Multitier Fog Computing With Large-Scale IoT Data Analytics for Smart Cities

Jianhua He, *Senior Member, IEEE*, Jian Wei, Kai Chen, Zuoyin Tang, Yi Zhou, and Yan Zhang, *Senior Member, IEEE*

***Abstract*—Analysis of Internet of Things (IoT) sensor data is a key for achieving city smartness. In this paper a multitier fog computing model with large-scale data analytics service is proposed for smart cities applications. The multitier fog is consisted of ad-hoc fogs and dedicated fogs with opportunistic and dedicated computing resources, respectively. The proposed new fog computing model with clear functional modules is able to mitigate the potential problems of dedicated computing infrastructure and slow response in cloud computing. We run analytics benchmark experiments over fogs formed by Rapsberry Pi computers with a distributed computing engine to measure computing performance of various analytics tasks, and create easy-to-use workload models. Quality of services (QoS) aware admission control, offloading, and resource allocation schemes are designed to support data analytics services, and maximize analytics service utilities. Availability and cost models of networking and computing resources are taken into account in QoS scheme design. A scalable system level simulator is developed to evaluate the fog-based analytics service and the QoS management schemes. Experiment results demonstrate the efficiency of analytics services over multitier fogs and the effectiveness of the proposed QoS schemes. Fogs can largely improve the performance of smart city analytics services than cloud only model in terms of job blocking probability and service utility.**

***Index Terms*—Data analytics, fog computing, Internet of Things (IoT), quality of services (QoS), Raspberry Pi, smart cities, Spark.**

I. INTRODUCTION

WITH greater access to public resource, such as education and health and more job opportunities, more and more people leave villages to live in cities. A rapid urbanization of the world's population was witnessed in the last decade. The global proportion of urban population was reported by United Nation to be 49% (3.2 billion) in 2005, and is expected to rise to 60% (4.9 billion) by 2030. However, the fast increasing urban population exacerbates the existing problems faced by modern cities, such as traffic congestion, pollution, low quality public services, insufficient public resource and budget for health and education. Smart cities is an ambitious vision to tackle the above city problems by making more efficient use of city resource and infrastructure and improve the quality of life for citizens. It is proposed to capitalize on the latest technology advances of Internet of Things (IoT), communication and networking, computing and big data analytics, to provide smartness on many sectors, such as transport and traffic management, health care, water, energy, and waste management.

IoT provides a vital instrument to sense and control the physical city environment [1]–[3]. IoT data analytics is a key in achieving and delivering the city smartness. With virtually unlimited computing and storage resource, clouds are thought to be the natural places for big data analytics [4], [5] and can provide easy management of IoT services [5]. However, with expansion of IoT systems and emerging big data from smart city applications and fast response requirement from applications, such as public safety and emergency response, there are problem with cloud-based solution due to real-time and reliable transport of enormous IoT traffic over communication networks, especially wireless access networks, which is well known with features of low bandwidth and high communication cost.

There are several edge computing models (including cloudlet, mobile edge computing and fog computing) proposed to tackle the data analytics problems in the cloud computing-based solution [3], [6]–[9]. The principle is moving computing and caching resources and analytics services closer to the things, where data is generated. However, it is noted that for the cloudlet, mobile edge computing and fog radio access networks-based solutions [7], [9], computing facilities are provided by the third party at fixed locations, which can be powerful for big data analytics but may not be flexible enough for on demand deployment when there is a need. And the wireless access bottleneck problem still exist for the IoT data traffic. Fog computing is gaining increasing research and development momentums but still in a very early stage. According to [3] and [8] end devices, such as smart phones and Wi-Fi access points can be used for data analytics when available and needed. But they are expected to take only very simple time-sensitive data processing tasks. Less time sensitive analysis and big data analytics are performed in the clouds. The original fog computing model does not solve the large scale data analytics problems faced by the IoT applications.

Manuscript received January 31, 2017; revised May 30, 2017; accepted June 14, 2017. Date of publication July 11, 2017; date of current version April 10, 2018. This work was supported by EU FP7 Project DETERMINE under Grant FP7-PEOPLE-2012-IRSES 318906. *(Corresponding author: Yi Zhou.)*

J. He, J. Wei, and Z. Tang are with the School of Engineering and Applied Science, Aston University, Birmingham B4 7ET, U.K. (e-mail: j.he7@aston.ac.uk; weij2@aston.ac.uk; z.tang1@aston.ac.uk).

K. Chen is with the Institute of Image Communication and Network Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: kchen@sjtu.edu.cn).

Y. Zhou is with the Computer Science Department, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: 744623881@qq.com).

Y. Zhang is with the Department of Informatics, University of Oslo, 0316 Oslo, Norway (e-mail: yanzhang@ieee.org).

Digital Object Identifier 10.1109/JIOT.2017.2724845

In addition, its network architecture and service model are not clearly specified.

It is noted that within the last several years we witnessed an explosive growth of mobile smart personal devices (e.g., smart phones and tablets). These smart personal devices with increasingly available computing and communication resources can be utilized to form small ad-hoc fogs. On the other hand, the number of small cell base stations and Wi-Fi-based home hotspots are also expected to grow fast. Dedicated computing resource can be deployed alongside these small base stations and home hotspots in addition to the macro cellular base stations to form dedicated fogs. With properly design quality of services (QoS) management schemes these multitier fogs can complement to each other and remote clouds to provide more effective and prompt response to fast changing circumstances of smart cities.

In this paper we propose a multitier fog computing-based large scale data analytics service for smart city applications. There are three main contributions.

1) A multitier fogs computing framework is proposed, which include both ad-hoc fogs with distributed opportunistic computing resources and dedicated fogs with specifically deployed computing resources. The fogs can utilize opportunistic and dedicated computing resources to mitigate the problem of huge initial fog infrastructure investment. Large scale analytics service can be run over multitier fog computing system with support of distributed computing engines.
2) Analytics benchmarking over multifogs is run over small size Rapspberry Pi computers with Spark computing engine to create workload models of various analytics jobs. In the existing offloading schemes the workload of computing jobs were usually represented by the instructions per second (e.g., [6]). By contrast, in this paper easy-to-use job level working load models are created and utilized in the design of practical QoS aware management schemes.
3) QoS aware service and resource management schemes are designed and developed for admission control, offloading and resource allocation, to provide real-time analytics services to smart city applications and improve utility for fog computing operators. The network bandwidth and latency, communication and computing costs, and computing time are all taken into account. to satisfy the QoS constraint of real time job completion and improve computing utility for the QoS aware analytics services. To the best of our knowledge, QoS issues for mobile edge computing and fog computing have rarely been touched in the literature.

