

# Context-Awareness for Mobile Sensing: A Survey and Future Directions

Ozgur Yurur, *Member, IEEE*, Chi Harold Liu, *Member, IEEE*, Zhengguo Sheng, *Member, IEEE*, Victor C. M. Leung, *Fellow, IEEE*, Wilfrido Moreno, and Kin K. Leung, *Fellow, IEEE*

**Abstract**—The evolution of smartphones together with increasing computational power have empowered developers to create innovative context-aware applications for recognizing user related social and cognitive activities in any situation and at any location. The existence and awareness of the context provides the capability of being conscious of physical environments or situations around mobile device users. This allows network services to respond proactively and intelligently based on such awareness. The key idea behind context-aware applications is to encourage users to collect, analyze and share local sensory knowledge in the purpose for a large scale community use by creating a smart network. The desired network is capable of making autonomous logical decisions to actuate environmental objects, and also assist individuals. However, many open challenges remain, which are mostly arisen due to the middleware services provided in mobile devices have limited resources in terms of power, memory and bandwidth. Thus, it becomes critically important to study how the drawbacks can be elaborated and resolved, and at the same time better understand the opportunities for the research community to contribute to the context-awareness. To this end, this paper surveys the literature over the period of 1991-2014 from the emerging concepts to applications of context-awareness in mobile platforms by providing up-to-date research and future research directions. Moreover, it points out the challenges faced in this regard and enlighten them by proposing possible solutions.

**Index Terms**—Context-Awareness, Middleware, Mobile Sensing

## I. INTRODUCTION

The continual development of sensor designs and deployment together with ever-increasing computing technologies in mobile device based embedded systems platforms have enabled to pervasively recognize the individual and social context that device users touch with. Hence, the inference of daily occurring human-centric actions, activities and interactions by a set of mobile device based sensors has drawn much interest in the research area of ubiquitous sensing community<sup>1</sup>. It is

This work is financially sponsored by National Natural Science Foundation of China (Grant No. 61300179).

O. Yurur and W. Moreno are with the Department of Electrical Engineering, University of South Florida, Tampa, FL, 33613. E-mail: oyurur@mail.usf.edu; wmoreno@usf.edu

C. H. Liu (Corresponding Author) is with the School of Software at Beijing Institute of Technology, Beijing, 100181, China. E-mail: chiliu@bit.edu.cn

Z. Sheng is with School of Engineering and Informatics, the University of Sussex, UK. E-mail: z.sheng@ieee.org

V. C. M. Leung is with Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC. E-mail: vleung@ece.ubc.ca

K. K. Leung is with the Department of Electrical and Electronic Engineering, Imperial College, London, SW7 2BT, UK. E-mail: kin.leung@imperial.ac.uk

<sup>1</sup>Paradigm of ubiquitous sensing is also described as pervasive computing, mobile computing, context-aware sensing, ambient intelligence or more recently, everywhere.

believed that introducing intelligence and situational awareness into recognition process of human-centric event patterns could give a better understanding of human behaviors, and it also could give a chance for proactively assisting individuals to enhance the quality of lives [1], [2].

Ubiquitous sensing was firstly envisioned by Weiser [3] as in providing the right information to the right person at the right time through an effective kind of technology via physical environment, yet making the relevant computing elements and inter-communication invisible to the user. Then, the term of context-awareness firstly used in [4] where the ability of a mobile user's applications to discover and react to changes in the environment they are situated in. Also, the definition of context or context-awareness were simplified and generalized first by [5], and later by [6] as in any information that can be used to characterize the situation of an entity, where an entity can be a person, place, or physical or computational object. In the latter, the complexity of context-awareness were linked with individual user activities by [7], and also were modeled in [8]. In addition, use of context-awareness within mobile sensing, and within the concept of smart spaces was introduced in [9]–[11] respectively. Earlier attempts of context-aware applications were also presented in [12]. Since then, the envisioned interaction between smart devices and users has become possible today and inevitable for future technologies. Therefore, the ubiquitous sensing has led to increase the demand for novel applications and services to provide any interested context at anytime and from anywhere.

The integration of sensing and advanced computing capability in network enabled mobile devices will produce sensory data and exchange information among local or system-wide resources by feeding the Internet at a social scale [13], [14]. This situation will emerge the concept of the Internet-of-Things (IoT, [15], [16]) to shift into a collection of autonomous, ambient intelligent and self-operated network nodes (e.g., independently acting smartphones) which are well aware of surrounding context, circumstances and environments. With these capabilities, the new network architecture would enhance data credibility, quality, privacy and share-ability by encouraging participation at personal, social and urban scales. It would also lead to discover the knowledge about human lives and behaviors, and environment interactions/social connections by leveraging the deployment capacity of smart things (e.g., smartphones, tablets) in order to collect and analyze the digital traces left by users.

It is well known that recognition of a human behavior highly depends on perception, context, environment and prior

knowledge of most recent event patterns. In other words, the understanding of human activity is based on the discovery of an activity pattern and accurate recognition of the activity itself. Therefore, researchers have focused on implementing computationally pervasive systems in order to create high-level conceptual models to infer activities, and low-level sensory models to extract context from unknown activity patterns. At this point, the creation of a generic model to represent the true nature of human behavior stands as a major challenge. In this aspect, the construction of a framework by distinct middleware technologies have been put forward to provide the required model for recognition of daily occurring human activities via observations acquired by various sensors built-in smartphones. These activities are inferred as outcomes of a wide range of sensory applications utilized in such diverse implementation areas ranging from environmental surveillance, assisting technologies for medical diagnosis/treatments, to the creation of smart spaces for individual behavior modeling. The key challenges that are faced in this concept is to infer relevant activities in such a system that takes raw sensor readings initially and processes them until obtaining a semantic outcome under some constrictions. These constrictions mostly stem from the difficulty of shaping exact topological structure, and also stem from modeling uncertainties in the observed data due to saving the energy wasted during physical sensor operations and process of sensory data.

Today's mobile devices have been becoming increasingly sophisticated, and the latest versions are now equipped with a rich set of powerful small size built-in sensors such as accelerometers, ambient light sensors, GPS<sup>2</sup>, magnetic compass, and Wi-Fi<sup>2</sup>. These sensors can measure various information belonging to physical world surrounding the mobile device; thereby, ubiquitously use of mobile devices in the society creates a new exciting research area for sensory data mining context-aware applications. Specifically, smartphones could provide a large number of applications within the defined research area. Since human beings are involved in a vast variety of activities within very diverse contexts along with the usage of mobile phones are getting more integrated into human lives throughout the day, a specific context, whose relevant data is acquired through built-in sensors can be extracted by a smartphone application. Eventually, a desired information within the context is inferred by successful computing implementations.

Context-aware sensing applications can be classified under two different categories: *personal/human-centric* and *urban including participatory/community/group or opportunistic*<sup>3</sup>. In personal sensing applications, device user is the point of interest. For instance, monitoring and recognition of user related posture and movement patterns for personal fitness log or for health care reasons is an active research topic in this field. On the other hand, participatory sensing relies on multiple deployment of mobile devices to interactively and intentionally share, gather and analyze of each local

knowledge that is not solely based on a human activity, but also based on surrounding environment. Hence, participatory sensing requires the active participation of each user into collecting of sensory data in order to result in a large-scale phenomena, which cannot be easily measured by a single participation. For example, delivering an intelligent traffic congestion report in case where many users provide their speed and location information while being in a transportation is a great example of implementation in participatory sensing. On the other hand, unlike participatory sensing, opportunistic sensing accepts sensory data collection in a fully autonomous way without active user interaction. This type of applications run in background mode without any user intervention in actual sensing, such as continuous location notification or ambient sound recognition. In summary, the generic idea of all possible sensing applications is to orchestrate the increasing capabilities of mobile devices (e.g., computing, communication and networking, and sensing) through a running software on an existing hardware platform at a right time and place in order to enable services to infer meaningful information for the benefit of individual and community use.

Besides the exciting development of context-aware applications, middleware systems/services in smart devices, however, only have very limited resources in terms of power, memory and bandwidth as compared to the capabilities of PCs and servers. Especially, energy efficiency is a major restriction imposed on context-aware application developments since the extraction and inference of user relevant sensory data requires continuous sensor operations. This requirement unfortunately shortens the device battery lifetime due to high energy consumption required by both sensor and processor operations. One solution is to take precautions on sensory operations while putting them into more sleeping mode to reduce power consumption. However, it turns into an accuracy problem that middleware services may produce while providing information to applications. This situation triggers the research topic of finding optimal solutions to balance a trade-off existing between power consumption and sensory data accuracy. Hence, the key goal lies under discovering the best characteristics of the target complex spatial phenomenon being sensed, meeting the demands of application, and satisfying the constraints on sensor usage.

Towards this end, this paper surveys the literature over the period of 1991-2014 from the emerging concepts to applications of context-awareness in mobile platforms by providing up-to-date research and future research directions. There are some successful surveys in the literature that take the phenomena of context-awareness in different perspectives. These works mostly consider the high-level (abstract) contextual information as a basis, and builds up the survey around it by introducing modeling schemes, types of applications, and their relations to different research areas. Accordingly, [17] surveys sensor utilization in mobile sensing, and types of context-aware application based on their sensing scales, such as within individual or large groups. It also gives information about mobile phone sensing architecture, and use of it as an sensing instrumentation. Finally, it discusses challenges faced in system architecture while sensing, learning and distributing

<sup>2</sup>Their utilizations as sensors are described in Section III-C

<sup>3</sup>Opportunistic sensing slightly differs from participatory sensing due to the autonomous sensory data collection.

high-level context. [18] presents a detailed explanation of context, context types along with sensor types. Then, it continues with context modeling. Also, it covers architectural design principles and models to implement context-aware systems, and conceptual framework design for context processing. [19] looks in depth at the types and modeling of context. It also browses applications and architectural models that adapt to changing context. [20] examines a category of applications implemented for crowdsensing by explaining their unique characteristics and challenges faced in design process. [21] discusses requirements set by context modeling and reasoning techniques. Also, it shows a variety of context information types with a comparison analysis, and high-level context abstractions in the existence of uncertainty. [22] highlights context-awareness from an IoT perspective, and presents the background of IoT paradigm, context-aware essentials and their relations to sensor networks. The work also points out the principles of context-aware management design by surveying a broad range of techniques, methods, models, functionalities, and applications related to abstract layer based context-awareness and IoT. [23], [24] survey context-awareness for recommender systems to build such intelligent systems that can better predict and anticipate the needs of users, and act more efficiently in response to their behaviors. The surveys explore contexts, context types and context modeling in the recommender systems, and defines future challenges to be faced in this research area. [25] takes context-awareness in a scope of mobile and wireless networking. It surveys traditional context-aware computing areas, and makes a connection between them with mobile and wireless networking notions. It defines the functionality of context-awareness in this term, and puts them in a precise taxonomy scheme.

This paper differs from other studies since it surveys context-awareness in mobile platforms by pointing out and proposing solutions to the challenges in terms of recognition process of both low and high level context. More specifically, the paper aims at enlightening possible solutions to enhance the existing tradeoffs in mobile sensing, especially between accuracy and power consumption, while context is being inferred under the intrinsic constraints of mobile devices and around the emerging concepts in context-aware middleware framework. In addition, the paper provides an overview of context-awareness in ubiquitous/mobile sensing, and a comprehensive introduction to the definition, representation and inference of context. However, unlike other surveys, this paper does not extensively cover basic definitions and essentials in context-awareness. Also, this paper not only intends to summarize and itemize some important works done under a specific research branch of context-awareness, but also it gives a detailed view and current trends in this branch by evaluating works and investigating further evolvement. In this sense, it categorizes and gives an inside-out look into context-aware applications depending on the interested context under the limitation of mobile sensing, and then identifies opportunities in the covered research areas. Moreover, the paper exposes the key elements that modern context-aware middleware and framework designs must have, and their connections to mobile computing. Apart from clearly pointing out the similarities and differences of all

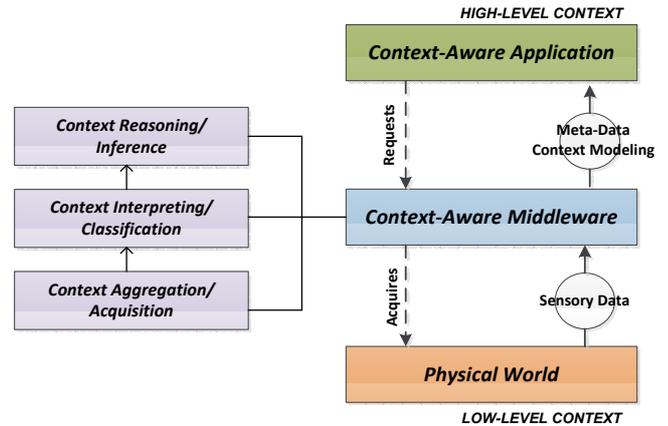


Fig. 1: The architecture of context-awareness system.

research efforts, and identifying the open challenges, it also evaluates future research trends and paves the way for the researchers to see emerging concepts in the defined research area. Finally, the content and the flow of the paper always intend to seek for motivation, identification of drawbacks and road-map to possible solutions of the covered topics, eventually summarize and present findings by projecting the future.

The remainder of this paper is organized as follows. Section II introduces the definition of context, context presentation and stages to context inference problem along with context modeling. This section also exposes the context-aware middleware and framework designs, and their key properties. Section III summarizes significant context-aware application domains, and categorizes them under the interested context. Section VIII puts emphasis on the challenges that are faced in context-awareness and system integration around this research topic, and evaluates possible solutions. Finally, conclusion is given in Section V.

## II. CONTEXT-AWARENESS ESSENTIALS

The context-aware systems aim at using a mobile device (e.g., a hand-held smartphone or attached/wearable device) integrated with smart sensors in order to monitor and measure individual or environmental phenomenas in the purpose for assisting or evaluating human lives to achieve a desirable quality in living standards. Fig. 1 shows the architecture of context-aware system including extraction of low-level context from unknown heterogeneous physical world information acquired by sensors, and then creation of high-level conceptual models based on such context inferences. In the following, we will provide details of key components and modeling processes of the system.

### A. Contextual Information

In the real world, being aware of context and communicating is a key part of human interaction. A context is defined as *a data source which can be sensed and used to characterize the situation of an entity*. In other words, the context describes a physical phenomenon in a real world environment. Hence,

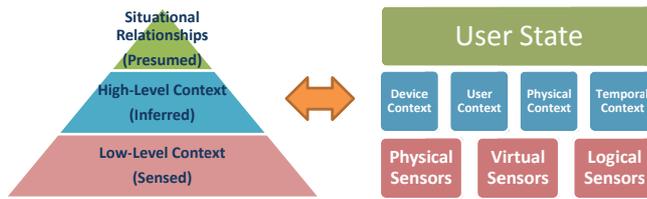


Fig. 2: The hierarchical definition of context representation.

the context can be described in a different way according to how equipped sensors are being used. The context can also be defined as a characterization of a specific entity situation such as user profile, user surrounding, user social interaction or user activity, etc. For instance, we can define the entity by user, and the context by location information. In this sense, context becomes a much richer and more powerful concept, particularly for mobile users by making sensor network services more personalized, and more useful. Therefore, context-awareness refers to the capability of an application being aware of its physical environment or situation, and responding proactively and intelligently based on such awareness [26].

### B. Context Representation

The property of context-awareness can be applied into mobile device based applications and systems in order to reduce human intervention by enabling autonomous proactive assistant services. Many context-aware applications provide this assistance by using logical context alone which is obtained through data mining techniques (e.g., stored information in profiles, databases or social websites). However, with the proliferation of wireless sensor-actuator networks, external physical factors (e.g., temperature, light, location etc.) are added into context-aware systems.

Fig. 2 shows the hierarchical definition of context representation. As can be seen, sensors are accepted as low-level context that is directly referred to a raw data. A sensor in context-aware applications is described not only a physical device, but also a data source that could be useful for context representation. The collected contextual information may range in a wide sense in terms of specification and representation of a phenomenon in real world onto an entity in cyber world. Hence, sensors can be classified as follows:

- *Physical sensors* refer sensors that can capture any physical world belonging data (e.g., GPS: location, accelerometer: activity etc.).
- *Virtual sensors* imply a source from software applications and/or services, and a semantic data obtained through cognitive inference (e.g., location info by manually entered place pinpoint through social network services or computation power of devices etc.).
- *Logical sensors* define combination of physical and virtual sensors with additional information obtained through various sources by user interactions (e.g., databases, log files etc.).

According to levels of abstractions, high-level context then is inferred from low-level contexts. Hence, a definition of

semantic meta-sensor/meta-data/meta-context implies a level of abstraction [27]. Unlike the sensors, the context can be divided into:

- *Device context*: including net connectivity, communication cost and resources, etc.
- *User context*: including profile, geographic position, neighbors, and social situation, etc.
- *Physical context*: including temperature, noise level, light intensity, traffic conditions, etc.
- *Temporal context*: including day, week, month, season, year, etc.

