A Tutorial on Internet of Behaviors: Concept, Architecture, Technology, Applications, and Challenges

Qinglin Zhao, *Senior Member, IEEE*, Guangcheng Li, Jincheng Cai, MengChu Zhou, *Fellow, IEEE*, and Li Feng

***Abstract*—In his blogs of 2012, Dr. Göte Nyman coined Internet of Behaviors (IoB). In his idea, people's behaviors are very good predictors of their needs, and hence technology companies can network human behaviors and construct behavior-based systems to deliver more timely, responsive, and intelligent services with privacy protection. In 2020, Gartner ranked IoB first among its top nine strategic technology trends and suggested that IoB would touch almost half of the world population. Built on his core idea of IoB, this paper is the first one to comprehensively, systematically, and deeply introduce the IoB concept, essential features, architecture, enabling technologies, applications, and open research issues. In this paper, we first give a formal definition and classification of IoB and reveal its fundamental difference from Internet of Things (IoT). Then, we propose a five-layer IoB architecture (consisting of behavior perception, behavior networking, behavior computing, service provision, and security/privacy) and provide the in-depth analysis of IoB enabling technologies. Particularly, we discuss functional requirements and possible fields of an IoB address (i.e., a human behavior identifier), present the networking and maintenance approaches to behaviors, explore four important implications of behavior computing, i.e., intention inference, behavior derivation, behavior programming, and behavior-chain optimization, as well as give a decentralized privacy-protection solution to Dr. Nyman's behavior-identity-separation idea. Next, we deeply investigate potential IoB applications in smart home/transportation/healthcare/business, and human-robot interaction. Finally, we provide insightful discussions on open research issues. This paper should help researchers and practitioners understand IoB quickly and promote future IoB development.**

***Index Terms*—Internet of Behaviors (IoB), Internet of Things (IoT), behavior perception, behavior networking, behavior computing, architecture, applications, security and privacy.**

Manuscript received 17 September 2022; revised 29 December 2022; accepted 30 January 2023. Date of publication 22 February 2023; date of current version 23 May 2023. This work was supported by the Science and Technology Development Fund, Macau, SAR, under Grant 0093/2022/A2, Grant 0076/2022/A2, Grant 0008/2022/AGJ, and Grant 0047/2021/A1. *(Corresponding author: MengChu Zhou.)*

Qinglin Zhao, Guangcheng Li, Jincheng Cai, and Li Feng are with the School of Computer Science and Engineering, Macau University of Science and Technology, Macau, China (e-mail: qlzhao@must.edu.mo; guangcheng.li@hotmail.com; jincheng.cai@outlook.com; lfeng@must.edu.mo).

MengChu Zhou is with the School of Information and Electronic Engineering, Zhejiang Gongshang University, Hangzhou 314423, China, and also with the Helen and John C. Hartmann Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: zhou@njit.edu).

Digital Object Identifier 10.1109/COMST.2023.3246993

I. Introduction

A. Background

TO DATE, Internet of Things (IoT) is extensively applied in various areas and IoT devices (such as sensors, actuators, and cameras) are becoming increasingly ubiquitous. International Data Corporation has predicted that by 2025, there would be 55.7 billion connected IoT devices worldwide [1]. These massive IoT devices constantly collect our personal data (such as blood pressure, heartbeat, and body action) and environmental data (such as temperature, humidity, and pollution level). By 2025, International Data Corporation predicts that the size of these collected data would be 73.1 Zettabytes [1]. These huge amounts of data contain a great deal of useful information for our life, business, government, etc. Hence, lots of ingenious artificial intelligence (AI) technologies (e.g., data-mining and machine-learning algorithms) have been designed to reveal such information. However, IoT systems (including their adopted AI algorithms) can only analyze the meaning of the data (we collect in advance) and perform our preset rules mechanically, but cannot understand why we collect these data and how we use these data. This inherent nature of IoT systems limits their capacity of serving human beings.

The DIKW (data, information, knowledge, wisdom) pyramid [2] has been widely recognized in providing a path from collecting data to maximizing their use. In the pyramid structure as shown in Fig. 1, data, a set of symbols, represent stimuli or signals (e.g., those collected from IoT devices). Information, the result of processing data, interprets the meanings of individual data points. Knowledge, which arises when information is synthesized into formal relationships and interconnections, involves recognizing patterns in information. Wisdom, the ability that applies knowledge to address problems, manifests the understanding of fundamental principles in knowledge. Only when a system can reach the wisdom layer in terms of processing capacity, it can better understand and use data, as well as provide the best service for us. From the angle of the DIKW pyramid, IoT systems typically touch only the data and information layers. Thus there is a huge gap from the information layer to the wisdom layer.