In the rest of this paper we present a framework of multitier fog computing system for smart city applications and the large-scale analytics service model in Section II. In Section III design and results on the benchmarking experiments over both ad-hoc fogs and dedicated fogs are reported. Design of QoS aware service and resource management schemes is presented in Section IV. Analytics services and the QoS schemes are evaluated and QoS performance is analyzed. Section V concludes this paper.

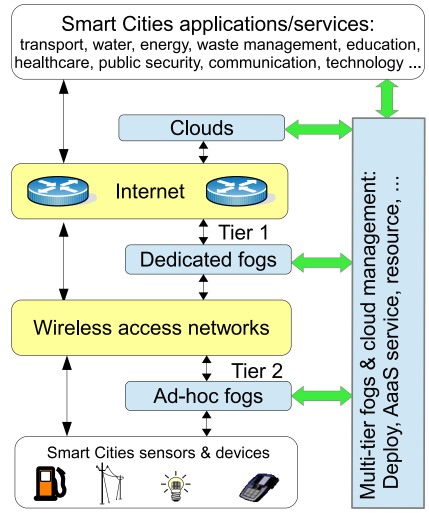


Fig. 1. Multitier fog cloud architecture.

II. Multitier Fogs Model and Scalable Analytics Service for Smart Cities

A. Proposed Multitier Fog Computing Model

In the Cisco fog computing model, fog aggregation nodes are not clearly defined. For example, it is not clear, where these nodes are located, how much computation power and storage resources they have, and by whom they may be deployed. The fog nodes are expected to analyze and act on the large volume of data generated by thousands of things across a large geographic area in less than one second [3]. It is not likely that the small size fog nodes like smart phones and video cameras can deliver the expected analytics services. But there is no discussions in the literature if dedicated fog nodes can complete the tasks. If the fog computing relies on the dedicated fog nodes and fog aggregation nodes for fast and reliable data analytics services, then there is little difference between fog computing and cloudlet models.

On the other hand, the fog nodes in the Cisco fog computing model are not expected to take complex and advanced data analytics. The majority of IoT data traffic still goes to the traditional data centers for big data analytics, which does not solve the bandwidth and prompt response problems faced by real time smart city applications. In addition, the connections of IoT devices to the Internet may not exist or have very limited network bandwidth, such as in the scenarios of emergency response and anti-terrorism events. Under theses conditions data analytics services for smart city applications may not be effectively delivered through the public clouds.

We propose a new multitier fogs computing model for smart city applications, which includes both ad-hoc fogs and dedicated fogs. Fig. 1 presents the architecture for the multitier fog computing model. In the hierarchical architecture, the Tier 1 fogs are dedicated fogs, which include the MEC and fogs supported by the dedicated routers and cellular network base stations. Tier 2 fogs are ad-hoc fogs, which are formed by opportunistic devices with computing and networking resources, such as smart phones, laptops, and vehicles.

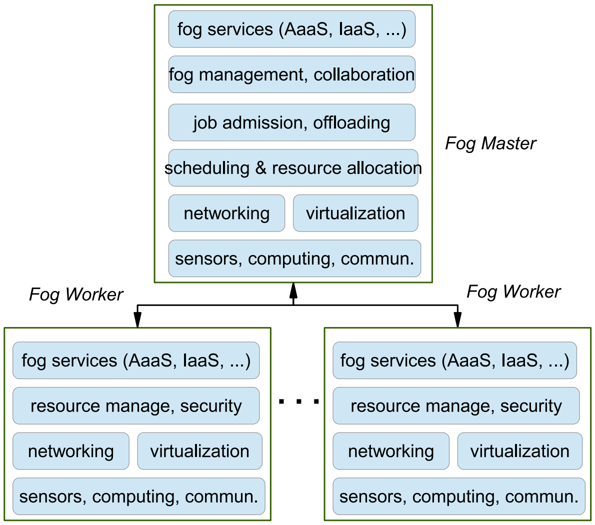


Fig. 2. Function model for fog nodes.

Fog nodes can share unused computing resources to provide data analytics services for both IoT applications and mobile applications. They can participate in a hierarchical cloud computing system, working with traditional remote clouds and optional cloudlets. They can also work in a stand alone mode.

With data analytics services from multitier fogs, analytics results are sent to the interested users of the analytics services. A large volume of IoT data from smart city applications may not need to be sent to the remote clouds. Therefore, the response latency and bandwidth consumption problems could be solved.

B. Fog Functional Model

Next, we present the function model for the fog nodes. Each fog is formed by a cluster of computers with a pool of computing resources. There are two types of fog nodes, i.e., fog master and fog worker. The functionalities of these nodes are illustrated by the functional models shown in Fig. 2.

In the ad-hoc fogs, fog workers are usually interconnected devices which join the fogs by invitation, e.g., from fog masters. The fog workers are responsible for sharing their computing resources, undertaking computing jobs, monitoring and reporting available computing and communication resources to fog masters, etc.

Each fog has at least one fog master. Multiple masters can be present in one fog for improved reliability. The masters may physically co-locate with the normal fog workers, or locate separately. The masters have the main responsibilities, such as fog creation, service management, and job scheduling.

1) Resource Module: The resource module is at the bottom of the function models for both fog master and fog workers. It represents physical resources of fog nodes, which may include sensing resources, computing resources and communication resources for connection to other fog nodes. It is noteworthy that apart from Wi-Fi technology, other communication technologies, such as cellular radio and visible light communication technologies can also be used for fog node communication.

2) Networking and Virtualization Module: Networking is a critical part of fog, especially for the scenarios, where ad-hoc fogs nodes are mobile and the wireless link bandwidth is limited. The fog master should keep tracking the mobility and network connection of the fog workers, and adaptively allocate the computing tasks to the fog workers to maximize the computing QoS.

In the fogs resource virtualization is optional but very important for the fog nodes which may have their own heavy computing tasks. With virtualization a part of computing resources can be reserved for the local computing tasks. And fog computing tasks can be run only in the isolated resources, by which local computing and security performance are ensured. The existing virtualization technologies can be applied with modifications for both ad-hoc and dedicated fogs.

3) Fog and Resource Management Module: Fogs can be formed on demand and managed by fog master nodes. Each fog has a life cycle of formation, maintenance, and release. Fog workers are responsible of monitoring and reporting computing resources and communication conditions to fog masters. Fog masters maintain the status of the available computing resources and communication conditions of the members in the fogs. Special incentive and reward schemes can be applied by fog masters to encourage interconnected devices to join fogs and share their unused computing resources. It is noted that mobility and security can have large impact on ad-hoc fogs. With the centralized fog and resource management framework, fog worker mobility and security could be handled effectively to achieve high level QoS.

