

# A Survey on Big Multimedia Data Processing and Management in Smart Cities

MUHAMMAD USMAN, Swinburne University of Technology, Australia

MIAN AHMAD JAN, Abdul Wali Khan University Mardan, Pakistan

XIANGJIAN HE, University of Technology Sydney, Australia

JINJUN CHEN, Swinburne University of Technology, Australia

Integration of embedded multimedia devices with powerful computing platforms, e.g., machine learning platforms, helps to build smart cities and transforms the concept of Internet of Things into Internet of Multimedia Things (IoMT). To provide different services to the residents of smart cities, the IoMT technology generates big multimedia data. The management of big multimedia data is a challenging task for IoMT technology. Without proper management, it is hard to maintain consistency, reusability, and reconcilability of generated big multimedia data in smart cities. Various machine learning techniques can be used for automatic classification of raw multimedia data and to allow machines to learn features and perform specific tasks. In this survey, we focus on various machine learning platforms that can be used to process and manage big multimedia data, generated by different applications in smart cities. We also highlight various limitations and research challenges that need to be considered when processing big multimedia data in real-time.

CCS Concepts: • **Networks** → Network types; • **Information systems** → Multimedia information systems; • **Computing methodologies** → Machine learning.

Additional Key Words and Phrases: IoMT, smart cities, multimedia, management, machine learning

## ACM Reference Format:

Muhammad Usman, Mian Ahmad Jan, Xiangjian He, and Jinjun Chen. 2019. A Survey on Big Multimedia Data Processing and Management in Smart Cities. 1, 1 (March 2019), 30 pages. <https://doi.org/10.1145/nnnnnnn>.

## 1 INTRODUCTION

Today, almost every second person is living in a city, and is connected to the Internet using one or more electronic devices [70]. As a result, within the infrastructure of a city in the modern world, billions of smart devices are communicating with different systems and applications. These smart devices include end-user devices, home appliances, surveillance cameras, traffic signals on streets and major roads, health monitoring systems, etc. The main idea behind the smart city concept is not only utilizing the unused communication, storage and computing resources to analyze and manage both multimedia and non-multimedia data generated by different applications in real-time,

---

This paper's research is partially supported by Australian Research Council projects of DP170100136 and LP140100816.

Authors' addresses: Muhammad Usman, Swinburne University of Technology, Hawthorn Campus, 3122, Melbourne, Victoria, Australia, [muhammad.usmanskk@gmail.com](mailto:muhammad.usmanskk@gmail.com); Mian Ahmad Jan, Abdul Wali Khan University Mardan, Garden Campus, 23200, Mardan, Khyber Pakhtunkhwa, Pakistan, [mianjan@awkum.edu.pk](mailto:mianjan@awkum.edu.pk); Xiangjian He, University of Technology Sydney, Broadway Campus, 2007, Sydney, New South Wales, Australia, [xiangjian.he@uts.edu.au](mailto:xiangjian.he@uts.edu.au); Jinjun Chen, Swinburne University of Technology, Hawthorn Campus, 3122, Melbourne, Victoria, Australia, [jinjun.chen@gmail.com](mailto:jinjun.chen@gmail.com).

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2019 Association for Computing Machinery.

XXXX-XXXX/2019/3-ART \$15.00

<https://doi.org/10.1145/nnnnnnn>

but also facilitating its inhabitants. The idea of smart cities can make the life of citizens easier and safer, and can help in establishing smart governments and ultimately improving the overall economy [115]. The fundamental objectives and solutions, e.g., production through smart factories and warehouses, energy production through smart grids, services through smart communities, healthcare, and hospitality, and traffic management through smart transportation systems, of smart cities are depicted in Fig. 1.

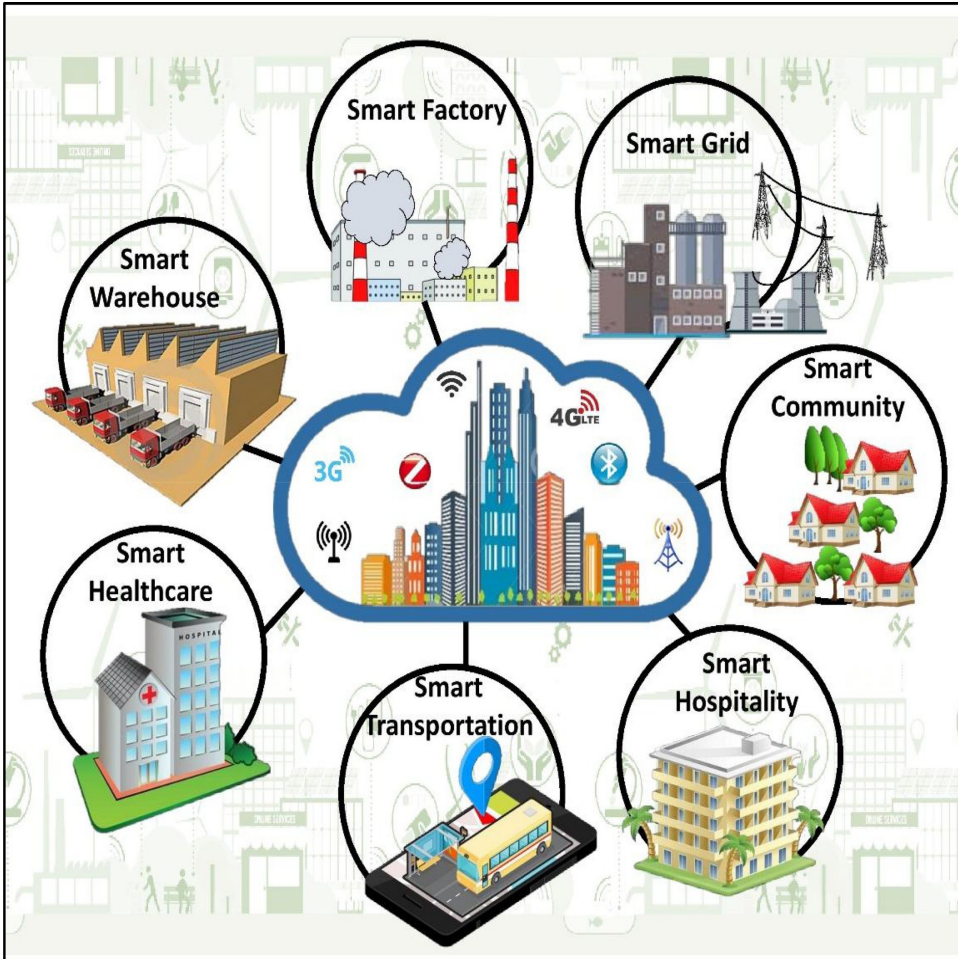


Fig. 1. Fundamental objectives and solutions for smart cities [87]

Most smart city applications, e.g., transportation management, power management, water management, waste management, health monitoring, and surveillance, generate big multimedia data. In these applications, the most important phase is data management. The data management phase can further be divided into sub-phases, i.e., data collection and processing [60, 65, 90]. The generated data can not only be used to improve the performance of these applications, but also can improve the lifestyle of inhabitants of smart cities [84, 102]. To achieve these tasks, there is a need for a unified framework to efficiently manage the generated big multimedia data. The generated data are always sensitive and time-critical, and need to be processed and managed in real-time.

Without proper management, the sensitive multimedia data can be leaked and misinterpreted, and may cause severe harms to public, and ultimately damage the reputation of the smart city government. In multimedia data management, it is always important to provide seamless data collection and processing services in an efficient and timely manner. Without proper management, the data collection and processing may not properly be done, and as a result, the goals of smart cities may be compromised.

The data collection services deal with the quality and rules related to data type and way of collection, respectively. The main purpose of the data collection rules is to maintain the consistency between the collected data, coming from various sources. In the smart cities, smart devices may be connected to Wireless Sensor Networks (WSNs), Wireless Local Area Networks (WLANs), or Mobile Ad-hoc Networks (MANETs). It is mandatory for data coming from different resources to be consistent in terms of format and type. The consistency of data helps in minimizing extra efforts required to perform a type conversion. The quality of data, on the other hand, helps in making effective decisions if the data cover many dimensions, and can be processed in clusters to extract the required information.

For data processing, data analysis tools play an important role. These tools are required to access various patterns in the input data for efficient processing and to improve the overall performance of smart city applications. For example, in a weather forecasting application, it is mandatory to extract various pieces of information, e.g., wind speed, direction, air pressure, humidity and current temperature, from the supplied data to provide an accurate weather forecast. The data may be supplied by an entire WSN, some parts of a WSN, or just a single node. Unlike the data coming from some parts of a WSN or a single node, the data coming from an entire WSN can be huge in volumes. This data need to be analyzed by different tools to find variations between different geographical locations. However, there is a trade-off between the data quality and analysis. As compared to multi-purpose data obtained from a single source, data obtained from multiple sources can create data inconsistency, and as a result, it may lead to various problems, e.g., communication and computational costs and data redundancy.

To improve the results of data processing, data analysis tools use machine learning algorithms to build predictive models for expected outcomes. In the machine learning domain, there are different types of algorithms, e.g., feature or representation learning, deep learning and data analytics algorithms. The representation learning algorithms allow a system to automatically discover representations required to detect and classify features from raw input data [10]. The automatic discovery of representations eliminates the need for manual learning and enables machines to automatically learn features and perform specific tasks based on the learned features. The deep learning, also known as a deep structured or hierarchical learning, category consists of machine learning algorithms that are based on learning data representations. Some deep learning algorithms are also a part of representation learning algorithms. The data analytics algorithms are used to examine raw input datasets and conclude useful information hidden in them [51, 77, 118]. The data analytics algorithms are widely used in commercial industries and help them in making informed decisions. These algorithms are also used by researchers and data scientists to verify models, theories, and hypotheses.

The algorithms in the deep learning and data analytics categories are divided into three main categories, i.e., supervised, semi-supervised, and unsupervised [52, 85]. The representation learning algorithms, on the other hand, are distributed into supervised, unsupervised and multilayer/deep architecture categories [31, 52, 64]. In these categories, the algorithms have the ability to analyze run-time behaviors of various objects and services before forwarding the data to the next phase. However, the outputs of representation learning algorithms depend on the amounts, nature, and dimensions of input data. Examples of the state-of-the-art representation learning algorithms in the supervised

category are supervised neural networks and supervised dictionary learning. The unsupervised category consists of  $k$ -means clustering, principal component analysis, local linear embedding, independent component analysis, and unsupervised dictionary learning. To avoid overlapping between the representation learning and deep learning, we discuss multilayer/deep architecture in the deep learning category. Well-known algorithms in this category are deep convolutional networks, deep belief networks, restricted Boltzmann machines, contractive autoencoders, and long short term memory networks. These algorithms are powerful and offer many predictive services. Popular algorithms in the data analytics category are support vector machines, Bayesian classifier,  $k$ -nearest neighbors classifier, linear discriminant analysis, learning vector quantization, and regression trees.

In this survey, we present a discussion on how different machine learning algorithms can be used to process and manage big multimedia data in the smart cities. We concentrate on most recent advances made in the past few years along with challenges and limitations. Various machine learning algorithms to facilitate computing technologies for big multimedia data in the smart cities are also highlighted in this survey. The rest of this survey is organized as follows. Contributions and recent surveys on smart cities are presented in Section 2. Deployment of various smart city applications generating big multimedia data is discussed in Section 3. Multimedia data collection and processing related issues are presented in Sections 4 and 5, respectively. Finally, the survey is concluded in Section 6.

## 2 SURVEYS AND CONTRIBUTIONS

In the past few years, many surveys were presented discussing various applications of smart cities. However, none of them has discussed the involvement of multimedia data in various smart city applications. Various state-of-the-art techniques to integrate Internet of Things (IoT) and cloud computing for big data management were reviewed and discussed in [21]. Extensive interdisciplinary reviews were conducted for smart sustainable cities of future in [12, 117]. These reviews also highlighted various critical issues in data collection and management. Various technical issues, e.g., data collection, aggregation, analytics and access, in smart city applications were discussed in [8]. Various data aggregation mechanisms for IoT applications were reviewed and analyzed in [75]. The studies presented in [8, 12, 21, 75, 117] discuss numerous research issues for data collection and management in various applications of smart cities. However, a discussion on multimedia data collection and management is still missing in these studies.