In his blog in 2012, Dr. Göte Nyman, a psychology professor at the University of Helsinki, created an idea of Internet of Behaviors (IoB) [4] to bridge the information-to-wisdom gap. In his idea, people's individual habits and behaviors are

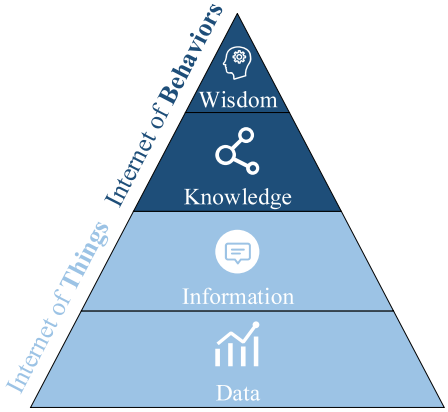


Fig. 1. IoB is built on the top of IoT and touches the knowledge and wisdom layers of the DIKW pyramid model [2], [3].

far better predictors of their situational and personal needs than the most sophisticated artificial intelligence and machine learning alone would ever be. By uniting the user data collected by smartphones, television sets, smart speakers, Web browsers and other devices, technology companies could build systems capable of analyzing the resulting stream of contextual information to develop more timely, responsive and intuitive products and services to benefit humans [5]. In his IoB concept, the separation of behavior data and identity is an inherent property of an IoB framework. IoB aims to utilize individual behavior data to provide powerful and situation-aware services, while protecting individual privacy. In his initiatives, companies and organizations can jointly collect all human behaviors by utilizing IoT systems, and then network these behaviors together and construct IoB systems. These IoB systems next perform behavior computing (i.e., various computational operations of modeling, analyzing and understanding human behaviors and their interactions; for example, intention inference for predicting human activities) by using the behavior data of IoB and AI technologies, and finally provide intelligent service with privacy protection according to human intention and promote the positive behaviors of humans. In this way, IoB systems can well avoid the blindness of data analysis and computation (as in IoT systems). At that time, Nyman had seen the great potential of IoB and believed that IoB might incubate unlimited possibilities "to be used in business, personal finances, education, work, collaboration, coordination, service provision, marketing, personalization, you name it" [4].

However, Nyman's forward-looking IoB idea has not been accepted until recently. In 2019, Gartner predicted that "by 2023, individual activities would be tracked digitally by an IoB to influence benefit and service eligibility for 40% of people worldwide" [6]. In 2020, IoB was listed as one of Gartner's nine strategic technology trends that would enable the plasticity or flexibility that resilient businesses would require in the significant upheaval driven by COVID-19 and the current economic state of the world [7]. With IoB, we may capture the "digital dust" of people's daily lives, the data that spans the digital and physical worlds, from a variety of sources, and then analyze and use the digital dust to influence human behaviors through feedback loops.

B. Motivation

IoB has received growing attention since Gartner listed it as a major future technological trend in 2020. However, at the time of writing, all learning materials on IoB are from Nyman's blogs that spanned from 2012 to 2021, which mainly shared the primitive concept, idea, and vision of IoB. There are few studies on IoB. Even the studies that are the most relevant to IoB, such as elderly healthcare [8], [9] and vehicle lane-change assistance [10], [11], are IoT-compliant and mainly involve the behavior perception, without touching the heart of IoB, i.e., behavior networking and computing. Therefore, there lacks a systematic and deep introduction to IoB. This study is devoted to such an introduction.

C. Our Contributions

Built on Nyman's core idea on IoB, this paper is the first to introduce IoB concept, essential features, architecture, enabling technologies, applications, and open research issues in a comprehensive and systematic manner. This study intends to make the following contributions:

1) It gives the formal definitions of the involved concepts including action, behavior, intention, and IoB. Particularly, we reveal the essential features of IoB, define two types of IoB: permissionless and permissioned IoB, and exemplify the difference between IoT and IoB.
2) It proposes a five-layer IoB architecture: i) behavior perception, ii) behavior networking, iii) behavior computing, iv) service provision, and v) security/privacy, and highlight their main functionalities.
3) It provides the in-depth analysis of IoB enabling technologies. We make an extensive survey on existing behavior perception methods and security/privacy mechanisms. Particularly, we discuss functional requirements and possible fields of an IoB address (i.e., a human behavior identifier), present the networking and maintenance approaches to behaviors, propose four important implications of behavior computing: i) intention inference, ii) behavior derivation, iii) behavior programming, and iv) behavior-chain optimization, as well as give a decentralized privacy-protection solution to Nyman's behavior-identify-separation idea.
4) It investigates potential IoB applications in smart home, smart transportation, smart health care, human-robot interaction and smart business.
5) It offers insightful discussions on technical challenges in terms of each layer of the proposed IoB architecture.

To date, IoB is in its infancy and is expected to play an important role that influences almost half of the world population. This study aims at providing a timely, systematic and deep introduction to IoB for researchers and practitioners, companies, institutions, organizations, and governments.

The rest of this paper is organized as follows. Section II reveals the difference between IoT from IoB. Section III defines related concepts of IoB. Section IV presents an IoB architecture. Section V provides the in-depth analysis of IoB enabling technologies. Section VI shows various potential IoB