4) Job Admission and Scheduling Module: When a computing job request is received (from smart city applications or other IoT applications), a fog master needs to assess the computing resources required to complete the job, and admit or reject the job request according to the available compute resources. If a job is accepted, it is scheduled to run over one or more fog workers depending on their available compute resources and network conditions. The fog master may communicate with other fog masters to jointly work on analytics tasks, or make decisions on offloading jobs to other fogs or remote clouds.

5) Services Module: There are three standard service models provided by traditional clouds, namely infrastructure as a service, platform as a service, software as a service. If the fog nodes are static and have powerful computing resources, a large computation resource pool can be created for fogs and the standard cloud service models can be offered by the fogs. However, due to the limitations on the computing power, bandwidth of wireless connections, and mobility of fog nodes, ad-hoc fogs may not be ideal to provide these standard cloud computing services. We propose to provide large scale analytics service over fogs. With the analytics service model the users of IoT applications can request analytics services from fog masters. The masters analyze the analytics service request, choose the required analytics algorithm and computing engine, and assess the service requirements on computing and communication resources. If the service request is admissible, fog member nodes and computing resources are scheduled to provide the service. Multiple fog member nodes

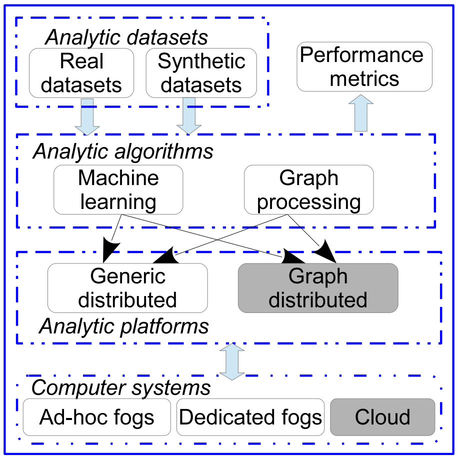


Fig. 3. Overall benchmarking framework.

may work collectively with distributed computing engines to provide advanced analytics if needed.

III. Benchmarking Experiments for Analytics Applications Over Fogs

As the multitier fog and cloud systems will undertake various analytics services from smart city applications with diverse QoS requirement, it is important to design and implement QoS aware service and resource management schemes for the analytics services. However, in order to do so, a key is to measure and model the workloads of different analytics services over the fogs with various computing and communication resources. In this section we present benchmarking experiments over ad-hoc fogs (A-Fogs) and dedicated fogs (D-Fogs), to provide basis for QoS scheme design.

A. Overall Experiment Methodology

For the benchmarking of analytics systems there could be three major dimensions of diversities that need to consider [10]: 1) analytics computing platform; 2) analytics algorithm; and 3) analytics job dataset. In our previous work, we have performed an intensive benchmarking of analytics systems over both private and public clouds, with various computing platforms and analytics algorithms. It was found from our previous work that Spark and GraphLab perform the best over the other computing platforms [11]–[13]. As GraphLab has not been updated for a long time, we use Spark as the only computing platform for benchmarking. The overall benchmarking experiment framework is shown in Fig. 3.

For the analytics algorithms, we present only experiments with logistic regression (LR) and support vector machine (SVM) for demonstration purpose only. LR and SVM are two typical machine learning algorithms for classification applications, which identify the category an object belongs to. It is noted that we have tested fog computing performance with various analytics algorithms as done in [10]. It is trivial to include more analytics algorithms and computing platforms to the benchmarking experiments.

TABLE I
Summary of Datasets for Ad-Hoc Fog Experiments

Application	Datasets	Size (MB)	# of vertices (10^6)
LR	DS-A-1	58.3	1
	DS-A-2	145.8	2.5
	DS-A-3	291.6	5
	DS-A-4	583.3	10
	DS-A-5	1770	30
SVM	DS-A-1	61.3	1
	DS-A-2	153.3	2.5
	DS-A-3	306.6	5
	DS-A-4	590	10
	DS-A-5	1770	30

B. Computing System Setup for A-Fog Benchmarking

In the benchmarking with A-Fogs, we consider a pool of computing resources with one desktop PC and eight Raspberry Pi three credit card sized micro computers, which forms an A-Fog environment [14]. The Raspberry Pis are connected to a Wi-Fi ad hoc network through their built-in wireless 802.11 module. One of the computers acts as fog master, while the rest act as fog workers. Virtual machines (VMs) are installed on the computers and each is allocated 700 MB RAM. Spark with the latest version 2.0 is installed in the VMs. Analytics job requests are sent from one of the fog nodes to the master node, which dispatches the jobs to the fog member nodes. Job completion time and resource consumption of the analytics jobs over Spark are recorded and used in the QoS aware resource management.

Raspberry Pi is a single-board computer with a 1.2 GHz 64-bit quad-core ARMv8 CPU and 1 GB RAM [14]. A 32 GB micro SD card with operating system Raspbian installed is slotted on each machine. Raspberry Pi has the features of low cost, low power consumption, small size but still good computing power. It has been used for many cost-effective entertainment, surveillance, mobile, and IoT applications. In addition computing power and storage of Raspberry Pi have the similar features of A-Fog nodes, such as smart phones, tablets, but has a better user-friendly programming environment. Therefore, Raspberry Pi is selected in our benchmarking instead of laptops.

Spark is an open source fast and general distributed computing engine for large scale data analytics [11]. It is very popular for big data analytics with significant performance enhancement over Hadoop [12], [13]. Spark has been evaluated and mainly used in large centrally controlled computer clusters. To the best of our knowledge, it has not been tested and evaluated in highly resource (computing and communications) constrained distributed computing environments.

