Incorporating Intelligence in Fog Computing for Big Data Analysis in Smart Cities

Bo Tang, *Student Member, IEEE*, Zhen Chen, Gerald Hefferman, Shuyi Pei, Tao Wei, Haibo He, *Senior Member, IEEE*, and Qing Yang, *Fellow, IEEE*

***Abstract*—Data intensive analysis is the major challenge in smart cities because of the ubiquitous deployment of various kinds of sensors. The natural characteristic of geodistribution requires a new computing paradigm to offer location-awareness and latency-sensitive monitoring and intelligent control. Fog Computing that extends the computing to the edge of network, fits this need. In this paper, we introduce a hierarchical distributed Fog Computing architecture to support the integration of massive number of infrastructure components and services in future smart cities. To secure future communities, it is necessary to integrate intelligence in our Fog Computing architecture, e.g., to perform data representation and feature extraction, to identify anomalous and hazardous events, and to offer optimal responses and controls. We analyze case studies using a smart pipeline monitoring system based on fiber optic sensors and sequential learning algorithms to detect events threatening pipeline safety. A working prototype was constructed to experimentally evaluate event detection performance of the recognition of 12 distinct events. These experimental results demonstrate the feasibility of the system's city-wide implementation in the future.**

***Index Terms*—Distributed computing, fiber optic sensor, fog computing, Internet of Things (IoT), smart cities, smart pipeline.**

I. Introduction

In THE past decade, the concept of *Smart City* has drawn great interest in both science and engineering fields as a means to overcome the challenges associated with rapidly growing urbanization. A smart city is an urbanized area, where multiple sectors cooperate to achieve sustainable outcomes through the analysis of contextual, real-time information. Smart cities reduce traffic congestion and energy waste, while allocating stressed resources more efficiently and improving quality of life. For instance, in 2013, Seattle partnered with Microsoft to launch its High-Performance Building program, allowing for real-time tracking of energy efficiency, reducing energy costs and carbon emissions [1]. Smart city technologies are projected to become massive economic engines in the coming decades, and are expected to be worth a cumulative 1.565 trillion dollars by 2020, and 3.3 trillion dollars by 2025. Today, companies are actively vying for a central role in the smart city ecosystem, creating an expanding number of technologies and employment opportunities. Already, IBM, Intel, GE, and many other companies have initiated projects to integrate their products and services into a smart city framework [2]. Hundreds of millions of jobs will be created to facilitate this smart city conversion; in June 2014, Intel and the city of San Jose, CA, USA, began collaborating on a project implementing Intel's Smart City Demonstration Platform, installing a network of air quality and climate sensors which alone fostered 25,000 new high-tech jobs in San Jose [3].

It is expected that more than 9 billion people will live on the planet in 2050, and about two-thirds of them in cities [4], [5]. This substantial population growth in urban areas will result in increasing demands on resources, services, and infrastructures in cities. Hence, to ensure the efficiency, sustainability and safety of urban communities, the cities should be becoming smarter with the intelligent technologies to integrate massive infrastructure components and services in the areas of energy, building, transportation, healthcare, education, real estate, and utilities [6]–[8].

While rapid urbanization provides numerous opportunities, building smart cities presents many challenges. First, it is essential to build accurate, real-time, and large-scale geospatially distributed sensing networks in future smart cities to monitor the structural health of critical infrastructure components, such as bridges, gas/oil/water pipelines, roads, and subways. Second, the widely distributed sensor networks generate a massive volume of data, which leads to a "Big Data" analysis challenge [9], [10]. Third, the machine-to-machine (M2M) communication among massive numbers of sensors will dominate future communication network traffic, namely Internet of Things (IoT) [11], [12], instead of traditional Internet of Contents in human-to-human and human-to-machine communication. Last but not least, the integration of infrastructure components and services in smart cities requires an efficient monitoring and a quick feedback control (intelligent decision making) system to ensure the safety of urban communities. For example, an automatic valve closure is necessary when a rupture along a segment of pipeline is detected, and an optimal distribution of municipal emergency services must be made when a natural disaster is taking place. In summary, the ubiquitous deployment of sensors in

Manuscript received March 3, 2016; revised August 8, 2016; accepted February 13, 2017. Date of publication March 8, 2017; date of current version October 3, 2017. This work was supported by the National Science Foundation under Grant CCF-1439011 and Grant CCF-1421823. Paper no. TII-16-0254. *(Corresponding author: Haibo He.)*

B. Tang is with the Department of Computer Science, Hofstra University, Hempstead, NY 11549 USA (e-mail: bo.tang@hofstra.edu).

Z. Chen, S. Pei, T. Wei, H. He, and Q. Yang are with Department of Electrical, Computer and Biomedical Engineering, University of Rhode Island, Kingston, RI 02881 USA (e-mail: chen@ele.uri.edu; spei@ele.uri.edu; wei@ele.uri.edu; he@ele.uri.edu; qyang@ele.uri.edu).

G. Hefferman is with the Warren Alpert Medical School, Brown University, Providence, RI 02903 USA (e-mail: ghefferm@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2017.2679740

smart cities requires a high-performance computing paradigm to support big data analysis with smart technologies and communications in IoT, providing location-awareness and latency-sensitive computing near the data sources (i.e., at the edge of the network).

Currently, the "pay-as-you-go" Cloud Computing paradigm is widely used in enterprises to address the emerging challenges of big data analysis because of its scalable and distributed data management scheme. However, data centers in the Cloud faces great challenges on the burden of exploding amount of big data and the additional requirements of location awareness and low latency at the edge of network necessary for smart cites. *Fog Computing* recently, proposed by Cisco, extends the Cloud Computing paradigm to run geodistributed applications throughout the network [13]. In contrast to the Cloud, the Fog not only performs latency-sensitive applications at the edge of network, but also performs latency-tolerant tasks efficiently at powerful computing nodes at the intermediate of network. At the top of the Fog, Cloud Computing with data centers can be still used for deep analytics.

In this paper, we introduce a hierarchical distributed Fog Computing architecture for big data analysis in smart cities [14]. Due to the natural characteristic of *geodistribution* in big data generated by massive sensors, we distribute intelligence at the edge of a layered Fog Computing network. The computing nodes at each layer perform latency-sensitive applications and provide quick control loop to ensure the safety of critical infrastructure components. Using smart pipeline monitoring as a use case, we implemented a prototypical four-layer Fog-based computing paradigm to demonstrate the effectiveness and the feasibility of the systems city-wide implementation in the future.

The rest of this paper is organized as follows. In Section II, we introduce the related work on future smart city computing architectures, fiber sensing networks, and smart computing technologies; in Section III, we give an overview of hierarchical distributed Fog Computing platform for various applications in smart cities, including computing power and communication performance; in Section IV, we describe the implementation of our prototype for smart pipeline monitoring; in Section V, experimental results are provided, showing the feasibility of our proposed architecture in smart cities; finally, conclusions and future work are discussed in Section VI.

II. Related Work

A. Computing and Communication Architecture for Smart Cities

The new challenges of big data analysis posed by smart cities demand that researchers investigate and develop novel and high-performance computing architectures. The rising of Cloud Computing and Cloud Storage in industry provides a solution to support dynamic scalability in many smart city applications, such as large scale data management for smart house [15], smart lighting [16], and video surveillance [17], and intensive business and academic computing tasks in education institutions [18]. However, the deployment of massive numbers of sensors in future smart cities requires location awareness and low latency, which are lacking in current commercial Cloud Computing models. To address this issue, one possible solution is to extend or modify current Cloud Computing approach. In [19], a volunteer "Sensing and Actuation as a Service" (SAaaS) Cloud Computing approach is proposed to ensure that users are able to obtain data from different heterogeneous sensors.