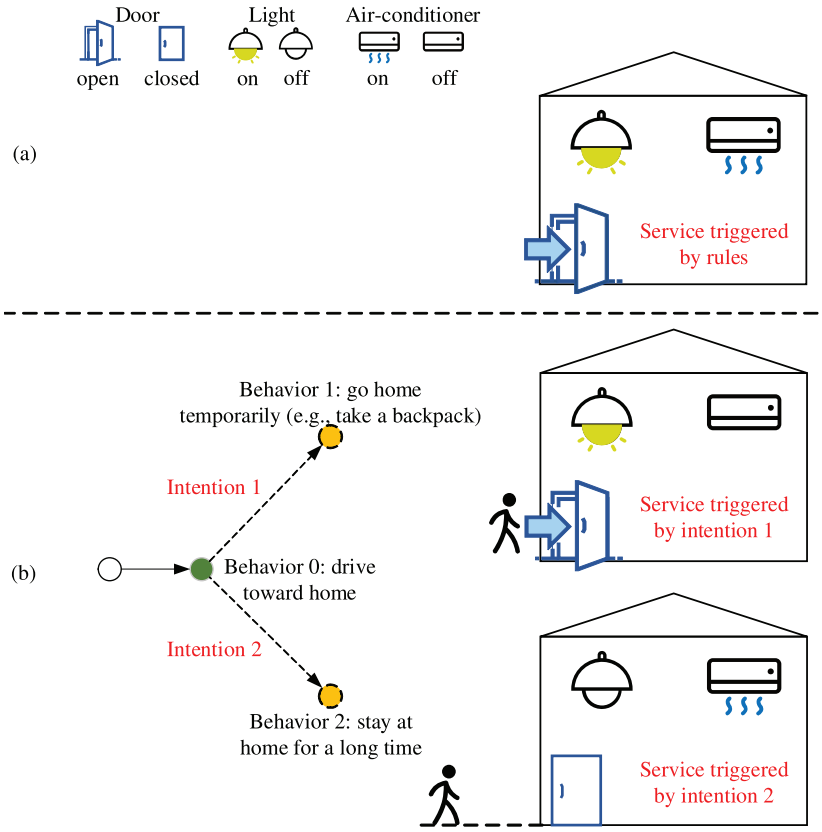


Fig. 2. An example of Go-Home: (a) IoT-based services, and (b) IoB-based services.

applications. Section VII discusses technical challenges of IoB. Finally, Section VIII concludes this survey paper.

II. An Example of Go-Home: IoT vs. IoB

Here, we first present an example of Go-Home and then discuss insights into this example.

A. Go-Home Example

Image that one day, Bob sets his driving navigation and then drives his car to go home. Also image that Bob's home has been equipped with various smart IoT devices such as smart door lock, light, and air-conditioner, which are connected to a home gateway. An IoT or IoB system may remotely control (e.g., switched on/off) these devices via the gateway for service provision. Consequently, the two systems may respectively offer two types of services: IoT and IoB-based services, when Bob reaches home.

IoT-based services: An IoT system provides rule-based services. In the Go-Home example in Fig. 2(a), the IoT system may define and implement a rule: automatically switch on lights and air-conditioners whenever the door is unlocked. Therefore, when Bob arrives at home, he may enjoy a comfortable service. The IoT system seems intelligent enough, but it cannot understand why Bob goes home. In reality, Bob may go home temporarily (e.g., fetch a bag) or go home and stay at home for a long time. In the former, switching on the air-conditioner may cause unnecessary energy waste. In the latter, switching on the air-conditioner (triggered by the door being unlocked) may be too late, since it needs a while to cool down the home.

IoB-based services: An IoB system provides intention-oriented services and therefore is more intelligent than an IoT system. Assume that Bob has run an IoB system in a cloud/edge server. This system maintains a behavior network, where each network node marks one of Bob's behaviors and Bob may add new behaviors and the relationship among behaviors to the network according to his work routines and preferences. It can constantly collect and identify Bob's behaviors through the interaction with IoT devices of Bob's whereabouts, e.g., office, vehicle, and home, associates these real-time behaviors with the nodes of the behavior network, infers Bob's real-time intention, and finally instructs these involved IoT devices to provide timely services for Bob. In the Go-Home example, as shown in Fig. 2(b), assume that Bob has defined three behaviors in his behavior network, namely, Behavior 0: drive toward home, Behavior 1: go home temporarily, and Behavior 2: stay at home for a long time. He has also defined two behavior transitions, namely, Behaviors 0 to 1 and Behaviors 0 to 2, to reveal his behavior patterns. When Bob drives his car, once his IoB system infers his intention of going home, the system marks his current behavior to behavior 0. Following the chain of Bob's behavior transitions, the IoB system infers Bob's next intention, i.e., go home temporarily (intention 1) or stay at home for a long time (intention 2), by learning about his daily routine and social activities via his agenda and chat history in his social software platform. For example, when finding that Bob plans to go hiking

with his friends soon, the system infers that Bob would go home temporarily (e.g., to take a backpack) and then take off. If the system infers that Bob would exhibit Behavior 1 and then opens the door and switches on the light when Bob reaches home. If the system infers that Bob would exhibit Behavior 2 (staying home for long) and then switches on the air-conditioner 15 minutes before he reaches home, and next opens the door and switches on the light when Bob reaches home.

B. Insights From Go-Home example

The above Go-Home example helps us gain insights into the IoT and IoB systems. The IoT system works mechanically without considering human intention. In the Go-Home example, regardless of a reason, as long as the door is unlocked, the system always and automatically switches on lights and air-conditioners. Also, the system does not need to access Bob's data trail of daily life. Note that the IoT system, according to some preset rules, connects physical devices together, collects and processes data, reveals the information from the collected data, and then provides a service for a user based on the revealed information. In contrast, the IoB system first infers human intention and then invokes IoT devices to provide services. In the Go-Home example, the system always first learns about Bob's daily routine and social activities to infer Bob' intention and then determines which IoT devices are invoked. In case that the system cannot access Bob's data trail of daily life, the system does not open the door for Bob automatically and does not switch on the light as well even if Bob opens the door manually. Note that the IoB system connects human behaviors, which are of societal implications and should be coded and addressable, and specifies the transitions among behaviors. To provide services in line with his intention, Bob has to define and update the behavior nodes of his behavior network by his preference and some schemes. The IoB system has to interpret the meaning of his specific behavior pattern.