Experiment datasets for analytics jobs are generated by the Spark datasets generator. The synthetic datasets are used for performance evaluation in this paper mainly for easy control of the dataset size. Datasets generated from real IoT applications can be used as well. For the benchmarking over A-Fogs, five datasets with different data sizes are used, with labels DS-A-1, DS-A-2, DS-A-3, DS-A-4, and DS-A-5. The letter A in the labels designates to the A-Fogs. Table I summarizes the datasets used for the algorithms LR and SVM. In this set of experiments the largest file size is 1.77 G, which is believed

TABLE II
COMPUTER SETTINGS USED IN THE A-FOG EXPERIMENTS

Label	A-1	A-2	A-3	A-4	A-5	A-6	A-7	A-8	A-9	A-10	A-11	A-12	A-13	A-14
Desktop	0	0	0	0	0	0	0	1	1	1	1	1	1	1
Raspberry	1	2	3	4	5	6	7	1	2	3	4	5	6	7

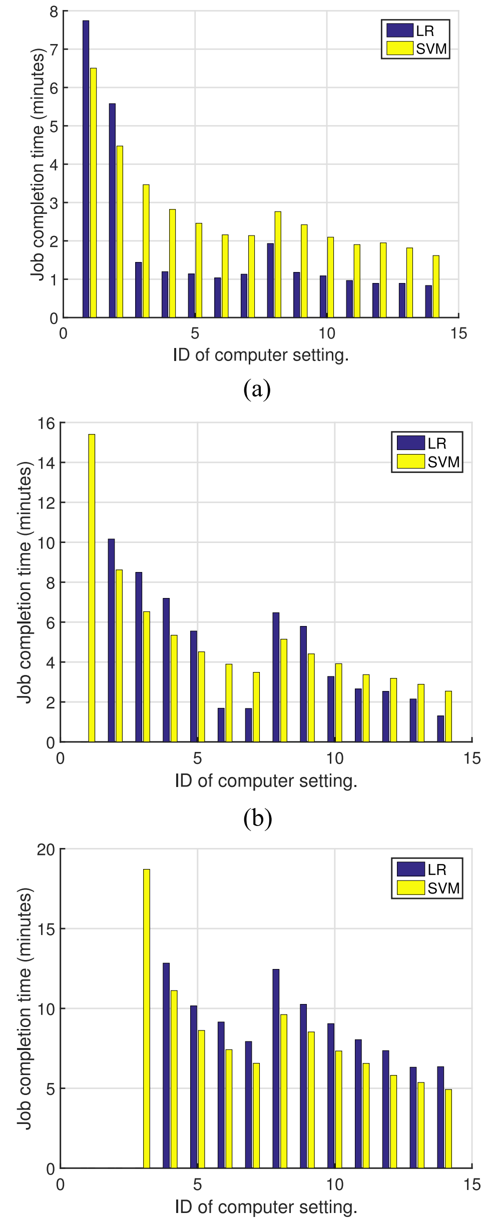
to be large enough for A-Fogs mainly consisted of devices with limited computing resource and energy. Larger datasets should be offloaded to and processed by dedicated fogs or remote clouds.

C. Benchmark Results With A-Fogs

Benchmarking results with Spark are presented over 14 A-Fog computer settings as shown in Table II. It is noted that the letter “A” in the computer setting labels designates to A-Fogs. As A-Fog nodes have only very limited computing resources, they should work together cooperatively and efficiently to process large datasets. It is unclear if Spark can run over distribute compute environment with small size and resource constrained micro computers, and how scalable it is to support large scale data analytics services over A-Fogs. Here, scalability is referred to the ability of a computing platform or fogs to improve its computing performance with increasing computing resources [15].

Ideally a scalable platform is supposed to improve its performance linearly with addition computing resources. As single fog node with computers like Raspberry Pi has limited computing power, a scalable computing platform is important to extend the analytics capabilities over fogs. We can study the feasibility of large scale analytics and system scalability from two different aspects: 1) the size of datasets that can be processed and 2) the service completion (or running) time with increasing number of fog nodes.

Fig. 4 presents the job completion time for LR and SVM jobs with DS-A-2, DS-A-3, and DS-A-4 datasets over the 14 A-Fog computer settings. It can be observed that the job completion time of both LR and SVM applications reduces quickly with increasing computing resources. The large datasets requires much more service time, and the computer setting A-1 fails to complete the jobs with dataset DS-A-4. The results show clearly the feasibility of running large scale analytics service over A-Fogs but also the necessity of distributed computing to process large analytics jobs. In addition the desktop computer shows a large impact on the analytics service capacity, which substantially reduces job completion time and extends the size of datasets that can be processed.



(a)

(b)

(c)

Fig. 4. Job completion time of LR and SVM algorithms versus A-Fog computer settings over different datasets. Letter A in the dataset labels designates to A-Fogs. (a) Dataset DS-A-2. (b) Dataset DS-A-3. (c) Dataset DS-A-4.

D. Benchmark Performance Over Dedicated Fogs

For the benchmarking experiments over D-Fogs, it is assumed that more powerful computing resources (including processing, memory, and storage) are available. We use four DELL R620 servers to set up a computer cluster to represent a D-Fog. In practical D-Fogs more powerful servers may be used to provide a much larger pool of computing resources. Each server consists of an Intel Xeon E5-2680v2 2.8 GHz CPU (dual core), a total memory of 256 GB, and a standard 240G SSD drive. An extra NFS server with 16 TB to store and access the input and output files consistently for all configurations.

In order to test the computing performance of the D-Fog, a pool of 16 VMs is created. The operation system installed on each VM is CentOS release 6.5 with the kernel version 2.6.32. To evaluate and compare the analytics computing performance under different experiment configurations, 12 computer settings representing the number of CPU cores and the memory

TABLE III
COMPUTER SETTINGS USED IN THE D-FOG EXPERIMENTS

Label	D-1	D-2	D-3	D-4	D-5	D-6	D-7	D-8	D-9	D-10	D-11	D-12
CPU cores	4	4	4	8	8	8	12	12	12	16	16	16
Memoery (GB)	4	8	16	4	8	16	4	8	16	4	8	16

TABLE IV
SYNTHETIC DATASETS FOR DEDICATED FOG EXPERIMENTS

Application	Datasets	DS-D-1	DS-D-2	DS-D-3	DS-D-4	DS-D-5	DS-D-6	DS-D-7	DS-D-8
	# of vertices (M)	20	50	100	175	250	350	450	600
LR	Size (GB)	1.14	2.81	5.7	9.97	14.24	19.94	25.63	34.18
SVM	Size (GB)	1.18	2.98	5.9	10.32	14.74	20.64	26.53	35.38

size are used, which are shown in Table III. It is noted that the letter "D" in the computer setting labels designates to D-Fogs.

Similar to the experiments over A-Fogs, Spark datasets generator and GTgraph are used to generate synthetic datasets for benchmarking experiments over D-Fogs. Table IV shows the labels of the datasets, number of vertices and dataset size used for LR algorithm and SVM algorithm. The dataset size can be adjusted by changing the number of vertices for machine learning algorithms. In total eight datasets with different data sizes are used for the experiments with D-Fogs, ranging from 1.14 GB to 35.38 GB.