A participatory sensing-based prototype platform to efficiently collect and manage data from smartphones for various smart city applications, e.g., transportation management, environmental monitoring, and public health, was discussed in [109]. A mobile cloud computing platform for big healthcare data sharing, processing and analysis in the smart cities was proposed in [40]. Large-scale distributed (dedicated and non-dedicated) smart city sensing systems for capturing, preprocessing and transmitting data to centralized centers were discussed in [27]. A large-scale data analytics platform for smart cities was proposed in [76]. An adaptive IoT and social sensing platform to support the selection of most appropriate data sources, and establish and control data communication in the smart cities was proposed in [74]. The problem of imperfection of collected data in the smart cities and its impact on information retrieval and data integration was discussed in [89]. The platforms presented in [27, 40, 74, 76, 89, 109] can be helpful to deal with various technical issues related to data processing in smart city applications. However, these solutions focus on the processing of non-multimedia data only, and as a result, may not be feasible for real-time multimedia data.

This survey is different from aforementioned studies due to its focus on multimedia data management in the smart city applications using various machine learning algorithms. The major contributions of this survey can be summarized as follows.

- We present and discuss three major applications of smart cities, e.g., transportation management, healthcare, and surveillance. These applications are discussed in terms of their implementation strategies and requirements.
- We provide an in-depth overview of various machine learning algorithms that can be used for big multimedia data management in terms of collection and processing.
- Finally, we highlight and discuss various research challenges that can be faced during big multimedia data management in the smart city applications.

### 3 SMART CITIES AND MULTIMEDIA APPLICATIONS

In this section, we discuss three major applications of smart cities, i.e., transportation management, healthcare, and surveillance. We also provide a discussion on various components used in these applications.

#### 3.1 Transport Management Systems

Transport management is one of the major applications of smart cities. It empowers its users to make smart decisions for parking and driving based on the current statuses of transport in surrounding areas. Various technologies, such as traffic signals at main junctions, vehicular navigation systems, and Close-Circuit Television Cameras (CCTV), are used to help drivers in making smart decisions. In these technologies, the CCTV is considered to be the most reliable one. It shows a live picture of surrounding environments and volumes of traffic at certain places. The smart transport management services are already implemented in various international cities, e.g., Boston [36], New York [39], and Louisville [37]. An example scenario of the smart transport management system is shown in Fig. 2.

As shown in the above figure, the data can be obtained from multiple road sensors, e.g., road-side sensors and in-road sensors. Other smart technologies, e.g., embedded sensors in vehicles and video cameras at major traffic junctions and streets, also provide data to manage the traffic. In the case of road sensors, whenever a vehicle passes through a WSN, its presence and direction are detected and determined and forwarded to a Traffic Management Controller (TMC). Based on the received information from the WSN, the TMC can easily decide about on-peak and off-peak traffic hours and estimate the total number of vehicles during specific traffic hours. While driving, many people use their smartphones for playing music, making calls, and finding routes to destination. The presence of smartphones in the vehicles can easily be detected when the vehicles pass through the WSNs installed at various highways, streets, and city roads. The installed WSNs can use the built-in sensors of smartphones to obtain the required information. Furthermore, the global positioning system can also help in finding the locations of smartphones. Travel speed of a vehicle can be estimated using the time and location information when the presence of a smartphone is detected for the first time and the time when the same smartphone is detected again at a different geographical location.

Speed monitoring and surveillance cameras are usually installed on the top of poles at highways, city roads, streets, and car parking to track speeds, entries, and exits of vehicles. These cameras work 24/7 and continuously record and forward videos to the a centralized server. In practical scenarios, the WSNs may be affected by various natural factors, e.g., bad weather and signal interference. The multimedia data captured by roadside cameras are always considered reliable and authentic. Furthermore, the cameras can record clearly and precisely at a very long range. Unlike the traditional WSNs, few cameras are enough to cover a large geographical area at different

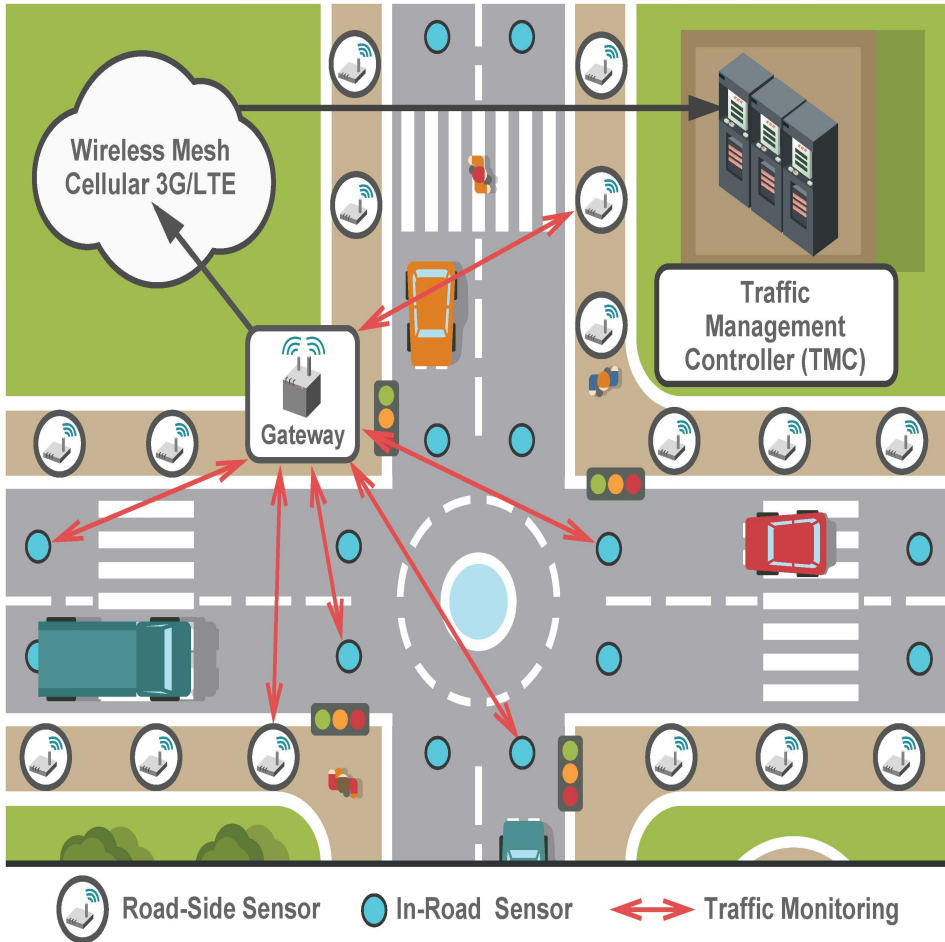


Fig. 2. A smart traffic management system [59]

angles. If one camera is compromised due to some technical fault, another camera in a nearby location can easily cover its area without any disruption in the recordings. If the recorded videos are used properly, it can be helpful for the TMC to not only manage traffic at different geographical locations, but also suggest drivers to take alternate routes by sharing recorded multimedia data in the form of images and videos.

### 3.2 Healthcare Systems

Inexpensive and high quality healthcare services are very important in the smart cities. However, in remote areas and towns, it is hard to find a medical facility that is available 24/7 with specialist doctors and nurses. This challenge poses a need for a smart healthcare system. In the smart healthcare system, objects and technologies like wearable sensors and Information and Communication Technologies (ICT), can help in providing necessary health services [7]. The smart healthcare services are already implemented in the Saensuk in Thailand [19]. An example smart healthcare system is depicted in Fig. 3.

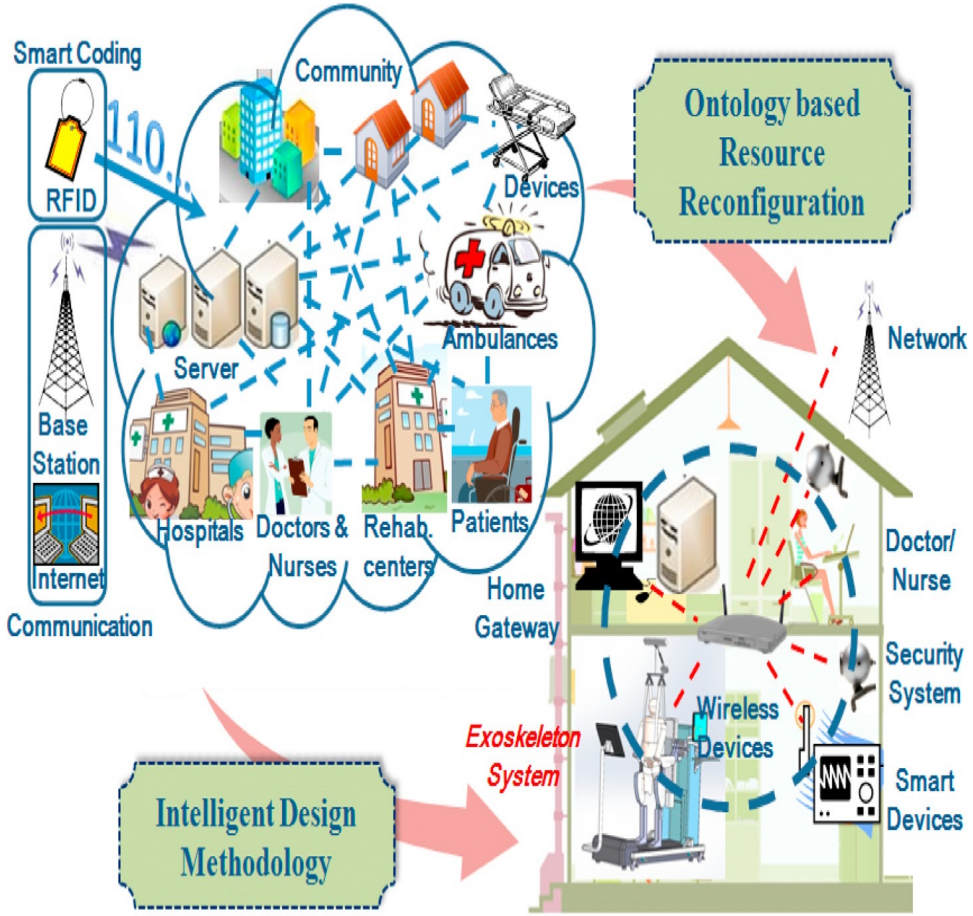


Fig. 3. A smart healthcare system [114]

As shown in the above figure, the smart healthcare systems also include rehab centers and remote patient monitoring through emergency and community services. Hospitals in the smart healthcare systems offer various services, e.g., virtual operating theaters and remote observations to handle patients with different requirements. In the virtual operating theaters, a surgeon can perform a delicate operation under the guidance of a team of experts from a remote hospital. The virtual operating theaters can be used to train new medical staff and interns by sharing live and stored videos of various operations. In the remote observation scenario, a patient can be at home and need to be observed remotely. The remote observation assists a carer to deal with patients with different medical conditions. Patients' data can be shared with different hospitals through the ICT and real-time decisions can be made by processing the shared data using various data analysis techniques.

In the traditional healthcare systems, sensing devices are usually deployed close to patients or attached with patients' bodies to get various measurements, e.g., heart rates, body temperatures, and muscle movements. In either case, the patients have to be within the range of sensing devices



to provide accurate measurements. However, it is not easy to carry sensing devices everywhere. Sometimes, the patients may not be comfortable in carrying the sensing devices with them everywhere. In this situation, an alternate option is to monitor activities and behaviors of patients using smart multimedia systems. Processing the sensed data generated by traditional healthcare systems and producing the final results of patients' medical situations may be time-consuming processes. Chances of errors in the generated results are also possible and cannot be ignored. On the other hand, the multimedia systems can help remote doctors to quickly diagnose a patient's symptoms and they can suggest a quick medical aid to the carer of the patient in emergency situations. Later, the doctors can get the final results from the processed sensed data and match with the multimedia data to make final and accurate decisions.

### 3.3 Surveillance Systems

In the smart cities, the main goal is to improve the overall sustainability and efficiency of urban areas through the ICT at low costs. In this context, surveillance is one of the major concerns in the urban areas. The inhabitants of smart cities can be monitored using different ICT-based technologies, e.g., global positioning systems based tracking and WSNs. The collected data can be processed and analyzed by various government organizations to improve the lifestyle of inhabitants of smart cities through better urban plannings and prevention of local crimes. The surveillance data (multimedia and non-multimedia data) captured and forwarded by various technologies can be used by government agencies to take necessary actions based on proactive and predictive policies [100]. The smart surveillance systems have been implemented in various international cities, e.g., Santa Cruz, Barcelona, Amsterdam and Stockholm [105]. An example of implementing the smart surveillance system in the smart cities is shown in Fig. 4. As shown in this figure, the security cameras mounted at poles can provide views of different spots in the smart cities.