III. Concepts of IoB

In this section, we first define IoB, and then specify its properties and types.

A. Definition of IoB

IoB is currently in its infancy stage. Most of its discussions are informal and mainly from Nyman's blogs [12]. To date, it has no formal definitions. We define IoB next.

Definition 1 [Internet of Behaviors (IoB)]: describes the technology that infers human intention from the network of human behaviors for promoting positive behaviors and providing personalized services in line with human intention.

In the above definition, *behavior* is the range of actions and mannerisms made by humans in conjunction with themselves or their environment. *Action* refers to the physical movement of a human, *environment* refers to the systems or organisms around humans as well as the (inanimate) physical environment, and *intention* is a mental state that represents a commitment to carrying out a behavior in the future. A method to infer users' intention should take their environment and context (e.g., time-space-dependency and scenario of their behavior occurrence) into account.

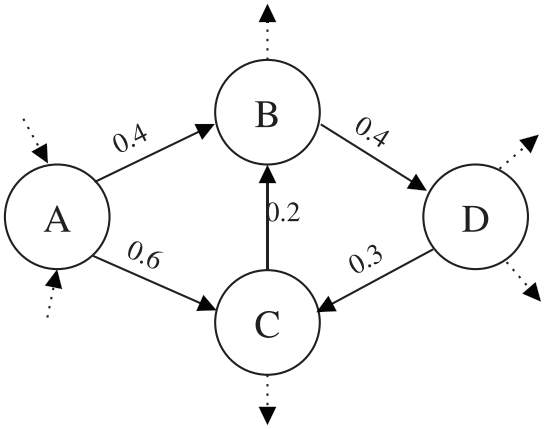


Fig. 3. An example of a behavior network.

There are mainly two different opinions about the concept of IoB: Private IoB (proposed by Nyman) and General IoB (described by Gartner). The former separates the behavior data and identity data of a person and therefore is more concerned with privacy and security. This IoB system requires specific architectures and platform solutions for such a separation. The latter is used "to link a person digitally to their actions" [6] and hence keeps the behavior data and identity data of a person as a useful data entity. This IoB system can be built almost directly on the existing ones including the privacy and security solutions.

B. IoB and Behavior Network

In this tutorial paper, IoB refers to the overall technology that exploits behavior data to infer human intentions. When it is clear from context, we also use IoB to denote the IoB system that implements the IoB technology. The IoB technology includes multiple components such as behavior perception, behavior network, and intention inference. In IoB, the behavior network is a fundamental component, and it refers to the set of human behaviors and the transition relationships among these behaviors. Built on the behavior network, an IoB system performs intention inference, service provision, etc.

In this work, we use a directed cyclic graph to represent a behavior network, where a vertex represents a behavior, and a directional edge between two vertex represents a directional behavior transition. The weight of an edge represents the probability of the behavior transition. For example, $\mathrm{A}\xrightarrow{0.4}\mathrm{B}$ and $\mathrm{A}\xrightarrow{0.6}\mathrm{C}$ denote that if Bob's current behavior is A, his next behavior is B with probability 0.4, and is C with probability 0.6. A behavior chain is a sequence of vertices $v_1, v_2, \ldots,$ and v_n, where $(v_i,\ v_{i+1})$ is an edge for $1 \leq i \leq N-1$. For example, the sequence, A, B, and D, is a behavior chain. People's intentions motivate their behavior transitions. For example, Ana is writing an article (i.e., her current behavior). She feels thirsty and wants to drink water (i.e., her intention) and hence she goes to buy a bottle of water (i.e., her next behavior).

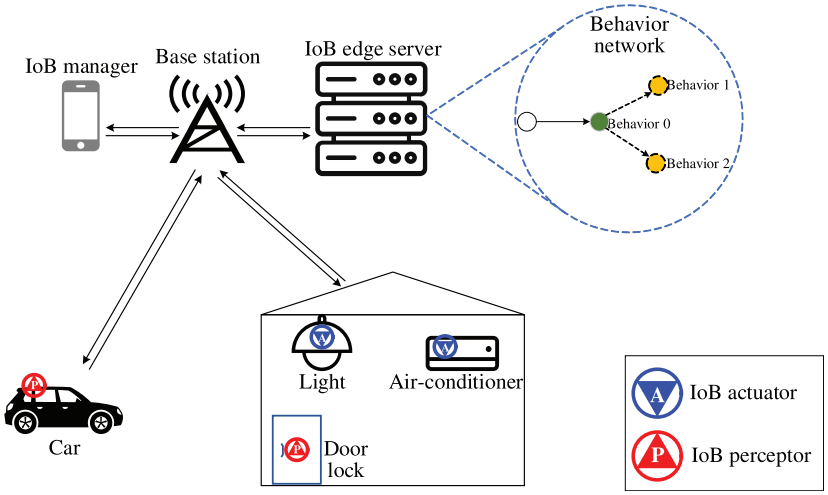


Fig. 4. How does IoB work?