Similar performance trends with D-Fogs benchmarking results are obtained as these for A-Fogs. First the job completion time decreases with computing resources (CPU and memory) scaling up. The size of dataset that can be processed by the D-Fogs also increases largely due to more computing power and memory. However, for the largest datasets DS-D-7 and DS-D-8, only the highest computer settings can complete the analytics jobs successfully.

IV. QoS AWARE SERVICE AND RESOURCE MANAGEMENT

In this section we presents the design of QoS aware job and resource management schemes for analytics services over multitier fogs, which is a core intelligent management module of fog masters introduced in Section II.

The overall service and job management framework is presented in Fig. 5. There are two major blocks, one for offline benchmarking and the other for online analytics service and jobs management.

1) *Offline Benchmarking:* In Section III, we have run extensive benchmarking experiments, which sent sample analytics jobs with different type of applications and dataset sizes to A-Fogs and D-Fogs with different computer settings. Workload models in terms of job completion time, computing, and communication resource consumption are created in relation to the job properties (e.g., analytics algorithm and dataset size) and computing resource settings. A table is created for each analytics algorithm from the workload models. In the tables each entry stores a job completion time for one dataset size and one computing resource setting.
2) *Online Analytics Service and Resource Management:* Real analytics jobs and computing resources are managed in this block. Each A-Fog has a cluster of fog workers, forming a pool of computing resources. For each arriving job request with associated QoS targets (e.g., target job completion time) admission control scheme is applied to decide if the job request can be accepted or rejected, based on the cost models for computing and communication resources, job completion time tables, available computing resource and network conditions. If a job request is accepted, computing resources at the A-Fog, D-Fog or cloud are allocated. The computing and QoS performance is recorded, and also feedback to the benchmarking block to update and enhance the workload models as well as the tables.

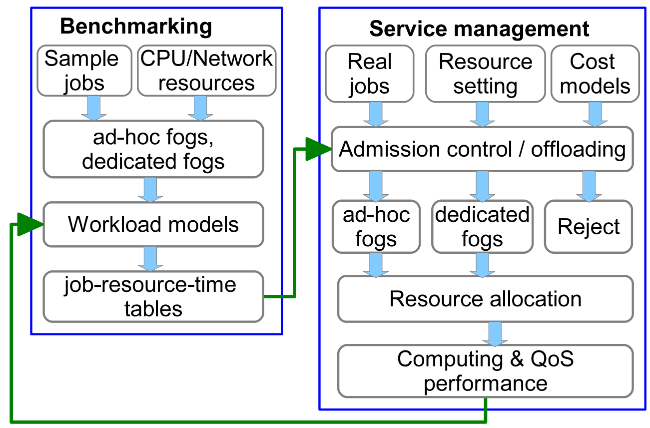


Fig. 5. Service management framework.

A. System Model

We consider the problem of managing analytics jobs over two tiers of fogs and a cloud. An example scenario is shown in Fig. 6. The A-Fog has a fog master and a cluster of computers as workers, which form a computing resource pool of N_d desktop computers and N_r Raspberry Pi computers, The D-Fog has a fog master and a computing resource pool of N_c CPU cores and N_m GB memory. The specification of computing resources for allocation to the analytics jobs is assumed to be the same as those used in the benchmarking experiments presented in Section III for A-Fog and D-Fogs, respectively. The remote clouds are assumed to have unlimited computing resources.

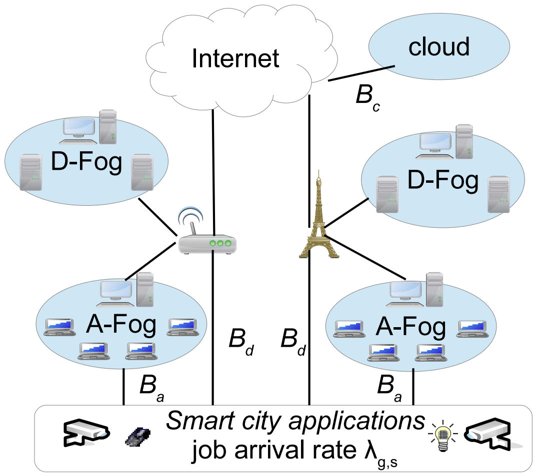


Fig. 6. Fog system scenario.

1) Job and Service Models: Analytics jobs are generated by a group of smart city IoT sensors and mobile devices. An analytics job is jointly characterized by the analytics algorithms and the dataset size. Let (g, s) denote a job type with g representing analytics algorithm, $g \in [1, N_G]$, and s representing the ID of dataset interval, $s \in [1, N_S]$. N_G and N_S denote the number of analytics algorithms and intervals, respectively. There are $N_G = 8$ analytics algorithms, which are mentioned in Section III. There are 13 dataset intervals ($N_S = 13$), corresponding to the five datasets used for A-Fog benchmark in Table I and the eight datasets used for D-Fog benchmark experiments in Table IV. The first dataset interval ($s = 1$) corresponds to the range of dataset sizes from 0 to the size of dataset DS-A-1.

Let $J_{g,s,j}$ denote the jth job of type (g, s). We assume the jobs generated from the IoT sensors and mobile devices follow Poisson distribution for all job types. The job generation rate per second for job type (g, s) is denoted by $\lambda_{g,s}$. The actual dataset size for a job $J_{g,s,j}$ is denoted by $D_{g,s,j}$ in bytes, which is assumed to be uniformly distributed in the dataset interval s.

Consider the target job completion time as the QoS metric of interest for analytics jobs, Each job is associated with a target job completion time and a service completion charge. The job completion time is counted from the time the job request is received to the completion of the job at a fog or the cloud, including the network latency, data communication time, possible queue delay, and job computation time. Data communication time is determined by network bandwidth and dataset size, while job computation time is determined by the analytics algorithm, dataset size, and allocated computing resource for the job. Let $T^e_{g,s,j}$ denote the target completion time in seconds, and $R_{g,s,j}$ denote the service completion charge in dollars, for job $J_{g,s,j}$, respectively. The service completion charge of a job is set proportional to the dataset size of the job. For the analytics service model, consider a simple business model: if a job request is accepted and is completed within the target completion time, the user of the analytics service for the job is charged by the operator (A-Fog, D-Fog, or the cloud) running the job, otherwise the service is not charged.

2) Networking Price Models: The round trip delay from the analytics jobs to the A-Fog, D-Fog, and the cloud is denoted by T_a, T_d, and T_c, respectively. The bandwidth of bottleneck link in the paths from the sensors to the A-Fog, D-Fog, and the cloud is denoted by B_a, B_d, and B_c, respectively. In general we have $T_a \leq T_d \leq T_c$.