### C. Context Modeling

Being associated with variant context sources, accurate representation of context with a high certainty under different conditions of measuring range and sampling methods is very important to assure the quality of contextual information. In this sense, context modeling is required to reason and interpret dynamic context representations at a high level abstraction in an unobtrusive way. A good context modeling aims at reducing complexity of applications for robustness and usability, and improving their adaptability and maintainability for future development. To be able to do that, it has to consider heterogeneity (i.e., imperfectionist dynamic nature), comparability (i.e., coexistence of similar context from different sources), and mobility (i.e., asynchronous, timeless data capture) of a large variety of context sources at any level of abstraction. It also considers relationships and dependencies among semantic entities such as accuracy in context provisioning versus remaining battery power. In this regard, many context modeling schemes have been proposed [21], [27]–[30]. Important ones are compared in Table I, and listed in the followings:

- Key-value models use a simplest matching algorithm that defines a list of attributes and their content/values describing specific context.
- Mark-up scheme based context models use a hierarchical data structure, mostly formed in XML [31], that consists of markup tags along with their attributes and contents. Therefore, it allows efficient data retrieval. Also, this type of schema can be used among different application domains to store temporary data and in-out data transfer. However, it is not feasible to make context reasoning in presence of multi-markup schemes due to the lack of interoperability among different schemes.
- Graphical models, such as Unified Role Modeling (UML) [32] and Object Role Modeling (ORM) [33], make connections among context attributes and values based on relationships. Especially, this model is widely used within database managements [34] that allow holding a massive amount of data, and perform quick data retrieval. Also, complex context relations can be managed easily through database queries.
- Object-oriented models [35] offer object-oriented techniques to be used in context modeling. Constructed object classes encapsulate or represent different context types, thereby reaching the context or processing its attributes

TABLE I: Important context modeling schemes

Context Modeling	Pros	Cons
Key Value	<ul style="list-style-type: none"> <li>· simple text string matching technique</li> <li>· easily manageable with small data size</li> <li>· mostly application bounded</li> </ul>	<ul style="list-style-type: none"> <li>· not scalable, better in less complexity</li> <li>· not applicable in hierarchical structure</li> <li>· lack of enabling efficient context retrieval and validation</li> </ul>
Mark-up Scheme	<ul style="list-style-type: none"> <li>· efficient data retrieval</li> <li>· applicable in hierarchical structure</li> <li>· provides partial validation</li> </ul>	<ul style="list-style-type: none"> <li>· no design criteria</li> <li>· complex context reasoning in multi-schemes</li> <li>· lack of interoperability with similar models</li> <li>· lack of richness and incompleteness</li> </ul>
Graphical	<ul style="list-style-type: none"> <li>· rich in context collection</li> <li>· allows relationship modeling</li> <li>· better in complex data management</li> </ul>	<ul style="list-style-type: none"> <li>· once designed, difficult to change later</li> <li>· no specific design structure</li> <li>· lack of validation and interoperability with similar models</li> </ul>
Object Oriented	<ul style="list-style-type: none"> <li>· allows more complex relationships and composition</li> <li>· easily designed, and run-time operable</li> <li>· applicable through programming languages</li> </ul>	<ul style="list-style-type: none"> <li>· no specific design structure, nontrivial to update and optimize</li> <li>· difficult data retrieval</li> <li>· mostly application bounded</li> <li>· hidden to other apps due to data encapsulation</li> </ul>
Logic based	<ul style="list-style-type: none"> <li>· designed for checking and resolving context inconsistency</li> <li>· easily designed, and run-time operable</li> <li>· co-operable with other models</li> <li>· high degree of formality</li> </ul>	<ul style="list-style-type: none"> <li>· lack of standardization</li> <li>· provides context reasoning and validation at a certain level</li> <li>· mostly application bounded</li> <li>· lack of richness and incompleteness</li> </ul>
Ontology based	<ul style="list-style-type: none"> <li>· allows knowledge share, integration and reuse</li> <li>· provides well defined, rich, quality and re-extendable abstract model and explicit relations</li> <li>· provides unique identification, redundancy, uncertainty handling and partial validation</li> </ul>	<ul style="list-style-type: none"> <li>· complex and computational expensive data retrieval</li> <li>· lack of handling heterogeneity, ambiguous and quality related issues</li> </ul>

are regulated with designed object-oriented class hierarchies and relationships. This model also provides reusability, inheritance, and polymorphism features into context or inter-context relationships. However, the model is suitable to be used for a dedicated application that employs its own context reasoning structure.

- Logic based models include formality based on facts, expressions and rules in order to set constraints, limitations, policies or preferences while defining context reasoning. It is powerful to manage richness in context definitions by allowing to add, remove, or update new set of rules. Therefore, it could cooperate with other context modeling techniques to enhance context reasoning efficiency.
- Ontology based modeling [36]–[38] uses semantic technologies to represent context related attributes and relationships. It is very widely used and promising instrument thanks to its highly formal expressiveness and conceptualization. Therefore, there are many development tools and engines, such as Resource Description Language (RDF) [39] and Web Ontology Language (OWL) [40], [41], available to apply ontology reasoning techniques. This model aims at providing simple, flexible, extensible, generic and explicitly well defined design objectives. However, with growing data size, context reasoning could be computationally expensive.

All context modeling approaches ultimately intends to provide solutions in context reasoning by seeking for capturing a variety of context types along with their relationships, dependencies, timeliness, and quality of content. They also intends to support accurate reasoning, and clear uncertainty on higher-level context abstractions. Therefore, there might not be a single context modeling technique to be used in a standalone fashion.

**OSI Reference Model**



**Context-Aware Middleware**



Fig. 3: Comparison of ISO/OSI reference model and context-aware middleware.

*D. Context-Aware Middleware*

The growing deployment of sensor technologies in smart devices and innumerable software applications utilizing sensors to sense the surrounding physical environment in order to offer a wide range of user-specific services have led the creation of a *layered* system architecture (i.e., context-aware middleware). In this way, the desired architecture can response effectively for optimal sensor utilization, large sensory data acquisitions as well as meeting ever-increasing application requirements, leveraging the pervasive context-processing software libraries, and considering mobile device resource constraints. Due to the ubiquity of these computing devices in a dynamic environment where the sensor network topologies actively change, it yields applications to behave opportunistically and adaptively with

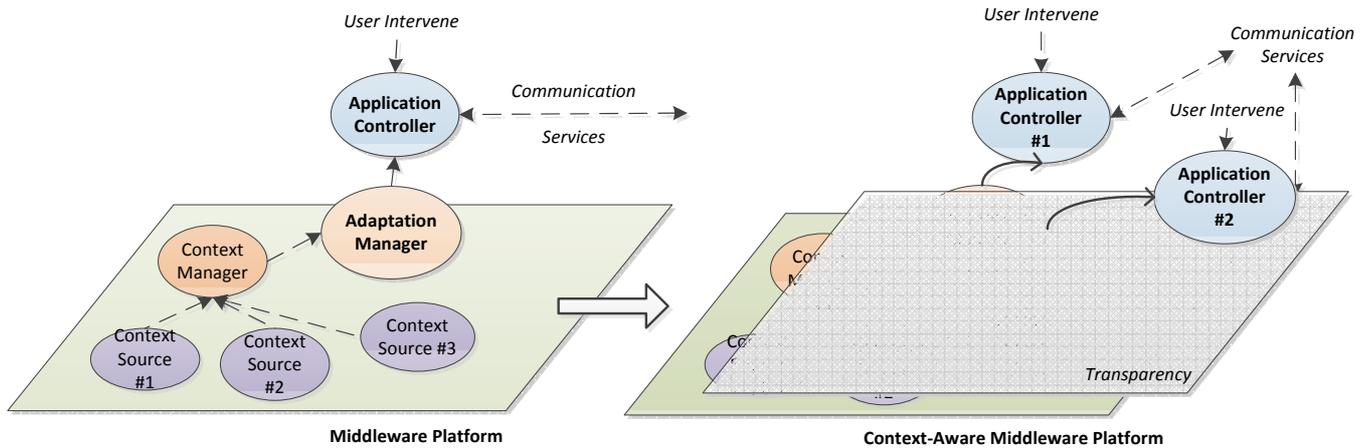


Fig. 4: A context-aware middleware is capable of wrapping, analyzing and delivering the physical world information to the application services in a transparent way.

no initial assumptions in response to the availability of diverse resources in physical world, and also to scalability, modularity, extensibility and interoperability of heterogeneous physical hardwares [42].

As shown in Fig. 3, within the ISO/OSI Reference Model, a conventional middleware takes place of the Session and Presentation Layers by providing a higher level of abstraction built over the network operating systems (OS), offering fault tolerant resource sharing, and masking out the problems to facilitate heterogeneity, stability and efficiency of distributed systems. On the other hand, the context-aware middleware is defined as an abstract layer between OS and up-running applications. It aims at dealing with the heterogeneity of physical world through edge technology, by adding more specialized mechanisms and services than an OS can provide. It is capable of wrapping (i.e., controlling physical devices and interacting with them to receive data), analyzing and delivering the physical world information (e.g., through sensor networks, embedded systems, RFID or NFC tags, etc.) to the application services in a transparent way, as shown in Fig. 4. This degree of transparency separates the application layer from the internal middleware operations and from the detailed implementations of lower layers directly. In essence, the middleware creates a shielded interface by both enhancing the level of abstraction support needed by the application, and intending to hide lower layer operations between the physical layer (i.e., hardware and communications) at the bottom and the application layer at the top. Furthermore, it allows the computational burden required for context management to shift from the application to the middleware by letting the developers only deal with implementation logic, and easily control the created entities (i.e., characterized context) by context management. In this regard, robust optimization in many system constraints (e.g., relative computational cost associated with entity relevant operations, limited battery power, insufficient information storage etc.) can be achieved. Moreover, the middleware will take the responsibility of all context-related entity management, and provide a complete global access into common resources needed by all applications residing on the

same host without any conflict [43], [44].

Fig. 4 also depicts the core components within the context-aware middleware design. Context Manager is responsible for collecting, processing and maintaining low-level context information (i.e., physical context) acquired through context sources. Basically, it converts low-level information to a high-level event (i.e., sensed context), handles context dissemination and inconsistency detection, and notifies the adaptation manager of the high-level event. Adaptation Manager queries, processes and regulates all contextual information/objects (i.e., inferred context) actively being used by each application, and also automatically receives a context change in case where a different context is observed due to the heterogeneity of context sources. In addition, it filters unnecessary information to have an optimal and effective result based on current context together with inclination or preference of user activity. Application controller has the highest-level context (i.e., presumed context) obtained through inter-working with the adaptation manager. It processes the final context and sends attribute information back to the context manager. Most importantly, the application controller does not have to interact with context sources, and even it does not know what context coming from which context sources at any time.

It is also worth noting that semantic metadata plays an important role in context-aware middleware, since it is defined at a high level of abstraction to represent contexts as structure and meaning of entities, and also to present context-related adaptation strategies, which enable the middleware to behave dynamically with a minimal human intervention. Having semantic metadata allows unambiguously specification of context models and knowledge share among entities without loss of meaning. Thanks to its interoperability and openness, it also allows to infer some other complex knowledge at the upper layers in presence of variant semantic metadata. However, the complexity of context resources in heterogeneous physical world, and also interactions among diverse context resources make it difficult to describe the relevant metadata explicitly.

Other properties supported by the context-aware middleware include:

- It either runs *standalone* for managing entire physical environments or accept the existence of an infrastructure which can deliver required services. This differentiation is caused by heterogeneous sensing environment.
- It can have a *reflective* property that represents the obtained entities through context as semantic *metadata*. The metadata may belong to application, middleware itself, a context, or interconnected contexts (i.e., *composition*). Note that a context can differ (e.g., asynchronously obtainable over different sensors), and inter-operate with other contexts. This reflection property allows the middleware to monitor its computation and detects a possible change in the semantic world, allowing the middleware model itself self-represented. For example, manipulation of its behavior may be changed. Hence, any change occurred at the meta-level can affect the underlying base level, or vice versa.
- *Adaptation* is an important design merit that empowers users to *customize* systems according to individual preferences. This adaptation is defined by an autonomous process triggered by a set of requirements to improve quality of service (QoS) at the application layer. It intends to sense the physical world, reason the obtained context, and react dynamically towards the changing context. It also supports proactive adaptations that describe the capability to envision future application requirements caused by the context change, and to adjust the functionality accordingly to prevent/minimize direct application interaction with neither interfering nor modifying the application logic.
- It may constitute entities from physical/virtual context for all types of applications, or it can provide an application specific information delivery (i.e., service provider).
- It must run smoothly with the underlying OS. Since mobile applications run on a resource limited devices with low memory size, slow CPU frequency, and low power supply, light-weight middleware systems need to be designed.

A middleware for context-awareness supports the application development task by enhancing the level of abstraction and providing services in dealing with context. By this means, there are many middleware studies in the literature. The notable ones are listed in the followings, and compared in Table II.

- Context Toolkit [45] is one of the earliest efforts in this domain. It delivers a combination of features and abstractions to capture and manage context source, and also to aggregate and share them among applications.
- Aura [46] is an architectural ubiquitous sensing framework. It provides context, application and task managements. Tasks are abstract representations of a collection of services. The framework detects environment changes, and migrates task operations into available service providers in the new environment. It is capable of adapting in the presence of dynamic resource variability, thereby it supports continuity of service for applications.
- CARISMA [47] provides a reflective middleware for mobile systems. It is also adaptive to dynamic environment changes. The tasks in context providing services are prioritized and resolved depending on importance ruled by applications, policies, and configurations under different environmental and user conditions.
- Gaia [48] is a distributed probabilistic based context-aware middleware that coordinates ontology based software entities and heterogeneous physical networking devices. It provides context management, detection of events, workload partitioning event handling, and virtual context file management.
- SOCAM [49] is a service oriented ontology based context-aware middleware. It supports semantic representation and reasoning of context. It also divides context into upper and lower level ontologies such as interpreted context through physical world, and memory and battery status respectively. It allows adaptability by listening, detecting and invoking events for application services.
- COSMOS [50] is a context-aware middleware that accepts contextual information as a context node, and organized many context nodes in a hierarchical structure. Each context node runs independently while collecting, processing and reasoning context. The middleware follows this distributed architectural model to create scalability by supporting many heterogeneous contextual sources, and their relations to each other.
- CoBrA [51] is a centralized middleware architecture that connects various context brokers. Each context broker runs independently, but the middleware creates a knowledgeable context network share. The middleware also addresses resource limitation and privacy issues in mobile computing.
- Hydra [52] is an IoT based middleware design to deliver solutions to wireless devices and sensor used in ambient awareness. It contains powerful reasoning toward various context sources including physical device based, semantic and abstract layer based. Therefore, it uses hybrid modeling scheme to represent low level context by object oriented modeling with key based approach, and high level context by ontologies.
- CASS [55] is a centralized middleware for context-aware mobile applications. Mobile clients connect to the middleware service through wireless network, and the middleware listens sensors on mobile clients, and gathers information.
- COSAR [53] is a context sharing architecture for mobile network services. It reasons human activity based context-awareness.
- QoS DREAM [56] is a component based middleware framework for the construction and management of context-aware multimedia applications. It also provides handling of location data derived from a variety of location-sensing technologies.
- Ubiware [54] is a self-managed middleware platform that utilizes different context agents in decentralized manner to manage mobility and scalability, enable autonomous context discovery, and configure complex functionalities such as composition and interoperability of relations

TABLE II: Notable context-aware middleware studies

Reference	Middleware	Architecture	Context Management	Context Source	Context Level	Reflection	Adaptation	Security and Privacy
[45]	Context Toolkit	Centralized	Key-value	Any type	High	-	Application based	✓
[46]	Aura	Distributed	Mark-up	Any type	High	✓	Middleware based	-
[47]	CARISMA	Distributed	Mark-up	Mobile	High	✓	Middleware based	-
[48]	Gaia	Distributed	Logic and Ontology	Any type	High	-	Application based	✓
[49]	SOCAM	Distributed	Ontology	Any type	High	-	Application based	✓
[50]	COSMOS	Distributed	Object Oriented	Physical	High	✓	Middleware based	-
[51]	CoBrA	Component	Ontology	Any type	High	-	Application based	✓
[52]	Hydra	Distributed	Ontology and Object	Physical	High	-	Application based	✓
[53]	COSAR	Stand-Alone	Ontology	Mobile	Low	-	Application based	✓
[54]	Ubiware	Stand-Alone	Ontology	Any type	High	✓	Middleware based	✓
[55]	CASS	Centralized	Logic and Object Oriented	Mobile	High	-	Application based	✓
[56]	QoSDREAM	Component	Logic and Object Oriented	Mobile	Low	✓	Middleware based	-
[57]	TinyRest	Centralized	Mark-up	Any type	High	-	Application based	-

among agents. Therefore, the middleware plans to create a collaboration among heterogeneous sources through semantic communication services.

- SALES [58] and CoMiHoc [59] are mobile environment based middleware platforms that support context management and situation reasoning through interconnection of various mobile devices.
- TinyRest [57] and Feel@Home [60] create an IoT based smart offices/homes by actuating wireless sensor networks through Internet connection. They act like a gateway to access different types of sensors and actuators, and fuse them to be able to support diverse application domains.

Like noted in Table II, the middleware design can accept different architectural structures. Centralized architecture, i.e. context server, offers a complete middleware design, and allows applications to be developed on top of it. The architecture connects to sensors and devices to provide rich resources and computational power. Communications among devices are handled by queries on the context server. The drawback for this type of middleware could stem from congestion of received queries. Stand-alone, or self-contained, architecture offers a direct access to sensors, but it does not allow context sharing of devices, connection to external services, or any type of device collaboration. It is ideally designed for small-scale solutions. Distributed architecture creates a hierarchical topology of connected many devices running independently and also having capable of their own context management services. It does not require the existence of a context server. Physical devices are connected through ad-hoc communication for information sharing, but the architecture lacks of processing computationally intensive resources. Component based architecture partitions entire middleware solutions on a few major components that interact each others. There are also some other design types available for middleware architecture such as node or client-server based. On the other hand, privacy issue arises when user related contexts are collected, used and shared autonomously. Therefore, many middleware designs try to secure the context by making queries as in who controls what information out of context. Basically, they introduce the concept of context ownership to determine which application needs to be granted or to be denied to access on a specific



Fig. 5: The processing stages of context inference.

concept.

### E. Context Inference

As one of the most important properties of context-aware middleware, context inference has drawn much interest in the research area. The middleware provides basic functionalities such as sensory data acquisition, processing and context recognition. The applied methodology may show differences in context modeling or reasoning. The context initially is called “low-level (i.e., atomic) context” since all required operations are carried out directly from data obtained by physical sensors. On the other hand, “high-level context” is obtained later through the combination of low-level and/or high-level contexts, which is called “composition”. Some mobile classifier development tools such as “Kobe” [61], “WEKA” [62], and former toolkits “The Context Toolkit” [45] can deal with low-level context acquisition from raw sensory. They infer high-level semantic outcomes while exhibiting efficient utilization of available resources, and achieving an optimal balance among energy, latency and accuracy tradeoffs. Fig. 5 shows the stages of context inference problem during the low-level process.

Accordingly, sensory readings are collected by a sliding window with a specific time interval and an overlap value. The length of windowing is an important design merit. The shorter windowing cannot seize the context properly, whereas the wider windowing would create a latency in detections, and puts additional workload in computations. Thus, obtained data segments by optimal windowing would provide more relevant information for context classification. In addition, the overlap value is important as well to detect any change in the context.

Preprocessing (e.g., context filtering or fusion) could be applied, if raw sensory data is too coarse-grained. It may also offer necessary modifications to correct deficiencies in the data due to the possible limitations on sensory operations (e.g., power concern).

It is generally very challenging to analyze, build a classification model, and infer any context from raw sensory data since it may consist of a large number of variant attributes, irrelevant information and additive noise distortion. Therefore, feature extraction is applied to exploit hidden information in the sensory data set, and remove direct effect of additive noise distortion. It also enables separability in the context classification algorithm while extracting and analyzing the spatial characteristics of sensory data in each window and assisting in identification of different context classes. A feature vector, as a representation of statistical characteristics in the contextual data, is then constructed by using diverse signal processing primitives, ranging from time space-based features such as mean, standard deviation, correlation etc. to frequency spectrum-based features such as entropy, Fast Fourier Transform (FFT) coefficients, power density etc., and also to wavelet transforms. Table III summarizes the elements of a feature vector in time and frequency domains. Time-domain feature extraction is the most-popular one in many practical real-time applications since sensory samplings are already obtained in a time-series way. Whereas, the frequency-domain features such as FFT coefficients require much computational power to discriminate such feature like periodicity of signals. On the other hand, sensors such accelerometers generate random signals in their nature, therefore, using time-domain features could be successful up to some limit since it is assumed that signals are mostly deterministic in time-domain analysis to make such a differentiation. As a result, it would be better to apply stochastic analysis in these cases in order to describe a suitable feature space. The purpose of the elements being used in the construction of feature vectors is as follows:

- *Mean* represents DC component of a signal.
- *Variance* shows dynamics of a signal activity. For instance, a low dynamic activity, or a stationary signal, will have a low value of variance.
- *Standard Deviation* basically shows the similar information like variance does. It also notifies how far signal samples are spread out from its mean value.
- *Energy or Root Mean Square (RMS)* captures the intensity of a signal.
- *Correlation among signals* helps distinguish a similar activity occurring through a single dimension or multiple dimensions.
- *Zero-Crossing Rate* captures the cyclic pattern of a signal. It could be seen as an approximation of frequency. Thereby, it requires less computation in time domain rather than having frequency value in spectrum analysis.
- *Spectral Peak* shows the dominant frequency of a signal activity.
- *Spectral Entropy* allows to specify whether or not energy is evenly distributed through different frequencies. For stationary signals, the entropy increases; whereas, non-deterministic signals gives out less entropy by having peaky look in spectrum.
- *Interquartile Range* is used where different signals have a similar mean value. It represents the dispersion of signal data except for taking the extreme values into account.