C. How Does IoB Work?

In IoB, the behavior network is a fundamental component. For each application area, IoB relies on human intelligence and knowledge and AI technology to create a network of behaviors. For example, the operators (e.g., individuals, social entities, and companies) enumerate and define all possible meaningful behaviors of users and give their societal implications, where the number of behaviors, determined by the granule size of a behavior, is finite. On this basis, an IoB system collects the behavior histories of users and learns the behavior-transition relationships and probabilities from these behavior histories via AI technology (e.g., if there is a transition from behavior A to behavior B in the behavior histories, IoB may establish one directional connection from behavior A to behavior B), and finally constructs a network of behaviors. Some behaviors of users might be different from or influenced by those of others. Therefore, they may tailor the network to adapt to themselves and update the network to reflect other users' influence on them.

Once creating a network of behaviors, IoB may use the network to track users' behavior in real time, infer their intention via AI technology and provide personalized services in line with their intention. Next, we proceed the example of Go-Home to specify how IoB works. As shown in Fig. 4, assume that Bob has an IoB edge server. The server stores a network of Bob's behaviors including Bob's predefined three behaviors, namely, Behavior 0: drive toward home, Behavior 1: go home temporarily, and Behavior 2: stay at home for a long time, where there is one directional connection each from Behavior 0 to Behaviors 1 and 2, respectively. The server also runs other IoB components for servicing Bob timely. Assume that Bob is now driving his car toward home. The IoB behavior perceptor, which is equipped in Bob's car, then notifies the server of Bob's driving behavior via a base station. When the server infers his intention of going home, it marks his current behavior to behavior 0 in his behavior network. From Behavior 0, the server knows that Bob would exhibit behaviors 1 and 2 next, and hence respectively calculates the intention probabilities from behavior 0 to behaviors 1 and 2, by learning about his daily routine and social activities via his agenda and chat history. Once the server identifies Bob's intention, it then invokes the IoB actuators via a base station, which are installed at Bob's home to provide services in line with Bob's intention. In this example, Bob's IoB system consists of an IoB edge server, IoB behavior perceptors, and IoB actuators, and an IoB manager that configures and manages the IoB system.

D. Difference Between IoT and IoB

In this section, we present the properties of IoB and compare them with those of IoT in Table I, in terms of the DIKW level, focused data, design principle, integration with AI, technology, applications, quality of experience, and development stage, etc.

The DIKW (data, information, knowledge, and wisdom) pyramid model presents a path from collecting data to maximizing their use. From the angle of the DIKW pyramid, IoB bridges the information-to-wisdom gap. The reasons are as follows. Human intention is a mental state that represents a commitment to carrying out a behavior in the future and hence can be used to predict human needs, while human behavior is an exhibition of human intention and hence can be used to infer human intention. Building on this idea, the design of an IoB system centers around intention inference to provide the best service for us. Roughly speaking, an IoB system collects physical or online behaviors of a person via IoT devices or software, and networks these behaviors together, then employs ingenious technology (e.g., AI) to reveal his/her behavior patterns and infer his/her intentions. On this basis, the IoB system predicts his/her next behavior and provide services in line with his/her intention. By exploiting human behavior data, IoB systems can understand human intentions (or human needs) and hence maximize the capacity of serving human beings, avoiding the blindness of service provision. In other words, they can gain knowledge on human needs and apply the knowledge to serve human beings. In the Go-Home example, the IoB system first infers Bob' intention and then determines which IoT devices are invoked, well meeting Bob's needs while wasting no energy. Therefore, in this sense, IoB bridges the information-to-wisdom gap. IoB aims at providing

TABLE I
DIFFERENCES BETWEEN IOT AND IOB

		IoT	IoB
DIKW level		Data → Information	Information → Knowledge → Wisdom
Focused data		Non-behavior data	Human-behavior data
Design principle		Information-centered	Intention-centered
Integration with AI		Learn information from data	Learn human intention from behavior data
Communication requirement		Moderate QoS	Stringent QoS
Computing requirement		Moderate	High
Technology	**Data collection**	Sensor-based	Sensor-based and software-based
	Networking	Things	Human behaviors
	Computing	Information-oriented	Human-intention-oriented
	Service provision	Preset-rules-based	In line with human intention
	Security and privacy	System failure, commercial confidentiality, ...	Personal information leakage, property damage, ...
Application		Smart home, smart transportation, smart health care, smart business, ...	
Quality of experience		Fair	Satisfactory
Development stage		Mature	In infancy

real-time and right services for human users. Therefore, IoB constantly collects users' behavior data via IoT devices of their whereabouts, and then transmits such data to IoB edge servers via communication technology, which usually invoke AI technology to infer users' intentions and transmit back instructions to those IoT devices for servicing users timely. Since users often move from one venue to another and IoB edge servers frequently interact with IoT devices of users' whereabouts, IoB usually relies on advanced wireless communication technology such as 4G/5G/6G that can meet stringent quality of service (QoS) requirements (e.g., high throughput, low latency, high reliability, and wide coverage). Furthermore, IoB constantly invokes AI technology to infer users' intentions, which requires data-intensive computing. IoB has huge application potential in smart home, smart transportation, etc. However, people seriously concern the security and privacy of their behavior data. Since people's behaviors might contain very sensitive and personal information, the leakage of these behavior data may cause harm to people themselves or lead to their property damage. To date, IoB is still in its infancy stage. It may take a long time to achieve IoB's applications in a wider scope and better solve the security and privacy issues.