Let $C^{u,z}_{g,s,j}$ denote the communication cost for the job $J_{g,s,j}$, which depends on the dataset size and the networking price. The letter z designates to the type of computing environments, where the job is executed, with $z = a$ for the A-Fog, $z = d$ for the D-Fog, and $z = c$ for the cloud. Let P_u, z denote the communication price in dollars per bytes, with z designating the type of computing environments. Then communication cost can be computed by $C^{u,z}_{g,s,j} = D_{g,s,j}P_{u,z}$. As the A-Fog is close to the IoT and mobile devices, the communication cost to the A-Fog is much lower than that to the D-Fog and the cloud.

3) Computing Price Models: Let $C^q_{g,s,j}(k)$ denote the computing resource cost for the job $J_{g,s,j}$ under the kth computer setting option. A computer setting option k implicitly determine the computing environment (the A-Fog, the D-Fog or the cloud). The computing cost depends on the computing resource price and the time used by the given analytics job. Let $P_{q,d}$ and $P_{q,r}$ denote the computing price per seconds for using a desktop and a Raspberry computer in the A-Fog, respectively. Let $P_{q,c}$ and $P_{q,m}$ denote the computing price per seconds for using a VM CPU core and one GB memory in the D-Fog or cloud, respectively. The computing resource price for the D-Fog and the cloud are set to be the same. Let $T_{g,s,j}(k)$ denote the computation time (in seconds) if the job $J_{g,s,j}$ is processed with computer setting option k. Then the computing cost for a given job can be computed according to the computing resource allocated for this job, the computation time with the allocated resources, and the computing resource price.

Let us define the overall cost for job $J_{g,s,j}$ processed with computer setting option k, which is denoted by $C_{g,s,j}(k)$, as the sum of the communication and computing costs for the job. We have $C_{g,s,j}(k) = C^{u,z}_{g,s,j} + C^q_{g,s,j}(k)$.

B. Job and Resource Management Schemes

For a new analytics job, the request is sent to the A-Fog master node. If an A-Fog is not available, the job request can be sent to the D-Fog or the cloud.

The fog masters need to make the following three decisions.

1) *Admission Control Decision:* Decide if to accept a job according to the existing computing resources, the target job completion time and the benefit of accepting a job.
2) *Offloading Decision:* Decide, where to run the job, the A-Fog, the D-Fog or the cloud.
3) *Resource Allocation Decision:* Decide how many and which computers for the A-Fog, or CPU cores and memory for the D-Fog and the cloud to run the job.

The three decisions are made by the A-Fog master. The A-Fog and D-Fog masters regularly exchange resource

TABLE V
SYSTEM CONFIGURATION ON DATA RATE AND RESOURCES

Variable	Values	Meaning
N_{d}	10	Number of desktops in A-Fog pool
N_{r}	40	Number of A-Fog Raspberry Pi
N_{c}	500	Number of D-Fog cores
N_{m} (GB)	1000	Size of D-Fog memory
T_{a} (ms)	30	Job to A-Fog end to end delay
T_{d} (ms)	50	Job to D-Fog end to end delay
T_{c} (ms)	500	Job to cloud end to end delay
B_{a} (Mbps)	10	Job to A-Fog bandwidth
B_{d} (Mbps)	2	Job to D-Fog bandwidth
B_{c} (Mbps)	1.5	Job to cloud bandwidth
$P_{\mathrm{u,a}}$ ($/GB)	0.1	Job to A-Fog communication price
$P_{\mathrm{u,d}}$ ($/GB)	1	Job to D-Fog communication price
$P_{\mathrm{u,c}}$ ($/GB)	2	Job to cloud communication price
$P_{\mathrm{q,d}}$ ($/hour)	2	A-Fog usage price per desktop
$P_{\mathrm{q,r}}$ ($/hour)	1	A-Fog usage price per Raspberry Pi
$P_{\mathrm{q,c}}$ ($/hour)	0.1 / 0.04	D-Fog / cloud usage price per CPU core
$P_{\mathrm{q,m}}$ ($/hour)	0.05 / 0.03	D-Fog / cloud usage price per GB memory

availability and resource prices models. The objective of decision making is to maximize the service utility of processing an analytics job. Let N_o denote the total number of computing resource configuration options. Let $U_{g,s,j}(k)$ denote the utility (or revenue) of completing a job $J_{g,s,j}$ with the kth computer setting, for $k \in [1, N_o]$, which is defined as

$$U_{g,s,j}(k) = \begin{cases} R_{g,s,j} - C_{g,s,j}(k), & T_k \le T^e_{g,s,j} \\ -C_{g,s,j}(k), & T_k > T^e_{g,s,j} \end{cases} \quad (1)$$

where T_k denotes the actual job completion time with computer setting k, including the data communication time and job computation time.

When a new request for job $J_{g,s,j}$ is received by the A-Fog master, it simply computes the utility $U_{g,s,j}(k)$ with (1) over all the possible computing resource configuration options with the available computing resources, and then find the best option k^* which maximizes $U_{g,s,j}(k)$ over these possible computing resource configuration options. If the utility $U_{g,s,j}(k^*)$ with the option k^* is positive, then the job request is accepted; otherwise, the job is rejected.

It is noted, if following the above procedure, a job request is accepted with option k^*, the offloading and resource allocation decisions are made implicitly as well, as the corresponding computing environment (A-Fog, D-Fog or cloud) and the resource allocation strategy are uniquely determined by the global computer setting option k^*.

C. Evaluation of Job Management Scheme

The proposed job management scheme and the analytics services over multitier fogs are evaluated. A discrete-event driven system level simulator is developed, which can be used to perform fast simulations with a large scale of computing resources in the A-Fogs and D-Fogs. Based on the simulator extensive experiments are run to obtain simulation results.

Table V shows the main system configurations. N_o is set to 38 in this paper. The first 14 computer settings are used at the A-Fog, corresponding to computer settings for A-Fog experiments shown in Table II and the following 12 computer setting are used at the D-Fog, corresponding to the settings in the D-Fog for the D-Fog experiments shown in Table III and the last 12 settings are used at the cloud, corresponding to the same computer settings used in the D-Fog.