For a successful implementation of smart surveillance systems, various factors need to be considered and documented. For example, within the smart cities, all public areas should be under surveillance observations to protect the inhabitants and public assets. All shops and shopping malls need to have CCTVs installed to monitor different activities. Other factors, such as installation, mounting angles, coverage, resolution, maintenance, upgradation, and connectivity to the Internet and back-end servers, also need to be considered.

The installation of smart surveillance systems at public places may also raise privacy issues. People usually do not feel comfortable in observed environments. Their recordings can be used for malicious purposes by third parties. Therefore, it is important to follow predefined standard privacy and security policies before installing smart surveillance systems. The recorded videos should be stored on secured servers. The transmission of recorded videos also needs to be protected. The privacy and security policies enforce various laws when recorded videos are shared between different organizations and agencies. Based on the importance and sensitivity of data, the recorded videos can be stored for shorter or longer time periods.

## 4 MULTIMEDIA DATA COLLECTION PLATFORMS

In the smart city applications, the most important factor is how to collect and manage data to meet different requirements. For data collection, various standards and quality aspects related to the use of collected data need to be considered. The data collection standards define rules about how to process similar data types. On the other hand, the data quality aspects define the usefulness of collected data for targeted applications. The use of collected data focuses on whether the collected data can be used by more than one application or not. In the smart cities, the data collected from various resources may include multimedia and non-multimedia contents. Therefore,





Fig. 4. A smart city with a security and surveillance system [23]

there is a need for a unified framework to support the collection and sharing of both multimedia and non-multimedia data.

In the following subsections, we provide an overview of three major technologies that can be used to collect data in the smart cities. These technologies can capture both multimedia and non-multimedia data. As a result, they can be used in different smart city applications. However, each technology has its own limitations. The benefits, challenges and limitations of these technologies are summarized in Table 1. We also discuss various issues that need to be considered when using these technologies in the smart city applications.

#### 4.1 Wireless Multimedia Sensor Networks

In Wireless Multimedia Sensor Networks (WMSNs), the sensor nodes are able to capture and process multimedia data through embedded components, e.g., cameras and microphones. These sensor nodes are known as Multimedia Sensor Nodes (MSNs). The MSNs can capture, sense and transmit both multimedia and non-multimedia data, simultaneously [1]. Due to this ability, the

Network Technology	Benefits	Challenges	Limitations
Wireless Multimedia Sensor Networks [2–4, 63, 73, 101, 116]	<ul style="list-style-type: none"> <li>• Low cost</li> <li>• Easy management</li> <li>• Easy deployment</li> <li>• Scalability</li> </ul>	<ul style="list-style-type: none"> <li>• Real-time multimedia transmission</li> <li>• High network concurrent access</li> </ul>	<ul style="list-style-type: none"> <li>• Standardization issues</li> <li>• Short-range wireless coverage</li> </ul>
Multimedia Ad hoc Networks [16, 20, 22, 24, 50, 58, 61, 69, 78, 95, 110]	<ul style="list-style-type: none"> <li>• Infrastructure-free</li> <li>• Fault tolerance</li> <li>• Scalability</li> </ul>	<ul style="list-style-type: none"> <li>• Real-time multimedia transmission</li> <li>• Use of multiple communication technologies</li> </ul>	<ul style="list-style-type: none"> <li>• Non-uniform node distribution</li> <li>• Unsecured</li> </ul>
Internet of Multimedia Things [5, 14, 33, 45, 46, 67, 80, 83, 86, 92–94, 97, 99, 103, 113]	<ul style="list-style-type: none"> <li>• Low-cost</li> <li>• Daily task automation</li> <li>• Tracking capabilities</li> </ul>	<ul style="list-style-type: none"> <li>• Real-time multimedia transmission</li> <li>• Data fusion and analysis</li> <li>• Scalability</li> </ul>	<ul style="list-style-type: none"> <li>• Standardization issues</li> <li>• Short-range wireless coverage</li> </ul>

Table 1. Communication network benefits and challenges

MSNs generate huge volumes of multimedia and non-multimedia data traffic and require concurrent access to network resources. The WMSNs are used in a wide variety of applications, e.g., military and civil applications, and provide accurate data. These networks are low cost and easy to manage. Based on the versatility feature in terms of data collection, the WMSNs can easily be deployed, scaled, and used in many smart city applications, e.g., surveillance, automated assistance to elderly people, industrial automation, and many more. However, due to the heterogeneous nature of MSNs, the WMSNs may face various standardization issues, e.g., changes in protocols, operating systems, applications, and hardware of devices. Furthermore, some devices may not be able to communicate over a long range and may be restricted to specific applications. In this section, we provide a discussion on the use of WMSNs in smart city applications.

For smart city applications, various types of sensors are suggested in the literature. However, our focus is on the use of WMSNs in the smart cities. A survey on various applications and testbeds of WMSNs in the smart cities was presented in [2]. In this survey, issues like controlling the network environment and enhancing the network performance are discussed in detail. A survey on multimedia streaming over WSNs was presented in [63]. In this survey, various technical issues related to multimedia streaming over low-bandwidth networks are highlighted and discussed in detail. A survey on recent developments in WMSNs was presented in [4]. This survey highlights current trends and future directions that need to be considered when using WMSNs in different applications. The surveys presented in [2, 4, 63] address a wide range of research challenges and technical issues related to WMSNs' deployment and can be helpful in improving the performance of smart city applications.

A framework based on wireless visual sensor network was proposed for smart cities in [73]. In this framework, mobile sinks are used to collect data from underlying sensor network to support various smart city applications, e.g., surveillance and road safety. A framework based on mobile MSNs was proposed in [3]. In this framework, mobile MSNs are used to deal with multi-hop communication to support applications like intelligent transport systems. A joint active duty scheduling and encoding rate allocation based framework was proposed for WMSNs in [101]. The main purpose of this framework is to support different smart city applications to reduce energy consumption in computationally complex tasks, e.g., video capturing, packetization, and transmission. A framework based on virtual streaming was proposed in [116]. In this framework, the streaming is based on the artificial fish-swarm technique, and is used to increase the coverage of WMSNs in the smart transport systems. The frameworks proposed in [3, 73, 101, 116] use both fix and mobile nodes and can be helpful in improving the performance of various smart city applications.

## 4.2 Multimedia Ad-hoc Networks

Due to the availability of high bandwidth wireless technologies and low-cost mobile devices with multimedia processing abilities, many popular real-time multimedia applications can now be streamed over multi-hop networks. Few popular examples are online gaming, video conferencing, mobile TV, peer-to-peer streaming, and video surveillance. The most popular and common example of multi-hop networks is multimedia ad-hoc networks. These networks are formed over multiple wireless technologies and consist of mobile devices. These networks are also known as infrastructure-free networks due to frequently changing network topology. Due to their infrastructure-free nature, they are easy to scale. Furthermore, due to the mobility of devices, the nodes may be distributed at different geographical locations in a non-uniform style. Although they are based on unreliable wireless channels and face issues like security, network congestion, and resource contention, they are still popular due to their fault tolerance ability [98]. In this section, we limit the discussion on the use of multimedia ad-hoc networks in the smart city applications.

A survey on technical challenges of using multimedia applications of smart cities on software-defined vehicular ad-hoc networks was presented in [16]. In this survey, the impact of software-defined networks over smart cities is discussed in detail. Another survey on multimedia streaming over information-centric networks was presented in [58]. In this survey, most recent advancements in the field of information-centric networking are highlighted as future research challenges for smart city applications. A communication-oriented perspective of smart transport management systems for smart cities was presented in [22]. In this survey, various networking technologies are discussed to deal with traffic management problem using multimedia and non-multimedia data. The concept of vehicular social networks in the context of Internet of vehicles to control traffic mobility using social multimedia information was discussed in [69]. In this survey, existing supporting communication technologies are discussed and various research challenges and open issues are highlighted. The surveys presented in [16, 22, 58, 69] discuss various communication technologies that can be used to support smart city applications. These surveys also highlight various technical challenges that need to be considered when streaming multimedia data over discussed communication technologies.

A priority enforced algorithm for multimedia transmission over wireless ad-hoc networks and its applications in the smart cities was proposed in [20]. In this paper, the issue of long waiting time to send a video frame on transmission links is addressed. The performance of scalable video streaming using H.264 standard over ad-hoc networks in the smart cities was evaluated in [24]. In this paper, a cross-layer routing protocol is introduced to estimate the available bandwidth. A combined framework to support video streaming and node authentication was proposed for mobile ad-hoc networks in [95]. In this framework, optimal transmission links between authorized nodes are selected to improve the quality of service. A scalable solution for unicast video streaming over vehicular ad-hoc networks to support smart city applications was proposed in [78]. In this paper, the problem of load balancing between relay nodes is targeted. A quality of service aware hierarchical web caching scheme to support online video streaming applications over vehicular ad-hoc networks in the smart cities was presented in [50]. In this paper, intelligent transport systems and mobile multimedia applications are targeted in vehicular ad-hoc networks. An information-centric multimedia streaming framework for vehicular ad-hoc networks was presented in [110]. In this paper, issues like quality of experience and energy efficiency, in the smart city applications are discussed in detail. A multimedia map-aware routing protocol to support video-reporting messages over vehicular ad-hoc networks in the smart cities was proposed in [61]. In this paper, a smart transport system was targeted to avoid the occurrence of road accidents. The approaches and frameworks presented in [20, 24, 50, 61, 78, 95, 110] deal with different problems when multimedia

data are streamed over communication networks. These approaches and frameworks can be useful to improve the overall quality of service of smart city applications.

### 4.3 Internet of Multimedia Things

Due to the popularity of multimedia applications in our daily lives, services like seamless integration, cooperative sensing, connectivity, and autonomy, are required in the IoT framework. Furthermore, popular applications of IoT, e.g., surveillance, healthcare, and transport management, generate huge volumes of real-time multimedia data. The generated data need to be processed, transmitted, fused, and analyzed in an efficient way to improve the performance of IoT applications. To deal with these challenges, the concept of Internet-of-Multimedia-Things (IoMT) has been introduced [43]. The IoMT framework consists of low-cost devices and applications that offer different services like daily task automation and tracking of suspicious objects. The IoMT devices and applications are able to deal with both multimedia and non-multimedia data, simultaneously [96]. The IoMT framework is easy to scale, however, it may face various standardization issues, e.g., changes in protocols, operating systems, applications, and hardware of devices. Furthermore, some devices may not be able to transmit over a long range, and as a result, they may be restricted to specific applications. In this section, we provide a discussion on the use of IoMT in smart city applications.

A survey on routing schemes and challenges in the WMSNs for IoMT applications was presented in [86]. In this survey, the routing schemes are classified based on their design and optimization objectives. A survey on the visions and challenges of IoMT was presented in [5]. In this survey, various aspects of IoMT, such as multimedia sensing, addressing, cloud, and multi-agent systems, are discussed. The concept of Internet of multimedia nano things was presented in [45]. In this concept, various research challenges and future research trends of multimedia data processing over nano-things are discussed. The surveys and concept presented in [5, 45, 86] highlight various aspects of IoMT and can be useful to improve the performance of smart city applications.

Algorithms to maintain a balance between video coding and the received video quality when maximizing the overall network lifetime in the WMSNs were proposed in [46, 97, 99]. These algorithms aim to support various applications of IoMT. These algorithms are dynamic in nature and can deal with channel transmission errors. A distributed algorithm to solve an inter-layer optimization problem based on network coding to support multimedia multicast traffic in IoT applications was proposed in [103]. This algorithm uses primal decomposition to provide global convergence and stability. A compressed sensing-based security approach for smart video surveillance in the smart cities was proposed in [67]. In this approach, the WMSNs are used as testbed to compute the security processing overhead. A cross-layer routing approach for efficient video dissemination in mobile multimedia applications of IoT was proposed in [80]. This approach relies on beaconless routing and uses multiple metrics for routing decisions. A comparison between the performances of various joint coding solutions based on application layer to support video transmission between mobile devices in IoT applications was provided in [14, 92–94]. In this comparison, the quality of video streaming is evaluated based on quality of experience. A quality-optimized multimedia information gathering scheme for a WSN-based IoT framework was proposed in [113]. In this scheme, power control and relay node selection strategies are jointly optimized. A routing scheme to support multimedia data transmission in IoT was proposed in [33]. In this scheme, spanning tree and shortest path routing algorithms are utilized to support multimedia data transmission. An adaptive protocol for multimedia data transmission in the IoT environment was proposed in [83]. This protocol is based on a real-time transport protocol and a real-time control protocol and focuses on wireless channel diversity and network status. The approaches presented in [14, 33, 46, 67, 80, 83, 92–94, 97, 99, 103, 113] are very useful in improving the overall quality of

service when streaming multimedia data over communication networks. These approaches can be used by smart city applications to ensure a smooth end-to-end delivery of multimedia data.