TABLE III: Feature selections

Feature Space	Features
Time Domain	mean, standard deviation, variance, magnitude, derivative, min-max, amplitude, histogram, interquartile range, mean absolute deviation, correlation between axes, peak counting, rms, sign, and kurtosis, zero-crossing rate
Frequency domain	Fourier Transform (FT), Discrete Cosine Transform, entropy, centroid, maximum frequency, FFT energy, FFT mean and standard deviation
Others	autoregressive coefficients, wavelet transforms

- *Mean Absolute Deviation* gives the averaged dispersion of signal data with respect to its mean value.
- *Spectral Centroid* is the balanced point of the spectral power distribution of signal.
- *Bandwidth* is the range of frequencies that occupy signal spectrum.
- *Normalized Phase Deviation* notifies the phase deviations of spectral frequency peaks, that are weighted by their magnitude.
- *Derivative* clears DC offset of a signal, and shows intensity of variations in signal data.
- *Histogram* captures the density of a signal.
- *Kurtosis* gives the peakedness (width of peak) of the density of a signal.
- *Discrete Cosine Transform* is similar to FFT that enables to have spectrum based analysis, but only using real numbers.
- *Autoregressive Coefficients* are used for filters to estimate characteristics of a signal.
- *Wavelet Transform* is essential to time vs. frequency analysis. It allows only changes in time extension in correspondence to frequency analysis.
- *Cepstral Coefficients* gives information about the rate of change in different spectrum bands.

The diverse characteristics of feature vectors enable to have training data classes (i.e., structural features) for classification algorithms. Thus, a training data class is employed by classifiers to build a classification model, which will allow an unknown feature vector to test for its membership to any class dependency. A confusion matrix can be used to measure the performance of a classifier. The process basically is to test one of classifiers to map a feature vector into a training contextual data class, called *supervised classification*.

In this regard, various classification algorithms can be used as a classifier to implement a context recognition system. Techniques include, but are not limited to, Naive Bayesian approaches and Decision Trees, pattern recognition techniques such as *k*-Gaussian Mixture Model (*k*-GMM), *k*-means, *k*-Nearest Neighbors (*k*-NN) search, Support Vector Machines (SVMs), and Multi-layer Neural or Fuzzy Logic Networks. In addition, a statistical tool based classification such as Hidden Markov Models (HMMs) or AutoRegressive (AR) Models is also widely applied; and pattern recognition toolkits such as WEKA provides powerful solutions to the context clustering

problem. Table IV classifies studies according to the applied classification methods. The followings are listed for widely-used classification algorithms, and for their roles in context-awareness:

- Naive Bayesian approach [63] assumes that each feature is conditionally independent in a given class definition, and estimates inter-class-conditional probability. Therefore, a Naive Bayes classifier fuses individual classification results to improve classification accuracy and robustness. This method uses the probability information residing in a training data in order to find the maximum probability of given hypothesis using the Bayes rule. On the other hand, Decision Trees [64] partition the feature space according to a tree structure. These structures fit the purpose of induction, and they are fast to be built for context inference on mobile devices.
- A multivariate-Gaussian ( $k$ -GMM) [65] is a maximum likelihood classifier based on mean vector and covariance matrix estimated from each class. Any feature vector (i.e. tested data set) can be drawn from this model to check for which data class encapsulates a given specific feature vector (i.e., training data set), *clustering problem*. This is also called density problem, in which each class represents a cluster that is assigned as a Gaussian model with its mean approximately in the middle of the cluster, and also with a standard deviation showing a measure of how far the cluster spreads out.
- $k$ -Nearest Neighbors ( $k$ -NN) search algorithm [66] assigns the nearest class set for the input feature vector by defining a dissimilarity function that measures the nearness between training data set and new data points in the feature vector. The dissimilarity function is generally defined by the squared Euclidean distance. However, the Euclidean distance does not consider how the data is spread out, and also may let the largest length scale between data points dominate the dissimilarity function. Therefore, the Mahalanobis distance is used where the covariance matrix rescales all length of scales to make them essentially equal.
- Support Vector Machine (SVM) [67] differentiates two classes from each other by using linear discrimination. SVM denotes each class as a binary data by labeling them +1/-1. The objective is to create a hyperplane that sets a rigid margin among data classes to achieve an optimal linear distance separation. Unfortunately, SVM cannot deal with multi-class classification directly. The multi-class classification problem is usually solved by decomposition of the problem into several two-class problems.
- Multi-layer Neural or Fuzzy Logic Networks [68], [69] create a multi-dimensional Gaussian memberships. They can also be decomposed into a number of one-dimensional Gaussian membership functions to correlate with the number of input feature data. Each class in a multi-dimensional feature space represents a member of the classification network. The output is obtained by checking for Gaussian memberships of each input according to majority vote in the network classifiers.
- $k$ -Means clustering algorithm [70] is associated with a specific case of Gaussian mixture models that stems from the limitation of covariance matrices such as them being equal, diagonal or small for each user state class.  $k$ -Means algorithm finds the members of each class from a given data, where the classes are represented by their centers, which also show updated/re-constructed mean values.
- A statistical tool shows dependencies of states at discrete time that are influenced directly by a state/states at prior discrete times. Discrete time is used to specify periodic sensor readings. Therefore, Hidden Markov Model (HMM) [71] is a mostly applied statistical tool that models time-series with spatial and temporal variability. In such statistical classifiers, sensor readings (i.e., extracted user contexts through mobile device based sensors) are seen as inputs. These readings undergo a series of signal processing operations, and eventually end up with a classification algorithm in order to provide desirable inferences. A required classification algorithm differs in terms of explanation of extracted context through a specific sensor. Outcomes of the algorithm are represented in a matrix whose elements show probability weights for possible context selections. Classification algorithms produce observations (i.e., *visible states*) of HMM. Among observations, only one observation is expected to provide the most likely differentiation in the selection of final context inference. On the other hand, desirable context inferences are defined as *hidden states* of HMM since they are not directly observable but only reachable over visible states. Therefore, each observation has cross probabilities to point any context inference. These cross probabilities build an emission matrix that basically defines decision probabilities of picking context from available observations.
- Auto-Regressive (AR) models [72] are used to show the correlation among various feature parameters for each context inference. AR models apply time series analysis in which a multi-dimensional vector is transformed into a number of coefficients to make the analysis much easier. By doing that, AR captures the evolution and inter-dependencies among time series.
- The feature vector constructed by any classification algorithm requires much computation. There is also no necessity to compute such features that are irrelevant or redundant to infer the context by providing insignificant improvement in accuracy. Hence, it is desirable to reduce the complexity and dimension of the feature set by retaining the core probability distribution spanned through feature vector spaces. In such cases, the dimension of feature vector can be reduced by using Principal Component Analysis (PCA) [73] or Linear/Quadratic Discriminant Analysis (LDA/QDA) [74], [75]. They help obtain sufficient statistics to model the context and allow lower computational complexity. Both methods seek for a projection vector that transforms the original features into a lower dimensional space by preserving the content of separability. Unlike the PCA, the LDA performs well in

TABLE IV: Classification algorithms

Classification Algorithm	References
Decision Tree (DT), Bayesian Network (BN), Naive Bayes (NB)	[76]–[82]
Multilayer Neural Networks (MNN)/ Meta Classifier Fusion	[83]–[87]
Fuzzy Logic	[84], [87], [88]
Gaussian Mixture Model (GMM)	[89]
$k$ -means/ $k$ -Nearest Neighbour ( $k$ -NN)	[90], [91]
Hidden Markov Models (HMM)	[78], [86], [92], [93]
AutoRegressive Models (AR-M)	[94], [95]
Support Vector Machine (SVM)	[82], [94], [96], [97]
Principal Component Analysis (PCA)	[84], [97]
Linear/Quadratic Discriminant Analysis (LDA/QDA)	[84], [85], [91]
HAAR Wavelet Models	[98]
Classifier ensembles: Boosting and Bagging	[99]
Toolkits: WEKA	[98], [100]–[102]

seeking a suitable projection for data discrimination by applying an effective separation in data transform into different classes.

The output of classifiers sometimes cannot resolve consistent discrimination in a time sequence of adjacent context inferences. In this case, a basic smoothing technique takes a majority voting scheme with a sliding window of a specific history length of context inferences. Hence, any inconsistency (i.e., false truthfulness) can be eliminated.

#### F. Online Context Processing

The context inference process may cause many drawbacks and tradeoffs, with respect to context classification algorithms. A statistical tool-based classification, mostly using Hidden Markov Models (HMMs) or using AutoRegressive (AR) models, is one of the foremost methods to infer context obtained via wearable or built-in smart device sensors. However, these studies mostly allow predefined user-manipulated system parameter settings, such as arbitrary formation of context transition matrix in HMMs, or building filtering coefficients in ARs, which is not suitable for online processing due to increasing computational workload while enlarging the data size. In addition, the quantitative nature of statistical tools makes it difficult to discriminate morphological bounded patterns and their interrelationships

Other methods rely on creating feature vectors that aim at exploiting signal characteristics of sensory data and then cluster these vectors according to specific data classes. These methods mostly intend to implement a framework which imposes a wide-range of context-aware workload, by using diverse signal processing primitives. The defined frameworks generally analyze spatial characteristics of the sensory signals, based on either time space or frequency spectrum. High-dimensional feature vectors are constructed by including many signal processing functions. Then, these large feature vectors undergo pattern recognition techniques. The major drawback of these attempts comes from the fact that when processing large data clusters in a resource limited hardware. Especially, for an online classification algorithm, one of the most important things is to reduce computational burden and stay away from large amount of data manipulations. For instance,

computational complexity time requires  $\mathcal{O}(\eta\mathcal{L}\mathcal{D}^2)$ ,  $\mathcal{O}(\mathcal{L}^2\mathcal{D})$ , and  $\mathcal{O}((\mathcal{L} + c)^2\mathcal{D})$  in a process of GMM,  $k$ -NN, and LDA respectively where  $\mathcal{L}$  is the length of feature vector,  $\mathcal{D}$  is the dimension,  $\eta$  is the number of iterations GMM algorithm, and  $c$  is the total number of user state classes. Increasing sizes of collected data, exploited dimensions, and extracted features affect the performance of online computation by adding additional matrix multiplications in pattern recognition algorithms.

In this sense, the desired approach to implement the context recognition process should provide the inference without considering *a priori* information, fixed thresholds, and initial training data classes. It also needs to show robustness in terms of any change in orientation of the device, dynamic profile in user context, and employ sufficient signal processing by causing non-redundant computational workload. Since supervised classifiers need extensive computations to generate models for training contextual data classes, and testing for unknown patterns, unsupervised learning is an active research area due to its nature of focusing on clustering or pattern discovery rather than classification. Therefore, the definition of self- or co-learning based semi- or un-supervised classifiers to actualize proactive context inferences without knowing any data class have been actively investigated [103]. Table V delivers a comparison table among classification algorithms used in context-awareness.

#### G. Context-Aware Framework Designs

The pervasive mobile computing, which captures and evaluates sensory contextual information in order to infer user relevant actions/activities/behaviors, has been becoming a well established research domain. Most studies rely on recognition of user activities (especially posture detection) and definition of common user behaviors by proposing and implementing numerous context modeling systems. In addition, researchers have been aware of the need for computational power while trying to infer sensory context accurately enough. However, most works provides some partial answers to the tradeoff between context accuracy and battery power consumption. It is difficult to say that power saving methods have been significantly taken at the low-level physical sensory operations. Especially, there is not a generic framework that intends to apply *adaptively changing* dynamic sensor management strategies. In contrast, most works for creating a context-aware application emphasize either to set a minimum number of sensors or to maximize power efficiency by solely applying less complexity in computations and/or changing transferring methods of obtained context to the outer network services.

From the standpoint of the creation of a generic framework design for context-aware middleware services, it would be notable to mention the following studies:

- “EEMSS” in [76] uses hierarchical sensor management strategy by powering a minimum number of sensors and applying appropriate sensor duty cycles so that the proposed framework could recognize user states through smartphone sensors while improving device battery lifetime. Unfortunately, sensors employ fixed duty cycles

TABLE V: Comparison among classification algorithms

Algorithm Name	Pros	Cons
Naive Bayesian	<ul style="list-style-type: none"> <li>· uncertainty handling</li> <li>· allows combinational reasoning</li> </ul>	<ul style="list-style-type: none"> <li>· probability bounded</li> <li>· numerical outcomes</li> </ul>
Supervised (e.g. SVM, k-NN, Ensembles)	<ul style="list-style-type: none"> <li>· many techniques available</li> <li>· discriminates morphologically interconnected patterns</li> <li>· provides more accurate inference</li> </ul>	<ul style="list-style-type: none"> <li>· computationally expensive</li> <li>· requires huge data set to have more accurate assumptions</li> <li>· training data set required to matching</li> <li>· challenging to find optimum feature set</li> <li>· mostly user intervention needed to specify training data</li> </ul>
Unsupervised (e.g. k-means)	<ul style="list-style-type: none"> <li>· no training data set needed</li> <li>· more machine learning included</li> <li>· robust and adaptive</li> </ul>	<ul style="list-style-type: none"> <li>· computationally expensive</li> <li>· complex system design</li> <li>· difficulty in validation</li> </ul>
Fuzzy Logic	<ul style="list-style-type: none"> <li>· simple and easily applicable</li> <li>· provides more understandable reasoning</li> <li>· uncertainty handling</li> </ul>	<ul style="list-style-type: none"> <li>· manually defined</li> <li>· prone to have false truthfulness</li> <li>· no quality check</li> </ul>
Decision Tree	<ul style="list-style-type: none"> <li>· simple and easily applicable</li> <li>· requires less computations</li> <li>· expandable</li> </ul>	<ul style="list-style-type: none"> <li>· manually defined threshold based</li> <li>· prone to have false truthfulness</li> <li>· no quality check</li> </ul>
Stochastic	<ul style="list-style-type: none"> <li>· provides accurate inference</li> <li>· allows combinational reasoning</li> <li>· quantitative features</li> <li>· uncertainty handling</li> </ul>	<ul style="list-style-type: none"> <li>· predefined expected probabilities</li> <li>· training data sets for coefficients</li> <li>· difficult to discriminate morphological patterns</li> <li>· probability bounded, ignores feature relationships</li> </ul>

whenever they are utilized, and also they are not adjustable to respond differently to variant user behaviors. Energy consumption is reduced by shutting down unnecessary sensors at any particular time. On the other hand, classification of sensory data is based on pre-defined test classification algorithms.

- The hierarchical sensor management system is also studied by introducing “SeeMon” system in [77] which achieves energy efficiency and less computational complexity by only performing continuous detection of context recognitions when changes occur during context monitoring. The framework also employs a bidirectional feedback systems in computations to detect similar context recognitions in order to prevent from redundant power consumptions.
- Similarly, “Sensay” in [104] is a context-aware mobile phone but as in form of an external sensor box which is mounted on the users’ hip area. It receives many different sensory data, and eventually determines to dynamically change cell phone ring tone, alert type and un-interruptible user states. However, it classifies user states offline, and the system does not have energy efficiency.
- “Darwin” studied in [89] proposes a system that combines classifier evolution, model pooling, and collaborative inference for mobile sensing applications. It is implemented for a speaker recognition application by using efficient but sophisticated machine learning techniques; however, there is no power saving method applied.
- “Jigsaw” presented in [105], a continuous sensing engine

for mobile phone applications, balances the performance needs of an application and resource demands. The engine employs each sensor under a processing pipeline. It performs all the sensing and classification processing exclusively on the mobile phone. It also uses sensor-specific pipelines that have been designed to cope with individual challenges experienced by each sensor. Duty cycling techniques are attached to adaptive pipeline process if applicable to conserve battery life.

- The study in [106] creates a general framework problem under an energy efficient location based sensing application. It is noted that there are four critical factors that affect energy efficiency in location-sensing through GPS. These factors are static use of location sensing mechanisms, absence of use of power-efficient sensors to optimize location-sensing, lack of sensing cooperation among multiple the similar applications, and unawareness of battery level. The framework solution is given by introducing *substitution* to find an alternative less power consuming location-sensing mechanism, *suppression* to use less power consuming sensor instead of GPS when user location is static, *piggybacking* to synchronize with other location-based applications to infer a collaborative location info, and *adaptation* to adjust system parameters such as time and distance longer when battery level is low.

There are also some other studies proposed in [86], [95], [107] in order to provide comprehensive solutions into creating a base framework for context-aware applications. Table VI

analyzes some important frameworks by breaking down each study and comparing them in terms of learning paradigm, applied algorithm, power efficiency, processing method, interested context, input sensor, platform where implementation is carried out, and accuracy of framework outcome.

### III. CONTEXT-AWARE APPLICATIONS

Mobile phones are equipped with sophisticated sensors. Most sensors currently available on smart devices are designed to perform some specific applications, such as accelerometers for detecting screen orientation, a microphone for voice conversations, a camera for capturing images and a GPS for displaying location. However, by introducing intelligence, situational awareness and context recognition into these devices, and given the right architecture within the context of ubiquitous sensing by enhancing and systematizing the existing methodologies, built-in sensors could be re-purposed and act as sensor nodes to proactively assist users in their daily activities by increasing the quantity, quality and credibility of community-gathered data. Hence, these smart devices could be used as instruments to collect data and provide meaningful observations belonging to user behaviors and surrounded environments. Some applied examples are activity measurement by accelerometer, ambient sound environment by microphone, and estimation of time and location a user spends indoors and outdoors by GPS. In addition, external sensors, such as biomedical sensors (e.g., ECG, BVP, GSR, and EMG<sup>4</sup>), can also be deployed with a wearable strap on human bodies. Hence, more than one sensor (multiple sensory system) would be available in ubiquitous sensing for health. Information obtained from different sensors can also be cross-linked and presented as a new valuable input. For instance, GPS and accelerometer actualizes Geographic Information Systems (GIS) with potentially providing insight as to how the proximity of recreational facilities affects physical activity levels, or how the relative accessibility of grocery stores and fast-food restaurants influence a diet program. Wi-Fi can be leveraged to determine relative proximity of individuals to each other or fixed locations, it could be used for a study to examine the spread of an infectious disease. Bluetooth, as well as ZigBee, can also be used for ambulatory data collection of more traditional signals, such as blood pressure, heart rate, respiration, and blood glucose level.