Rather than analyzing all original data from massive sensor networks, an event-driven platform can greatly reduce the computational burden of smart cities. In [20], Filipponi *et al.* introduced an event driven architecture to monitor public areas and infrastructures with heterogeneous sensors. Their smart city architecture consists of two major components: a knowledge processor and a semantic information broker. While the knowledge processor produces notifications of events detected by the sensors, the semantic information broker conveys the notifications to all consumer processors in a publish-and-subscribe manner. The advantage of this architecture is its efficient information broadcasting in a city-wide environment; however, it lacks a quick response mechanism for emergency events that may threaten public safety.

A number of new network architectures have also been developed to support the M2M communication. For example, in [21], Jin *et al.* presented four different network architectures for smart cities to support IoT and Internet of Services, including autonomous network architecture, ubiquitous network architecture, application-layer overlay network architecture, and service-oriented network architecture. Among these existing networks, most of them are the IP-based connectivity models [22]. Unlike the hierarchical networks, IP-based architectures have the disadvantages of a lack of modularity and separation of concerns [23].

More recently, the concept of Fog Computing [13] is introduced by extending the Cloud Computing paradigm to the edge of M2M network to support the IoT. Many characteristics have been found in Fog Computing, including low-latency, location awareness, mobility, and wide-spread geographical distribution, etc., which makes it suitable for many smart city applications. The work described in this paper develops this Fog Computing concept further, and the new paradigm will be described in detail in Section II-B. The key idea of Fog Computing architecture is to distribute computing tasks throughout the network such that the communication overhead of the network is reduced and the high-performance computing capability is achieved. In-network processing architecture is the other type of architecture to reduce the communication overhead of the network, in which intermediate proxy nodes are chosen to execute data transformation function to eliminate data redundancy for feature extraction. However, compared to the Fog Computing architecture, only lightweight sensor data processing is performed without the incorporation of intelligent algorithms, which usually require intermediate computing power, and thus quick response may not be provided in the in-network processing architecture. The hierarchical architecture proposed in this paper can support multilevel data association and processing with different scales of coverage, and its nice scalability also enables it suitable for large-scale and computing-intensive systems.

B. Sensor Networks in Smart Cities

Sensors in future smart cities will be heterogeneous; thus, it is necessary to have a senor network providing efficient information exchange among interconnected sensors. Two types of sensor networks could be adopted: one is active wireless sensor network (WSN) and the other is passive fiber optic sensor network (FOSN). WSNs have been widely used for IoT applications [24]. The feasibility of using WSNs on large civil engineering infrastructure components, such as smart bridges and smart tunnels, is studied in [24]. However, the lifetime of such sensor system is limited to the battery of sensors, and frequent battery maintenance is necessary. Interference, transmission range, and other limiting factors further detract from the practicality of WSNs for smart cities, especially for monitoring critical infrastructure components, which are not easy to access, such as underground oil/gas pipelines.

The two well-known examples of WSNs are ad hoc networks and radio frequency identification systems [25]–[27]. WSNs provide many distinct advantages, such as high radio coverage and standard communication protocol. However, the lifetime of such sensor system is limited to the battery of sensors, and frequent battery maintenance is necessary. Interference, transmission range, and other limiting factors further detract from the practicality of WSNs for smart cities, especially for monitoring critical infrastructure components, which are not easy to access, such as underground oil/gas pipelines. Unlike the WSNs, the FOSNs embed optical fibers as sensors within infrastructure components, providing a passive and low-maintenance sensing [28], [29]. The optical fiber sensors offer a series of characteristics that are suitable for the citywide sensing, such as compactness, lightweight, high spatial and temporal resolution, immunity to electromagnetic inference, low cost, and multiplexing capability. In this paper, FOSNs within the sensing layer are built, the base layer of the proposed architecture, to measure temperature changes in a pipeline monitoring system. The detailed description of our FOSNs implementation will be presented in the sections that follow.

C. Smart Computing Technologies in Smart Cities

In addition to the large-scale data storage, the “smartness” of infrastructure in future smart cities requires intelligent data analysis for smart monitoring and actuation to achieve automated decision making, thereby ensuring the reliability of infrastructure components and the safety of public health. Such “smartness” in smart cities derives from the employment of many advanced artificial intelligence algorithms or the combination of several of them, including density distribution modeling [30], supervised and nonsupervised machine learning algorithms [31], [32], and sequential data learning [33], to name a few. The employment of these artificial intelligence algorithms could be centralized or distributed. One example of centralized employment is the well-known Cloud Computing. The use of Cloud Storage offers artificial intelligence experts a convenient way to access the data and perform data analysis. However, the centralized methods typically lack real-time computation properties and fail to provide quick feedback responses and control if anomalies are detected. Waiting for control commands from a centralized center may result in missing the opportunity to avoid damage from hazardous events.

The wide use of heterogeneous senors leads to another challenge to extract useful information from a complex sensing environment at different spatial and temporal resolutions [34]. Current state-of-the-art methods usually shallow this problem: they first apply supervised learning algorithms to identify predefined patterns and use unsupervised learning algorithms to detect data anomalies [35]. Then, sequential learning methods with spatial-temporal association are employed to infer local activities or predefined events. Complex city-wide spatial and longer temporal activities or behaviors could be further detected at a higher layer [34]. It is worth noting that the proposed hierarchical architecture in this paper is suitable for such distributed employment of artificial intelligence algorithms across multiple layers.

III. Hierarchic Distributed Fog Computing Platform for Smart Cities

A. Multilayer Fog Computing Architecture

The big data in smart cities exhibits a new characteristic: *geodistribution* [36]. This new dimension of big data requires that the data needs to be processed near the sensors at the edge, instead of the data centers in traditional Cloud Computing paradigm. Meanwhile, the major computing task of big data analysis in smart cities is to identify potential anomalies and hazardous events. It is necessary to offer low latency responses to protect the safety of critical infrastructure components when anomalies and hazardous events are detected. Fog Computing is a suitable paradigm by extending the Cloud Computing to the edge of network. Because the data is processed at the edge, quick control loops are feasible using the Fog Computing model.

We show a four-layer Fog Computing architecture in Fig. 1. At the edge of network, layer 4, is the sensing network which contains numerous sensory nodes. Those sensors are noninvasive, highly reliable, and low cost; thus, they can be widely distributed at various public infrastructures to monitor their condition changes over time. Note that massive sensing data streams are generated from these sensors that are geospatially distributed, which have to be processed as a coherent whole.

The nodes at the edge forward the raw data into the next layer, layer 3, which is comprised of many low-power and high-performance computing nodes or edge devices. Each edge device is connected to and responsible for a local group of sensors that usually cover a neighborhood or a small community, performing data analysis in a timely manner. The output of the edge device has two parts: the first are reports of the results of data processing to an intermediate computing node at its next upper layer, while the second is simple and quick feedback control to a local infrastructure to respond to isolated and small threats to the monitored infrastructure components.

Layer 2 consists of a number of intermediate computing nodes, each of which is connected to a group of edge devices at layer 3 and associates spatial and temporal data to identify potential hazardous events. Meanwhile, it makes quick response to control the infrastructure when hazardous events are detected.