#### 4.4 Research Challenges in Data Collection

In Subsections 4.1,4.2,4.3, we have discussed various networking technologies and techniques that can be used to collect multimedia data in IoT-related frameworks. In this section, we highlight various issues that need to be considered when using these technologies and techniques to collect multimedia data in the smart cities.

**4.4.1 Management and Coordination Issues.** Each technology uses a different technique to collect data. For their deployment in the smart cities, it is very important for technologies to be synchronized and coordinated. Proper synchronization and coordination help in making smart decisions in real-time. For example, in the smart surveillance and transport systems, different technologies can be combined together to track suspicious objects. The suspicious objects can be human beings or vehicles. Visual sensors and cameras are mainly deployed on main roads, junctions, and highways. Small streets are usually uncovered. In this scenario, technologies like wireless networks, can be used to track suspicious objects. However, without proper coordination and synchronization between organizations controlling wireless networks, laws may be violated and suspicious objects may not be tracked well. As a result, serious public safety issues may occur.

**4.4.2 Data Integrity and Quality Issues.** Smart city applications generate huge volumes of data. In the case of multimedia data, facts like quality and integrity of the generated data, need to be ensured by various technologies. The technologies should be well-scaled and able to identify useful data quickly in order to minimize the redundancy in the generated data. For example, in the smart surveillance systems, cameras generate multimedia data 24/7. The generated multimedia data are huge in volumes when captured through high-resolution colored cameras. However, it is possible that in some areas, no suspicious activity may happen for several hours or days. In this case, the captured multimedia data are redundant and need not to be processed. On the other hand, if any suspicious activity happens, the cameras should be able to identify and immediately respond to the back-end servers for quick actions. In this situation, if the quality and integrity of captured multimedia data are compromised, some serious incidents may happen.

Another example is the smart transport systems where the quality and integrity of generated data play important roles in optimizing the traffic flows in different areas of the smart cities. A timely management of data and quick response help in making life of the inhabitants of smart cities easier by managing traffic at peak hours. Drivers can report traffic situations and jams at places where smart cameras are not deployed. Drivers can also suggest about alternate routes, thus helping other drivers by informing them current situations of traffic at different places in the smart cities.

**4.4.3 Computing Platform Issues.** Data processing requirements vary from one application to another in the smart cities. Data collection and management organizations need to decide whether the collected data should be processed locally or sent to nominated cloud platforms. This decision depends on various factors, e.g., nature of applications, time deadlines to make decisions, and processing capabilities of local computing platforms. For example, in the smart transport management systems, the collected data can be processed locally for an immediate action to control the traffic and signals at various locations. On the other hand, the data generated from smart surveillance and healthcare systems need to be sent to the cloud platforms. This data are always huge in volumes and need to be shared with other organizations dealing with public safety and healthcare.

## 5 MULTIMEDIA DATA PROCESSING

Multimedia data collected from various smart city technologies and applications are always heterogeneous in nature and bulk in volumes. It becomes a challenging task to process such big data in real-time and extract meaningful information [29, 54, 71]. In this section, we discuss how the collected multimedia data can be processed efficiently using representation learning, deep learning, and data analytics algorithms.

### 5.1 Data Processing Using Representation Learning Algorithms

The concept of representation learning was motivated by the concept of classification in the field of machine learning. The classification often requires input that are mathematically and computationally convenient to process. However, the multimedia data do not contain algorithmically defined features set. The best way to discover the required features from multimedia data is through representations and examinations without relying on explicit algorithms and techniques. In the representation learning domain, various algorithms are developed and used to infer patterns from collected multimedia data. The inferred patterns can later be used by computer programs to adjust actions accordingly. In general, the representation learning algorithms are classified into three major categories, i.e., supervised, unsupervised and multilayer/deep architectures [31, 52, 64]. In the supervised learning category, a training dataset is required to make a final decision. For example, an initial input and an expected output are always provided in the supervised learning algorithms. The algorithms infer patterns from provided datasets and apply the same inferred patterns on newly provided input and output datasets. In the unsupervised learning category, the algorithms work without a labeled training dataset. The algorithms need to find hidden patterns in the provided data without any instructions and guidance. In the multilayer/deep architectures, the algorithms are inspired by human neurological systems and stack multiple layers of learning nodes. These architectures are based on distributed representations. Multiple interactions based on different factors are performed between different levels to generate the observed data. Each intermediate layer generates an output that can be viewed as a representation of the original data. The output of each level is used as an input for next levels to produce a new representation, i.e., the original data are the input of the first level and final representation of features is the output of the last level.

**5.1.1 Representation Learning Algorithms.** Each category, i.e., supervised, unsupervised and deep architecture, consists of multiple representation learning algorithms. In this section, we discuss literature on supervised and unsupervised algorithms only. The algorithms in the deep architecture category are discussed in Section 5.2. The supervised learning category consists of supervised dictionary learning and supervised neural networks. The unsupervised learning algorithms category consists of  $k$ -means clustering, principal component analysis, local linear embedding, independent component analysis, and unsupervised dictionary learning. In the following subsections, we discuss these algorithms in detail.

**Supervised Dictionary Learning.** In the dictionary learning based approaches, a set/dictionary of representative elements is developed from input data. The input data are represented as a weighted sum of representative elements to minimize the average representation errors. In the case of multimedia data, the Supervised Dictionary Learning (SDL) algorithms exploit the structures of input multimedia data and labels of dictionary elements to formulate the minimization problem with minimum classification errors. In literature, there exist surveys that discuss various SDL algorithms along with their architecture and performances from different perspectives. For example, surveys on SDL algorithms was presented in [30]. In this survey, different learning algorithms are explained, analyzed and evaluated based on various performance metrics. Based on features, the SDL

algorithms are distributed into four categories, i.e., reconstructive approach, generative approach, discriminative approach, and multiclass extension [57]. In each category, the SDL algorithms follow different mathematical models for learning, preventing overfitting, and minimizing the overall cost.

**Supervised Neural Networks.** Supervised Neural Networks (SNNs) are just like simple neural networks with a supervised feature [34]. The SNNs consist of multiple layers with interconnected nodes. The SNNs comprise of multiple edges each with an associated weight. Multiple computational rules are defined in each SNN to pass input data from input to output layers. The relationship between input and output layers is defined through a network function which is parameterized by some weights. Various learning tasks can easily be performed if network functions are properly defined. A most common example of multilayer SNNs is Siamese networks [11].

***k*-means Clustering.** In the *k*-means clustering approach, the input is distributed into *k* clusters in a way that each member of the input belongs to the cluster with the closest mean [28]. This algorithm is usually used to distribute unlabeled sets of inputs into multiple clusters. The centroid of formed clusters are used to produce features of unlabeled data. There are different ways to generate the features out of which the most simple one is based on the addition of *k* binary features [17]. It is also believed that some variants of *k*-means clustering algorithm are similar to sparse coding algorithms [18]. Furthermore, it is noted that the *k*-means clustering algorithm can outperform autoencoders and restricted Boltzmann machines in certain tasks, e.g., image classification. It can also be used in the natural language processing domain for named-entity recognition and can perform better than the Brown clustering algorithm [91].

**Principal Component Analysis.** The Principal Component Analysis (PCA) is used to reduce the dimension of input data [107]. For an unlabeled set of *n* input data vectors, the PCA generates *m* right singular vectors where the dimension of *m* is much smaller than the dimension of the input data. The generated *m* vectors correspond to a data matrix of large singular values. The generated *m* singular vectors are called feature vectors that are learned from the input data. These feature vectors are used to represent the directions along which the input data has the largest variations. The PCA is a linear feature learning approach and the *m* vectors are generated through simple iterative algorithms. The PCA has several limitations. For example, it is based on an assumption that the directions with large variance are the most important ones which may not be true. Secondly, the PCA exploits only the first and second-order moments of input data which may not well characterize data distribution. Thirdly, the PCA can reduce the dimensions only if the input data vectors are correlated which may not be true in most cases.

**Local Linear Embedding.** The Local Linear Embedding (LLE) is a nonlinear learning approach [81]. It is used to generate low-dimensional representations from unlabeled high-dimensions input data. There are two major steps in the LLE approach. In the first step, each input data point is reconstructed from weighted sums of *k* nearest neighbor data points. The optimal weights are computed by minimizing the average squared reconstruction error. In the second step, the dimensions of input data are reduced by minimizing the representation error through optimized weights computed in the first step. In the first step, the weights are optimized with fixed data. On the other hand, the lower-dimensional points are optimized with fixed weights in the second step. Unlike the PCA, the LLE is a powerful and useful approach to exploit underlying data structures.

**Independent Component Analysis.** The Independent Component Analysis (ICA) technique is used to form data representations through weighted sums of independent non-Gaussian components [35]. The independent Gaussian components are found by maximizing the statistical independence of estimated components. There are two broadest definitions of independence for



ICA, i.e., minimization of mutual information and maximization of non-Gaussianity. In the minimization of mutual information type, maximum entropy is used to measure the components. On the other hand, the maximization of non-Gaussianity type uses negentropy to achieve the target. The ICA algorithms use techniques like centering, whitening, and dimensionality reduction, as preprocessing steps to reduce the complexity of a problem. There are many practical applications of the ICA algorithms, e.g., blind signal separation, face recognition, prediction of stock market prices, and analysis of EEG data.

**Unsupervised Dictionary Learning.** The Unsupervised Dictionary Learning (UDL) algorithms do not utilize labels and rely on the exploitation of structures of input multimedia data. Sparse coding is a well-known example of the UDL algorithms [112]. A survey on UDL algorithms was presented in [104]. In this survey, different UDL algorithms are explained, analyzed and evaluated based on various performance metrics. The UDL algorithms are used for different purposes, e.g., signal reconstruction and classification [108]. A major difference between SDL and UDL algorithms is overcompleteness, i.e., the learned dictionaries in the SDL category have more elements than the signal dimensions [56]. Stochastic gradient descent algorithms are well-suited to UDL category if their learning rate is well tuned [55].

**5.1.2 Computing Platforms for Representation Learning.** Big multimedia data are streaming from different sources over the Internet, such as WMSNs, ad hoc networks, and IoMT. There are many companies, such as IBM, Google, Microsoft and Amazon, that help their clients to process big multimedia and non-multimedia data using an Application Programming Interface (API). These APIs are based on various representation learning algorithms and help developers to create and develop applications in an easy way. These APIs abstract complexities involved in implementing representation learning models and allow developers to focus on manipulating data and designing, experimenting and delivering of results. These APIs also help developers to integrate various representation learning algorithms for real-time smart city applications without worrying about scaling and implementation issues. In this section, we discuss five different computing platforms from well-known global organizations. These platforms make use of various representation learning algorithms and also help programmers to develop new ones. Examples include IBM Watson API, Microsoft Azure Machine Learning API, Google Prediction API, Amazon Machine Learning API, and BigML API. A comparison between these computing platforms is summarized in Table 2.

**IBM Watson API.** The IBM Watson API offers a cognitive service to simplify the process of data preparation and analysis. It also offers various visual tools to display analysis results. It is a public API and available for public use through IBM's Bluemix cloud service platform [38]. The developers can use these APIs to develop applications, services and products with cognitive skills for various smart city applications. There are more than 25 APIs powered by 50 technologies under the IBM Watson platform. Some desirable features offered by the IBM Watson API are listed below.

- Machine translation service to interpret text in different language pairs
- Finding popularity of a specific word or phrase with predetermined audience using message resonance
- Providing answers to questions triggered by primary document resources
- Predicting social characteristics of someone from a given text

By using these features, audio data collected from various sensors or extracted from videos can be converted into simple text and analyzed for various purposes.