In this section, context-aware applications are categorized in terms of the application fields that they are designed for. The categorization, as shown in Fig. 6, introduces different application fields that researchers have been studying extensively.

#### A. Healthcare and Well-being

Previously, the use of mobile devices within the context of the ubiquitous sensing has been successfully integrated in zoology and veterinary medicine to study the feeding habits and social behaviors of some types of animals from zebra to

<sup>4</sup>Electrocardiography (ECG), Blood volume pulse (BVP), Galvanic skin response (GSR) and Electromyography (EMG).

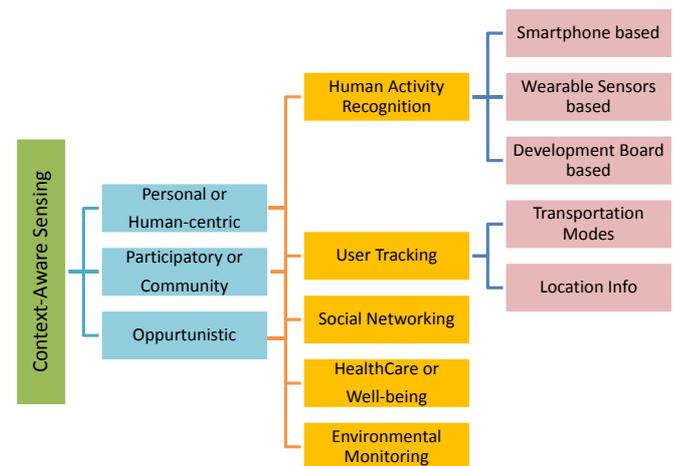


Fig. 6: The categories of context-aware sensing.

whales; whereas, the adaptation of this technology to human health has been paid attention recently.

With the advancements and increasing deployment of micro-sensors and low-power wireless communication technologies within the Personal/Body Area Network (PAN/BAN), the studies conducted under ubiquitous computing have grown interest in healthcare domain. Besides high demands for applying and understanding Human Activity Recognition (HAR) based systems, the integration of monitoring and analyzing vital sign data (e.g., heart rate, blood sugar level and pressure level, respiration rate, skin temperature, etc.) through sensors also more likely enable to change assessment, treatment and diagnostic methodologies in healthcare domain since traditional methodologies have been based on self-reports, clinic visits and regular doctor inspections [119].

With the integration of emerging technologies in healthcare domain, sensor-enabled autonomous mobile devices can help caretakers continuously monitor patients, record their wellbeing process, and report any acute situation in case where abnormal behavior is detected. Thereby, it would be more easier and efficient to monitor and manage the lifestyles and well-beings of patients with chronic diseases, the elderly people, the rehab taking patients, the patients dealing with obesity, the patients with cognitive disorders, children, and even more significantly to monitor and rescue the emergent vitals and status notifying soldiers in combat zone.

The home-based health care monitoring by mobile based devices is defined under smart home applications. The studies in [120]–[122] are carried out in order to create a smart home environment for treatment procedures of patients (e.g., having cardiac problem [123], or diabetics) based on collecting data through different wearable physiological sensors (e.g., body temperature, heart rate, blood pressure, blood oxygen values, respiration level, and ECGs) and also reporting feedbacks remotely to the healthcare givers. The wearable sensors (including accelerometers, heart rate monitors and many others) have been also studied in [124]–[127] to recognize activity patterns for measuring fitness level, and discovering frequentness of body movement against obesity and weight

TABLE VI: Analysis of some notable framework designs

Reference	Learning Paradigm	Algorithms	Power Efficiency	Processing	Interested Context	Input Device	Platform	Results Accuracy
[102]	Supervised	$k$ -NN, DT and NB	N/A	Offline	Activities	ACC	Wearable Sensors	50-80%
[108]	Supervised	NB and HMM	N/A	Offline	Activities	ACC	Development Board	90%
[109]	Unsupervised	$k$ -means	N/A	Offline	Activities	ACC	Wearable	77%
[105]	Supervised	DT, GMM, SVM and NB	MDP based DSS	Online	Activities and Ambient Sound	ACC, MIC and GPS	Smartphone	94% for act. 84% for sound
[110]	Semi-Supervised	Multiple SVMs	N/A	Offline	Activities	ACC	Wearable	70-80%
[111]	Supervised	DT, SVM, NB, AdaBoost	N/A	Offline	Activities	ACC	Wearable	77%
[76]	Supervised	DT and FE	DSS and DC	Online	Activities and Ambient Sound	GPS, MIC, Wi-Fi and ACC	Smartphone	73-100% for act. 70-94% for sound
[112]	Unsupervised	SVM, $k$ -means and DT	N/A	Online	Activities and Environment	ACC, Wi-Fi and MIC	Smartphone	87%
[83]	Supervised	FE, HMM, NB and NN	DSS	Offline	Activities	ACCs	Wearable Sensors	80-90%
[78]	Supervised	FE, HMM and NB	N/A	Offline	Activities	ACC and Proximity	Wearable Sensors	88-94%
[79]	Supervised	FE and NB	AS	Online	Activities	ACC, BT and MIC	Smartphone	70-90%
[94]	Supervised	FE, AR and SVM	N/A	Offline	Activities	ACC	Development Board	92.25%
[96]	Supervised	FE and SVM	N/A	Offline	Activities	ACC	Smartphone	91-95%
[90]	Semi-Supervised	FE, $k$ -means, PCA and AdaBoost	N/A	Offline	Location	MIC	Smartphone	88.7%
[84], [85]	Supervised	FE, LDA and FBF	N/A	Offline	Activities	ACC	PC	93%
[100], [101]	Supervised	FE and WEKA Toolkit	N/A	Offline	Activities	ACC and HR	PC	80-94%
[80]	Supervised	FE	N/A	Online	Activities	ACC	Wearable	82-97%
[81]	Supervised	FE, NB, $k$ -NN	N/A	Offline	Activities	ACC and HR	Development Board	75-95%
[113]	Semi-Supervised	DT and FE	N/A	Online	Ambient Sound	MIC	Smartphone	78-93%
[98]	Supervised	FE, HAAR and WEKA	N/A	Offline	Activities	ACC	Development Board	90-94%
[114]	Unsupervised	DT	N/A	Online	Activities	GPS, Wi-Fi and GSM	Smartphone	90%
[86]	Supervised	FE, HMM and NN	N/A	Offline	Activities	ACC	Smartphone	87-90%
[115]	Unsupervised	FE and DT	N/A	Online	Activities and Location	ACC, GPS and WiFi	Smartphone	90%
[93]	Supervised	FE and HMM	N/A	Offline	Activities	ACC and MIC	Development Board	77-85%
[91]	Supervised	FE, QDA and $k$ -NN	N/A	Offline	Activities	ACC	Smartphone	90%
[87]	Supervised	FE and FN	N/A	Offline	Activities	ACC	Smartphone	97%
[89]	Semi-Supervised	FE and GMM	N/A	Online	Location	MIC	Smartphone	80-90%
[82]	Supervised	FE, DT and SVM	DC	Online	Activities and Social Context	ACC, MIC and GPS	Smartphone	90%
[116]	Unsupervised	FE, DT and HMM	DC and AS	Online	Activities	ACC	Smartphone	75-96%
[117]	Unsupervised	FE, DT and HMM	POMDP based DC and AS	Online	Activities	ACC	Smartphone	95%
[118]	Semi-Supervised	FE, AR and Estimation	RO	Online	Location	WiFi and GSM	Smartphone	90%

- DT: Decision Tree; FE: Feature Extraction; HMM: Hidden Markov Model; NB: Naive Bayesian; BN: Bayesian Network; SVM: Support Vector Machine; AR: AutoRegressive Model; PCA: Principal Component Analysis; LDA: Linear Discriminant Analysis; FN: Fuzzy Networks;  $k$ -NN:  $k$  Nearest Neighbour; GMM: Gaussian Markov Model; SVM: Support Vector Machine; NN: Neural Networks

- DSS: Dynamic Sensor Selection; DC: Duty Cycling; AS: Adaptive Sampling; MDP: Markov Decision Process; RO: Radio Optimization

- ACC: Accelerometer; BT: Bluetooth; MIC: Microphone; HR: Heart Rate Monitor

- POMDP: Partially Observable Markov Decision Process

loss programs [128], diagnosing insidious diseases [129] (e.g., hypotension), and understanding emotional states [130], [131] (e.g., stress level). Besides, smartphones can be used as a reminding systems [120] for aging related cognitive disorders such as Alzheimer treatment. Also, like a study called UbiFit in [132], smartphone can capture user relevant physical activity level and corresponds the obtained information to personal fitness goals by presenting feedback reports back to the user.

There are many commercial products available at the market to give ubiquitous computing solutions in the healthcare domain. These products are mostly concentrated on assisting people on controlling dietary programs/weight management, discovering fitness level, measuring burnt calorie or energy level, counting step numbers, and recognizing activities. Philips Directlife, FitBit Zip and BodyMedia GoWear are some device examples produced for tracking activity patterns, counting steps, measuring calorie burnt, and calculating distance traveled. In addition, Impact Sports ePulse proposes heart pulse monitoring system. Many other products can also be found for measuring heat flux, galvanic skin response and skin temperature.

### B. Human Activity Recognition

Recognizing human-centric activities and behaviors have been an important topic in pervasive mobile computing. Human Activity Recognition (HAR) intends to observe human related actions in order to obtain an understanding of what type of activities/routines that individuals perform within a time interval. By providing accurate information about HAR relevant data history could assist individuals on having better well-being, fitness level and situational-awareness [133]–[135]. For example, patients with diabetes, obesity, or heart disease are suggested to follow a predefined fitness program as a part of their treatment [136], [137]. In this case, information corresponding to human postures (e.g., lying, sitting, standing, etc.) and movements (e.g., walking, running, etc.) can be inferred by a HAR system in order to provide useful feedbacks to the caregiver about a patient's behavior analysis. In addition, by the attachment of external sensor devices, e.g., Heart Rate (HR) monitor, patients with abnormal heart beats can be tracked easily and notified to caregivers in case of emergency in order to prevent undesirable consequences [138]. In practical, HAR has only interest on single person activity detection; however, it can be extended to be multiple person recognition, which is called Activity of Daily Living (ADL). ADL is a way to describe functional status of a person, and his/her interaction with others. Hence, ADL becomes an essential part of community sensing especially for community health-care concerns (e.g., finding stress level in a group of people [139]).

Studies for HAR, as shown in Table VII, can be divided into sub-categories based on the platform that a context-aware system is built on:

- *A smartphone based:* Activity recognition basically concerns about human beings and/or their surrounding environment. The constant monitoring of activity recognition was used to carry out by deployment of cameras with high

cost or personal companion devices with no easy use. In addition, aggregation of monitored data was very complicated and impractical. However, since mobile devices are carried by people throughout the day, it makes them appear to be an ideal platform to be used in purpose of human-centric sensing. Especially, the accelerometer sensor, which can return a real-time measurement of acceleration through all coordinate spaces, is commonly used for HAR. It is employed either as a pedometer to measure steps counts and calorie consumption or as a monitor to recognize user physical activities such as postures and movements. Most measured events/actions/attributes are related to human posture or movement (e.g., using accelerometers or GPS/Wi-Fi/Cell Tower), environmental variables (e.g., using temperature and humidity sensors, microphone and cameras), or physiological signals (e.g., attachment of external devices such as heart rate or electrocardiogram, finger pulse, etc.). In this aspect, there are many studies [96] proposed to use smartphones to monitor users' daily physical activities according to their lifestyles.

- *Wearable sensors based:* Wearable sensors, i.e., multiple-sensor multiple-position solutions, have been put forward to recognize complex activities and gestures within the HAR concept. It basically introduces multiple-sensor placement on multiple location of human body to well capture some specific target activities (e.g., brushing teeth, arm and wrist movements while folding laundry, etc.) which a smartphone cannot detect by itself. With the use of wearable sensors, sensory context is extracted from miniature sensors integrated into garments, accessories, or straps. Especially, traditional accelerometer based HAR solutions cannot provide recognitions at finer granularities for differentiation of some postures such as sitting and lying down since there are some drawbacks observed such as mis-adjustment of device orientation and position or insufficient number of sensors to have enough spatial information. Hence, wearable sensors with utilization of heterogeneous sensor deployment has been an active research area to respond a growing demand for HAR systems in the health care domain, especially elder care support, assisting the cognitive disorders, and fitness/well-being management [78], [100]–[102].

Heterogeneous sensors are connected to each other with wired/wireless communication (mostly Bluetooth). Smartphone can be used as a center position for external sensor attachments. Proximity sensors decide the distance between sensor nodes (i.e., topology of sensor placement) by measuring the received signal strength indication (RSSI) of radio frequency in dBm. On the other hand, the deployment of heterogeneous sensors entail high cost and brings about some constraints in computationally since it requires intensive supervised classification algorithms. These algorithms are mostly carried out in offline analysis, which also makes the solution impractical. The constraints may also stem from sensor degradation, interconnection failures, and jitter in

TABLE VII: HAR in mobile devices

Platform	Reference	Sensors
Smartphone	[87], [91], [96], [98], [140]	ACC
	[90]	MIC
	[95]	ACC, GPS
	[79]	ACC, MIC, BT
	[118], [141]	WiFi, GPS
	[105]	ACC, MIC, GPS
	[76], [82], [89]	ACC, MIC, WiFi, GPS
Wearable Devices	[83], [86], [94], [102]	ACC
	[78]	ACC, Proximity
	[100]	ACC, HR
	[80], [81]	ACC, BT
Mobile Development Boards	[84], [94], [101], [142]	ACC
	[98]	ACC, BT
	[77], [85], [104]	ACC, Temperature, Light
	[80], [93]	ACC, MIC, Compass, Temperature, Light

- ACC: Accelerometer; BT: Bluetooth; MIC: Microphone; HR: Heart Rate Monitor

the sensor placement. Hence, the reduction of sensor dimension is highly important for node interconnection, and make the system stay still unobtrusive.

- *An embedded platform based:* In a HAR based system, higher classification accuracy is always desired. Especially, this implies a large number of sensor placements over the body in wearable sensors based applications. Variations in sensors and center device placements must let the system act robust to diverse feature extractions, and set specific classification models to make a context differentiation in any condition. Compared to the multiple-sensor multiple-position solutions, creating a development platform consisting of multiple sensors could be a more practical way for HAR based applications since the attachment of sensors on specific body locations could return in similar reflection of feature signal characteristics on different activities, and it could lack of distinguishing diverse contexts. Therefore, like using smartphones, researchers have been studied mounting one-board based embedded hardware platforms with a predefined device position and orientation in order to investigate the effect of sensor placement on activity recognition performance, and make a better differentiation in context by fusing multiple sensor information with *a priori* experiment setups and unchanging conditions of training contextual data.

### C. Transportation and Location

Location-based sensing [143] aims at tracking people over a period of time by recognizing their activities in terms of specifying transportation modes (e.g., walking, running, vehicle etc. when user is outside). Especially, since GPS receivers have been an integral hardware component in smartphones, data collected by GPS becomes handy to be used for network connected applications. Thereby, GPS is employed as an instrument for location-based sensing in order to inspect for the habits and general behaviors of individuals and communities [144]–[146].

The localization is inferred with the help by GPS delivered speed and location information as well as a large amount of available data (e.g., street maps). GPS provides 2D data by

setting a resolution value (e.g., generally 10 m) per certain distance within two successive data points (i.e., unit difference). Hence, consecutive GPS readings are grouped based on their spatial relationships in order to create distinctive segmentations among GPS traces. Then, GPS traces are associated with available street maps, which are represented as directed graphs where an edge represents a street and a vertex represents the intersection of streets.

GPS cannot penetrate through walls, and thereby the received data gets degraded. Thus, the usage of GPS for location-based sensing is valid for outdoors. Once GPS times out because of the lost satellite signals, Wi-Fi scan can be performed for indoors by checking for surrounding wireless access points. Wi-Fi could be used for outdoors either since it covers a range of 20-30 m as radius. Indeed, smartphones apply a hybrid localization scheme by using GPS with network-based triangulation by leveraging wireless access points for achieving coarse positioning [43]. The network-based triangulation collects information from RF signal beacons around reachable wireless cell towers, from Wi-Fi access points or even from Bluetooth (which is not effective but could be used indoor environment in presence of multiple users around), and then it uses received RF signal strength to measure relative distance through the physics of signal propagation among network nodes (e.g., utilization of local and mobile base stations). Hence, by measuring sequential RSSI data, transportation modes for users can be identified. In addition, during the Wi-Fi scan, MAC address (i.e., BSSID) of wireless access points might have already been tagged as a point of interest, which yields to retrieve automatically that user is in a familiar environment (e.g., office, home, gym etc.). Although GPS could detect some postures such as sitting or standing, the accelerometer sensor is rather used for such static activities due to GPS may not provide a concise solution for differentiation of user state classes at similar speed. Besides, the efficiency in power consumptions would be more healed in case where the accelerometer sensor is used.

Investigation of mobility patterns to extract places and activities from GPS traces have been generally implemented in a hierarchical structure [145]–[147]. According to the structure, the lower level begins with association of GPS traces with

street maps, and the structure rises up by inferring activity sequences; and eventually, the structure ends up with discovering significant places from activity pattern with the help of spent time within each activity. By taking a log of recent history of transportation modes belonging to individuals throughout the daily life as well as mapping their location history, a general physical activity report can be documented, and also the goals of future activity plan can be reconfigured for the purpose of health and fitness monitoring. For instance; from physiological perspective, driving behaviors are investigated in [148] by taking consideration of trip destinations, trip times and driving efficiency.

By actuating community sensing, it could be possible to monitor highways for real-time traffic conditions, and forecast probabilistic traffic congestions, thereby the traffic flow could be re-routed in such cases [147], [149]–[151]. This scenario can also be applied into biking [152], [153], thereby bikers can share their routes and let noisiness of the bike trails be known, and also they can take ride statistics for fitness documentation. Besides, most significantly, crowdedness level of metropolitan areas can be investigated in terms of daily visitor density [154], [155]. Meanwhile, existence of multiple users in a specific area could also give a help to track and notify air pollution level for environmental monitoring.

