Papers in Economics Working Paper No. 00-06, Sep. 2000.

[29] A. Karasaridis and D. Hatzinakos, "Network heavy traffic modeling using alpha-stable self-similar processes," *IEEE Trans. Commun.*, vol. 49, no. 7, pp. 1203–1214, Jul. 2001.

[30] F. Qian, Z. Wang, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, "Characterizing radio resource allocation for 3g networks," in *Proc. ACM SIGCOMM 2010*, New York, NY, USA, May 2010.

[31] F. P. Tso, J. Teng, W. Jia, and D. Xuan, "Mobility: A Double-Edged Sword for HSPA Networks: A Large-Scale Test on Hong Kong Mobile HSPA Networks," in *Proc. ACM Mobihoc 2010*, Sep. 2010.

[32] G. Samorodnitsky, *Stable Non-Gaussian Random Processes: Stochastic Models with Infinite Variance*. New York: Chapman and Hall/CRC, Jun. 1994. [Online]. Available: http://www.amazon.com/Stable-Non-Gaussian-Random-Processes-Stochastic/dp/0412051710

[33] J. R. Gallardo, D. Makrakis, and L. Orozco-Barbosa, "Use of alpha-stable self-similar stochastic processes for modeling traffic in broadband networks," in *Proc. SPIE Conf. P. Soc. Photo-Opt. Ins*, vol. 3530, Boston. Massachusetts, Nov. 1998.

[34] X. Ge, G. Zhu, and Y. Zhu, "On the testing for alpha-stable distributions of network traffic," *Comput. Commun.*, vol. 27, no. 5, pp. 447–457, Mar. 2004.

[35] W. Song and W. Zhuang, "Resource Reservation for Self-Similar Data Traffic in Cellular/WLAN Integrated Mobile Hotspots," in *Proc. IEEE ICC 2010*, Cape Town, South Africa, May 2010.

[36] J. C.-I. Chuang and N. Sollenberger, "Spectrum resource allocation for wireless packet access with application to advanced cellular Internet service," *IEEE J. Sel. Area. Comm.*, vol. 16, no. 6, pp. 820–829, Aug. 1998.

[37] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, Aug. 1998.

[38] Y. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in *Proc. ACSSC 1993*, Pacific Grove, CA, USA, Nov. 1993.

[39] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[40] X. Zhou, Z. Zhao, R. Li, Y. Zhou, and H. Zhang, "The Predictability of Cellular Networks Traffic," in *Proc. IEEE ISCIT 2012*, Gold Coast, Australia, Oct. 2012.

[41] J. H. McCulloch, "Simple consistent estimators of stable distribution parameters," *Commun. Stat. Simulat.*, vol. 15, no. 4, pp. 1109–1136, Jan. 1986.

[42] X. Zhou, Z. Zhao, R. Li, Y. Zhou, J. Palicot, and H. Zhang, "Understanding the Nature of Social Mobile Instant Messaging in Cellular Networks," *IEEE Commun. Lett.*, vol. 18, no. 3, pp. 389 – 392, Mar. 2014.

[43] A. N. Kolmogorov, K. L. Chung, and B. V. Gnedenko, *Limit distributions for sums of independent random variables*, rev. ed. ed. Reading, Mass: Addison-Wesley, 1968. [Online]. Available: https://openlibrary.org/books/OL19738039M/Limit_distributions_for_sums_of_independent_random_variables

[44] R. Fang, T. Chen, and P. C. Sanelli, "Towards robust deconvolution of low-dose perfusion CT: sparse perfusion deconvolution using online dictionary learning," *Med. Image Anal.*, vol. 17, no. 4, pp. 417–428, May 2013.

[45] Wikipedia, "Augmented Lagrangian method," Oct. 2014. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Augmented_Lagrangian_method

[46] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK ; New York: Cambridge University Press, Mar. 2004.

[47] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Ann. Stat.*, vol. 32, pp. 407–499, 2004.

[48] R. Sutton and A. Barto, *Reinforcement learning: An introduction*. Cambridge University Press, 1998. [Online]. Available: http://webdocs.cs.ualberta.ca/~sutton/book/ebook/