The A-Fog computing resource prices are set higher than those of D-Fog and cloud. The computing resource prices for the D-Fog and cloud are set in line with the current Amazon EC2 service prices, where virtually unlimited computing resources are available. The main consideration is that the A-Fog nodes are usually energy constrained and computing resource limited. Higher computing resource price is used to provide incentives for sharing unused computing resources. On the other hand, the price of communication to the D-Fog and clouds are much higher than that to A-Fog. According to our market research, the currently cheapest mobile broadband data tariff in the U.K. is 2 pounds per GB mobile data traffic. The communication price for the job to D-Fog is set according to the market price. Part of the analytics job data may be delivered to the D-Fog through low cost wireline access networks.

Analytics job arrival rates $\lambda_{g,s}$ are set to decrease with dataset s, but remain the same for any given algorithm g. For simplicity the traffic arrival rates for dataset types except the dataset DS-A-2 are fixed and set to be relatively small, representing background traffic in the simulations. The main analytics jobs are generated from different algorithms with dataset DS-A-2. The traffic arrival rates $\lambda_{g,1}$ is varied to generate light and heavy traffic loads. Each job has a random target completion time, which is the communication time to the cloud times a uniformly distributed random number in the range of [1] and [3]. The job service charge in dollars is assumed to be the communication cost to the cloud times a random value in the range [1], [2].

With the above system parameters configuration, we can compute the communication costs for jobs to the A-Fog, D-Fog, and cloud, and the computing cost for the sample jobs running with all the 38 computer settings. The total costs of communication and computing versus the computer settings are sorted in increasing order and stored in a table for each job type. In the simulations we can simply pick the best option for a given job as the first available computer setting from the corresponding cost table, which can significantly speed up the simulation process.

Four different cloud computing architectures are compared: 1) with A-Fog only (label "A-Fog" in the following figures); 2) with all the three computing systems (labeled "All"); 3) with only D-Fog and cloud (labeled "D-Fog+cloud"); and 4) with only cloud.

For the data analytics service and the QoS aware job management schemes, the major concern from the data analytics service users is on the service quality, such as job blocking probability; while for the service operators of the fogs and the clouds they are more concerned on the utility (or revenue) generated from the analytics services. Job blocking probability and service utility are chosen as the main performance metrics. Typical results of blocking probability and service utility are presented in Fig. 7(a) and (b), respectively. It is noted that each result shown in the figures is obtained by averaging over ten simulations. Each simulation stops until 10 000 jobs are successfully completed.

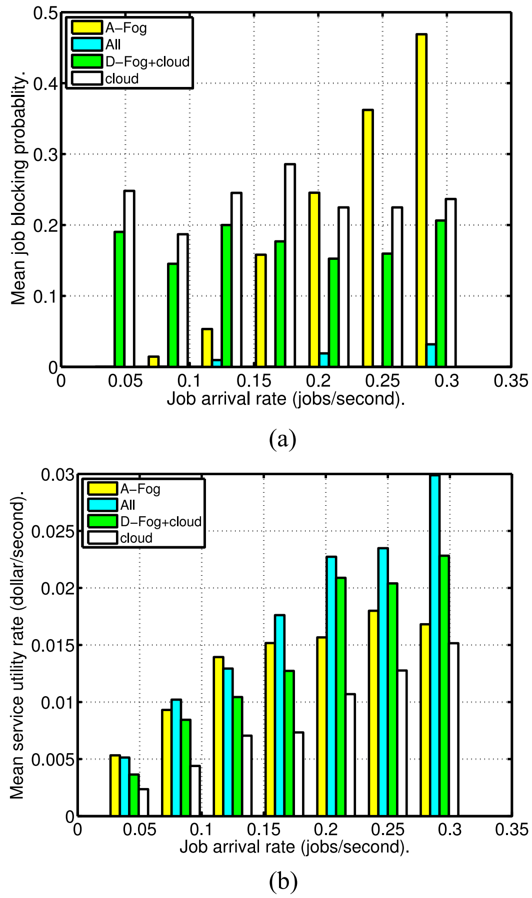


Fig. 7. Analytics services performance over multitier fog computing platform against job arrival rate. (a) Job blocking probability. (b) Service utility.

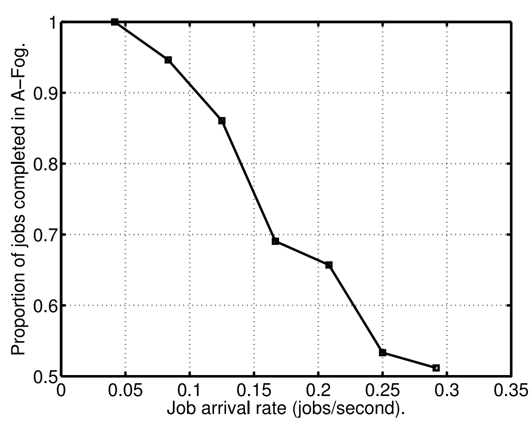


Fig. 8. Proportion of jobs completed in the A-Fog under the All computing architecture.

Further details on the proportion of the jobs completed at the A-Fog under the computing architecture All is presented in Fig. 8.

From the results presented in Figs. 7 and 8, we have the following observations.

1) Fog computing can provide good analytics service quality. Although the cloud has virtually unlimited computing services, it does not provide satisfactory service quality. The job blocking probability with cloud only architecture is larger than 0.2 irrespective of job arrival rates.
2) On the other hand, the A-Fog only computing architecture can provide quite good service quality until job arrival rate is larger than 0.2 jobs per second. Due to the limited computing resources at the A-Fog, the blocking probability increases fast with job arrival rates larger than 0.2, which indicates strong need of dedicated computing resources at D-Fog and/or clouds.
3) When the multitier computing architecture is used (with A-Fog, D-Fog, and cloud), the overall analytics service has the lowest blocking probability, which is less than 0.03 for most job arrival rates.
4) The multitier fog computing system achieves the highest service utility for most job arrival rates, which is almost double of the cloud only system. The A-Fog only computing system can also deliver a high service utility rate with job arrival rate of less than 0.2.
5) Compared to the A-Fog only architecture, the D-Fog+cloud architecture has lower utility rate with low job arrival rates, but higher utility rate at high job rates. The job blocking rate is relatively high, which is more than 0.15 for most job arrival rates. According to Fig. 8, the jobs offloaded to the D-Fog or the cloud increase linearly with job arrival rate due to the limited computing resource at the A-Fog.

According to the results, we can conclude that large scale analytics service over fogs computing is feasible. The A-Fogs formed with opportunistic distributed computing resource can provide good analytics service quality alone when the analytics job arrival rate is low. With increasing analytics service popularity additional D-Fogs with dedicated and larger pool of computing resources can be deployed to provide better analytics services.