#### D. Social Networking

The ubiquity of Internet usage have enable people to exchange innumerable different form of information at a global scale. This situation have resulted in explosive growth in the creation of social network platforms (e.g., Facebook, Twitter etc.) where people can describe and share their personal interests, preferences and information. With the emergence of sophisticated sensors equipped smartphones, the integration of smartphones and social networks have leveraged data collection capability, and led the born of exciting context-aware applications as well as the evolution of the Internet of Things. However, the question of how the inference of a human relevant context can incorporate with social network platforms in an autonomous way is still the most exciting research topic. In this sense, researchers have been trying to create context-aware systems where diverse and large data streams (e.g., image, video, user location, user transportation mode) are automatically sensed and logically fused together for social interaction of individuals or groups of people, which is sometimes called *crowdsensing* or *crowdsourcing*. “CenceMe” [156] is the foremost study which enables to infer user relevant activities, dispositions, habits and surroundings, and then to inject these information into social networks platforms. The fusing of sensor and social data for context-aware computing is also studied in [157]. A detailed study for the current state and future challenges of the crowdsensing is given in [20], [158]. In addition, some exciting futuristic project ideas can be obtained through [www.funf.org](http://www.funf.org).

On the other hand, privacy, security and resource considerations unfortunately limit the expansion of community-based sensing applications since cyber-stalking [159] by tracing the revealed user information could harm mobile users by economically, physically, and legally. In the absence of relevant

concerns, some websites/applications such as [160] can be used for reducing risks of community sensing.

#### E. Environmental

Environmental monitoring, on one hand, aims at sensing and collecting information about surrounding environment by basically providing personalized environmental scorecards at the human level; on the other hand, it creates an impact toward environmental exposure by contributing environmental solutions at the community level. The surrounding environment is either a small scale area (e.g., indoor) or a large one (e.g., outdoor). For indoor environments, applications to monitor HVAC systems and building maintenance can be considered [161], [162]. For instance, one can use a smartphone to measure room temperature, and then smartphone can adjust heater or ventilator automatically to change air balance in a smart home environment. Moreover, it would be more reasonable to apply environmental monitoring in the context of community sensing. The studies in [163]–[169] provides applications for environmental monitoring to track and notify hazard exposures such as carbon emission level, air pollution, waste accumulation, water intoxication level, etc. In addition, noise pollution and ambience fingerprinting (fusion of sound, light and color) are other topics that have been studied in this content [112], [170].

### IV. CHALLENGES AND FUTURE TRENDS

Mobile, smart devices supporting emerging pervasive applications will constitute a significant part of future mobile technologies by providing highly proactive services requiring continuous monitoring of user related contexts. However, a major challenge standing up to these sensor-rich smart devices is the limited computational, storage and energy resources. Table VIII summarizes awaiting challenges in design process of context-awareness in mobile sensing.

In the following, we identify some interesting research opportunities for future context-aware development.

#### A. Energy Awareness

Because mobile devices operate on a finite supply of energy contained in their batteries, energy awareness is one of the key resource management issues in mobile sensing. Specifically, continuously capturing user context through sensors imposes heavy workload in physical and computational capacity aspects of the device working process, and it drains the battery power rapidly.

To understand this issue better, an application example [171] can be examined. Accordingly, the accelerometer sensor built-in HTC Touch Pro is employed at a fixed sampling frequency. When the phone samples the accelerometer, overall power consumption on device increases by 370 mW; whereas, according to the data sheet of the accelerometer, it should consume less than 1 mW when active. Even if the accelerometer itself wastes very little power to operate its functionality, the phone with its main processor and other hardware components causes much more power consumption during the operation to accomplish a

TABLE VIII: Awaiting challenges in design of context-awareness in mobile sensing

Research Area	Challenges	
Energy Awareness	<ul style="list-style-type: none"> <li>· creation of energy profiles</li> <li>· radio optimization</li> <li>· energy-efficient routing</li> <li>· battery characterization</li> </ul>	<ul style="list-style-type: none"> <li>· energy estimation</li> <li>· data reduction</li> <li>· sensing scheduling</li> </ul>
Sensing Management	<ul style="list-style-type: none"> <li>· dynamic sensor selection</li> <li>· opportunistic workload division</li> </ul>	<ul style="list-style-type: none"> <li>· adaptive sampling</li> <li>· optimal sensing</li> </ul>
Battery Behavior	<ul style="list-style-type: none"> <li>· non-linearity</li> <li>· effects of usage patterns</li> </ul>	<ul style="list-style-type: none"> <li>· estimation of energy delivery</li> <li>· battery discharge profiles</li> </ul>
Data Acquisition	<ul style="list-style-type: none"> <li>· data calibration</li> <li>· distortion, noise</li> </ul>	<ul style="list-style-type: none"> <li>· orientation change in device</li> <li>· device placement</li> </ul>
Context Inference	<ul style="list-style-type: none"> <li>· learning paradigm</li> <li>· online processing</li> </ul>	<ul style="list-style-type: none"> <li>· computational complexity</li> <li>· redundancy check</li> </ul>
Framework Design	<ul style="list-style-type: none"> <li>· generalization</li> <li>· adaptability</li> <li>· estimation/prediction</li> <li>· robust processing</li> </ul>	<ul style="list-style-type: none"> <li>· inhomogeneous physical world</li> <li>· tradeoff handling</li> <li>· time-variant sensing</li> <li>· optimization in sensor sensing</li> </ul>
Middleware Design	<ul style="list-style-type: none"> <li>· collection of async. heterogeneous context</li> <li>· interoperability</li> <li>· creation of an abstract layer</li> <li>· generic infrastructure, standardization</li> <li>· fault tolerance</li> </ul>	<ul style="list-style-type: none"> <li>· full transparency</li> <li>· scalability</li> <li>· decentralization</li> <li>· dynamic adaption, auto/self configuration</li> <li>· smartness</li> </ul>
Mobile Cloud	<ul style="list-style-type: none"> <li>· transparently partitioning data, offloading</li> <li>· adaptability, self-awareness</li> <li>· scalability</li> <li>· frequent network disconnections</li> <li>· augmentation process</li> </ul>	<ul style="list-style-type: none"> <li>· reconfigurability</li> <li>· resource scarcity</li> <li>· mobility</li> <li>· fault tolerance</li> <li>· resource optimization, inter-context relations</li> </ul>
General	<ul style="list-style-type: none"> <li>· limited power, bandwidth and storage</li> <li>· complex device architectures</li> </ul>	<ul style="list-style-type: none"> <li>· richness in context sources</li> <li>· security, privacy and trust issues</li> </ul>

contextual sensory data extraction. Another example provided in [172] reports that today’s smart devices are not durable to employ all sensors at the same time by giving an example of Samsung Galaxy SIII smartphone with a fully-charged battery. It is experimentally examined that the smartphone consumes 805 mW in idle mode, whereas the same device puts extra 573 mW power consumption while employing GPS connectivity. The experiment shows that constant sensor usage almost leads to reduce battery life by half. Given examples conclude that any sensor employment in mobile devices would draw more current from the device battery than it happens during a regular device run. Therefore, the mobile device battery will not last a long time to support device operations.

Energy awareness can be integrated by creating energy profiles and energy estimation to present maps of power usage allowing the analysis of power draining caused by physical sensor hardware, computations in coding, and transmission of data. Using this analysis and having more statistics on

behavioral characterization of mobile device infrastructure, context-aware applications can benefit more efficient process environment. Also, user intervene is important for securing energy awareness. In order to make power efficient decisions, users need to understand when power drain mostly occurs on their mobile devices.

On the other hand, increasing usage of mobile device application features along with network connectivity all the time will force the need for enhancing energy efficiency. For instance, constant connectivity in social networking platforms require continuous sensing that affects on the device battery life. Cellular and Wi-Fi are the most widely used wireless transmission technologies. Due to the increase in mobile data traffic and ever-demanding popularity of mobile applications, energy consumption on wireless mobile data transmission/communications grows rapidly. Especially, energy cost of wireless networking operations in addition to the expectations of future context-aware applications requiring

more use of multimedia, image and video rendering, data compressions/decompressions, complex web service connections will be extremely high. Therefore, there are some actions needed to be taken to decrease energy consumption and to prolong battery lifetime as in the followings:

- *Radio optimization* could reduce energy dissipation due to wireless communications by enhancing some radio parameters such coding, modulation schemes, power transmission and antenna direction.
- *Data reduction* would be a solution to reduce the amount of data to be processed or transferred. Methods such as adaptive sampling, network coding and data compression could help removing unnecessary information in sensing task.
- *Energy-efficient routing* aims at ensuring connectivity and coverage, and exploiting redundancy in topology control protocols by dynamically adapting network settings with respect to application needs.
- *Sensing scheduling, battery characterization and energy modeling* are examined in detail in upcoming subsections.

### B. Adaptive and Opportunistic Sensory Sampling

To address power efficiency in context-awareness, efficient sensor management systems infusing low level sensory operations need to be considered. An example method could be illustrated in Fig. 7. The first stage starts with dynamically selecting a sufficient number of sensors [76], [77], called “Dynamic Sensor Selection”, while a context-aware application is running. Thereby, sensors can be put in an order according to their power consumption levels and application relevance depending on an interested context. In addition, the best energy saving algorithm would be the one that manipulates the frequentness of sensory sampling intervals. In this regard, different duty cycling approaches would be the next stage of the ladder by tuning the wave form to power a sensor for a desired power efficiency. Besides that, adaptively changing sensor sampling periods can also be the final stage to achieve a certain level of power efficiency [79]. By adjusting sampling periods in sensory operations as needed, the total number of sampling occurrences either increases or decreases. As a result, relevant power consumption will adjust accordingly.

Above of all, an adaptive sensor management mechanism/system to assign a mixture pair of duty cycles and sampling periods simultaneously would be a cure of consuming less power while running context-aware applications accurately enough [173]. However, intervening sensory operations to achieve power efficiency may jeopardize the accuracy of context-aware services, and thereby it creates a tradeoff between power consumption caused and accuracy provided by these services.

In this sense, optimization of physical sensor hardware operations during data acquisition needs to be created in a collective way by employing sufficient number of sensor sets to characterize contextual information, creating work division among chosen sensors, scheduling time-variant duty cycles and setting adaptively changing sampling periods.

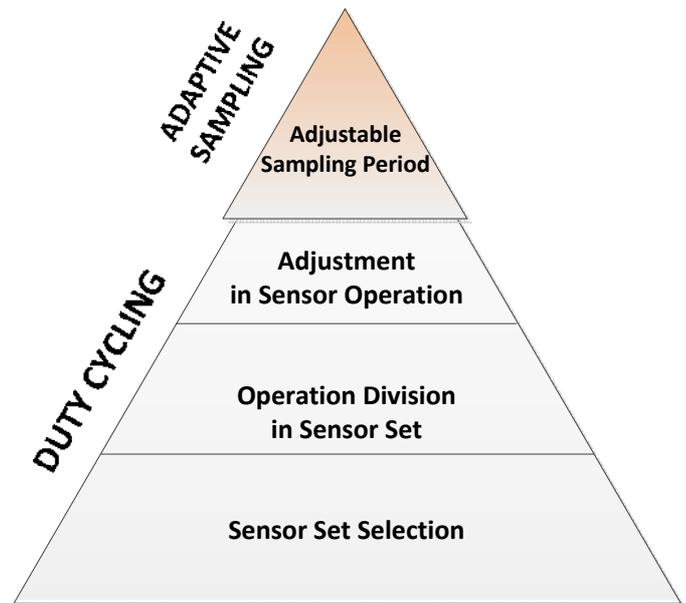


Fig. 7: A power efficient sensor management system for future context-aware applications.

### C. Energy Estimation and Modeling

The use of smart devices is constrained by limited battery. The slow growth in energy densities of battery technologies compared with the increasing computing power requirement and hardware capabilities is now driving the need for accurately modeling power consumption profiles. With an high accurate energy model on device performance, energy efficient applications or running operating systems can be designed. However, the diversity in architectural designs within mobile devices and their components along with the differences in usage patterns present a challenge to profile energy consumption. To do this, we need to explicitly consider the impact of different usage patterns, and their relationships with the projected effect on power consumptions.

To find the relationship between user activity and energy cost, special applications are developed to gather data on user behavior by tracing usage pattern on device. Traditional methods [174] use external equipments such as power meters to model energy estimation based on measurements of device operation in different activity modes. On the contrary, recent studies remove the necessity of using extra equipments by runtime monitoring to track key operating system parameters and hardware components. They first determine a measurable power cost of using a specific device component, and characterizes its effect on battery drain over a unit of time. Then, they collect power consumption related data through use of applications, and broke this data down onto premeasured component base statistics collected in applications in order to create a proper energy model. The final step aggregates all data collected by use of different applications and components, and generates power cost coefficients to anticipate online power consumption or battery drain for future smartphone operations. There exist many tools to measure energy consumption of mobile software entities. PowerBooster [175] constructs an

automated power model technique that uses built-in measurement sensors and knowledge of battery discharge status to monitor power consumption on individual device components. The study also provides an online smartphone application called PowerTutor by associating with proposed technique to estimate system-level power consumption, and then inform software developers and end-users power-efficiency-wise application use. Another study in [176] uses an application tool called Carat that explores the energy drainage behavior on mobile applications. The study classifies applications as either having an energy bug, or being an energy hog. Informally, an application is an energy hog when the application drains the battery much faster than its average use. Whereas, an application has an energy bug when some running instances of the application drain the battery much faster than other instances of the same application. Trepan Profiler by Qualcomm is a diagnostic tool that measures mobile phone system performance and power consumption. It is capable of system and application level of energy profiling that might helpful to developers to be aware of power optimization in application developments.

However, these studies lack of some drawbacks that affect on accuracy of energy estimations. CPU time is mostly used as a proxy for energy, whereas mobile devices may interact with multiple hardware components at the same time for a specific application, and it causes a very variant voltage discharge on the device battery. Also, background running applications or services could consume a significant amount of energy that could distort the energy attributed to the application and create false truthfulness in energy estimation process; therefore, estimation techniques require carefully fine-tuned software environments. Moreover, power modeling tools like Trepan Profiler uses information obtained from internal device components to perform the battery level measurements. However, many mobile devices may not include components inside to support this feature, and power modeling lacks from sufficient measurements on power management. Finally, running energy modeling application tools itself requires involved process and algorithms to retrieve accurate measurements with a very low error margin.

#### D. Battery Discharge Modeling

Energy modeling methods that involve bypassing the device battery characteristics would not be reliable. Even though some technological advances has be made to improve density in battery capacity and charging cycles, it is not suitable to be used in the end products [177]. Thereby, the modeling of battery's non-linearities [178], [179], and understanding the correlation between usage patterns and battery depletion will lead to successful discovery of optimal energy reduction strategies that will eventually help maximize the wasted energy consumption to improve the QoS of context-aware applications. In this regard, the topics such as extension of battery lifetimes, estimation of energy delivery or battery discharge, and optimal energy management have drawn much research interest in mobile computing. The examination of non-linear battery behaviors becomes crucial in terms of creating optimal

sensor management systems. Correspondingly, battery lifetime mostly depends on energy consumption rate, discharge profile, i.e., usage pattern, and battery non-linearities. At the high energy consumption rate, the effective residual battery capacity degrades and results in having a shorter battery lifetime. However, any precautionary change in the usage pattern could extend the battery lifetime. More importantly, the physical non-linearities in the batteries could recover the lost capacity while energy consumption decreases. Accurately estimating the remaining battery capacity [180] and reporting of the battery state-of-charge becomes a difficult task due to the nonlinear battery behavior, and also the time-varying nature of mobile device operation that would cause different amount of power need at any time.

The future research trend should investigate mobile device based battery behavior with respect to variant sensory operations in smart devices. Thereby, the linkage between battery discharge and power consumption caused by the sensors can be analyzed, and most importantly, a fine power efficiency can be objected to achieve while satisfying continuity of mobile device based context-aware services.

#### E. Data Calibration and Robustness

Because of mobility, the outputs of inertia or ambient sensors are prone to having false truthfulness such as the quality of sound and picture samples. More significantly, constant sensor displacement in motion-based activity recognition systems is a serious disadvantage that causes decrease in application accuracy. Any change in orientation of the mobile device, such as rotation, is an important design drawback for the most of classification algorithms, especially for those which solely rely on exploiting feature vectors through a specific axis information. In case where the sensor is not placed fixed, suppose an accelerometer is used in a HAR based application, it would produce some distortion over acceleration axes. Upward or downward position change of the device causes x-axis flipped to y-axis or vice versa. Therefore, an adaptive context inference scheme needs to be employed to detect the sensor position/orientation or device position (e.g., in a purse, or a packet) in order to satisfy the robustness toward various practical usage conditions. This scheme also should select the most relevant context inference strategy dynamically. In addition, orientation-independent features should be considered during the context inference process. Finally, to reduce data redundancy, noise, and jitter in instantaneous sensor readings, a calibration and normalization process needs to be applied on sensory sampling operations to find proper filtering, sampling offsets and scaling factors on sensory data streams.

#### F. Efficient Context Inference Algorithms

Besides power consumption and battery density considerations, analysis and inference process of contextual sensory data has many drawbacks. Many studies can be found in which a framework is proposed to capture and evaluate sensory data. Most studies rely on recognition of user activities and definition of common user behaviors. The applied methods

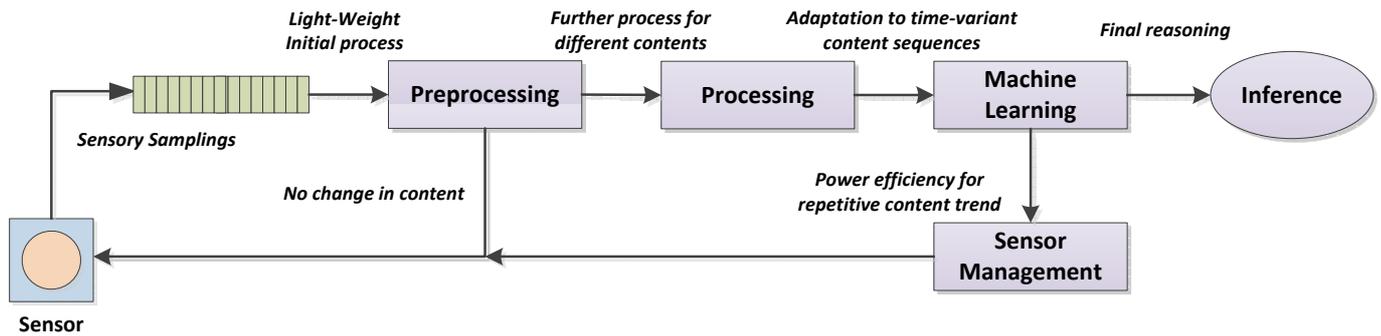


Fig. 8: The futuristic context-aware framework designs seek for finding a fine balance between power efficiency and application accuracy, managing adaptability to time-variant user preferences and behaviors, and applies lesser computational processing workload.

in relevant studies are based on using statistical models, predefined feature extraction and classification algorithms. The similar issue is valid with studies done for mobile event/user tracking and localization problems. The studies use clustering algorithms to derive the most significant user based location traces, and specify consecutive location points aggregated within a cluster radius through a certain of time period. Finally, they use map matching and reverse geo-coding to create geographical dictionaries for users, thereby this information could be used for predicting user mobility or trajectory. However, none of these studies engage themselves to model a common framework in order to construct a base structure for future context-aware applications. They would rather have canalized solutions to solve their own unique applications instead of a generalized approach. Therefore, these studies mostly focus on a specific sensor to discover possible target applications in order to exploit the contextual data.