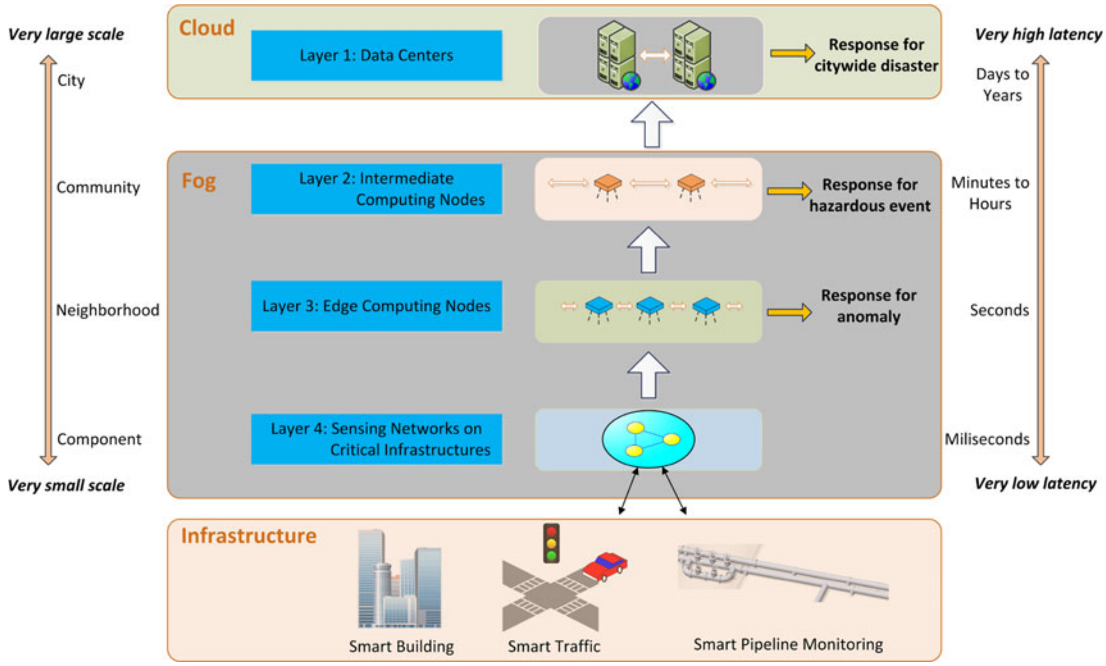


Fig. 1. Four-layer Fog Computing architecture in smart cities, in which coverage and latency sensitive applications run near the edge.

The quick feedback control provided at layers 2 and 3 acts as localized "reflex" decisions to avoid potential damage [37]. For example, if one segment of gas pipeline is experiencing a leakage or a fire is detected, these computing nodes will detect the threat and shutdown the gas supply to this section. Meanwhile, all the data analysis results at these two layers are reported to the top layer, for more complex and large-scaled behavior analysis and condition monitoring.

The top layer is a Cloud Computing data center, providing city-wide monitoring and centralized controlling. The high-performance distributed computing and storage capacity allows us to perform complex, long-term (days to years), and city-wide behavior analysis at this layer, such as large-scale event detection, long-term pattern recognition, and relationship modeling, to support dynamic decision making. This allows municipalities to perform city-wide response and resource management in the case of a natural disaster or a large-scale service interruption.

In summary, the four-layer Fog Computing architecture supports quick response at neighborhood-wide, community-wide, and city-wide levels, providing high computing performance and intelligence in future smart cities.

B. High-Performance Computing and Communication

The hierarchical Fog Computing architecture offers high-performance computing and communication to address the challenges of big data analysis and provides quick response in smart cities. It is shown that the workloads of data analysis can be parallelized on massive edge devices and computing nodes. Each edge device or intermediate computing node only performs light-weight computing tasks; thus, their massive use in parallel offers high-performance computing power for city-wide data analysis. More importantly, such parallel computing mechanisms can easily balance the throughput and load among all edge devices and computing nodes to avoid potential computing bottlenecks.

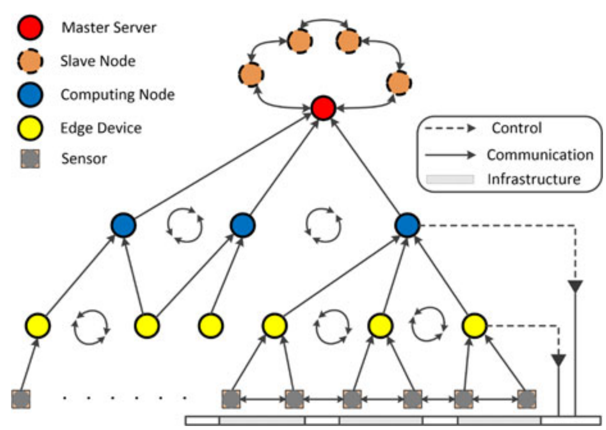


Fig. 2. Data and control flow in hierarchical Fog Computing architecture.

While the massive parallelization of Fog Computing offers high-performance computing, its hierarchical architecture also reduces burdens on communication bandwidth. We show the data and control flow among all devices in Fig. 2. The massive amount of data is generated from widely distributed sensors at layer 4. Instead of transmitting the raw data to the Cloud directly, the hierarchical distributed edge devices and intermediate computing nodes at layers 3 and 2 only upload high-level data representation while performing associated computing tasks, which can greatly reduce the data size transmitted to the Cloud. The communication within the same layer is also allowed, which enables sensor nodes or computing nodes to exchange the data with their neighbors. The communication between two adjacent layers and within the same layer could be wireless and/or wired, depending on specific smart city applications. While wireless

TABLE I
THREE LEVELS OF EMERGENCY DETECTION IN SMART PIPELINE

Layer in Fog Computing	Emergency Level	Emergency Description	Intelligent Algorithm Category
Layer 3: Edge devices	3rd Level	Disturbances (leakage, corrosion, etc.)	Pattern recognition
Layer 2: Intermediate computing nodes	2nd Level	Significant perturbation (damaged pipeline, approaching fire, etc.)	Spatial-temporal association
Layer 1: Data center on the Cloud	1st Level	Long-term and large-scale emergency events (earthquake, extremely cold or hot weather, etc.)	Complex system-wide behavior analysis

networks have many important applications in remote and mobile monitoring, wired networks are able to provide robust and reliable communication among different devices. Moreover, the computing devices in each layer can output control signal (as the dashed line signifies) in a timely manner to the monitored infrastructures to ensure their safety when anomalies and hazardous events are detected. At the top layer, a number of computers are connected via the Internet to build Clouds, performing city-level computing tasks and data storage.

IV. PROTOTYPE OF SMART PIPELINE

Pipelines play important role in resource and energy supplying and are essential infrastructure components in cities. However, several threats endanger the integrity of pipeline, such as aging and sudden environmental changes. Those threats lead to corrosion, leakage, and failure of pipelines, with serious economic and ecologic consequences [38], [39]. A smart pipeline system would be able to monitor the safety of pipeline by detecting potential hazardous events. In this section, we show that the proposed four-layer Fog Computing architecture is suitable for accurate and real-time monitoring of city-wide pipelines and provides quick responses when predefined threats and hazardous events are detected. Threat pattern recognition and hazardous event detection are performed with advanced machine learning algorithms in computing nodes. More specifically, we consider that a smart pipeline system will detect three possible levels of emergency as given in Table I. The detection of these three levels of emergency can be performed at each corresponding hierarchical layer with different intelligent algorithms. It is worth noticing that the task allocation would be different in many other smart city applications. The detailed implementation of each individual layer and the employed technologies is described in the rest of this section.

A. Layer 4: Fiber Optic Sensing Networks

At layer 4, optical fibers are used as sensors to measure the temperature along the pipeline. Optical frequency domain reflectometry (OFDR) system is applied to measure the discontinuity of the regular optical fibers [40]. With the continuous sweep method, the Rayleigh scatter (~ -80 dB) as a function of length along the fiber under test can be obtained via the Fourier transform. With the time-domain filter and cross correlation method, the extracted frequency patterns at certain locations can be used to detect the ambient physical change, such as strain, stress and temperature [41]. Enhanced sensitivities can

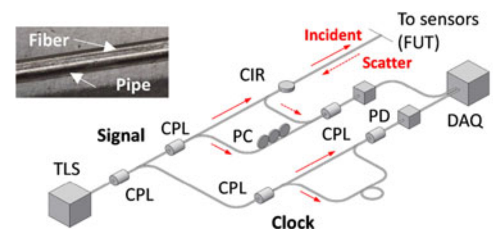


Fig. 3. Schematic of the sensor interrogation system (TLS: tunable laser source, CPL: coupler, PC: polarization controller, CIR, BPD: balanced photodiode, DAQ: data acquisition card).

be achieved by modifying the regular communication fiber with higher reflectivity [42]. Different cases, such as fire and gas leakage, would introduce the temperature change, which can be monitored by the fiber sensing network.