V. Conclusion

In this paper we proposed a multitier fog computing model based analytics service for smart city applications. In the multitier fog computing model there are both ad-hoc fogs with opportunistic computing resources and dedicated fogs with dedicated computing resources. Detailed functional modules of fog nodes were designed. QoS aware job admission control, offloading and resource allocation schemes were designed and developed to provide QoS support for large scale data analytics services over multitier fogs. To support the QoS aware resource management schemes, extensive benchmark experiments over both ad-hoc and dedicated fogs were run to measure computing performance of analytics jobs over fogs and create workload models for QoS management schemes. Analytics services over fogs and the proposed QoS aware resource management schemes were evaluated with various computing architectures with and without fogs. Simulation results demonstrated the efficiency of analytics services model over multitier fogs and the effectiveness of the proposed QoS schemes.

REFERENCES

[1] J. A. Stankovic, "Research directions for the Internet of Things," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 3–9, Feb. 2014.

[2] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, Feb. 2014.

[3] "Fog computing and the Internet of Things: Extend the cloud to where the things are," White Paper, Cisco, San Jose, CA, USA, 2015.

[4] A. Fox *et al.*, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, Apr. 2010.

[5] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: Architecture, applications, and approaches," *Wireless Commun. Mobile Comput.*, vol. 13, no. 8, pp. 1587–1611, Dec. 2013.

[6] C. M. S. Magurawalage, K. Yang, L. Hu, and J. Zhang, "Energy-efficient and network-aware offloading algorithm for mobile cloud computing," *Comput. Netw.*, vol. 74, pp. 22–33, Dec. 2014.

[7] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Comput.*, vol. 8, no. 4, pp. 14–23, Oct./Dec. 2009.

[8] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proc. ACM MCC*, Helsinki, Finland, 2012, pp. 13–16.

[9] X. Wang, S. Leng, and K. Yang, "Social-aware edge caching in fog radio access networks," *IEEE Access*, vol. 5, pp. 8492–8501, Apr. 2017.

[10] J. Wei, J. He, K. Chen, and Y. Zhou, "Benchmarking of distributed computing engines spark and graphlab for big data analytics," in *Proc. IEEE BigDataService*, Oxford, U.K., 2016, pp. 10–13.

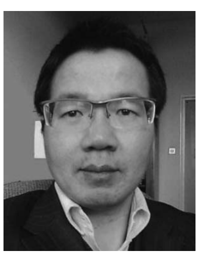
[11] *Spark*. Accessed on May 1, 2016. [Online]. Available: http://spark.apache.org

[12] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets," in *Proc. 2nd USENIX Conf. Hot Topics Cloud Comput.*, Boston, MA, USA, Jun. 2010, p. 10.

[13] M. Zaharia *et al.*, "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing," in *Proc. NSDI*, San Jose, CA, USA, 2012, p. 2.

[14] *Raspberry Pi Foundation*. Accessed on Jul. 1, 2016. [Online]. Available: http://www.raspberrypi.org

[15] Y. Guo *et al.*, "How well do graph-processing platforms perform? An empirical performance evaluation and analysis," in *Proc. IEEE IPDPS*, Phoenix, AZ, USA, 2014, pp. 395–440.



Jianhua He (A'07–M'07–SM'12) received the B.Sc. and M.Sc. degrees from the Huazhong University of Science and Technology, Wuhan, China, in 1995 and 1998, respectively, and the Ph.D. degree from Nanyang Technological University, Singapore, in 2002.

He is a Lecturer with Aston University, Birmingham, U.K. He has authored or co-authored over 100 technical papers in major international journals and conferences. His current research interests include 5G and beyond technologies, machine to machine communications, Internet of Things system and technologies, and big data analytics.

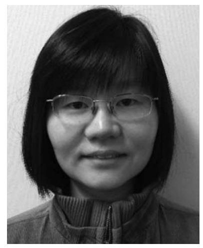
Dr. He has been an/a Editor/Guest Editor of several journals and the TPC Chair of several international conferences. He is a TPC member of many international conferences including IEEE Globecom and IEEE ICC.

Jian Wei received the B.E. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2013. He is currently pursuing the Ph.D. degree in engineering and applied science at Aston University, Birmingham, U.K.

His current research interests include machine learning and data mining, modeling and scheduling in distributed computing systems, and fog computing for IoT big data analytics.

Kai Chen received the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, China.

He is with the Institute of Image Communication and Network Engineering, Shanghai Jiao Tong University. He is the key member of the Institute on Network Engineering Research. He was the principle investigator of several key nation projects and many industry-academia-research projects. His current research interests include information retrieving, object recognition, and big data mining.



Zuoyin Tang received the Ph.D. degree from the University of Bath, Bath, U.K., in 2008.

She is currently a Lecturer with the School of Engineering and Applied Science, Aston University, Birmingham, U.K. She has authored or co-authored over 40 technical papers in major international journals and conferences. Her current research interests include resource management for cellular networks, Internet of Things, wireless sensor networks, and fog computing.

Yi Zhou received the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, China, in 2010.

She is with the Computer Science Department, Shanghai Jiao Tong University. Her research on big data analytics and Chinese characters recognition has been supported by Huawei and the National Natural Science Foundation of China. Her current research interests includes big data mining and object recognition.

Yan Zhang (M'05–SM'10) received the Ph.D. degree from the School of Electrical & Electronics Engineering, Nanyang Technological University, Singapore.

He is a Full Professor with the Department of Informatics, University of Oslo, Oslo, Norway. His current research interests include next-generation wireless networks leading to 5G, and green and secure cyber-physical systems (e.g., smart grid, healthcare, and transport).

Dr. Zhang is an Associate Technical Editor of the *IEEE Communications Magazine*, an Editor of the IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, *IEEE Communications Surveys & Tutorials*, and IEEE INTERNET OF THINGS JOURNAL, and an Associate Editor of IEEE ACCESS. He serves as the Chair for a number of conferences, including IEEE GLOBECOM 2017, IEEE VTC-Spring 2017, IEEE PIMRC 2016, IEEE CloudCom 2016, IEEE ICCC 2016, IEEE CCNC 2016, IEEE SmartGridComm 2015, and IEEE CloudCom 2015. He serves as a TPC member for numerous international conference, including IEEE INFOCOM, IEEE ICC, IEEE GLOBECOM, and IEEE WCNC. He is an IEEE Vehicular Technology Society Distinguished Lecturer. He is a Senior Member of the IEEE ComSoc, IEEE CS, IEEE PES, and IEEE VT Society. He is a Fellow of the IET.