Another important system attribute to consider is to prevent the use of supervised learning strategy. Most systems take predefined models or classifiers where a training data is obtained through several repetition of a similar experiment setup. However, it yields to have a large amount of data in return to process, and makes the subsequent analysis to be carried out offline. Especially, obtaining training data classes to feed statistical models, classification or machine learning algorithms in a supervised learning strategy is an expensive real-time operation for smart devices, and it is impractical when considering the computational manner. That is, acquiring and analyzing of data, resource management by storing training samples, scalability problem by labeling data, and bandwidth problem by exchanging large amount of information. Therefore, the utilization of sensors must be lightweight and unobtrusive, and also the applied classification/clustering and machine-learning algorithms must be applied without computationally expensive, human-intervening offline methods. For example, the study in [181] provides a light-weight, unsupervised and online classification method to detect HAR based user context by collecting data from smartphone accelerometer sensor. The solution applies a sufficient number of signal processing and statistical techniques (*light-weightiness*) without receiving any *a priori* information related to user state classes, and setting any predefined/fixed thresholds over any specific acceleration

spaces (*unsupervised learning*) in order to differentiate user activities. As a result, the similar type of implementation mentality needs to be applied into context-aware applications in order to meet requirements set by resource-constrained mobile platforms.

#### G. Generic Context-Aware Framework Designs

A generic framework that fulfills requirements set by all types of context-aware applications has not yet clearly identified. The problem often comes from the difficulty in building a reliable data set to represent a specific context interest, since the obtained sensory data can vary significantly under different circumstances (e.g., human speech with a variant background noise or placement of the mobile device). As a result, classifiers would not be practical toward varying sensing conditions, and eventually it would perform poorly. Hence, the adaptation problem becomes an important system attribute to consider. This issue even turns into a severe problem in case of resulting in different inference assumptions by multiple co-located mobile devices on a similar sensing event where a participatory sensing application takes place. One solution is to take advantage of cloud computing technologies, enabling to share information and ensemble situational resources among co-located mobile devices.

In this research content, the studies in [116], [117] enlighten future research directions by presenting a generic system framework within the area of mobile device based context-aware applications. The research focuses on the inhomogeneity and the user profile adaptability while examining the trade-off between accuracy in contextual inference through sensory data and required power consumption during data processing. The inhomogeneity is characterized by time-variant system parameters, and the user profile adaptability challenge is modeled using the convergence of entropy rate in conjunction with the inhomogeneity. Accordingly, an implemented smartphone application demonstrates how entropy rate converges in response to distinctive time-variant user profiles under different sensory sampling operations. In addition, user related context is either recognized in the presence of sensory data or estimated while various energy saving strategies are being applied. During the recognition process, a sufficient number of signal processing

techniques is applied to find out the best context-exploiting methods on the sensory signal instead of applying computationally harsh pattern recognition methods. Moreover, power efficiency is taken at the low-level sensory operations rather than just applying less complexity in computations or changing transferring methods of data packets in the application.

Futuristic context-aware framework design, as illustrated in Fig. 8, must address to provide a generic solution that aims at achieving a fine balance between power efficiency and application accuracy. Accordingly, the framework initially acquires sensory observations and make them undergone into a preprocessing structure. This structure basically filters out required information from raw sensory data, and applies basic signal processing so that it may help decrease redundant computational operations by not letting go further processing in case where there is not much change in desired context in the sensory data. Processing structure is reserved for context inference. A required context inference algorithm could differ according to the interested context through a specific sensor. Machine learning is applied later to obtain a better realization in context-awareness in order to create adaptability to time-variant user preferences and behaviors, estimate missing context inferences in presence of idle sensory operations, and also preserve the functionality against aperiodically received sensory observations. Most importantly, machine learning structure regulates sensor management by estimating user preference trends, and opportunistically finding out stable moments. Thereby, sensor management structure could use this information to figure optimal sensing policies, and change sensor sampling settings so as to power efficiency could be achieved while satisfying the accuracy of context-aware application services. In addition, sensor management structure needs to be aware of co-existence of similar context providing sensors, and create a coordination among them with respect to choices by processing structure.

#### H. Standard Context-Aware Middleware Solutions

A standard context-aware middleware needs to cover all types of application settings, provides an adaptation toward changing context, and acquire a collection of asynchronous heterogeneous context to create different abstract entities. It also needs to succeed to have a full transparency, that eliminates direct involvement of an application into context modeling process. In this direction, gathering diverse and asynchronous information, and presenting it to the application would be the future work in mobile computing middleware research.

Futuristic context-aware middleware also must deal with

- *interoperability* challenge that expects to collaborate among heterogeneous context providing devices as well as awareness of resource state at different layers and abstraction levels to optimize network-layer protocols.
- increasing amount in number of context providers, e.g., today's built-in sensors or future products, that bring about *scalability* problem. This results in many devices to interact on a small-scale place. Therefore, efficient information management and exchange is required.

- creation of an *abstraction layer*, not a centralized coordination, because context-aware middleware handles operations through various level of heterogeneous input and output hardware devices, software interfaces and protocols, data streams and transfer. A central entity cannot guarantee of managing all various system level activities, especially while having spontaneous inter-operation and de-coupled coordination among connecting devices.
- *generic infrastructure* where any context resources can be manageable, and sudden interactions of context providers are handled, such as plug and play of physical sensors. In addition, since meta-contexts like social contexts could be incorporated with physical contexts, a notable effort is needed toward this type of generic integration into a wide-scale opportunistic future ubiquitous computing.
- *standardization* for different types of domains and applications to address common requirements. This might be impossible to have but at least standardization needs to be specified for a certain domain.
- *dynamic adaptation* and *auto-configuration* support to context changes or context lost due to the underlying network is transient and fragile to have disconnections, as well as due to physical hardware that frequently join and leave context inference process. Also, auto-configuration means integration of different protocols, algorithms, and solutions on the fly depending on application requirements, mobile device settings, and available network infrastructures.
- *asynchronous* communication capability, especially in a mobile cloud use, to delegate and monitor data intensive and time consuming tasks.
- *fault tolerance* that determines the reliability and safety of ongoing processes. It is triggered by incomplete, interrupted or delayed tasks, e.g., a sensor operation to infer a context.
- *smartness* that help acting as an autonomous and intelligent delegate by being robust to mobility and context prediction while reasoning context. Especially, a fair and objective dissemination, classification and elimination is needed to concurrent or similar context representation and management.
- *resource management* to effectively support multiple applications at a time while allocating device components/resources and making them work coordinated at accommodating concurrent application requests.

#### I. Mobile Cloud Computing

Cloud computing, i.e., on demand computing, refers to the applications delivered as services through the Internet, and aggregation of computing as a utility together with required hardware and software provides those services [182]. Enabling cloud computing within resource constraint mobile devices along with context-aware sensing have led to the creation of a new paradigm called mobile cloud computing. Thereby, it is intended that computationally expensive and more resource demanding jobs are transparently partitioned and offloaded to the cloud in order to remove the obstacle that mobile devices

encounters in terms of limited battery and processing power, limited connectivity, and low memory storage. In addition, another significant merit in the creation of a mobile cloud is that mobile devices themselves are not only contextual resource providers to the cloud but also could connect a mobile peer-to-peer network by collecting resources of the various providers in a local vicinity. The idea of building a shared pool of configurable computing resources benefits from provisioned services with a minimal management effort and redeeming the disadvantages of having limited connectivity to remote servers and limited power to burst long range communications.

Despite increasing trend to mobile cloud computing, there are certain requirements such as adaptability, resource scarcity, scalability, mobility, frequent network disconnections, and self-awareness need to be met. Therefore, a mobile computing cloud has to be aware of resource scarcity, availability and quality of its service to enable diverse mobile computing entities located in an efficient scale by considering some aspects of mobility, low connectivity and finite power source, and finally dynamically engage with these entities depending on their requirements and workloads. In addition, fault-tolerance becomes very important due to mobility, and its effects on network signal losses. Most importantly, adaptability while employing mobile cloud based computing resource is not an easy task. Especially, augmentation process [183] and optimum selection of resource types stand as a big challenge to enhance and optimize computing capabilities of mobile devices that perform context-aware applications on a resource constraint platform. This challenge would be more difficult where plug and play smart objects are deployed in an environment that are willing to communicate other smart objects around through an inter-operable backbone. Moreover, mobile cloud computing will have a highly heterogeneous networks in terms of wireless connectivity. Different mobile nodes using variant radio access technologies will bring about some important issues such as low bandwidth, service congestion, network failures and latency in data transfer. To overcome these shortcomings, widely used fourth generation networks have emerged as promising technologies by increasing bandwidth capacity for mobile subscribers. Also, Femtocells [184] could create a connection to the cloud for short-range areas by enabling mobile subscribers to gain access to the network that results in a highly economical network structure to use only sufficient resources in meeting user demands at any given access point without impacting the large-scale network immediately.

### J. Security, Privacy and Trust

Autonomous operation of real-life related ubiquitous objects creates a huge potential to interfere people's trust, security and privacy. Especially, context-aware based applications could give a feeling of users are being monitored all the time. This type of surveillance needs to be addressed as one of main functions during design process of context-aware middleware to manage and protect of security and confidentiality. Together with the deployment of IoT and mobile cloud, the scale of interactions, complexity, mobility and heterogeneity will grow drastically, and that makes the cloud of things hard

to be controlled and be open to security threats such as application, web, network or physical based attacks. For this purpose, innovative encryption, cryptography and enforcement on data stream access control technologies need to be used for securing offloading data management and exchange, and detecting/removing malware. However, this will impose to utilize computational expensive and energy harvesting algorithms. On the other hand, for privacy protection, perception of the general public is still immature. Since mobile environments are dynamic and unpredictable, it is important for mobile users to have transparency and choices in order to control over their personal information, and also to have knowledge of data collection being operated by authorized services by authorized service providers. Moreover, privacy-preserving context-aware technologies are still an open subject for resource-restricted devices on what measures that privacy is secured. In this sense, the issue with contextual data ownership in a collaborative cloud networks along with data anonymity suffers from privacy of user, and needs to establish trust and authentication. Trust is only established if security policies are modeled to regulate accesses to resources and credentials.

### V. CONCLUSION

This survey provides an overview of the context-awareness in ubiquitous/mobile sensing. It provides a comprehensive introduction to the definition, representation and inference of context. It also points out the importance of context-aware middleware design, and the challenges that are faced during design process and system integration. Moreover, the paper categorizes and gives an inside-out look into context-aware applications depending on the interested context and identifies opportunities in this research area. Looking into the future, we tend to believe that with the evolution of smartphones, software developers have empowered to create context-aware applications for recognizing human-centric or community based innovative social and cognitive activities in any situation and at any location. This leads to the exciting vision of forming a society of "Internet-of-Things". With the highlights of this survey, we intend to enlighten current trends and future directions to enhance the existing tradeoffs and drawbacks in mobile sensing while context is being inferred under the intrinsic constraints of mobile devices and around the emerging concepts in context-aware framework technologies.

### REFERENCES

- [1] A. Wood, J. Stankovic, G. Virone, L. Selavo, Z. He, Q. Cao, T. Doan, Y. Wu, L. Fang, and R. Stoleru, "Context-aware wireless sensor networks for assisted living and residential monitoring," *IEEE Networks*, vol. 22, no. 4, pp. 26–33, july-aug. 2008.
- [2] K. K. Rachuri, M. Musolesi, C. Mascolo, P. J. Rentfrow, C. Longworth, and A. Aucinas, "Emotionsense: a mobile phones based adaptive platform for experimental social psychology research," in *ACM Ubi-comp'10*, pp. 281–290.
- [3] M. Weiser, "The computer for the 21st century," *Scientific american*, vol. 265, no. 3, pp. 94–104, 1991.
- [4] B. N. Schilit and M. M. Theimer, "Disseminating active map information to mobile hosts," *IEEE Networks*, vol. 8, no. 5, pp. 22–32, 1994.
- [5] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggle, "Towards a better understanding of context and context-awareness," in *Handheld and ubiquitous computing*. Springer, 1999, pp. 304–307.

- [6] A. K. Dey, "Understanding and using context," *Personal and ubiquitous computing*, vol. 5, no. 1, pp. 4–7, 2001.
- [7] P. Prekop and M. Burnett, "Activities, context and ubiquitous computing," *Computer Communications*, vol. 26, no. 11, pp. 1168–1176, 2003.
- [8] K. Henriksen, J. Indulska, and A. Rakotonirainy, "Modeling context information in pervasive computing systems," in *Pervasive Computing*. Springer, 2002, pp. 167–180.
- [9] T. Rodden, K. Cheverst, K. Davies, and A. Dix, "Exploiting context in hci design for mobile systems," in *Workshop on human computer interaction with mobile devices*. Citeseer, 1998, pp. 21–22.
- [10] C. D. Kidd, R. Orr, G. D. Abowd, C. G. Atkeson, I. A. Essa, B. MacIntyre, E. Mynatt, T. E. Starner, and W. Newstetter, "The aware home: A living laboratory for ubiquitous computing research," in *Cooperative buildings. Integrating information, organizations, and architecture*. Springer, 1999, pp. 191–198.
- [11] I. A. Essa, "Ubiquitous sensing for smart and aware environments," *IEEE Personal Communications*, vol. 7, no. 5, pp. 47–49, 2000.
- [12] P. Brown, J. Bovey, and X. Chen, "Context-aware applications: from the laboratory to the marketplace," *IEEE Personal Communications*, vol. 4, no. 5, pp. 58–64, Oct 1997.
- [13] P. TalebiFard and V. C. M. Leung, "Context-aware dissemination of information and services in heterogeneous network environments," pp. 775–787, Dec. 2014.
- [14] K. Shafiee, V. C. M. Leung, and R. Sengupta, "Request-adaptive packet dissemination for context-aware services in vehicular networks," in *IEEE Vehicular Technology Conference*, 2012, pp. 1–5.
- [15] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.
- [16] J. Zheng, D. Simplot-Ryl, C. Bisdikian, and H. Mouftah, "The internet of things," *IEEE Commun. Mag.*, vol. 49, no. 11, pp. 30–31, nov 2011.
- [17] N. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. Campbell, "A survey of mobile phone sensing," *IEEE Comm. Mag.*, vol. 48, no. 9, pp. 140–150, 2010.
- [18] M. Baldauf, S. Dustdar, and F. Rosenberg, "A survey on context-aware systems," *Int'l Jour. of Ad Hoc and Ubiquitous Comput.*, vol. 2, no. 4, pp. 263–277, 2007.
- [19] G. Chen and D. Kotz, "A survey of context-aware mobile computing research," Hanover, NH, USA, Tech. Rep., 2000.
- [20] R. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Commun. Mag.*, vol. 49, no. 11, pp. 32–39, nov 2011.
- [21] C. Bettini, O. Brdiczka, K. Henriksen, J. Indulska, D. Nicklas, A. Ranganathan, and D. Riboni, "A survey of context modelling and reasoning techniques," *Pervasive and Mobile Comput.*, vol. 6, no. 2, pp. 161–180, 2010.
- [22] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the internet of things: A survey," *IEEE Comm. Surv. & Tut.*, vol. PP, no. 99, pp. 1–41, 2013.
- [23] K. Verbert, N. Manouselis, X. Ochoa, M. Wolpers, H. Drachler, I. Bosnic, and E. Duval, "Context-aware recommender systems for learning: A survey and future challenges," *IEEE Trans. on Learning Technologies*, vol. 5, no. 4, pp. 318–335, Oct 2012.
- [24] M. del Carmen Rodriguez-Hernandez and S. Ilarri, *Towards a Context-Aware Mobile Recommendation Architecture*, ser. Mobile Web Information Systems. Springer, 2014, vol. 8640.
- [25] P. Makris, D. Skoutas, and C. Skianis, "A survey on context-aware mobile and wireless networking: On networking and computing environments' integration," *IEEE Communications Surveys Tutorials*, vol. 15, no. 1, pp. 362–386, First 2013.
- [26] M. Sama, D. S. Rosenblum, Z. Wang, and S. Elbaum, "Model-based fault detection in context-aware adaptive applications," in *ACM SIGSOFT'08*, pp. 261–271.
- [27] R. Tan, J. Gu, Z. Zhong, and P. Chen, "Metadata management of context resources in context-aware middleware system," *Web Info. Sys. and Mining*, pp. 350–357, 2012.
- [28] T. Strang and C. Linnhoff-Popien, "A context modeling survey," in *Workshop Proceedings*, 2004.
- [29] M. Baldauf, S. Dustdar, and F. Rosenberg, "A survey on context-aware sys." *Int'l Jour. of Ad Hoc and Ubiquitous Comput.*, vol. 2, no. 4, pp. 263–277, 2007.
- [30] M. Román, C. Hess, R. Cerqueira, A. Ranganathan, R. H. Campbell, and K. Nahrstedt, "A middleware infrastructure for active spaces," *IEEE Pervasive Computing*, vol. 1, no. 4, pp. 74–83, 2002.
- [31] M. Knappmeyer, S. Kiani, C. Fra, B. Moltchanov, and N. Baker, "Contextml: A light-weight context representation and context management schema," in *IEEE Int'l Symp. Wireless Pervasive Computing (ISWPC)*, May 2010, pp. 367–372.
- [32] K. Henriksen, J. Indulska, and A. Rakotonirainy, "Generating context management infrastructure from high-level context models," in *Int'l Conf. on Mobile Data Management (MDM) - Industrial Track*, 2003, pp. 1–6.
- [33] T. Halpin, "Object-role modeling: principles and benefits," *International Journal of Information System Modeling and Design (IJISMD)*, vol. 1, no. 1, pp. 33–57, 2010.
- [34] J. Han, E. Haihong, G. Le, and J. Du, "Survey on nosql database," in *Int'l Conf. on Pervasive Computing and Applications*, Oct 2011, pp. 363–366.
- [35] C. Bolchini, C. A. Curino, E. Quintarelli, F. A. Schreiber, and L. Tanca, "A data-oriented survey of context models," *SIGMOD Rec.*, vol. 36, no. 4, pp. 19–26, Dec. 2007.
- [36] R. Krummenacher and T. Strang, "Ontology-based context modeling," in *Proc. of Context-Aware Proactive Systems*, 2007, p. 22.
- [37] D. Ejigu, M. Scuturici, and L. Brunie, "An ontology-based approach to context modeling and reasoning in pervasive computing," in *IEEE Int'l Conf. on Pervasive Computing and Communications Workshops*, March 2007, pp. 14–19.
- [38] D. Allemang and J. Hendler, *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2008.
- [39] J. Forstadius, O. Lassila, and T. Seppanen, "Rdf-based model for context-aware reasoning in rich service environment," in *IEEE Int'l Conf. Pervasive Computing and Communications Workshops*. IEEE, 2005, pp. 15–19.
- [40] R. Hoekstra, "Representing social reality in owl 2," in *In OWLED 2010, 7th International Workshop*, 2010.
- [41] X. Wang, D. Q. Zhang, T. Gu, and H. Pung, "Ontology based context modeling and reasoning using owl," in *IEEE Conf. on Pervasive Computing and Communications Workshops*, March 2004, pp. 18–22.
- [42] D. Bandyopadhyay and J. Sen, "Internet of things: Applications and challenges in technology and standardization," *Wireless Pers. Comm.*, vol. 58, no. 1, pp. 49–69, 2011.
- [43] M. Kim and D. Kotz, "Extracting a mobility model from real user traces," in *IEEE INFOCOM'06*.
- [44] K. E. Kjær, "A survey of context-aware middleware," in *Int'l Multi-Conf. on IASTED: Software Eng.*, 2007, pp. 148–155.
- [45] D. Salber, A. K. Dey, and G. D. Abowd, "The context toolkit: aiding the development of context-enabled applications," in *ACM SIGCHI'99*, 1999, pp. 434–441.
- [46] D. Garlan, D. Siewiorek, A. Smailagic, and P. Steenkiste, "Project aura: toward distraction-free pervasive computing," *IEEE Pervasive Computing*, vol. 1, no. 2, pp. 22–31, April 2002.
- [47] L. Capra, W. Emmerich, and C. Mascolo, "Carisma: context-aware reflective middleware system for mobile applications," *IEEE Trans. on Software Engineering*, vol. 29, no. 10, pp. 929–945, Oct 2003.
- [48] M. Roman, C. Hess, R. Cerqueira, A. Ranganathan, R. Campbell, and K. Nahrstedt, "A middleware infrastructure for active spaces," *IEEE Pervasive Computing*, vol. 1, no. 4, pp. 74–83, Oct 2002.
- [49] T. Gu, H. K. Pung, and D. Q. Zhang, "A service oriented middleware for building context aware services," *Journal of Network and Computer Applications*, vol. 28, no. 1, pp. 1–18, 2005.
- [50] P. Bellavista, R. Montanari, and D. Tibaldi, "Cosmos: A context-centric access control middleware for mobile environments," in *Mobile Agents for Telecommunication Applications*. Springer Berlin Heidelberg, 2003, vol. 2881, pp. 77–88.
- [51] H. Chen, T. Finin, A. Joshi, L. Kagal, F. Perich, and D. Chakraborty, "Intelligent agents meet the semantic web in smart spaces," *IEEE Internet Computing*, vol. 8, no. 6, pp. 69–79, Nov 2004.
- [52] A. Badii, M. Crouch, and C. Lallah, "A context-awareness framework for intelligent networked embedded systems," in *Advances in Human-Oriented and Personalized Mechanisms, Technologies and Services (CENTRIC), 2010 Third International Conference on*, Aug 2010, pp. 105–110.
- [53] D. Riboni and C. Bettini, "Cosar: Hybrid reasoning for context-aware activity recognition," *Personal Ubiquitous Comput.*, vol. 15, no. 3, pp. 271–289, Mar. 2011.
- [54] A. Katasonov, O. Kaykova, O. Khriyenko, S. Nikitin, and V. Y. Terziyan, "Smart semantic middleware for the internet of things," in *ICINCO-ICSO*. INSTICC Press, 2008, pp. 169–178.
- [55] P. Fahy and S. Clarke, "Cass a middleware for mobile context-aware applications," in *Workshop on Context Awareness, MobiSys*, 2004.