The Schematic of the OFDR interrogation system is illustrated in Fig. 3 [43]. Light from a tunable laser source (TLS) is split into two paths, "clock" and "signal." "Clock" is an interferometer used to calibrate the nonlinear sweep effect of the TLS by providing a corrected time base for a data acquisition card (DAQ) during frequency sweep. A comparator circuit was designed to receive "clock" signal to trigger the DAQ card to sample "signal" data. Light in the "signal" section is split between the reference and measurement arms of an interferometer via a 50/50 coupler (CPL); in the measurement path, an optical circulator (CIR) further splits the light to interrogate the low-reflection intrinsic Fabry–Perot interferometer (IFPI) array and returns the reflected light. A polarization controller (PC) is used to tune the state of polarization in the system. Another 50/50 CPL then recombines the measurement and reference fields. In this setup, the TLS sweeps from 1525 to 1555 nm at a speed of 100 nm/s, covering a total bandwidth of 3.79 THz. The fiber sensors along the pipeline system are connected to the interrogation system for real-time data collection and analysis.

B. Layer 3: Edge Device for Feature Extraction

Layer 3 is composed of parallelized small computing nodes, or edge devices. Each edge device usually performs two computing tasks. The first task is to identify potential threat patterns on the incoming data streams from sensors and to output control signal when the threat is detected. Supervised machine learning algorithms, such as neural network, support vector machine, nearest neighbors, and others, can be used to identify predefined patterns, and nonsupervised machine learning algorithms, such as clustering, can be used to detect data anomalies. Learning models are typically trained offline with experimental data.

The second computing task is to perform feature extraction and to report the results to the intermediate computing nodes at the upper layer for further analysis. Considering a region governed by one edge device with a total length of hundreds of meters, millions of temperature sensors in our high-resolution sensing network produce massive data streams and lead to a high data rate. Instead of transmitting the raw sensor data to layer 2, it is necessary to reduce the communication load between the edge devices and the intermediate computing nodes. Thereafter, raw sensor data can be discarded. In our current system implementation, only the second computing task is performed on field-programmable gate array (FPGA) as the edge device. The interested readers are referred to [37], where support vector machine is implemented on FPGA to detect potential threat patterns.

C. Layer 2: Intermediate Computing Node for Event Recognition With Spatial-Temporal Association

The intermediate computing nodes at layer 2 are connected to tens and hundreds of edge devices, governing the community-level sensors. The data streams from these edge devices represent measurements at various locations. The key is to associate the spatial and temporal data and to identify potential hazardous events.

Assume an intermediate computing node connects n edge devices, and denote a $m \times 1$ vector $\mathbf{s}_i(t)$ by the features outputted from the ith edge device at time t. Since the sensors are static, the features output from each edge device contains the geospatial information. After receiving all the features from n edge devices, we combine these n groups of feature vectors into a $mn \times 1$ feature vector $\mathbf{x}(t)$. Hence, from time 1 to time t, this intermediate computing node receives the data sequences $X = \{\mathbf{x}(1), \ldots, \mathbf{x}(t)\}$, and the task of event recognition at this layer is to recognize the event pattern given its previous data sequences.

We apply hidden Markov model (HMM) for modeling the spatial-temporal pattern of each event in a probabilistic manner. HMM is a powerful probabilistic tool to represent a dynamic process characterized by a stochastic Markov chain with unobserved or hidden states. From Markov chain theory, the hidden variable moves from one state to another (or to the same state) with a certain probability at each discrete time interval. HMM is a suitable learning model for event recognition, as one can assert that the occurrence of an event is determined by the underlying unobserved variable and different events have different hidden variable transition probabilities.

Two basic procedures are needed in HMM for classification: learning and evaluation. The occurrence of each of the N events is represented by an unique HMM model, denoted by $\text{hmm}_i(\mathbf{A}_i, \theta_i)$, $i = 1, 2, \ldots, N$. The model parameters $\mathbf{A}_i$ and θ_i of each HMM is trained offline with the Baum Welch algorithm using the collected experimental data. The Baum Welch learning algorithm is an expectation-maximization method. At the evaluation stage, the probability of observing a sequence under each individual HMM model is calculated. By applying the maximum a posterior rule, we assign a new observed sequence $\mathbf{x}_{1:t}$ to the ith event if

$$\begin{aligned} i &= \underset{i=\{1,2,\ldots,N\}}{\arg\max}\; p(\text{hmm}_i(\mathbf{A}_i, \theta_i)|\mathbf{x}_{1:t}) \\ &= \underset{i=\{1,2,\ldots,N\}}{\arg\max}\; p(\mathbf{x}_{1:t}|\text{hmm}_i(\mathbf{A}_i, \theta_i))p(\text{hmm}_i(\mathbf{A}_i, \theta_i) \end{aligned} \quad (1)$$

where $p(\text{hmm}_i(\mathbf{A}_i, \theta_i))$ is the prior probability of the ith event. If the occurrence of each event has a uniform distribution, the above decision rule can be written as

$$i = \underset{i=\{1,2,\ldots,N\}}{\arg\max}\; p(\mathbf{x}_{1:t}|\text{hmm}_i(\mathbf{A}_i, \theta_i)). \quad (2)$$

We use the efficient forward algorithm to calculate the likelihood of observing the sequence $\mathbf{x}_{1:t}$ under each HMM, where the joint probability of hidden state and observing sequence is recursively updated.

D. Layer 1: Cloud for Data Management

The top layer is at data centers of the Cloud, which collects data and information from each intermediate computing node on layer 2. We build the Cloud using the open source Hadoop, taking advantage of the power of clusters and high-performance computing and storage. The Hadoop framework consists of two major parts: MapReduce and Hadoop Distributed File System. It provides automatic parallelization of large-scale data analysis workloads and automatic data recovery from one of multiple data copies. Such Cloud models hold many desired properties: *fault tolerance* has *efficiency*. When a machine fails, the data analysis job is transparently assigned to another machine, and the data stored in this failed machine is automatically recovered with its copies from the other machine. For high efficiency, one data analysis task is executed in multiple machines, and it is marked as completed when either of these executions is completed.

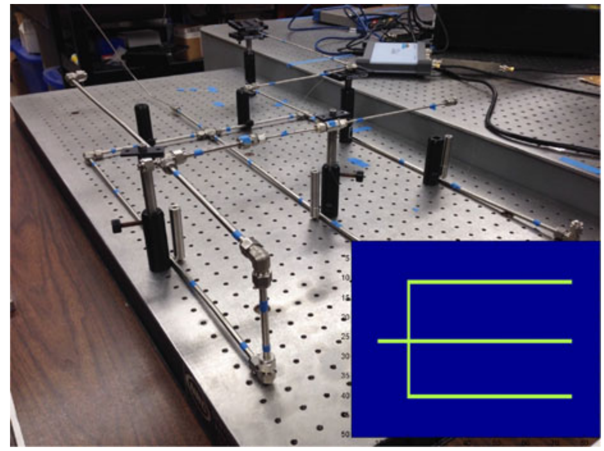


Fig. 4. Layout of prototype pipeline system.