In contrast, from the angle of the DIKW pyramid, IoT only establishes a connection from data to information. The reasons are as follows. Typically, IoT devices are deployed to collect environmental data for acquiring information. Therefore, the design of an IoT system centers around information acquisition and the IoT system usually works mechanically according to some preset rules. Roughly speaking, an IoT system connects IoT devices to collect data and then employs ingenious technology (e.g., AI) to reveal the information from the collected data, and finally provides services according to preset rules without considering human intention. In the Go-Home example, the signal that the door is unlocked, is interpreted as the information of Bob reaching home. Then according to the preset rule, the IoT system always and automatically switches on lights and air-conditioners, no matter what reason Bob goes home; as a result, the system might fail to meet Bob's needs and lead to energy waste. On the other hand, compared with IoB, IoT has moderate communication and computing requirements since IoT is not always required to provide real-time services, is relatively mature and has widely been applied in

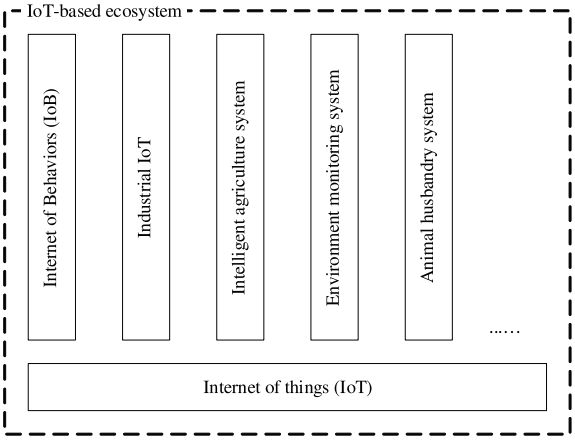


Fig. 5. An IoT-based ecosystem.

smart home, smart transportation, etc. In IoT, people generally concern the security and privacy of non-behavior data. For example, industrial IoT aims to utilize real-time industrial data, which are those of inventory, storage, distribution pace, and forecasted demand in industry (instead of human behavior data) and which are collected via interconnected sensors, instruments, and other devices, to improve the production levels of industrial applications including manufacturing and energy management. Since these industrial data contain the information of industrial systems, the leakage of these industrial data may lead to serious industrial production accidents and commercial confidentiality or financial damage.

In addition, IoB and IoT have different design goals. In IoB, human behaviors are networked together and uniquely identified by behavior IDs. By specifying the behavior-transition probabilities in a behavior network and searching behavior-transition chains, and applying AI technology, people may utilize IoB to infer users' intention. In contrast, in IoT, things are networked together and uniquely identified by IP addresses. By introducing routing algorithms to IoT, people may utilize IoT to transmit data.

Finally, IoB is built on IoT and is one system of the whole IoT-based ecosystem. It focuses on utilizing human behavior data to infer human intention and providing services in line with human intention. Besides IoB, there are other systems built on IoT, as shown in Fig. 5. For example, we have

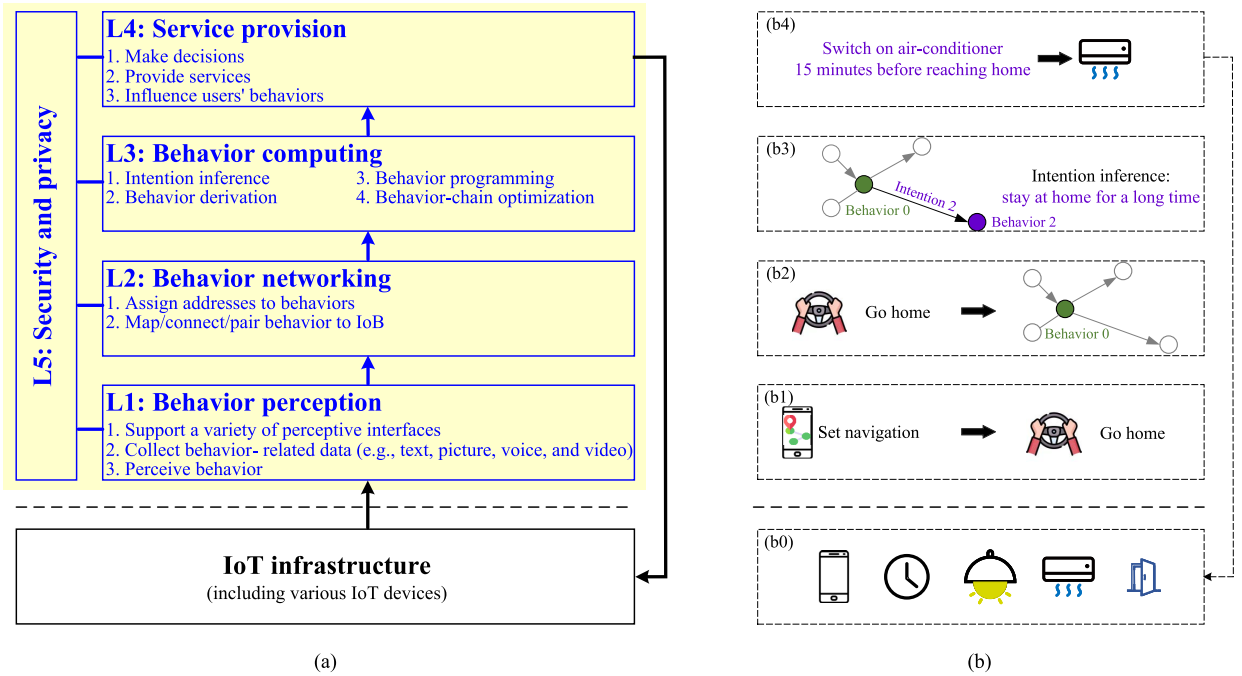


Fig. 6. Proposed 5-layer IoB architecture: (a) a generic model and (b) a Go-Home example.

industrial IoT [13], [14], [15] utilizing industrial data, intelligent agriculture systems [16], [17], [18] utilizing agricultural data, environment monitoring systems [19], [20], [21] utilizing environmental data, and animal husbandry systems [22], [23], [24] utilizing animal behavior data.