- [56] O. Davidyuk, J. Rieki, V. mikko Rautio, and J. Sun, "Context-aware middleware for mobile multimedia applications," in *MUM 04: Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia*. ACM Press, 2004, pp. 213–220.
- [57] T. Luckenbach, P. Gober, S. Arbanowski, A. Kotsopoulos, and K. Kim, "Tinyrest - a protocol for integrating sensor networks into the internet," in *Proc. of REALWSN*, 2005.
- [58] A. Corradi, M. Fanelli, and L. Foschini, "Implementing a scalable context-aware middleware," in *IEEE Symp. on Computers and Communications*, July 2009, pp. 868–874.
- [59] W. Wibisono, A. Zaslavsky, and S. Ling, "Comihoc: A middleware framework for context management in manet environment," in *IEEE Int'l Conf. on Advanced Information Networking and Applications (AINA)*, April 2010, pp. 620–627.
- [60] B. Guo, L. Sun, and D. Zhang, "The architecture design of a cross-domain context management system," in *IEEE Int'l Conf. on Pervasive Computing and Communications Workshops*, March 2010, pp. 499–504.
- [61] D. Chu, N. D. Lane, T. T.-T. Lai, C. Pang, X. Meng, Q. Guo, F. Li, and F. Zhao, "Balancing energy, latency and accuracy for mobile sensor data classification," in *ACM SenSys'11*, 2011, pp. 54–67.
- [62] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten, "The weka data mining software: an update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.
- [63] T. D. Nielsen and F. V. JENSEN, *Bayesian networks and decision graphs*. Springer, 2009.
- [64] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.
- [65] J.-L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains," *IEEE Trans. on speech and audio processing*, vol. 2, no. 2, pp. 291–298, 1994.
- [66] K. Weinberger, J. Blitzer, and L. Saul, "Distance metric learning for large margin nearest neighbor classification," *Advances in neural information processing systems*, vol. 18, p. 1473, 2006.
- [67] M. A. Hearst, S. Dumais, E. Osman, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their Applications*, vol. 13, no. 4, pp. 18–28, 1998.
- [68] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.
- [69] L. A. Zadeh, "Fuzzy logic, neural networks, and soft computing," *Communications of the ACM*, vol. 37, no. 3, pp. 77–84, 1994.
- [70] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl *et al.*, "Constrained k-means clustering with background knowledge," in *ICML*, vol. 1, 2001, pp. 577–584.
- [71] S. R. Eddy, "Hidden markov models," *Current opinion in structural biology*, vol. 6, no. 3, pp. 361–365, 1996.
- [72] G. Deodatis and M. Shinozuka, "Auto-regressive model for nonstationary stochastic processes," *Journal of Engineering Mechanics*, vol. 114, no. 11, pp. 1995–2012, 1988.
- [73] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2005.
- [74] C. J. Huberty, *Discriminant analysis*. Wiley Online Library, 2005.
- [75] J. H. Friedman, "Regularized discriminant analysis," *Journal of the American statistical association*, vol. 84, no. 405, pp. 165–175, 1989.
- [76] Y. Wang, J. Lin, M. Annaram, Q. A. Jacobson, J. Hong, B. Krishnamachari, and N. Sadeh, "A framework of energy efficient mobile sensing for automatic user state recognition," in *ACM MobiSys'09*, pp. 179–192.
- [77] S. Kang, J. Lee, H. Jang, H. Lee, Y. Lee, S. Park, T. Park, and J. Song, "Seemon: scalable and energy-efficient context monitoring framework for sensor-rich mobile environments," in *ACM MobiSys'08*, pp. 267–280.
- [78] M. Quwaider and S. Biswas, "Body posture identification using hidden markov model with a wearable sensor network," pp. 19:1–19:8.
- [79] K. K. Rachuri, M. Musolesi, and C. Mascolo, "Energy-accuracy trade-offs in querying sensor data for continuous sensing mobile systems," pp. 65–76, 2012.
- [80] M. Ermes, J. Parkka, and L. Cluitmans, "Advancing from offline to online activity recognition with wearable sensors," in *IEEE EMBS'08*, aug., pp. 4451–4454.
- [81] L. C. Jatoba, U. Grossmann, C. Kunze, J. Ottenbacher, and W. Stork, "Context-aware mobile health monitoring: Evaluation of different pattern recognition methods for classification of physical activity," in *IEEE EMBS'08*, aug., pp. 5250–5253.
- [82] E. Miluzzo, N. D. Lane, H. Lu, and A. T. Campbell, "Research in the app store era: Experiences from the cenceme app deployment on the iphone," 2010.
- [83] P. Zappi, C. Lombriser, T. Stiefmeier, E. Farella, D. Roggen, L. Benini, and G. Trister, "Activity recognition from on-body sensors: Accuracy-power trade-off by dynamic sensor selection," in *European Conf. on Wireless Sens. Netw.*, 2008, pp. 17–33.
- [84] Y.-P. Chen, J.-Y. Yang, S.-N. Liou, G.-Y. Lee, and J.-S. Wang, "Online classifier construction algorithm for human activity detection using a tri-axial accelerometer," *Appl. Math. and Comput.*, vol. 205, no. 2, pp. 849–860, 2008.
- [85] U. Maurer, A. Smailagic, D. Siewiorek, and M. Deisher, "Activity recognition and monitoring using multiple sensors on different body positions," in *Int'l Workshop on Wearable and Impl. Body Sens. Netw.*, april 2006, pp. 4 pp. –116.
- [86] C. Zhu and W. Sheng, "Human daily activity recognition in robot-assisted living using multi-sensor fusion," in *IEEE ICRA'09*, may, pp. 2154–2159.
- [87] M. Berchtold, M. Budde, D. Gordon, H. Schmidtke, and M. Beigl, "Actiserv: Activity recognition service for mobile phones," in *Wearable Comp. (ISWC)*, oct. 2010, pp. 1–8.
- [88] T. Kao, C. Lin, and J. Wang, "Development of a portable activity detector for daily activity recognition," in *IEEE ISIE'09*, pp. 115–120.
- [89] E. Miluzzo, C. T. Cornelius, A. Ramaswamy, T. Choudhury, Z. Liu, and A. T. Campbell, "Darwin phones: the evolution of sensing and inference on mobile phones," in *ACM MobiSys '10*, pp. 5–20.
- [90] D. Peebles, H. Lu, N. D. Lane, T. Choudhury, and A. T. Campbell, "Community-guided learning: Exploiting mobile sensor users to model human behavior." M. Fox and D. Poole, Eds. AAAI Press, 2010.
- [91] P. Siirtola and J. Rning, "Recognizing human activities user-independently on smartphones based on accelerometer data." *IJIMAI*, no. 5, pp. 38–45.
- [92] L. T. Vinh, S. Lee, H. X. Le, H. Q. Ngo, H. I. Kim, M. Han, and Y.-K. Lee, "Semi-markov conditional random fields for accelerometer-based activity recognition," *Appl. Intell.*, vol. 35, no. 2, pp. 226–241, Oct. 2011.
- [93] J. Lester, T. Choudhury, and G. Borriello, "A practical approach to recognizing physical activities," in *ACM Pervasive'06*, 2006, pp. 1–16.
- [94] Z.-Y. He and L.-W. Jin, "Activity recognition from acceleration data using ar model representation and svm," in *Int'l Conf. on Mach. Learn. and Cyber. '08*, vol. 4, july 2008, pp. 2245–2250.
- [95] D. Riboni and C. Bettini, "Cosar: hybrid reasoning for context-aware activity recognition," *Pers. Ubiquitous Comput.*, vol. 15, no. 3, pp. 271–289, Mar. 2011.
- [96] L. Sun, D. Zhang, B. Li, B. Guo, and S. Li, "Activity recognition on an accelerometer embedded mobile phone with varying positions and orientations," in *UIC'10*, pp. 548–562.
- [97] Z. He and L. Jin, "Activity recognition from acceleration data based on discrete cosine transform and svm," in *IEEE SMC'09*, oct. 2009, pp. 5041–5044.
- [98] Y. Hanai, J. Nishimura, and T. Kuroda, "Haar-like filtering for human activity recognition using 3d accelerometer," in *IEEE DSP/SPE'09*, jan. 2009, pp. 675–678.
- [99] D. Minnen, T. Westeyn, D. Ashbrook, P. Presti, and T. Starner, "Recognizing soldier activities in the field," in *Int'l Workshop on Wear. and Impl. Body Sens. Netw.* Springer, 2007, pp. 236–241.
- [100] E. M. Tapia, S. S. Intille, W. Haskell, K. Larson, J. Wright, A. King, and R. Friedman, "Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor," in *IEEE ISWC'07*, 2007, pp. 1–4.
- [101] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *SIGKDD Explor. Newsl.*, vol. 12, no. 2, pp. 74–82, Mar. 2011.
- [102] L. Bao and S. Intille, "Activity recognition from user-annotated acceleration data," in *Pervasive Computing*, ser. Lecture Notes in Computer Science, A. Ferscha and F. Mattern, Eds. Springer Berlin Heidelberg, 2004, vol. 3001, pp. 1–17.
- [103] X. Zhu, "Semi-supervised learning literature survey," 2006.
- [104] D. Siewiorek, A. Smailagic, J. Furukawa, A. Krause, N. Moraveji, K. Reiger, J. Shaffer, and F. L. Wong, "Sensay: A context-aware mobile phone," in *IEEE ISWC'03*, 2003, pp. 248–.
- [105] H. Lu, J. Yang, Z. Liu, N. D. Lane, T. Choudhury, and A. T. Campbell, "The jigsaw continuous sensing engine for mobile phone applications," in *ACM SenSys'10*, pp. 71–84.
- [106] Z. Zhuang, K.-H. Kim, and J. P. Singh, "Improving energy efficiency of location sensing on smartphones," in *ACM MobiSys'10*, pp. 315–330.

- [107] M. Shin, P. Tsang, D. Kotz, and C. Cornelius, "Deamon: Energy-efficient sensor monitoring," in *IEEE SECON'09*, pp. 1–9.
- [108] K.-H. Chang, M. Y. Chen, and J. Canny, "Tracking free-weight exercises," in *UbiComp 2007: Ubiquitous Computing*. Springer, 2007, pp. 19–37.
- [109] T. Huynh, M. Fritz, and B. Schiele, "Discovery of activity patterns using topic models," in *Proc. of Int'l Conf. on Ubiquitous computing*. ACM, 2008, pp. 10–19.
- [110] M. Stikic, D. Larlus, S. Ebert, and B. Schiele, "Weakly supervised recognition of daily life activities with wearable sensors," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 33, no. 12, pp. 2521–2537, Dec 2011.
- [111] Z. Yan, D. Chakraborty, A. Misra, H. Jeung, and K. Aberer, "Samplle: Detecting semantic indoor activities in practical settings using locomotive signatures," in *IEEE Int'l Symp. Wearable Computers (ISWC)*. IEEE, 2012, pp. 37–40.
- [112] M. Azizyan, I. Constandache, and R. Roy Choudhury, "Surroundsense: mobile phone localization via ambience fingerprinting," in *ACM MobiCom'09*, 2009, pp. 261–272.
- [113]
- [114] A. Papliatseyeu and O. Mayora, "Mobile habits: Inferring and predicting user activities with a location-aware smartphone," ser. *Advances in Soft Computing*, J. Corchado, D. Tapia, and J. Bravo, Eds. Springer Berlin Heidelberg, 2009, vol. 51, pp. 343–352.
- [115] Y. Chon and H. Cha, "Lifemap: A smartphone-based context provider for location-based services," *IEEE Pervasive Computing*, vol. 10, no. 2, pp. 58–67, April 2011.
- [116] O. Yurur, M. Labrador, and W. Moreno, "Adaptive and energy efficient context representation framework in mobile sensing," *IEEE Trans. on Mob. Comp.*, 2013.
- [117] O. Yurur, C. H. Liu, X. Liu, K. K. Leung, and W. Moreno, "Energy efficient and context-aware smartphone sensor employment," *accepted and to appear in IEEE Trans. Vehicular Technology*, 2014.
- [118] M.-R. Ra, J. Paek, A. B. Sharma, R. Govindan, M. H. Krieger, and M. J. Neely, "Energy-delay tradeoffs in smartphone applications," in *Proc. of Int'l Conf. on Mobile Systems, Applications, and Services*, ser. *MobiSys '10*, 2010, pp. 255–270.
- [119] S. C. Mukhopadhyay and O. A. Postolache, *Pervasive and Mobile Sensing and Computing for Healthcare: Technological and Social Issues*. Springer Verlag, 2012, vol. 2.
- [120] D. Craig, "Cognitive prosthetics in alzheimer's disease: A trial of a novel cell phoned- based reminding system," *Elsevier Alzheimer's & dementia*, vol. 6, no. 4, pp. 1552–5260, 2010.
- [121] M. Skubic, "A ubiquitous sensing environment to detect functional changes in assisted living apartments: The tiger place experience," *Elsevier Alzheimer's & dementia*, vol. 6, no. 4, pp. 1552–5260, 2010.
- [122] M. Fahim, I. Fatima, S. Lee, and Y.-K. Lee, "Daily life activity tracking application for smart homes using android smartphone," in *Int'l Conf. on Adv. Commun. Tech. '12*, feb. 2012, pp. 241–245.
- [123] S. Hu, H. Wei, Y. Chen, and J. Tan, "A real-time cardiac arrhythmia classification system with wearable sensor networks," *Sensors*, vol. 12, no. 9, pp. 12 844–12 869, 2012.
- [124] C. Liolios, C. Doukas, G. Fournalas, and I. Maglogiannis, "An overview of body sensor networks in enabling pervasive healthcare and assistive environments," in *ACM PETRA'10*, 2010, pp. 43:1–43:10.
- [125] A. Rehman, M. Mustafa, I. Israr, and M. Yaqoob, "Survey of wearable sensors with comparative study of noise reduction ecg filters," *Int. J. Com. Net. Tech.*, vol. 1, no. 1, pp. 45–66, 2013.
- [126] "Centinela: A human activity recognition system based on acceleration and vital sign data," *Pervasive and Mobile Computing*, vol. 8, no. 5, pp. 717 – 729, 2012.
- [127] A. Pantelopoulos and N. Bourbakis, "A survey on wearable sensor-based systems for health monitoring and prognosis," *IEEE Trans. on Syst., Man, and Cyber.*, vol. 40, no. 1, pp. 1–12, jan. 2010.
- [128] C.-W. Lin, Y.-T. Yang, J.-S. Wang, and Y.-C. Yang, "A wearable sensor module with a neural-network-based activity classification algorithm for daily energy expenditure estimation," *IEEE Trans. on Info. Tech. in Biomed.*, vol. 16, no. 5, pp. 991–998, sept. 2012.
- [129] D.-W. Jang, B. Sun, S.-Y. Cho, S. Sohn, and K.-R. Han, "Implementation of ubiquitous health care system for active measure of emergencies," in *Int'l Conf. on Adv. Lang. Proc. and Web Info. Tech. '07*, aug. 2007, pp. 420–425.
- [130] R. Picard, E. Vyzas, and J. Healey, "Toward machine emotional intelligence: analysis of affective physiological state," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 10, pp. 1175–1191, oct 2001.
- [131] E. Jovanov, A. O'Donnell Lords, D. Raskovic, P. Cox, R. Adhami, and F. Andrasik, "Stress monitoring using a distributed wireless intelligent sensor system," *IEEE Engineering in Medicine and Biology Mag.*, vol. 22, no. 3, pp. 49 – 55, may-june 2003.
- [132] S. Consolvo, D. W. McDonald, T. Toscos, M. Y. Chen, J. Froehlich, B. Harrison, P. Klasnja, A. LaMarca, L. LeGrand, and R. Libby, "Activity sensing in the wild: a field trial of ubifit garden," in *ACM SIGCHI'08*, pp. 1797–1806.
- [133] E. Jovanov and A. Milenkovic, "Body area networks for ubiquitous healthcare applications: Opportunities and challenges," *Jour.l of Med. Sys.*, p. 110, 2011.
- [134] P. Yan, I. Lin, M. Roy, E. Seto, C. Wang, and R. Bajcsy, in *Wireless Internet Conf. (WICON)*, march 2010.
- [135] E. M. Berke, T. Choudhury, S. Ali, and M. Rabbi, "Objective measurement of sociability and activity: Mobile sensing in the community," *The Annals of Family Med.*, vol. 9, no. 4, pp. 344–350, 2011.
- [136] K. G. Stanley and N. D. Osgood, "The potential of sensor-based monitoring as a tool for health care, health promotion, and research," *The Annals of Family Med.*, vol. 9, no. 4, pp. 296–298, 2011.
- [137] J. Blum and E. Magill, "M-psychiatry: Sens. netw. for psychiatric health monitoring," in *9th Annual Postgraduate Symp. The Convergence of Telecomm., Networking and Broadcasting*, june 2008.
- [138] O. Lara and M. Labrador, "A mobile platform for real-time human activity recognition," in *IEEE CCNC'12*, jan. 2012, pp. 667 –671.
- [139] J. Choi and R. Gutierrez-Osuna, "Using heart rate monitors to detect mental stress," in *Int'l Workshop on Wear. and Impl. Body Sens. Netw. '09*, june 2009.
- [140] A. Henprasertae, S. Thiemjarus, and S. Marukatat, "Accurate activity recognition using a mobile phone regardless of device orientation and location," in *Int'l Conf. on Body Sensor Networks (BSN)*, May 2011, pp. 41–46.
- [141] F. Ben Abdesslem, A. Phillips, and T. Henderson, "Less is more: energy-efficient mobile sensing with senseless," in *ACM MobiHeld'09*, 2009, pp. 61–62.
- [142] C. Randell and H. Muller, "Context awareness by analysing accelerometer data," in *Int'l Symp. on Wearable Computers*, Oct 2000, pp. 175–176.
- [143] S. Dhar and U. Varshney, "Challenges and business models for mobile location-based services and advertising," *Communications of the ACM*, vol. 54, no. 5, pp. 121–128, May 2011.
- [144] H. Yoon, Y. Zheng, X. Xie, and W. Woo, "Social itinerary recommendation from user-generated digital trails," *Personal and Ubiquitous Computing*, vol. 16, pp. 469–484, 2012.
- [145] V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang, "Towards mobile intelligence: Learning from gps history data for collaborative recommendation," *Artif. Intell.*, vol. 184-185, pp. 17–37, Jun. 2012.
- [146] L. Liao, D. Fox, and H. Kautz, "Extracting places and activities from gps traces using hierarchical conditional random fields," *Int. J. Rob. Res.*, vol. 26, no. 1, pp. 119–134, Jan. 2007.
- [147] X. J. Ban and M. Gruteser, "Towards fine-grained urban traffic knowledge extraction using mobile sensing," in *ACM UrbComp'12*, pp. 111–117.
- [148] J. Krumm and E. Horvitz, "Predestination: inferring destinations from partial trajectories," in *ACM UbiComp'06*, pp. 243–260.
- [149] J. Herrera, D. Work, R. Herring, X. Ban, Q. Jacobson, and A. Bayen, "Evaluation of traffic data obtained via GPS-enabled mobile phones: The Mobile Century field experiment," *Transportation Research Part C*, vol. 18, no. 4, pp. 568–583, August 2010.
- [150] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson, "Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones," in *ACM SenSys'09*, 2009, pp. 85–98.
- [151] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: rich monitoring of road and traffic conditions using mobile smartphones," in *ACM SenSys'08*, pp. 323–336.
- [152] S. Reddy, K. Shilton, G. Denisov, C. Cenizal, D. Estrin, and M. Srivastava, "Biketastic: sensing and mapping for better biking," in *ACM SIGCHI'10*, pp. 1817–1820.
- [153] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell, "The bikenet mobile sensing system for cyclist experience mapping," in *ACM SenSys'07*, pp. 87–101.
- [154] N. Husted and S. Myers, "Mobile location tracking in metro areas: malnets and others," in *ACM CCS'10*, pp. 85–96.
- [155] G.-S. Ahn, M. Musolesi, H. Lu, R. Olfati-Saber, and A. T. Campbell, "Metrotrack: predictive tracking of mobile events using mobile phones," in *ACM DCROSS'10*, pp. 230–243.