V. Experimental Results and Analysis

A. Sensor Data Collection

In our experiment, we built a prototype of pipeline monitoring system. The layout of pipeline structure is shown in Fig. 4.

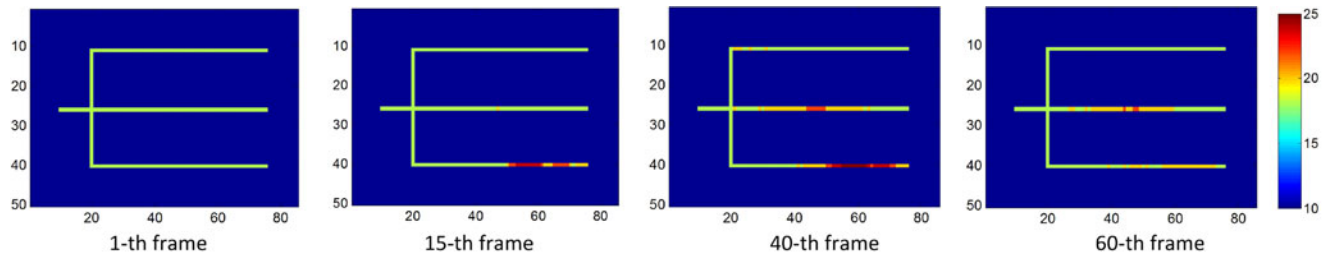


Fig. 5. One example of spatial-temporal event pattern. A colormap is used to represent the temperature.

The optical fiber sensors were distributed along this the pipeline such that the temperature of pipeline is measured. The real-time data was collected from the fiber sensor network along the prototypical pipeline system. Each sensing element contains a 1 cm sensing region, with a temperature resolution of 7.6 °C [44]. The temperature shift was extracted for each sensing element. The step between the neighboring sensing elements was 1 cm and temperature data were gathered along the pipeline system from a total of 205 sensing elements. The system-level update rate was 1 Hz with a color mapping display. In summary, the system had a temporal resolution of 1 s and a spatial resolution of 0.01 m.

We simulated multiple events around the pipeline and collected the resulting pipeline temperature sensing data. Each event includes a heating and a cooling process. A heat source was placed nearby, blowing the hot air toward the pipeline system. In each experiment, total 100 frames of data were gathered, where in the first 10 frames the system remained stable, from 11 to 40 frames the heat source was on and from 41 to 100 the heat source was off. One example of a spatial-temporal event pattern is shown in Fig. 5, where the 1st, 15th, 40th, and 60th frames are given. In each event the heat source was placed with different angle and distance to generate different patterns, and repeated 10 times with an identical setup. In total, 12 spatial-temporal events were generated for detection.

B. Feature Extraction for Sequential Learning

Features are important for machine learning algorithms. The raw sensor data usually is high-dimensional and redundant. In this prototypical system, the dimension of raw sensing data is 205, among which strong redundancy exists due to the fact that the temperature measured by one sensor node is closely correlated with the one by its neighbors. The large number of redundant and irrelevant features would lead to "overfitting" and decrease classification accuracy during testing, and meanwhile, the high dimension of data can bring high computational burdens, which may inhibit quick decision making for hazardous events. Hence, instead of using raw sensor data, two data statistics, the mean and the variance of the data sensed by a group of sensor nodes, measuring the central tendency and the dispersion, respectively, were extracted as features at each edge device and were reported to the intermediate computing nodes.

C. Spatial-Temporal Event Recognition

Each edge device connects to sensors of a neighborhood. We first assume that six edge devices are used to monitor each segment of the pipeline, as shown in Fig. 6, and evaluate their recognition performance of the 12 spatial-temporal events. As shown in Fig. 6, the intermediate computing node at layer 2 receives 12 spatial features at one time. We train a HMM model for each event. Each HMM model has Q hidden states, and the observation probability distribution is modeled by a Gaussian mixture model (GMM) with K Gaussian components. We perform tenfold cross validation to evaluate the recognition performance. All the following reported results are averaged over tenfolds. For each test data, we run online inference, i.e., at time frame t, an inference decision is made based on its currently and previously observed sequence $\mathbf{x}_{0:t}$.

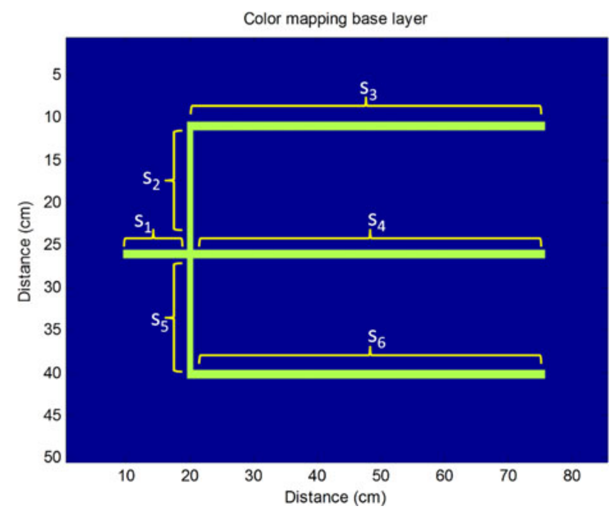


Fig. 6. Fiber optic sensors governed by six edge devices. Each edge device is connected to a neighborhood of sensors.

The online recognition performance with different number of hidden states is shown in Fig. 7, when $K = 2$ Gaussian components are used in GMM, and the performance with different number of components in GMM is given in Fig. 8, when $Q = 2$ hidden states are used. The results in Figs. 7 and 8 illustrate that using more hidden states and Gaussian components in HMM would increase the inference performance due to the growing capacity of HMMs. However, the complex HMM models need more training data for model parameters estimation and the computational complexity is increased. The results also show that we are able to obtain more than 90% accuracy to classify 12 events at the end of the heating process.

We next examined the impact of spatial distribution of feature on the inference performance. More specifically, we considered the other four modes of sensor connection to the edge devices,

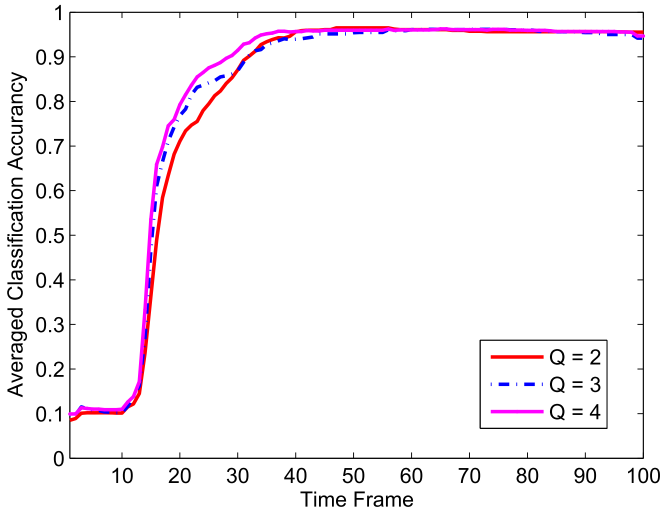


Fig. 7. Online inference performance with different number of hidden states in each HMM: $Q = 2$, $Q = 3$, and $Q = 4$, when two components GMMs are used ($K = 2$).