E. IoB Types

IoBs can be permissionless or permissioned. A permissionless IoB allows any user to join an IoB network and access/add/modify behaviors according to certain protocols. It has many participants and contains plenty of behaviors, thereby facilitating behavior computing (e.g., intention inference). However, this leads to slow synchronization and face security and privacy challenges, e.g., how to verify the reasonableness of newly added behaviors and how to prevent malicious tampering to IoB.

A permissioned IoB allows certain users to join an IoB network and restricts their rights to access/add/modify behaviors. It is usually used and maintained by individuals or organizations where only permissioned members are participants of IoB. It is more secure and reliable, because its level of security, authorizations, permissions, and accessibility is completely controlled by these individuals and organizations. However, it is not conducive to behavior computing, because it has a limited number of participants and contains a limited number of behaviors.

IV. IoB Architecture

To date, there is not an IoB architecture since IoB is in its very early stage. Both academia and industry do not have a comprehensive and clear understanding of IoB.

In this section, we propose a general IoB architecture shown in Fig. 6(a). It is a summary and refinement of existing IoB-related architectures, which are more compliant to IoT. Building on existing IoT infrastructures, it consists of five layers, namely, L1: behavior perception, L2: behavior networking, L3: behavior computing, L4: service provision, and L5: security and privacy. It implements the conversion from data to wisdom as shown in Fig. 1. Existing IoT infrastructures include three types of IoT devices: a) devices for sensing and collecting data, e.g., cameras, smart bracelets, and infrared sensors, b) devices for providing services, e.g., air conditioners, TVs, and lights, and c) devices for both sensing and service purposes, e.g., smartphones, and PCs. In our architecture, the first layer, i.e., the behavior perception layer, utilizes these existing IoT infrastructures to perceive/collect/extract human behaviors from human activities. On this basis, the other layers connect behaviors together, infer human intention, and finally invoke IoT devices to provide secure and intelligent services for human users. Next, we present each layer's functionalities of our IoB architecture and exemplify them by using the Go-Home example.

A. Behavior Perception

This layer collects users' raw data, e.g., signal, text, image, voice, and video, from IoT infrastructures, and then perceives and recognizes user behaviors using pattern recognition or AI algorithms.

In the Go-Home example, this layer collects Bob's data, e.g., navigation data, current time and location, from his automotive navigation system and smartphone, and then recognizes Bob's driving behavior to go home.

B. Behavior Networking

This layer assigns addresses to behaviors and maps/connects these behaviors to the corresponding IoB nodes. Roughly

speaking, it receives the perceived user behaviors from the lower layer, then calculates or assigns an address to each behavior according to an IoB addressing scheme to be explained later. Next, it checks if each perceived behavior has already existed in IoB according to the IoB address. If not, it adds the new behavior to IoB and chains the behavior to others according to their behavior transition relationships; otherwise, it associates the behavior with the corresponding IoB node for behavior computing. In the Go-Home example, this layer maps Bob's driving behavior to the Behavior-0 node of IoB.

C. Behavior Computing

Behavior computing refers to all computational operations for modeling, analyzing, understanding and predicting human behaviors, behavioral interactions and relationships, behavioral patterns, behavior generation and decomposition, etc. In this paper, we propose four forms of behavior computing: intention inference, behavior derivation, behavior programming, and behavior-chain optimization. Intention inference infers users' intention from their historical and current behaviors. With the inferred intention, we may use IoB to infer their next behavior from their current one, just like that we currently use Google Map to assist navigation. Behavior derivation generates new behaviors by existing IoB behaviors to reduce the learning time of new behaviors. Behavior programming assembles new behavior sequences according to existing IoB behaviors, to achieve a new purpose, e.g., choreograph a new dance routine by existing dance actions. Behavior-chain optimization searches behavior sequences that optimize user experience. In the Go-Home example, this layer infers that Bob would stay at home for a long time.

D. Service Provision

This layer makes decisions based on the results of behavior computing, and then provides intelligent services to users by invoking IoT devices, as well as influences users' behaviors. In the Go-Home example, if the behavior-computing layer infers that Bob would stay at home for a long time, this service provision layer switches on the air-conditioner 15 minutes before Bob reaches home, and next open the door and switch on the light when Bob reaches home.

E. Security and Privacy

Besides the functional requirements explained in the above four layers, IoB has many non-functional requirements, such as security and privacy, scalability, reliability, interoperability, usability, and maintainability. Since IoB involves collecting, storing, and processing human behavior data which might contain very sensitive and personal information, people raise great concern on the privacy and security of their behavior data. This greatly hinders the development and promotion of IoB. Therefore, in this tutorial paper, we mainly focus on the privacy and security issues of IoB.

This layer separates behavior and identity data, and makes private data (e.g., user identity and intention) safe. In the Go-Home example, this layer protects Bob's behavior (i.e., driving) and intention (e.g., the intention that Bob will stay at home for a long time).