- [156] E. Miluzzo, N. D. Lane, S. B. Eisenman, and A. T. Campbell, "Cenceme: injecting sensing presence into social networking applications," in *EuroSSC '07*, pp. 1–28.
- [157] A. Beach, M. Gartrell, X. Xing, R. Han, Q. Lv, S. Mishra, and K. Seada, "Fusing mobile, sensor, and social data to fully enable context-aware computing," in *ACM HotMobile'10*, pp. 60–65.
- [158] G. Chatzimilioudis, A. Konstantinidis, C. Laoudias, and D. Zeinalipour-Yazti, "Crowdsourcing with smartphones," *IEEE Internet Comput.*, vol. 16, no. 5, pp. 36–44, sept.-oct. 2012.
- [159] K. Shilton, "Four billion little brothers?: privacy, mobile phones, and ubiquitous data collection," *Commun. of the ACM*, vol. 52, no. 11, pp. 48–53, Nov. 2009.
- [160] H.-L. Truong and S. Dustdar, "A survey on context-aware web service systems," *Int'l J. of Web Info. Sys.*, vol. 5, no. 1, pp. 5–31, 2009.
- [161] A. Purohit, Z. Sun, F. Mokaya, and P. Zhang, "Sensorfly: Controlled-mobile sensing platform for indoor emergency response applications," in *Int'l Conf. on Info. Proc. in Sens. Netw. '11*, april 2011, pp. 223–234.
- [162] C. Gomez-Otero, R. Martinez, and J. Caffarel, "Climapp: A novel approach of an intelligent hvac control system," in *Iberian Conf. on Info. Syst. and Tech. '12*, june 2012, pp. 1–6.
- [163] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, and P. Boda, "Peir: the personal environmental impact report, as a platform for participatory sensing systems research," in *ACM/USENIX MobiSys '09*.
- [164] J. Froehlich, T. Dillahunt, P. Klasnja, J. Mankoff, S. Consolvo, B. Harrison, and J. Landay, "Ubigreen: investigating a mobile tool for tracking and supporting green transportation habits," in *Int'l Conf. on Human factors in Comput. Syst.*, 2009, pp. 1043–1052.
- [165] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden, "Cartel: a distributed mobile sensor computing system," in *ACM SenSys'06*, 2006, pp. 125–138.
- [166] R. Honicky, E. Brewer, E. Paulos, and R. White, "N-smarts: networked suite of mobile atmospheric real-time sensors," in *ACM SIGCOMM'08*, pp. 25–30.
- [167] R. Murty, A. Gosain, M. Tierney, A. Brody, A. Fahad, J. Bers, and M. Welsh, "Citysense: A vision for an urban-scale wireless networking testbed," in *IEEE Int'l Conf. on Tech. for Homeland Security*, 2008.
- [168] "Environmental sensor networks: A revolution in the earth system science?" *Earth-Science Reviews*, vol. 78, no. 34, pp. 177–191, 2006.
- [169] R. K. Ganti, N. Pham, H. Ahmadi, S. Nangia, and T. F. Abdelzaher, "Greengps: A participatory sensing fuel-efficient maps application," in *Int'l Conf. on Mobile Sys., Apps, and Serv.*, 2010, pp. 151–164.
- [170] N. Maisonneuve, M. Stevens, M. E. Niessen, P. Hanappe, and L. Steels, "Citizen noise pollution monitoring," in *Int'l Conf. on Digital Government Research: Social Networks: Making Connections between Citizens, Data and Government*, 2009, pp. 96–103.
- [171] B. Priyantha, D. Lymberopoulos, and J. Liu, "Littlerock: Enabling energy-efficient continuous sensing on mobile phones," *IEEE Pervasive Computing*, vol. 10, no. 2, pp. 12–15, april-june 2011.
- [172] A. Carroll and G. Heiser, "The systems hacker's guide to the galaxy energy usage in a modern smartphone," in *Proc. of the 4th Asia-Pacific Workshop on Systems*. ACM, 2013, p. 5.
- [173] O. Yurur, C. H. Liu, X. Liu, and W. Moreno, "Adaptive sampling and duty cycling for smartphone accelerometer," in *Mobile Ad-Hoc and Sensor Systems (MASS), 2013 IEEE 10th International Conference on*. IEEE, 2013, pp. 511–518.
- [174] A. Carroll and G. Heiser, "An analysis of power consumption in a smartphone," in *Proc. of the USENIX Conf. on USENIX Annual Technical Conference*, ser. USENIXATC'10, 2010, pp. 21–21.
- [175] L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R. P. Dick, Z. M. Mao, and L. Yang, "Accurate online power estimation and automatic battery behavior based power model generation for smartphones," in *Proc. of the IEEE/ACM/IFIP Int'l Conf. on Hardware/Software Codesign and System Synthesis*, ser. CODES/ISSS '10. ACM, 2010, pp. 105–114.
- [176] A. J. Oliner, A. Iyer, E. Lagerspetz, S. Tarkoma, and I. Stoica, "Collaborative energy debugging for mobile devices," in *Proc. of USENIX Conference on Hot Topics in System Dependability*, ser. HotDep'12. Berkeley, CA, USA: USENIX Association, 2012, pp. 6–6.
- [177] B. Scrosati and J. Garche, "Lithium batteries: Status, prospects and future," *Journal of Power Sources*, vol. 195, no. 9, pp. 2419–2430, 2010.
- [178] V. Rao, G. Singhal, A. Kumar, and N. Navet, "Battery model for embedded systems," in *Int'l Conf. VLSI Design*, 2005, pp. 105–110.
- [179] L. Cloth, M. R. Jongerden, and B. R. Haverkort, "Computing battery lifetime distributions," in *Int'l Conf. Dependable Syst. and Netw. (DSN)*, 2007, pp. 780–789.
- [180] P. Notten and D. Danilov, "From battery modeling to battery management," in *IEEE Int'l Telecommunications Energy Conference*, Oct 2011, pp. 1–8.
- [181] O. Yurur, C. H. Liu, and W. Moreno, "Unsupervised posture detection by smartphone accelerometer," *Electronics Letters*, vol. 49, no. 8, pp. 562–564, 2013.
- [182] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, Apr. 2010.
- [183] S. Abolfazli, Z. Sanaei, E. Ahmed, A. Gani, and R. Buyya, "Cloud-based augmentation for mobile devices: Motivation, taxonomies, and open challenges," *IEEE Communications Surveys Tutorials*, vol. 16, no. 1, pp. 337–368, 1 2014.
- [184] V. Chandrasekhar, J. Andrews, and A. Gatherer, "Femtocell networks: a survey," *IEEE Communications Magazine*, vol. 46, no. 9, pp. 59–67, September 2008.



**Özgür Yürür** (M'13) received a double major from the Department of Electronics Engineering and the Department of Computer Engineering at Gebze Institute of Technology, Kocaeli, Turkey, in 2008, and MSEE and PhD from the Department of Electrical Engineering at the University of South Florida, Tampa, Florida, in 2010 and 2013, respectively. He is currently with RF Micro Devices, responsible for research and design of new test development strategies and also for the implementation of hardware, software, and firmware solutions for 2G, 3G, 4G, and wireless-based company products. In addition, Dr. Yurur conducts research in the field of mobile sensing. His research area covers ubiquitous sensing, mobile computing, machine learning, and energy-efficient optimal sensing policies in wireless networks. The main focus of his research is on developing and implementing accurate, energy efficient, predictive, robust, and optimal context-aware algorithms and framework designs on sensor-enabled mobile devices.



**Chi Harold Liu** (M'10) receives his Ph.D. degree from Imperial College, UK, and his B.Eng. degree from Tsinghua University, China. He is currently a Full Professor at the School of Software, Beijing Institute of Technology, China. He is also the Head of Department of Software Service Engineering, the Director of IBM Mainframe Excellence Center (Beijing), Director of IBM Big Data Technology Center, and Director of National Laboratory of Data Intelligence for China Light Industry. Before moving to academia, he joined IBM Research - China as a staff researcher and project manager, after working as a postdoctoral researcher at Deutsche Telekom Laboratories, Germany, and a visiting scholar at IBM T. J. Watson Research Center, USA. His current research interests include the Internet-of-Things (IoT), big data analytics, mobile computing, and wireless ad hoc, sensor, and mesh networks. He received the Distinguished Young Scholar Award in 2013, IBM First Plateau Invention Achievement Award in 2012, and IBM First Patent Application Award in 2011 and was interviewed by EEWeb.com as the Featured Engineer in 2011. He has published more than 60 prestigious conference and journal papers and owned more than 10 EU/U.S./China patents. He serves as the editor for KSII Trans. on Internet and Information Systems and the book editor for four books published by Taylor & Francis Group, USA. He also has served as the general chair of IEEE SECON'13 workshop on IoT Networking and Control, IEEE WCNC'12 workshop on IoT Enabling Technologies, and ACM UbiComp'11 Workshop on Networking and Object Memories for IoT. He served as the consultant to Asian Development Bank, Bain & Company, and KPMG, USA, and the peer reviewer for Qatar National Research Foundation, and National Science Foundation, China. He is a member of IEEE and ACM.



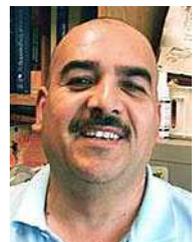
**Zhengguo Sheng** (M'11) is a lecturer at School of Engineering and Informatics, the University of Sussex, UK. He is also the visiting faculty of University of British Columbia (UBC) and co-founder of WRTnode. Previously, he was with UBC as a research associate, and with France Telecom Orange Labs as the senior researcher and project manager in M2M/IoT. He also worked as a research intern with IBM T. J. Watson Research Center, USA, and U.S. Army Research Labs. Before joining Orange Labs, he received his Ph.D. and M.S. with distinction at

Imperial College London in 2011 and 2007, respectively, and his B.Sc. from the University of Electronic Science and Technology of China (UESTC) in 2006. He has published more than 30 prestigious conference and journal papers. He is also the recipients of Auto21 TestDRIVE Competition Award 2014 and Orange Outstanding Researcher Award 2012. His current research interests cover IoT/M2M, cloud/edge computing, vehicular communications, and power line communication (PLC).



**Victor C. M. Leung** (F'03) is a Professor and holder of the TELUS Mobility Research Chair in Advanced Telecommunications Engineering in the Department of Electrical and Computer Engineering at the University of British Columbia in Vancouver, Canada. His research interests are in the broad areas of wireless networks and mobile systems, in which he has co-authored more than 600 technical papers in international journals and conference proceedings. Several papers co-authored by him and his students have won best paper awards.

Dr. Leung is a registered professional engineer in the Province of British Columbia, Canada. He is a Fellow of IEEE, the Engineering Institute of Canada, and the Canadian Academy of Engineering. He has served on the editorial boards of the IEEE Journal on Selected Areas in Communications Wireless Communications Series, IEEE Transactions on Wireless Communications and IEEE Transactions on Vehicular Technology, and is serving on the editorial boards of the IEEE Transactions on Computers, IEEE Wireless Communications Letters, Journal of Communications and Networks, Computer Communications, as well as several other journals. He has guest-edited many journal special issues, and served on the technical program committees of numerous international conferences. He has contributed to the organization of many conferences and workshops.



**Wilfrido Moreno** received his B.S.E.E., M.S.E.E. and Ph.D. degrees from the University of South Florida (USF) in Tampa, Florida. Dr. Moreno is a Professor in the Department of Electrical Engineering at USF and is the director for the Ibero-American Science and Technology Educational Consortium at USF (ISTEC-USF). In addition, Dr. Moreno has served as Director of the R&D Initiative for ISTEC Inc. since the fall of 2003. In the area of research, Dr. Moreno is a founding member of the former Center for Microelectronics Research, (CMR), which is

currently the Nanotechnology Research & Education Center, (NREC). Dr. Moreno's current areas of interest are oriented towards system integration by providing "off-the-shelf" hardware/software solutions to industrial applications. Application areas of interest include Digital Signal Processing, Communications, Industrial Power & Controls, Nano/Micro-electronics, Biomedical Engineering and Multimedia Solutions, and Optimal Sensing. Dr. Moreno's broad educational background in Communication, Systems & Controls and Micro/Nano-electronics has allowed him to work on multidisciplinary research projects where concepts and expertise in all of these areas comes together for system design, fabrication and test. Currently, Dr. Moreno has supervised industrial projects applying System of Systems Engineering methodologies. Dr. Moreno has published over a hundred technical papers, has participated in over one hundred and fifty International conferences and seminars around the World, and has organized over fifty hands-on workshops in the area of Electronic Design Automation, Systems Engineering & Controls, Nano-electronics, and Digital Signal Processing.



**Kin K. Leung** (F'01) received his B.S. degree from the Chinese University of Hong Kong in 1980, and his M.S. and Ph.D. degrees from University of California, Los Angeles, in 1982 and 1985, respectively.

He joined AT&T Bell Labs in New Jersey in 1986 and worked at its successor companies, AT&T Labs and Bell Labs of Lucent Technologies, until 2004. Since then, he has been the Tanaka Chair Professor in the Electrical and Electronic Engineering (EEE), and Computing Departments at Imperial College in London. He serves as the Head of Communications

and Signal Processing Group in the EEE Department at Imperial. His research interests focus on networking, protocols, optimization and modeling issues of wireless broadband, sensor and ad-hoc networks. He also works on multi-antenna and cross-layer designs for the physical layer of these networks.

He received the Distinguished Member of Technical Staff Award from AT&T Bell Labs in 1994, and was a co-recipient of the 1997 Lanchester Prize Honorable Mention Award. He was elected as an IEEE Fellow in 2001. He received the Royal Society Wolfson Research Merits Award from 2004 to 2009 and became a member of Academia Europaea in 2012. Along with his co-authors, he also received several best paper awards at major conferences. He has actively served on many conference committees. He serves as a member (2009-11) and the chairman (2012-15) of the IEEE Fellow Evaluation Committee for Communications Society. He was a guest editor for the IEEE Journal on Selected Areas in Communications (JSAC), IEEE Wireless Communications and the MONET journal, and as an editor for the JSAC: Wireless Series, IEEE Transactions on Wireless Communications and IEEE Transactions on Communications. Currently, he is an editor for the ACM Computing Survey and International Journal on Sensor Networks.