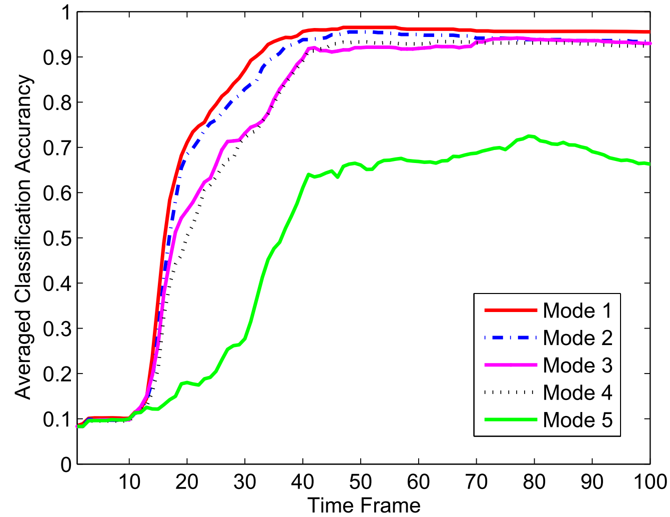


Fig. 9. Online inference performance for different sensor connections to the edge devices.

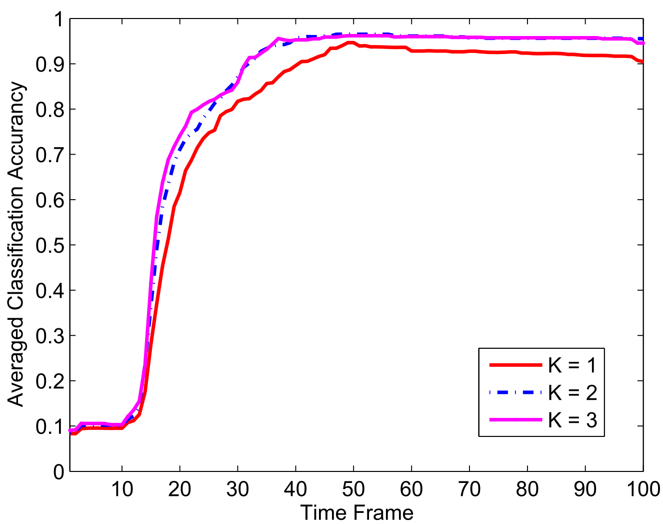


Fig. 8. Online inference performance with different number of Gaussian components in the observation distribution of each HMM: $K = 1$, $K = 2$, and $K = 3$, when two hidden states are used ($Q = 2$).

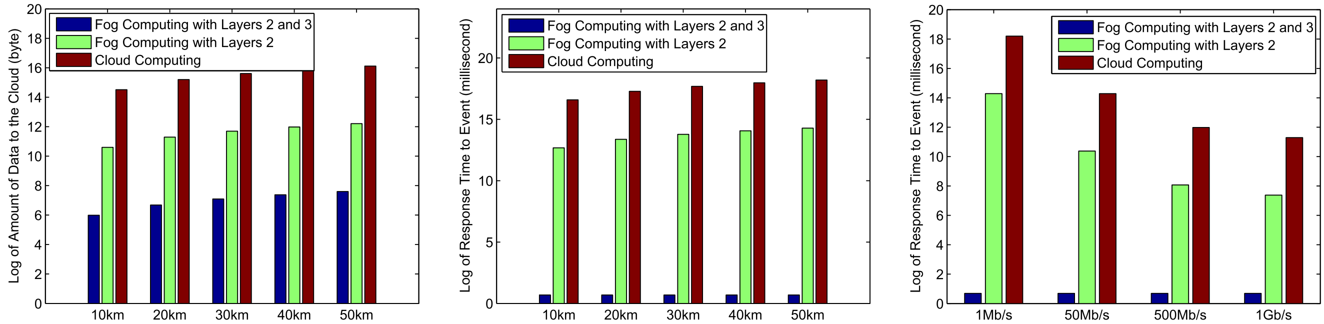
in addition to the one shown in Fig. 6, where the sensors of each pipeline segment are connected to one edge device. Denoted the ith edge device by e_i and the jth pipeline segment by s_j, the total five modes were configured as followings:

1) *Mode 1:* Six edge devices: $e_1 \leftarrow s_1$, $e_2 \leftarrow s_2$, $e_3 \leftarrow s_3$, $e_4 \leftarrow s_4$, $e_5 \leftarrow s_5$, $e_6 \leftarrow s_6$.
2) *Mode 2:* Four edge devices: $e_1 \leftarrow [s_1\ s_2\ s_5]$, $e_2 \leftarrow s_3$, $e_3 \leftarrow s_4$, $e_4 \leftarrow s_6$.
3) *Mode 3:* Four edge devices: $e_1 \leftarrow s_1$, $e_2 \leftarrow [s_2\ s_3]$, $e_3 \leftarrow s_4$, $e_4 \leftarrow [s_5\ s_6]$.
4) *Mode 4:* Three edge devices: $e_1 \leftarrow [s_1\ s_4]$, $e_2 \leftarrow [s_2\ s_3]$, $e_3 \leftarrow [s_5\ s_6]$.
5) *Mode 5:* One edge device: $e_1 \leftarrow [s_1\ s_2\ s_3\ s_4\ s_5\ s_6]$.

The averaged online inference performance for these five modes is shown in Fig. 9, when $Q = 2$ and $K = 2$. The performance comparison among these five modes shows that the performance decreases when less edge devices are used. For the pipeline with fixed length, less edge devices mean that some of them would cover more sensors. If only the global data statistics, such as the mean and variance, are extracted as features, the spatial resolution of observation in HMM modeling would be reduced. Thus, if the sensors connected to one edge device covers a long pipeline, this result suggests that it is necessary to segment sensing data and extract the statistics of each segment as features to improve the spatial resolution for spatial-temporal learning. However, it is worth noting that high spatial resolution also leads to feature irrelevance and a heavier computational burden. For example, one can get the maximum spatial resolution if the raw sensing data are used for spatial-temporal learning.

D. Discussion

The Fog Computing architecture has significant advantages over the Cloud Computing architecture for smart city monitoring. First, the distributed computing and storage nodes of Fog Computing ideally suited to support the massive numbers of sensors distributed throughout a city to monitor infrastructure and environmental parameters. If Cloud Computing alone is used for this task, huge amounts of data will need to be transmitted to data centers, necessitating massive communication bandwidth and power consumption. Specifically, suppose that we use current sensing setup with 1 cm spatial-resolution and 0.5 s time-resolution, and that each edge device covers 10-m pipeline and each computing node connects five edge devices. That is to say, each edge device at layer 3 will connect to 1,000 sensor nodes at layer 4 with the total sensing data receive rate of 4 kB/s, where each sensing data is represented by 16 bits. For each edge device, only two data statistics (features), each of which is represented by 16 bits, are reported to the computing node at layer 2; hence, each computing node has the total data receive rate of 40 B/s in our hierarchical Fog Computing architecture. It can be shown that the communication load is also reduced in a hierarchical manner. For many other applications,



(a) (b) (c)

Fig. 10. Comparisons of the amount of data transmitted to the Cloud and the response time for hazardous events within three different architectures. The log values in the y-axis are used to clearly illustrate the comparisons. (a) The amount of data that are sent to the Cloud per second. (b) The response time for hazardous events, when the Internet bandwidth is 1 Mb/s. (c) The response time for hazardous events with different Internet bandwidths.

notice that it is also possible to upload the raw data to the top layer, data center on the Cloud, for data storage and retrieve. In the prototypical system, the raw data are discarded after extracting the features at layer 3. Considering the total pipeline length L ranging from 10 to 50 km, we compare the size of data that needs to be sent to the Cloud per second in Fig. 10(a) for the following three cases: our current Fog Computing architecture with layers 2 and 3, the Fog Computing architecture with only layer 3 by removing the computing tasks at layer 2 to the Cloud, and the traditional Cloud Computing architecture in which both computing tasks at layers 2 and 3 are executed at Cloud. To clearly illustrate the difference among these three architectures, we plot log values of data size. The results in Fig. 10(a) show that using Fog Computing, the data transmitted are about 0.02% of the total size, significantly reducing transmission bandwidth and power consumption.