**Microsoft Azure API.** The Microsoft Azure API helps data scientists to quickly analyze their data and skip the time-consuming process of developing complex representation learning models. By

Features	IBM Watson [38]	Microsoft Azure [62]	Google Prediction [25]	Amazon [6]	BigML [13]
Support for different data sources	Advanced	Advanced	No support	No support	Advanced
Data transformation and cleaning capabilities	Advanced	Advanced	Basic	Basic	Advanced
Support for different machine learning algorithms	Advanced	Advanced	Advanced	Basic	Advanced
Support for creation of algorithms	Advanced	Advanced	No support	No support	Advanced
Support for algorithm performance evaluation	Advanced	Advanced	No support	Advanced	Advanced
Support for computed result evaluation	Advanced	Advanced	Advanced	No support	Advanced
Support for parameter tuning	Advanced	Advanced	Basic	Basic	Advanced

Table 2. Supported features of different computing platforms

using this API, the data scientists can analyze multimedia data collected from smart city applications and predict various abnormal events in different scenarios using predictive models. This API offers representation learning abilities and is available in different Microsoft products, such as Xbox and Bing [62]. There are many features offered by this API. Some of them are listed below.

- It allows data scientists to create a customized and configurable model to train and predict tasks based on their own R language code.
- It allows data scientists to include Python language scripts using different Python libraries, e.g., SciPy, Pandas and NumPy. Other popular Python tools, e.g., iPython Notebook and Visual Studio-based tools, are also supported by this API.
- It allows data scientists to train petabyte of data using PCA or support vector machines to predict malicious behaviors in video data captured through surveillance cameras.
- It supports other data processing platforms, such as Hadoop and Spark, to process huge volumes of big multimedia and non-multimedia data.

By using these features, the multimedia data collected from various smart city applications can be processed quickly to make real-time decisions.

**Google Prediction API.** The Google Prediction API helps its users to process big multimedia data and generate results for real-time applications. This API is a cloud computing based facility and can be utilized in many applications, e.g., behavior analysis, spam detection, classification of data, and prediction of certain events [25]. This API uses classifiers to predict results and allows its users to crunch big multimedia data with little programming and machine learning knowledge. The multimedia data need to be uploaded to the Google cloud storage. This API reads the uploaded data using the BigQuery. This API has been used in many smart applications. Two popular examples are listed below.

- Ford is a company that designs and sells vehicles. Ford is using this API in its research laboratories to facilitate customers in improving their driving skills. This API helps drivers to map their daily-routine routes. Once the route and location information are saved, the API helps vehicles to determine automatically where the driver wants to drive based on previous driving knowledge at a specific time on a particular day of a week. This API also monitors the behaviors of drivers during driving using various sensors.
- Pondera Solutions is a USA-based company to detect frauds. This company uses advanced machine learning algorithms on cloud platforms to address various government issues, such as fraud, abuse and waste in different public sectors. For machine learning computations, the Pondera Solutions uses the Google Prediction API.

If properly utilized, this API can be used to analyze multimedia data collected from various technologies to monitor citizens' behaviors in different situations and can be helpful in various smart city applications, e.g., transportation management, surveillance and healthcare.

**Amazon API.** The Amazon API is another popular facility to simplify the process of predictions. It abstracts the processes of model building, data filtering, and statistical analysis [6]. The Amazon machine learning model deals with prediction problems only. The speed of processing big multimedia data depends on the ratio of prediction error. The Amazon API offers visualization tools to its users to get a better insight of the processed data. This API has certain restrictions in terms of user interface and representation learning algorithms. However, it is still user-friendly and easy to use. This API can be utilized in a wide variety of applications. Some of them are summarized below.

- It helps data analysts in finding genre of songs by analyzing features of signal levels.
- To recognize human actions and activities from multimedia and non-multimedia data. In the case of non-multimedia data, geo-location information can be utilized to approximate users' activities.
- It helps to analyze payments in the first week in any shopping center to predict whether the users will pay using cash or by card.
- It helps to detect fake users and identities by analyzing web activities.

From above use cases, it is clear that this API can be used to analyze data and predict events. The accuracy of prediction depends on the amount of provided data. Hence, this API can be used in many smart city applications, e.g., surveillance, transportation management, and healthcare.

**BigML.** The BigML is a user-friendly API for representation learning developers and is based on decision trees. It makes predictive analytics tasks easy and understandable for its users. It offers features like understanding business requirements and analysis reports. There are three main modes in this API, i.e., RESTful API, Web Interface, and Command Line Interface, among which the Web Interface is the most popular one [13]. This API offers many useful features to data analysts for making smart decisions in real-time. Some well-known examples are listed below.

- It helps in dealing complex forms of multimedia data collected from various resources, finding relationships between different attributes of input data, and predicting about features of similar objects so that a user can efficiently utilize specific objects rather than using all objects.
- It helps its users to create predictive models based on some past examples and the same input dataset. The predictive models can be designed in a way that various data instances can be processed in batches to save computational time and perform accurate analysis.
- It offers a remote access facility so that once the predictive model is developed, it can be accessed remotely via a Command Line Interface.

From above examples, it can clearly be seen that this API offers many useful features and can support various forms of data. Due to its versatility in terms of data support, it can be used to process multimedia and non-multimedia data collected from various resources in the smart cities. However, the accuracy of predictions depends on the amount of provided data. Once the predictive model is developed, it can be accessed remotely via a Command Line Interface to perform batch processing tasks for various organizations and agencies dealing with different smart city applications.

## 5.2 Data Processing Using Deep Learning

Although multilayer/deep learning is considered as a part of the representation learning paradigm, we discuss deep learning algorithms and computational platforms separately in this section due to their increasing importance in academia and research domains and smart city applications.

**5.2.1 Deep Learning Algorithms.** Deep learning algorithms are unique and suitable for a diverse range of applications. They are based on multiple layers of non-linear processing units. The layers usually include artificial neural networks. The non-linear units work in a cascading style and are used to extract features and perform transformations. The non-linear processing units can work in a supervised or an unsupervised manner. The deep learning algorithms learn from multiple levels of representations corresponding to different levels of abstraction.

In practice, there are various machine learning models in the deep learning domain. Examples include deep convolutional networks [88], deep belief networks [53], restricted Boltzmann machines [66], contractive autoencoders [68, 79], and long short term memory networks [32]. These models are briefly discussed in the following subsections.

**Deep Convolutional Network.** The Deep Convolutional Networks (DCNs), also known as a feed-forward artificial neural networks, are mainly used to analyze visual imagery. They use a variation of multi-level perceptrons to minimize the preprocessing overhead. The DCNs consist of multiple layers including input, output and hidden layers. The hidden layers can be convolutional, pooling or fully connected. In the convolutional type, a convolution operation is applied to input data and results are forwarded to the next layer in sequence. In the pooling type, the output of neuron clusters at one layer can be combined into a single neuron in the next layer in sequence. In the fully connected type, every neuron of one layer is connected to every neuron in the next layer. A network of this type is similar to traditional multi-layer perceptron neural network. The DCNs can be used in many useful applications, such as image recognition, video analysis, natural language processing, drug discovery, and checker games. Due to their applications in the multimedia domain, they can be applied to multimedia data generated by many applications of smart cities.

**Deep Belief Network.** The Deep Belief Networks (DBNs) are generative graphical models and composed of multiple layers with hidden units. The layers can be connected to each other, however, there is no connection between the hidden units in each layer. The DBNs can be viewed as a simple unsupervised network where the hidden units of one layer serve as visible layers for other layers. In this composition, a layer-by-layer unsupervised training procedure becomes very fast and layers can be utilized to detect different features of objects. For classification, the DBNs can be trained with a suitable supervision. There are many applications of DBNs in real-life, such as drug discovery and electroencephalography. Due to the feasibility of DBNs for medical applications, they can be applied to smart healthcare and chemical industries in the smart cities.

**Restricted Boltzman Machine.** The Restricted Boltzman Machines (RBMs) are based on generative stochastic artificial neural networks and learns from probability distribution over the set of inputs. Training of the RBMs can be supervised or unsupervised. The RBMs have many technical applications, such as modeling, learning of features, collaborative filtering, and dimension reduction.

Neurons in the RBMs form bipartite graphs. These graphs consists of visible and hidden nodes. These nodes may or may not have a connection between them. By default, hidden and visible units are based on binary values. A DBN can be constructed by stacking an RBM and a fine-tuned deep network together. Due to their popularity, the RBMs can be utilized to process big multimedia data generated by many smart city applications and learn various features to predict certain events.

**Contractive AutoEncoders.** The Contractive AutoEncoders (CAEs) belong to a family of autoencoders and is designed to make learned representations robust towards small changes around training examples. To achieve this robustness, the CAEs apply different penalties on learned representations. The CAEs are very similar to denoising autoencoders in terms of robustness. The denoising autoencoders achieve the robustness in the reconstruction phase while the CAEs achieve it during the encoding phase. Due to the encoding phase operation, the CAEs perform better than the denoising autoencoders when dealing with classification problems. Another major difference between these two encoders is the way of achieving robustness. In the denoising autoencoders, the robustness is achieved by adding the noise in the input data. On the other hand, the CAEs achieve robustness analytically by applying penalties on learned representations. The CAEs can be used in many applications, such as data clustering, reduction of dimensions, prediction of specific structures, and anomaly detection. When applied to the collected multimedia data, the CAEs can be used to make smart decisions in real-time with small variations in the input data.

**Long Short Term Memory Network.** The Long Short Term Memory (LSTM) network is based on a recurrent neural network architecture. During arbitrary intervals in the training process, the LSTM network remembers values and does not modify those values as the learning process proceeds. It also allows forward and backward connections between neurons. It can easily process and predict time series events apart from variable duration between important time events. Due to its successful applications in text compression and recognition, the LSTM network is adopted by many well-known companies, such as Google, Apple and Microsoft, in their fundamental products. The LSTM network usually consists of multiple LSTM units which can further be distributed into multiple LSTM blocks to facilitate parallel processing. The LSTM block may contain more than one LSTM unit. Each LSTM block has multiple gates to control the flow of information. There are many applications of the LSTM network, e.g., robotics, time-series prediction, speech recognition, music composition, handwriting detection, human action recognition, protein homology detection, and prediction of subcellular localization of proteins. Due to its usage in a wide variety of applications, the LSTM network can be used in many smart city applications, such as smart healthcare, smart surveillance, and chemical industry.

**5.2.2 Computing Platforms for Deep Learning.** In the field of deep learning, many software products are introduced in the market to facilitate deep learning developers. These products offer different facilities and can support multiple baseline platforms, i.e., operating systems. Most of them are open-source and can be used with popular programming languages, e.g., Python, C++, R and Matlab. A comparison between multiple deep learning computing platforms is summarized in Table 3.

### 5.3 Data Processing Using Data Analytics

In the field of data analytics, input datasets are examined to draw conclusion about the information contained in the input datasets [118]. Data analytics techniques are widely used in business and research industries to make informed decisions and verify theories and scientific models, respectively. The data analytics techniques can be classified as quantitative or qualitative [72]. In the quantitative category, numerical data are analyzed based on quantifiable variables. The

Platform	Open Source	Operating System	Written in	Interface
Caffe	Yes	<ul style="list-style-type: none"><li>• Linux</li><li>• Mac</li><li>• Windows</li></ul>	<ul style="list-style-type: none"><li>• C++</li></ul>	<ul style="list-style-type: none"><li>• Python</li><li>• Matlab</li></ul>
Microsoft Cognitive Toolkit	Yes	<ul style="list-style-type: none"><li>• Linux</li><li>• Windows</li></ul>	<ul style="list-style-type: none"><li>• C++</li></ul>	<ul style="list-style-type: none"><li>• Python</li><li>• C++</li><li>• Command Line Interface</li></ul>
TensorFlow	Yes	<ul style="list-style-type: none"><li>• Linux</li><li>• Mac</li><li>• Windows</li></ul>	<ul style="list-style-type: none"><li>• C++</li><li>• Python</li></ul>	<ul style="list-style-type: none"><li>• Python</li><li>• C/C++</li><li>• Java</li><li>• Go</li><li>• R</li></ul>
Theano	Yes	<ul style="list-style-type: none"><li>• Cross-platform</li></ul>	<ul style="list-style-type: none"><li>• Python</li></ul>	<ul style="list-style-type: none"><li>• Python</li></ul>
Torch	Yes	<ul style="list-style-type: none"><li>• Linux</li><li>• Mac</li><li>• Windows</li><li>• Android</li><li>• iOS</li></ul>	<ul style="list-style-type: none"><li>• C</li><li>• Lua</li></ul>	<ul style="list-style-type: none"><li>• Lua</li><li>• C/C++</li><li>• OpenCL</li></ul>
Wolfram Mathematica	No	<ul style="list-style-type: none"><li>• Linux</li><li>• Mac</li><li>• Windows</li><li>• Cloud computing</li></ul>	<ul style="list-style-type: none"><li>• C++</li></ul>	<ul style="list-style-type: none"><li>• Wolfram language</li></ul>

Table 3. Comparison of deep learning computing platforms [106]

qualitative category is more interpretive and focuses on the contents of non-numerical data, e.g., texts, images, audio and videos.

**5.3.1 Data Analytics Algorithms.** Data analytics projects consist of multiple phases, i.e., pre-processing, visualization, correlation, regression, forecasting, classification, and clustering [82]. In each phase, different techniques are applied to input data to extract the required information. From representation learning perspective, last two phases, i.e., classification and clustering, are the most important ones. The representation learning techniques are applied in these phases to produce final outputs. Remaining phases are mostly used to filter input data so that the classification and clustering phases can produce accurate outputs. In this section, we mainly focus on classification and clustering phases and highlight algorithms used in them.

**Classification.** The classification is considered as a supervised learning phase where labeled data are used to classify various data objects. The performance of a classifier is assessed by various indicators that are used in pairs. In smart city applications, the classification of data plays an important role. Multimedia and non-multimedia data collected from different resources are always huge in volumes and the classification can help in reducing the processing overhead. Due to the shortage of computing and storage resources, there must be some techniques to find redundancy in the collected data. The classification techniques can help in finding the redundant data that need to be discarded. In this section, we discuss various classifiers along with their applications. Some well-known classifiers are support vector machine, Bayesian classifier, *k*-nearest neighbor, linear discriminant analysis [41], learning vector quantization [48], and regression trees [15].

**Support Vector Machine** The Support Vector Machine (SVM), also known as a support vector network, is based on supervised learning models and is used to analyze data for regression and classification purposes. During a training process, a set of training examples is provided to an SVM learning model where the training examples are classified to one of the two classes and make the SVM learning model a non-probabilistic binary linear classifier. Along with linear classification, the SVM can also perform a non-linear classification using kernel tricks. The supervised learning is not possible with non-labeled data. Therefore, when input data are not labeled, an unsupervised algorithm is applied to form data groups before the SVM is applied. This algorithm is known as Support Vector Clustering (SVC) [9]. The SVM and SVC are used in many industrial applications where either the data are not labeled properly or some of the data are labeled. These techniques can be applied in many real-life applications, e.g., text and hypertext categorization, classification of images, handwriting detection, and biomedical applications.

**Bayesian Classifier** The Bayesian classifier is a simple probabilistic classifier based on a well-known Bayes' theorem with naive assumptions to classify data using various features. The Bayesian classifier is highly scalable and requires linear parameters in learning problems. The Bayesian model is used to assign class labels to various data features. The assigned labels are drawn from a set of finite values. Unlike other classification algorithms, the Bayesian classifier involves multiple algorithms with common principles. Despite of oversimplified assumptions, the Bayesian classifier has successfully been used to solve complex real-world problems. There are many applications of this classifier, e.g., text retrieval, text categorization, activity recognition, image and video classification, and medical diagnosis.

**$k$ -Nearest Neighbors Classifier** The  $k$ -Nearest Neighbors ( $k$ -NN) is a non-parametric method and is used in regression and classification problems. In both cases, the input consists of  $k$ -closest training examples in the feature space. However, the output depends on whether the  $k$ -NN is used for regression or classification. The  $k$ -NN is an instance-based learning technique where the classification function is approximated locally and the computations are deferred until the classification is done. For both regression and classification, different weights are assigned to neighbors based on their contributions so that the nearest neighbors should contribute more on average as compared to the others. The neighbors are picked from well-known classes and can be considered as training samples. In the case of big multimedia data, the data may be redundant and may need to be transformed into a reduced form to lower the computational overhead. This transformation can be performed using various features. This procedure is known as feature extraction. The feature extraction can efficiently be performed by the  $k$ -NN algorithm. A well-known example of the  $k$ -NN is face recognition where different facial features are used to identify facial images. There are many practical applications of the  $k$ -NN, e.g., text mining, weather forecasting, financial modeling, stock exchange, and healthcare.

**Linear Discriminant Analysis** The Linear Discriminant Analysis (LDA) is a machine learning technique used for classification of objects and events. The successful classification outcomes can later be used as linear classifiers for further classifications. Default implementation of an LDA technique requires all data samples be known in advance. However, there may be situations where the input data are provided as streams. In this situation, it depends on the feature extraction capability of the LDA technique to observe data streams without running actual classification algorithm. There are many practical applications of the LDA technique, e.g., bankruptcy prediction, face recognition, marketing and finance, earth science, and biomedical studies.

**Learning Vector Quantization** The Learning Vector Quantization (LVQ) is a prototype-based supervised classification algorithm. Sometimes, it is considered as a special case of an artificial neural network. Due to its prototype-based nature, it is easy for experts to use in various application domains and can be applied to multi-class classifications. Initially, the choice of an appropriate value



for training and classification purposes was a major issue. However, in recent times, many techniques have been developed to compute a parameterized distance measure during a training process [47]. The LVQ can be used in many practical applications, e.g., classification of text documents, speech recognition, robotics, automated synthesis of digital systems, and data compression.

**Regression Trees** The regression trees are based on decision trees that take continuous values in the targeted variables. The regression trees-based learning is one of the predictive modeling approaches used in the representation learning domain to classify various objects. In the regression trees, the main goal is to create a model to predict values of a targeted variable based on multiple input variables. In the decision trees, each node is labeled with input and selected features belong to finite discrete domains. In the regression trees, the learning process can be performed by splitting trees recursively into multiple subsets based on an attribute value test. The recursion stops when the subset on a node has the same values of a targeted variable or further splitting adds on unique values to predict. In the data mining domain, the regression trees are used for providing description, categorization and generalization of given datasets. The decision and regression trees have been applied in many real-life applications, e.g., agriculture, astronomy, control systems, financial analysis, biomedical engineering, object recognition, power stability predictions in power systems, and remote sensing.

**Clustering.** The clustering is a method of identifying similar groups of data in a dataset. Entities within one data group share similar properties. The clustering is an unsupervised representation learning approach. However, it can be used to improve the performance of supervised representation learning algorithms by dividing datasets into clusters with proper labels as independent variables [42]. The clustering domain can broadly be classified into two sub-domains, i.e., soft and hard clustering. In the soft clustering, instead of assigning each data point to a cluster, a probability value is assigned to data points to indicate their possible relationships with one of the defined data clusters. In the hard clustering, instead of assigning a probabilistic value, the data points are assigned to one of the defined data clusters.

The task of clustering is subjective in nature and there are many techniques to achieve this task. Each technique follows a different set of rules to define similarity between data points. In literature, there exist many clustering algorithms. However, the most popular ones are connectivity models [44], centroid models [26], distribution models [111], and density models [49]. Each of these models follow different strategies to form data clusters. The connectivity models are based on the fact that data points in close proximity share similar attributes as compared to those that are lying further away. The hierarchical clustering algorithm and its variant belong to this category. The centroid models are based on iterative algorithms and the similarity of data points is found by closeness of data points to the center of a data cluster. The  $k$ -means clustering is a popular algorithm in this category. The distribution models are based on the fact that how probable all data points within the same cluster belong to the same distribution. Popular expectation-maximization algorithm belongs to this category. In the density models, data space is searched to estimate varied densities of data points, isolate multiple density regions, and assign data points to isolated regions within the same clusters. OPTICS and DBSCAN are popular examples in this category.

The clustering has a number of practical applications and can be applied to various scientific and technical domains. Some of the most popular applications of clustering are recommendation engines in websites, market segmentation, social network analysis, grouping of search results, medical imaging, image segmentation, and anomaly detection.

#### 5.4 Research Challenges in Data Processing

In Subsections 5.1,5.2,5.3, we have discussed different machine learning techniques to process big multimedia and non-multimedia data generated by various data sources in an IoT environment. However, each technique has its own limitations. In this section, we discuss various issues that need to be considered when using these techniques to process multimedia data produced by smart city applications.

**5.4.1 Data Processing Issues.** Multimedia data generated by smart city applications are always huge in volumes and heterogeneous in nature. Due to the heterogeneous nature and large size, it becomes critical for data processing platforms to process the generated data in real-time. For example, during some important public events or ceremonies in a smart city, the transportation management system needs to process huge volumes of data to efficiently manage traffic at different places. Smart cameras installed at major routes produce big multimedia data. Multimedia sensor nodes deployed at different streets report vehicles and people movements. To estimate total number of people visiting a smart city for a particular event, social sites can also be monitored. To track movements of pedestrians, wireless and mobile networks can also be utilized. If data collected from all these resources are properly handled and processed, the smart transportation management systems can efficiently manage the traffic loads in the smart cities, the smart surveillance systems can ensure proper security measures, and the smart healthcare systems can alert medical staff for emergency situations.

**5.4.2 Learning Issues.** In the case of supervised learning techniques in the representation learning, deep learning, and data analytics domains, the availability of labeled data is a challenging task. Multimedia and non-multimedia data collected from different resources in the smart cities are always unlabeled and it costs a lot to label such big and heterogeneous data. Furthermore, it is important to select right data and performance metrics for training. A wrong decision can lead to a performance degradation. In these situations, the only implementable solution is to use semi-supervised learning techniques that can take benefits from labeled and unlabeled data by utilizing features of supervised and unsupervised learning techniques. For example, in public events mentioned in above subsection, the data are collected from heterogeneous data sources and are a combination of both labeled and unlabeled data. The smart transportation management system may find hidden factors from a combination of labeled and unlabeled data by applying semi-supervised learning techniques. However, the main challenge is to select right learning techniques to perform semi-supervised learning and computing platforms to process data in real-time. Similar cases may be found in other smart city applications, e.g., smart surveillance and smart healthcare systems.

**5.4.3 Dependency Issues.** It is very important for smart city applications to determine data dependency before processing the collected data. Some data sections may be dependent on critical factors, such as time and location. Ignoring such factors may lead to bad decisions in critical situations like natural disasters and medical emergencies. For example, in smart medical emergency systems, two factors, i.e., time and location, play important roles. Data processing platforms need to be updated from time-to-time by data collecting sources to keep medical staff well-aware of medical situations. Moreover, the locations of patients need to be reported so that an initial medical treatment can be suggested to the attendants of patients or a person in a nearby location of patients. In the case of an incident where an attendant of a patient does not have sufficient medical knowledge, various symptoms on a patient's body need to be reported to make an exact approximation. When data processing platforms are able to identify such dependencies, emergency medical staff can quickly and accurately respond to the situation.

**5.4.4 Deployment Issues.** For data processing platforms, two most important factors are quality and interpretation of meaningful information. These factors are difficult to achieve when collected data are huge and heterogeneous. To address the data quality, an information management system needs to be deployed before data processing systems start processing the collected data. Due to limited processing capabilities, data collection technologies cannot guarantee the quality of collected multimedia data. Therefore, there is a need to maintain the quality of collected multimedia data in the smart cities. The interpretation of meaningful information obtained from collected multimedia data depends on data representation techniques. If data representation techniques can efficiently summarize and present the processed data, it becomes easy for decision-making organizations to understand the interpreted results and take proper actions. If the processed data are represented in a hierarchical format where data with similar attributes are presented in one group, it becomes easy to analyze and understand. For example, in the smart transportation management system, if the processed data are interpreted in a meaningless way, it can affect decisions of transportation management authorities and can result in an unsatisfactory performance causing traffic management problems.