Second, Fog Computing supports real-time interactions. Because of high burdens on data transmission, Cloud Computing fails to provide real-time control. To quantify the response time for hazardous events under the above three computing architectures, we assume that the execution speed in computing node is $1\,G$IPS, and we omit the memory access time for simplifying our analysis. The comparison of response time for these three architectures is shown in Fig. 10(b), when the Internet bandwidth connecting to the Cloud is 1 Mb/s. It is seen that the response time is dominated by the data transmission in Cloud Computing. Fig. 10(c) also shows the response time when different Internet bandwidths are considered.

As shown in Fig. 1, different levels of latency of response can be provided in the Fog Computing, which is distinct from the batch processing of Cloud Computing. These results illustrate that Fog Computing addresses the big data analysis challenge by distributing computing tasks to the edge devices and computing nodes at the edge of network, thus offering optimal responses to changes in city environment.

VI. Conclusion

In this paper, we introduce a hierarchical Fog Computing architecture for big data analysis in smart cities. In contrast to the Cloud, the Fog Computing parallelizes data processing at the edge of network, which satisfies the requirements of location awareness and low latency. The multilayer Fog Computing architecture is able to support quick response at neighborhood-wide, community-wide, and city-wide levels, providing high-computing performance and intelligence in future smart cities. We further enhance the "smartness" of city infrastructure by employing advanced machine learning algorithms across all system layers. To verify the effectiveness of this architecture, we have implemented a prototypical system for smart pipeline monitoring. A sequential learning method, HMM, was successfully used for hazardous event detection to monitor pipeline safety. These observed performance of the hierarchical Fog Computing architecture indicates its substantial potential as a method of future smart city monitoring and control.

Acknowledgment

The authors would like to thank the anonymous reviewers for providing comments and suggestions that improved the quality of the paper. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] "Smart buildings." [Online]. Available: http://www.psrc.org/econdev/programs/smart-buildings. Accessed on: Jul. 26, 2015.
[2] "Smart cities." [Online]. Available: http://www.ibm.com/smarterplanet/us/en/smarter_cities/overview/. Accessed on: Jul. 26, 2015.
[3] "Smart cities USA." [Online]. Available: http://smartamerica.org/teams/smart-cities-usa/. Accessed on: Jul. 26, 2015.
[4] M. R. Montgomery, "The urban transformation of the developing world," *Science*, vol. 319, no. 5864, pp. 761–764, 2008.
[5] K. Davis, "The urbanization of the human population," in *The City Reader*. New York, NY, USA: Routledge, 2011, pp. 2–11.
[6] V. C. Güngör *et al.*, "Smart grid technologies: Communication technologies and standards," *IEEE Trans. Ind. Informat.*, vol. 7, no. 4, pp. 529–539, Nov. 2011.
[7] W. Su, H. Eichi, W. Zeng, and M.-Y. Chow, "A survey on the electrification of transportation in a smart grid environment," *IEEE Trans. Ind. Informat.*, vol. 8, no. 1, pp. 1–10, Feb. 2012.
[8] H. Chourabi *et al.*, "Understanding smart cities: An integrative framework," in *Proc. Int. Conf. Syst. Sci.*, 2012, pp. 2289–2297.
[9] Nature, "Big data," 2008. http://www.nature.com/news/specials/bigdata/index.html
[10] Z. Zhou, N. Chawla, Y. Jin, and G. Williams, "Big data opportunities and challenges: Discussions from data analytics perspectives," *IEEE Comput. Intell. Mag.*, vol. 9, no. 4, pp. 62–74, Nov. 2014.

[11] L. Da Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.

[12] L. Jiang, L. Da Xu, H. Cai, Z. Jiang, F. Bu, and B. Xu, "An IoT-oriented data storage framework in cloud computing platform," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1443–1451, May 2014.

[13] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proc. 1st Edition MCC Workshop Mobile Cloud Comput.*, 2012, pp. 13–16.

[14] B. Tang, Z. Chen, G. Hefferman, T. Wei, H. He, and Q. Yang, "A hierarchical distributed Fog Computing architecture for big data analysis in smart cities," in *Proc. ASE Big Data Social Informat. 2015*, 2015, pp. 28:1–28:6.

[15] K. Su, J. Li, and H. Fu, "Smart city and the applications," in *Proc. Int. Conf. Electron., Commun. Control*, 2011, pp. 1028–1031.

[16] M. Castro, A. J. Jara, and A. F. Skarmeta, "Smart lighting solutions for smart cities," in *Proc. Int. Conf. Adv. Inf. Netw. Appl. Workshops*, 2013, pp. 1374–1379.

[17] S. Dey, A. Chakraborty, S. Naskar, and P. Misra, "Smart city surveillance: Leveraging benefits of cloud data stores," in *Proc. IEEE Conf. Local Comput. Netw. Workshops*, 2012, pp. 868–876.

[18] S. Yamamoto, S. Matsumoto, and M. Nakamura, "Using cloud technologies for large-scale house data in smart city," in *Proc. IEEE 4th Int. Conf. Cloud Comput. Technol. Sci.*, 2012, pp. 141–148.

[19] N. Mitton, S. Papavassiliou, A. Puliafito, and K. S. Trivedi, "Combining cloud and sensors in a smart city environment," *EURASIP J. Wireless Commun. Netw.*, vol. 2012, no. 1, pp. 1–10, 2012.

[20] L. Filipponi, A. Vitaletti, G. Landi, V. Memeo, G. Laura, and P. Pucci, "Smart city: An event driven architecture for monitoring public spaces with heterogeneous sensors," in *Proc. Int. Conf. Sensor Technol. Appl.*, 2010, pp. 281–286.

[21] J. Jin, J. Gubbi, T. Luo, and M. Palaniswami, "Network architecture and QoS issues in the Internet of Things for a smart city," in *Proc. Int. Symp. Commun. Inf. Technol.*, 2012, pp. 956–961.

[22] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, 2010.

[23] A. Dunkels, F. Österlind, and Z. He, "An adaptive communication architecture for wireless sensor networks," in *Proc. 5th Int. Conf. Embedded Netw. Sensor Syst.*, 2007, pp. 335–349.

[24] F. Stajano, N. Hoult, I. Wassell, P. Bennett, C. Middleton, and K. Soga, "Smart bridges, smart tunnels: Transforming wireless sensor networks from research prototypes into robust engineering infrastructure," *Ad Hoc Netw.*, vol. 8, no. 8, pp. 872–888, 2010.

[25] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "Landmarc: Indoor location sensing using active RFID," *Wireless Netw.*, vol. 10, no. 6, pp. 701–710, 2004.

[26] G. De Vita and G. Iannaccone, "Design criteria for the RF section of UHF and microwave passive RFID transponders," *IEEE Trans. Microw. Theory Techn.*, vol. 53, no. 9, pp. 2978–2990, Sep. 2005.

[27] W. Zhu, J. Cao, H. Chan, X. Liu, and V. Raychoudhury, "Mobile RFID with a high identification rate," *IEEE Trans. Comput.*, vol. 63, no. 7, pp. 1778–1792, Jul. 2014.

[28] B. Lee, "Review of the present status of optical fiber sensors," *Opt. Fiber Technol.*, vol. 9, no. 2, pp. 57–79, 2003.

[29] R. Perez-Herrera and M. Lopez-Amo, "Fiber optic sensor networks," *Opt. Fiber Technol.*, vol. 19, no. 6, pp. 689–699, 2013.

[30] B. Tang, H. He, Q. Ding, and S. Kay, "A parametric classification rule based on the exponentially embedded family," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 2, pp. 367–377, Feb. 2015.