## 6 CONCLUSION

The main goal of a smart city is to improve the lifestyle of its inhabitants by providing them a sustainable environment at low costs. Realizing the concept of a smart city deployment requires an understanding of various components, e.g., data collection and processing. In this survey, we have discussed the concept of smart cities from the perspective of multimedia data and have discussed various data collection technologies, data processing techniques and platforms. Data collection technologies are dependent on various industry standards to support the connectivity of heterogeneous devices, systems and applications. Various data processing techniques and platforms are implemented to process huge multimedia data, reduce volumes of redundant data, and extract meaningful information. Throughout this survey, we have highlighted benefits and limitations of various data collection and processing techniques, presented in the literature. Although there are many valuable surveys on the concept of smart cities, the use of representation learning, deep learning and data analytics techniques and platforms to process multimedia data in the smart cities has not been highlighted yet. The research challenges point out that there exist many open research areas in the smart cities and need to be focused and solved by efficiently utilizing the representation learning, deep learning, and data analytics techniques and platforms to process and manage the generated multimedia data.

## REFERENCES

- [1] Ian F Akyildiz, Tommaso Melodia, and Kaushik R Chowdhury. 2007. A survey on wireless multimedia sensor networks. *Computer networks* 51, 4 (2007), 921–960.
- [2] Ian F Akyildiz, Tommaso Melodia, and Kaushik R Chowdhury. 2008. Wireless multimedia sensor networks: Applications and testbeds. *Proc. IEEE* 96, 10 (2008), 1588–1605.
- [3] Fadi Al-Turjman, Ayman Radwan, Shahid Mumtaz, and Jonathan Rodriguez. 2017. Mobile Traffic Modelling for Wireless Multimedia Sensor Networks in IoT. *Computer Communications* (2017).
- [4] Islam T Almkaw, Manel Guerrero Zapata, Jamal N Al-Karaki, and Julian Morillo-Pozo. 2010. Wireless multimedia sensor networks: current trends and future directions. *Sensors* 10, 7 (2010), 6662–6717.
- [5] Sheeraz A Alvi, Bilal Afzal, Ghalib A Shah, Luigi Atzori, and Waqar Mahmood. 2015. Internet of multimedia things: Vision and challenges. *Ad Hoc Networks* 33 (2015), 87–111.
- [6] Amazon. 2015. Amazon API. <https://aws.amazon.com/documentation/machine-learning/>
- [7] Fabrice Axisa, Pierre Michael Schmitt, Claudine Gehin, Georges Delhomme, Eric McAdams, and André Dittmar. 2005. Flexible technologies and smart clothing for citizen medicine, home healthcare, and disease prevention. *IEEE Transactions on information technology in biomedicine* 9, 3 (2005), 325–336.

- [8] C Badii, P Bellini, D Cenni, A Difino, P Nesi, and M Paolucci. 2017. Analysis and assessment of a knowledge based smart city architecture providing service APIs. *Future Generation Computer Systems* 75 (2017), 14–29.
- [9] Asa Ben-Hur, David Horn, Hava T Siegelmann, and Vladimir Vapnik. 2001. Support vector clustering. *Journal of machine learning research* 2, Dec (2001), 125–137.
- [10] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence* 35, 8 (2013), 1798–1828.
- [11] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. 2016. Fully-convolutional siamese networks for object tracking. In *European conference on computer vision*. Springer, 850–865.
- [12] Simon Elias Bibri and John Krogstie. 2017. Smart sustainable cities of the future: An extensive interdisciplinary literature review. *Sustainable Cities and Society* (2017).
- [13] BigML. 2011. BigML API. <https://bigml.com/api>
- [14] Igor Bisio, Chiara Garibotto, Fabio Lavagetto, and Andrea Sciarrone. 2017. Performance Evaluation of Application Layer Joint Coding Solutions for Video Transmissions between Mobile Devices over the Internet of Things. *Computer Communications* (2017).
- [15] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. 1984. *Classification and regression trees*. CRC press.
- [16] Manisha Chahal, Sandeep Harit, Krishn K Mishra, Arun Kumar Sangaiah, and Zhigao Zheng. 2017. A Survey on software-defined networking in vehicular ad hoc networks: Challenges, applications and use cases. *Sustainable Cities and Society* (2017).
- [17] Adam Coates, Andrew Ng, and Honglak Lee. 2011. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. 215–223.
- [18] Adam Coates and Andrew Y Ng. 2012. Learning feature representations with k-means. In *Neural networks: Tricks of the trade*. Springer, 561–580.
- [19] DELL. 2016. Saensuk Pilots Smart City in Collaboration with DELL and Intel. <https://www.dell.com/learn/us/en/uscorp1/press-releases/2016-07-26-saensuk-smart-city-pilots-first-healthcare-iot-project-with-dell-intel>
- [20] Der-Jiunn Deng. 2011. PSSB: Priority enforced slow-start backoff algorithm for multimedia transmission in wireless ad-hoc networks. *Journal of Network and Computer Applications* 34, 5 (2011), 1468–1473.
- [21] Manuel Díaz, Cristian Martín, and Bartolomé Rubio. 2016. State-of-the-art, challenges, and open issues in the integration of Internet of things and cloud computing. *Journal of Network and Computer Applications* 67 (2016), 99–117.
- [22] Soufiene Djahel, Ronan Doolan, Gabriel-Miro Muntean, and John Murphy. 2015. A communications-oriented perspective on traffic management systems for smart cities: Challenges and innovative approaches. *IEEE Communications Surveys & Tutorials* 17, 1 (2015), 125–151.
- [23] dnp systems. 2017. Physical Security and Access Control Systems. <http://www.dnpsys.com/solution.php>
- [24] Santiago González, Wilder Castellanos, Paola Guzmán, Pau Arce, and Juan Carlos Guerri. 2016. Simulation and experimental testbed for adaptive video streaming in ad hoc networks. *Ad Hoc Networks* 52 (2016), 89–105.
- [25] Google. 2013. Google Prediction API. <https://cloud.google.com/prediction/docs/>
- [26] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. 1998. CURE: an efficient clustering algorithm for large databases. In *ACM Sigmod Record*, Vol. 27. ACM, 73–84.
- [27] Hadi Habibzadeh, Zhou Qin, Tolga Soyata, and Burak Kantarci. 2017. Large Scale Distributed Dedicated-and Non-Dedicated Smart City Sensing Systems. *IEEE Sensors Journal* (2017).
- [28] John A Hartigan and Manchek A Wong. 1979. Algorithm AS 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 28, 1 (1979), 100–108.
- [29] Ibrahim Abaker Targio Hashem, Victor Chang, Nor Badrul Anuar, Kayode Adewole, Ibrar Yaqoob, Abdullah Gani, Ejaz Ahmed, and Haruna Chiroma. 2016. The role of big data in smart city. *International Journal of Information Management* 36, 5 (2016), 748–758.
- [30] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2009. Overview of supervised learning. In *The elements of statistical learning*. Springer, 9–41.
- [31] Geoffrey E Hinton and Terrence Joseph Sejnowski. 1999. *Unsupervised learning: foundations of neural computation*. MIT press.
- [32] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
- [33] Jun Huang, Qiang Duan, Yanxiao Zhao, Zhong Zheng, and Wei Wang. 2017. Multicast Routing for Multimedia Communications in the Internet of Things. *IEEE Internet of Things Journal* 4, 1 (2017), 215–224.
- [34] Don R Hush and Bill G Horne. 1993. Progress in supervised neural networks. *IEEE signal processing magazine* 10, 1 (1993), 8–39.
- [35] Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. 2004. *Independent component analysis*. Vol. 46. John Wiley & Sons.

- [36] IBM. 2012. IBM's Smarter Cities Challenge: Boston Report. <https://www.smartercitieschallenge.org/assets/cities/boston-united-states/documents/boston-united-states-full-report-2012.pdf>
- [37] IBM. 2012. IBM's Smarter Cities Challenge: Louisville Summary Report. <https://www.prd-ibm-smarter-cities-challenge.s3.amazonaws.com/applications/louisville-united-states-summary-2012.pdf>
- [38] IBM. 2013. IBM Watson API. <https://www.ibm.com/watson/developercloud/doc/index.html>
- [39] IBM. 2014. IBM's Smarter Cities Challenge Report: Suffolk County, New York, United States. [https://www.suffolkcountyny.gov/Portals/0/planning/.../IBM\\_SuffolkCounty\\_Report2014r.pdf](https://www.suffolkcountyny.gov/Portals/0/planning/.../IBM_SuffolkCounty_Report2014r.pdf)
- [40] Md Mofijul Islam, Md Abdur Razzaque, Mohammad Mehedi Hassan, Walaa Nagy, and Biao Song. 2017. Mobile Cloud-Based Big Healthcare Data Processing in Smart Cities. *IEEE Access* (2017).
- [41] Alan Julian Izenman. 2013. Linear discriminant analysis. In *Modern multivariate statistical techniques*. Springer, 237–280.
- [42] Anil K Jain. 2010. Data clustering: 50 years beyond K-means. *Pattern recognition letters* 31, 8 (2010), 651–666.
- [43] Mian Ahmad Jan, Muhammad Usman, Xiangjian He, and Ateeq Ur Rehman. 2018. SAMS: A Seamless and Authorized Multimedia Streaming framework for WMSN-based IoMT. *IEEE Internet of Things Journal* (2018).
- [44] Stephen C Johnson. 1967. Hierarchical clustering schemes. *Psychometrika* 32, 3 (1967), 241–254.
- [45] Josep Miquel Jornet and Ian F Akyildiz. 2012. The internet of multimedia nano-things. *Nano Communication Networks* 3, 4 (2012), 242–251.
- [46] Nesrine Khernane, Jean-François Couchot, and Ahmed Mostefaoui. 2017. Optimal power/rate trade-off for internet of multimedia things lifetime maximization under dynamic links capacity. *Future Generation Computer Systems* (2017).
- [47] Teuvo Kohonen. 1990. Improved versions of learning vector quantization. In *Neural Networks, 1990., 1990 IJCNN International Joint Conference on*. IEEE, 545–550.
- [48] Teuvo Kohonen. 1995. Learning vector quantization. In *Self-organizing maps*. Springer, 175–189.
- [49] Hans-Peter Kriegel, Peer Kröger, Jörg Sander, and Arthur Zimek. 2011. Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 1, 3 (2011), 231–240.
- [50] Neeraj Kumar, Sherali Zeedally, and Joel JPC Rodrigues. 2015. QoS-aware hierarchical web caching scheme for online video streaming applications in internet-based vehicular ad hoc networks. *IEEE Transactions on Industrial Electronics* 62, 12 (2015), 7892–7900.
- [51] Steve LaValle, Eric Lesser, Rebecca Shockley, Michael S Hopkins, and Nina Kruschwitz. 2011. Big data, analytics and the path from insights to value. *MIT sloan management review* 52, 2 (2011), 21.
- [52] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444.
- [53] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y Ng. 2009. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th annual international conference on machine learning*. ACM, 609–616.
- [54] Chiehyeon Lim, Kwang-Jae Kim, and Paul P Maglio. 2018. Smart cities with big data: Reference models, challenges, and considerations. *Cities* (2018).
- [55] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. 2010. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research* 11, Jan (2010), 19–60.
- [56] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman. 2009. Non-local sparse models for image restoration. In *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2272–2279.
- [57] Julien Mairal, Jean Ponce, Guillermo Sapiro, Andrew Zisserman, and Francis R Bach. 2009. Supervised dictionary learning. In *Advances in neural information processing systems*. 1033–1040.
- [58] Muhammad Faran Majeed, Syed Hassan Ahmed, Siraj Muhammad, Houbing Song, and Danda B Rawat. 2017. Multimedia Streaming in Information-Centric Networking: A Survey and Future Perspectives. *Computer Networks* (2017).
- [59] Pavel Masek, Jan Masek, Petr Frantik, Radek Fudjak, Aleksandr Ometov, Jiri Hosek, Sergey Andreev, Petr Mlynek, and Jiri Misurec. 2016. A harmonized perspective on transportation management in smart cities: the novel IoT-driven environment for road traffic modeling. *Sensors* 16, 11 (2016), 1872.
- [60] Yasir Mehmood, Farhan Ahmad, Ibrar Yaqoob, Asma Adnane, Muhammad Imran, and Sghaier Guizani. 2017. Internet-of-Things-Based Smart Cities: Recent Advances and Challenges. *IEEE Communications Magazine* 55, 9 (2017), 16–24.
- [61] Ahmad Mohamad Mezher and Mónica Aguilar Igartua. 2017. Multimedia Multimetric Map-aware Routing protocol to send video-reporting messages over VANETs in smart cities. *IEEE Transactions on Vehicular Technology* (2017).
- [62] Microsoft. 2015. Microsoft Azure API. <https://azure.microsoft.com/en-us/services/cognitive-services/>
- [63] Satyajayant Misra, Martin Reisslein, and Guoliang Xue. 2008. A survey of multimedia streaming in wireless sensor networks. *IEEE communications surveys & tutorials* 10, 4 (2008).
- [64] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. 2012. *Foundations of machine learning*. MIT press.
- [65] M Victoria Moreno, Fernando Terroso-Sáenz, Aurora González-Vidal, Mercedes Valdés-Vela, Antonio F Skarmeta, Miguel A Zamora, and Victor Chang. 2017. Applicability of big data techniques to smart cities deployments. *IEEE*

*Transactions on Industrial Informatics* 13, 2 (2017), 800–809.

- [66] Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*. 807–814.
- [67] S Aasha Nandhini and S Radha. 2017. Efficient compressed sensing-based security approach for video surveillance application in wireless multimedia sensor networks. *Computers & Electrical Engineering* (2017).
- [68] Andrew Ng. 2011. Sparse autoencoder. *CS294A Lecture notes* 72, 2011 (2011), 1–19.
- [69] Zhaolong Ning, Feng Xia, Noor Ullah, Xiangjie Kong, and Xiping Hu. 2017. Vehicular social networks: enabling smart mobility. *IEEE Communications Magazine* 55, 5 (2017), 16–55.
- [70] OECD. 2015. The OECD Model Survey on ICT Access and Usage by Households and Individuals. <https://www.oecd.org/sti/ieconomy/>
- [71] Ahmed M Shahat Osman, Ahmed Elragal, and Birgitta Bergvall-Kåreborn. 2017. Big Data Analytics and Smart Cities: A Loose or Tight Couple?. In *10th International Conference on Connected Smart Cities 2017 (CSC 2017), Lisbon, 20-22 July 2017*. IADIS, 157–168.
- [72] Michael Quinn Patton. 2005. *Qualitative research*. Wiley Online Library.
- [73] João Paulo Just Peixoto and Daniel G Costa. 2017. Wireless visual sensor networks for smart city applications: A relevance-based approach for multiple sinks mobility. *Future Generation Computer Systems* (2017).
- [74] Javier Poncela, Panagiotis Vlachas, Raffaele Gialfreda, Suparna De, Massimo Vecchio, Septimiu Nechifor, Raquel Barco, Mari Carmen Aguayo-Torres, Vera Stavroulaki, Klaus Moessner, et al. 2014. Smart cities via data aggregation. *Wireless personal communications* 76, 2 (2014), 149–168.
- [75] Behrouz Pourghhebleh and Nima Jafari Navimipour. 2017. Data aggregation mechanisms in the Internet of things: a systematic review of the literature and recommendations for future research. *Journal of Network and Computer Applications* (2017).
- [76] Dan Puiu, Payam Barnaghi, Ralf Toenjes, Daniel Kümper, Muhammad Intizar Ali, Alessandra Mileo, Josiane Xavier Parreira, Marten Fischer, Sefki Kolozali, Nazli Farajidavar, et al. 2016. Citypulse: Large scale data analytics framework for smart cities. *IEEE Access* 4 (2016), 1086–1108.
- [77] Wullianallur Raghupathi and Viju Raghupathi. 2014. Big data analytics in healthcare: promise and potential. *Health information science and systems* 2, 1 (2014), 3.
- [78] Cristiano Rezende, Azzedine Boukerche, Heitor S Ramos, and Antonio AF Loureiro. 2015. A reactive and scalable unicast solution for video streaming over VANETs. *IEEE Trans. Comput.* 64, 3 (2015), 614–626.
- [79] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. 2011. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th international conference on machine learning (ICML-11)*. 833–840.
- [80] Denis Rosário, Zhongliang Zhao, Aldri Santos, Torsten Braun, and Eduardo Cerqueira. 2014. A beaconless opportunistic routing based on a cross-layer approach for efficient video dissemination in mobile multimedia IoT applications. *Computer Communications* 45 (2014), 21–31.
- [81] Sam T Roweis and Lawrence K Saul. 2000. Nonlinear dimensionality reduction by locally linear embedding. *science* 290, 5500 (2000), 2323–2326.
- [82] Thomas A Runkler. 2012. *Data Analytics*. Springer.
- [83] Omar Said, Yasser Albagory, Mostafa Nofal, and Fahad Al Raddady. 2017. IoT-RTP and IoT-RTCP: Adaptive Protocols for Multimedia Transmission over Internet of Things Environments. *IEEE Access* 5 (2017), 16757–16773.
- [84] Johannes M Schleicher, Michael Vögler, Schahram Dustdar, and Christian Inzinger. 2016. Application Architecture for the Internet of Cities: Blueprints for Future Smart City Applications. *IEEE Internet Computing* 20, 6 (2016), 68–75.
- [85] Jürgen Schmidhuber. 2015. Deep learning in neural networks: An overview. *Neural networks* 61 (2015), 85–117.
- [86] Hang Shen and Guangwei Bai. 2016. Routing in wireless multimedia sensor networks: A survey and challenges ahead. *Journal of Network and Computer Applications* 71 (2016), 30–49.
- [87] Bhagya Nathali Silva, Murad Khan, and Kijun Han. 2018. Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities. *Sustainable Cities and Society* 38 (2018), 697–713.
- [88] Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations*.
- [89] Hatem Ben Sta. 2017. Quality and the efficiency of data in “Smart-Cities”. *Future Generation Computer Systems* 74 (2017), 409–416.
- [90] Paula Ta-Shma, Adnan Akbar, Guy Gerson-Golan, Guy Hadash, Francois Carrez, and Klaus Moessner. 2017. An ingestion and analytics architecture for iot applied to smart city use cases. *IEEE Internet of Things Journal* (2017).
- [91] Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*. Association for Computational Linguistics, 384–394.

- [92] Muhammad Usman, Xiangjian He, Kenneth KM Lam, Min Xu, Jinjun Chen, Syed Mohsin Matloob Bokhari, and Mian Ahmad Jan. 2017. Error Concealment for Cloud-based and Scalable Video Coding of HD Videos. *IEEE Transactions on Cloud Computing* (2017).
- [93] Muhammad Usman, Xiangjian He, Kin-Man Lam, Min Xu, Syed Mohsin Matloob Bokhari, and Jinjun Chen. 2016. Frame interpolation for cloud-based mobile video streaming. *IEEE Transactions on Multimedia* 18, 5 (2016), 831–839.
- [94] Muhammad Usman, Mian Ahmad Jan, and Xiangjian He. 2017. Cryptography-based secure data storage and sharing using HEVC and public clouds. *Information Sciences* 387 (2017), 90–102.
- [95] Muhammad Usman, Mian Ahmad Jan, Xiangjian He, and Muhammad Alam. 2018. Performance evaluation of High Definition video streaming over Mobile Ad Hoc Networks. *Signal Processing* 148 (2018), 303–313.
- [96] Muhammad Usman, Mian Ahmad Jan, Xiangjian He, and Jinjun Chen. 2018. A mobile multimedia data collection scheme for secured wireless multimedia sensor networks. *IEEE Transactions on Network Science and Engineering* (2018).
- [97] Muhammad Usman, Mian Ahmad Jan, Xiangjian He, and Priyadarsi Nanda. 2016. Data sharing in secure multimedia wireless sensor networks. In *2016 IEEE Trustcom/BigDataSE/ISPA*. IEEE, 590–597.
- [98] Muhammad Usman, Mian Ahmad Jan, Xiangjian He, and Priyadarsi Nanda. 2018. QASEC: A secured data communication scheme for mobile Ad-hoc networks. *Future Generation Computer Systems* (2018).
- [99] Muhammad Usman, Ning Yang, Mian Ahmad Jan, Xiangjian He, Min Xu, and Kin-Man Lam. 2018. A joint framework for QoS and QoE for video transmission over wireless multimedia sensor networks. *IEEE Transactions on Mobile Computing* 17, 4 (2018), 746–759.
- [100] Rosamunde Van Brakel and Paul De Hert. 2011. Policing, surveillance and law in a pre-crime society: Understanding the consequences of technology based strategies. *Technology-Led Policing* 20 (2011), 165–92.
- [101] Nguyen-Son Vo, Thanh-Hieu Nguyen, and Huu Khuong Nguyen. 2017. Joint Active Duty Scheduling and Encoding Rate Allocation Optimized Performance of Wireless Multimedia Sensor Networks in Smart Cities. *Mobile Networks and Applications* (2017), 1–11.
- [102] Michael Vögler, Johannes M Schleicher, Christian Inzinger, Schahram Dustdar, and Rajiv Ranjan. 2016. Migrating smart city applications to the cloud. *IEEE Cloud Computing* 3, 2 (2016), 72–79.
- [103] Junyi Wang, Zhihui Liu, Yuxiang Shen, Hongbin Chen, Lin Zheng, Hongbing Qiu, and Shi Shu. 2016. A distributed algorithm for inter-layer network coding-based multimedia multicast in Internet of Things. *Computers & Electrical Engineering* 52 (2016), 125–137.
- [104] Xiaolong Wang and Abhinav Gupta. 2015. Unsupervised learning of visual representations using videos. In *Proceedings of the IEEE International Conference on Computer Vision*. 2794–2802.
- [105] Wikipedia. 2017. Surveillance Issues in Smart Cities. [https://en.wikipedia.org/wiki/Surveillance\\_issues\\_in\\_smart\\_cities](https://en.wikipedia.org/wiki/Surveillance_issues_in_smart_cities)
- [106] Wikipedia. 2018. Comparison of deep learning software. [https://en.wikipedia.org/wiki/Comparison\\_of\\_deep\\_learning\\_software](https://en.wikipedia.org/wiki/Comparison_of_deep_learning_software)
- [107] Svante Wold, Kim Esbensen, and Paul Geladi. 1987. Principal component analysis. *Chemometrics and intelligent laboratory systems* 2, 1-3 (1987), 37–52.
- [108] John Wright, Allen Y Yang, Arvind Ganesh, S Shankar Sastry, and Yi Ma. 2009. Robust face recognition via sparse representation. *IEEE transactions on pattern analysis and machine intelligence* 31, 2 (2009), 210–227.
- [109] Zhe Xiao, Hock Beng Lim, and Loganathan Ponnambalam. 2017. Participatory Sensing for Smart Cities: A Case Study on Transport Trip Quality Measurement. *IEEE Transactions on Industrial Informatics* 13, 2 (2017), 759–770.
- [110] Changqiao Xu, Wei Quan, Hongke Zhang, and Luigi Alfredo Grieco. 2016. GrIMS: green information-centric multimedia streaming framework in vehicular ad hoc networks. *IEEE Transactions on Circuits and Systems for Video Technology* (2016).
- [111] Xiaowei Xu, Martin Ester, H-P Kriegel, and Jörg Sander. 1998. A distribution-based clustering algorithm for mining in large spatial databases. In *Data Engineering, 1998. Proceedings., 14th International Conference on*. IEEE, 324–331.
- [112] Jianchao Yang, Kai Yu, Yihong Gong, and Thomas Huang. 2009. Linear spatial pyramid matching using sparse coding for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 1794–1801.
- [113] Runan Yao, Wei Wang, Mahdi Farrokh-Baroughi, Honggang Wang, and Yi Qian. 2013. Quality-driven energy-neutralized power and relay selection for smart grid wireless multimedia sensor based IoTs. *IEEE Sensors Journal* 13, 10 (2013), 3637–3644.
- [114] YIN Yuehong, Yan Zeng, Xing Chen, and Yuanjie Fan. 2016. The internet of things in healthcare: An overview. *Journal of Industrial Information Integration* 1 (2016), 3–13.
- [115] Andrea Zanella, Nicola Bui, Angelo Castellani, Lorenzo Vangelista, and Michele Zorzi. 2014. Internet of things for smart cities. *IEEE Internet of Things journal* 1, 1 (2014), 22–32.



- [116] Ke Zhang, Haitao Jia, and Haibin Lv. 2016. Coverage-enhancing approach in multimedia directional sensor networks for smart transportation. *Multimedia Tools and Applications* 75, 24 (2016), 17593–17615.
- [117] Kaile Zhou, Chao Fu, and Shanlin Yang. 2016. Big data driven smart energy management: From big data to big insights. *Renewable and Sustainable Energy Reviews* 56 (2016), 215–225.
- [118] Paul Zikopoulos, Chris Eaton, et al. 2011. *Understanding big data: Analytics for enterprise class hadoop and streaming data*. McGraw-Hill Osborne Media.