[31] B. Tang and H. He, "ENN: Extended nearest neighbor method for pattern recognition [research frontier]," *IEEE Comput. Intell. Mag.*, vol. 10, no. 3, pp. 52–60, Aug. 2015.

[32] B. Tang, Q. Ding, H. He, and S. Kay, "Hybrid classification with partial models," in *Proc. Int. Joint Conf. Neural Netw.*, Jul. 2014, pp. 3726–3731.

[33] B. Tang, S. Khokhar, and R. Gupta, "Turn prediction at generalized intersections," in *Proc. IEEE Intell. Vehicles Symp.*, 2015, pp. 1399–1404.

[34] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013.

[35] R. V. Kulkarni, A. Forster, and G. K. Venayagamoorthy, "Computational intelligence in wireless sensor networks: A survey," *IEEE Commun. Surveys Tut.*, vol. 13, no. 1, pp. 68–96, Jan.–Mar. 2011.

[36] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog computing: A platform for internet of things and analytics," in *Big Data and Internet of Things: A Roadmap for Smart Environments*. New York, NY, USA: Springer, 2014, pp. 169–186.

[37] J. Kane *et al.*, "Reflex-tree: A biologically inspired parallel architecture for future smart cities," in *Proc. 44th Int. Conf. Parallel Process.*, 2015, pp. 360–369.

[38] B. Anifowose, D. Lawler, D. Horst, and L. Chapman, "Evaluating interdiction of oil pipelines at river crossings using environmental impact assessments," *Area*, vol. 46, no. 1, pp. 4–17, 2014.

[39] R. Alzbutas, T. Iešmantas, M. Povilaitis, and J. Vitkutė,"Risk and uncertainty analysis of gas pipeline failure and gas combustion consequence," *Stoch. Environ. Res. Risk Assessment*, vol. 28, no. 6, pp. 1431–1446, 2014.

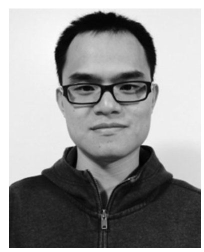
[40] M. Froggatt and J. Moore, "High-spatial-resolution distributed strain measurement in optical fiber with Rayleigh scatter," *Appl. Opt.*, vol. 37, no. 10, pp. 1735–1740, 1998.

[41] Z. Chen, G. Hefferman, and T. Wei, "Multiplexed oil level meter using a thin core fiber cladding mode exciter," *IEEE Photonics Technol. Lett.*, vol. 27, no. 21, pp. 2215–2218, Nov. 2015.

[42] Z. Chen, L. Yuan, G. Hefferman, and T. Wei, "Ultraweak intrinsic Fabry–Perot cavity array for distributed sensing," *Opt. Lett.*, vol. 40, no. 3, pp. 320–323, 2015.

[43] Z. Chen *et al.*, "FiberID: Molecular-level secret for identification of things," in *Proc. 2015 Nat. Conf. Parallel Comput. Technol.*, 2015, pp. 84–88.

[44] Z. Chen, L. Yuan, G. Hefferman, and T. Wei, "Terahertz fiber Bragg grating for distributed sensing," *IEEE Photon. Technol. Lett.*, vol. 27, no. 10, pp. 1084–1087, May 2015.



Bo Tang (S'14) received the B.S. degree in geo-informatics from Central South University, Changsha, China, in 2007, the M.S. degree in signal and information processing from the Institute of Electronics, Chinese Academy of Science, Beijing, China, in 2010, and the Ph.D. degree in electrical engineering from University of Rhode Island, Kingston, RI, USA, in 2016.

He is currently an Assistant Professor in the Department of Computer Science, Hofstra University, Hempstead, NY, USA. His research interests include statistical machine learning, high-performance computing for big data, robotics, and cyber-physical systems.

Zhen Chen received the B.S. degree in optics and optical science from Nanjing University of Science and Technology, Nanjing, China, in 2012 and the M.S. degree in electrical engineering from Hong Kong University of Science and Technology, Hong Kong, in 2013. He is currently working toward the Ph.D. degree in electrical engineering from the Department of Electrical, Computer, and Biomedical Engineering at the University of Rhode Island, Kingston, RI, USA.

In 2013, he was a Research Assistant in the Next Generation Sensing Technology Lab, University of Rhode Island in Kingston. His research interests include instrumentation, fiber-optics, and laser techniques.

Gerald Hefferman received the B.I.D. degree in industrial design from the Pratt Institute in Brooklyn, New York, NY, USA, in 2009 and the Diploma in Premedical Studies from the Harvard Extension School, Cambridge, MA, USA, in 2013. He is currently working toward the M.D. degree in medical technology and innovations from the Warren Alpert Medical School of Brown University, Providence, RI, USA.

Since 2013, he has been an Engineering Researcher in the Next Generation Sensing Technology Laboratory, University of Rhode Island, Kingston, RI, USA. He is also the CEO and co-founder of Sciolux Technologies, LLC, Kingston, RI, USA. His research interests include the development of photonics-based distributed sensor systems and their application to clinical medicine.

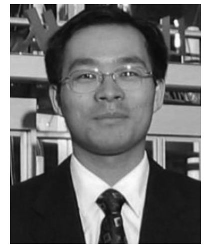
Shuyi Pei received the B.S. degree in computer science and technology from Huazhong University of Science and Technology, Wuhan, China, in 2010. She received the M.S. degree in electrical engineering from the University of Rhode Island, Kingston, RI, USA, in 2016, where she is currently working toward the Ph.D. degree in computer engineering from the Department of Electrical, Computer, and Biomedical Engineering.

In 2013, she was a Research Assistant in the High-Performance Computing Lab, University of Rhode Island. Her research interests include computer architectures, and memory and storage systems.

Tao Wei received the B.S. degree in mechanical engineering from Nanjing Technology University, Nanjing, China, in 2006, and the M.S. and Ph.D. degrees in electrical engineering from the Missouri University of Science and Technology, Rolla, MO, USA, in 2008 and 2011, respectively.

He is currently an Assistant Professor of electrical engineering at the University of Rhode Island, Kingston, RI, USA. His research interests include photonic and microwave devices for various sensing applications.

Dr. Wei received the Office of Naval Research Young Investigator Program Award 2017.



Haibo He (SM'11) received the B.S. and M.S. degrees in electrical engineering from Huazhong University of Science and Technology, Wuhan, China, in 1999 and 2002, respectively, and the Ph.D. degree in electrical engineering from Ohio University, Athens, OH, USA, in 2006.

He is currently the Robert Haas Endowed Chair Professor of electrical engineering at the University of Rhode Island, Kingston, RI, USA. From 2006 to 2009, he was an Assistant Professor in the Department of Electrical and Computer Engineering, Stevens Institute of Technology. His current research interests include adaptive dynamic programming, computational intelligence, self-adaptive systems, machine learning, data mining, embedded intelligent system design (VLSI/FPGA), and various applications such as smart grid, cognitive radio, sensor networks, and others.

Prof. He is the Editor-in-Chief of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS.

Qing Yang (M'88–SM'96–F'09) received the B.S. degree in computer science from Huazhong University of Science and Technology, Wuhan, China, in 1982, the M.S. degree in electrical engineering from the University of Toronto, ON, Canada, in 1985, and the Ph.D. degree in computer engineering from the Center for Advanced Computer Studies, University of Louisiana, Lafayette, LA, USA, in 1988.

He is currently a Distinguished Engineering Professor in the Department of Electrical, Computer, and Biomedical Engineering, University of Rhode Island, Kingston, RI, USA, where he has been a faculty member since 1988. His research interests include computer architectures, memory systems, I/O and data storage architectures, storage networking (SAN, NAS, and LAN), parallel and distributed computing (software and hardware), embedded computer systems and applications, and computer applications in biomedical systems.

Prof. Yang is a member of the ACM SIGARCH.