V. IoB Enabling Technologies

In this section, we survey various IoB enabling technologies at each layer. In particular, we propose our approach to and insight into behavior networking, behavior computing, service provision, and security/privacy since most of them have not been studied yet.

A. Behavior Perception

In behavior perception, we collect behavior data from IoT infrastructures, then adopt some data-processing technologies (e.g., pre-processing, feature extraction, and classification) to process these collected data, and finally identify specific behaviors. Here, we consider two types of human behaviors: physical and online ones. The former involves actions generated by body parts or organs, e.g., eye, ear, mouth, hand, and leg. The latter involves actions over Internet-connected devices or online platforms, e.g., smartphone, laptop, PC, Quora, Twitter, and Facebook, for generating information that is delivered over the Internet. Below we present these perception methods of physical and online behaviors.

1) Physical-Behavior Perception: To perceive a physical behavior, we need to place perceptive devices in appropriate locations. According to the type count of perceptive devices for identifying specific behaviors, we may classify existing physical-behavior perception methods into two categories: single and multi-type-device-based perceptions. Tables II and III summarize these behavior perception studies of the two categories, respectively.

a) Single-type-device-based behavior perception: This method perceives human behaviors through a single type of perceptive devices, e.g., accelerometer or camera. Their placement varies with tasks. For example, Gao et al. [31] propose adopting multiple sensors of the same type to identify human physical activities of daily living, to study the impact of these activities on people's health. They recruit 8 community-dwelling older adults with various diseases; for each adult, four accelerometer-based sensors are placed in 4 body locations, i.e., the chest, left under-arm, waist and thigh. In their experiment, these adults perform 8 normal activities as shown in Table IV, each activity being repeated 3 times and hence totaling 192 activities. After obtaining these accelerometer signals, they first adopt signal pre-processing algorithms to eliminate the impact of sensor displacement and calibrate the signal dynamically, then extract signal features, e.g., mean, variance, spectral energy, and signal magnitude area. Finally, they feed these extracted features into five classifiers: artificial neural network (ANN) [65], decision tree [66], k-nearest neighbors (KNN) [67], naive Bayes [68] and support vector machine (SVM) [69], to identify each activity and compare classification accuracy.

In intelligent vehicles, to study the drivers' intention for helping generate assistant and collaborative control strategies,

TABLE II
SINGLE-TYPE-DEVICE-BASED BEHAVIOR PERCEPTION

Dev. type	Equip Pos.	Analysis: Pre-processing	Analysis: Feature extraction	Analysis: Behavior classification	Behaviors to perceive	Ref.	Scenario
accelerometer	head body arm back	normalization	high frequency(0.2s interval)	hidden Markov model	standing, walking, running	[25]	activity recognition
	back	segmentation	time and frequency domain, average, standard deviation	decision tree	lie, sit, stand, dynamic, standing, walk, run, cycle	[26]	
	chest thigh ankle	segmentation	mean, standard deviation, the skewness, the kurtosis	the multiple regression model with a hidden logistic process	stair descent, stair ascent, walking, sitting, standing up, sitting on the ground	[27]	
	hip wrist ankle arm thigh	filtering, segmentation	time and frequency domain, statistical functions	decision tree, KNN, NB, SVM	walking, running, cycling, standing still, sitting and relaxing, lying down and relaxing, stretching climbing, strength-training	[28]	
	ankle body leg	segmentation	sliding window, mean, variance, max, min, mean crossing rate, frequency domain, entropy	SVM	walking, cleaning table, cleaning window	[29]	
	pelvic	normalization	mean, standard deviation, energy, correlation	decision tables, SVM, decision trees, KNN, naive Bayes, hidden Markov models	standing, walking, climbing up stairs, climbing down stairs, sit-ups, vacuuming, running, brushing teeth	[30]	
	chest arm thigh waist	complex algorithms, light-weight algorithms	mean, standard deviation, variance	decision tree, KNN, naive Bayes, SVM	standing, sitting, lying, walking, transition	[31]	
	wrist	segmentation	time and frequency domain	random forest	sedentary, stationary, walking, and running	[32]	
	waist	segmentation, normalization	means	SVM	stand, sit, walking, lying, bent, sit to lying, lying to sit, bend down, bend up	[33]	healthcare
	waist wrist ankle arm chest knee	filtering	variance, entropy, frequency features	KNN	lying, preparing food, eating and drinking, socialising, reading, getting dressed, wiping, running, cycling, sitting	[34]	
	hip wrist	segmentation	summary statitics, sliding window	SWEM, SVM	lying down, sitting, standing, walking, running, basketball and dance, household chores	[35]	
	body	filtering, segmentation	min, max, mean, root mean square, variance, standard deviation, kurtosis, skewness, entropy, median, zero crossing rate, mean cross rate	WITS	sitting, standing, lying, walking, arm movement, kicking, crouching, falling, taking medicine eating, watching TV, reading magazine, cleaning table	[36]	smart home
camera	body	conversion into grayscale, Gaussian blur	hardware setup, input acquisition	CNN	motion, abnormal breathing	[37]	healthcare
	eye body head	filtering, segmentation	N/A	LSTM, ANN, Bayesian network, naive Bayes	gaze direction, body movement, head turning	[38]	smart transportation
	body eye hand head	filtering, segmentation	N/A	LSTM-RNN	body gestures, eye gaze, hand motion, head pose	[10]	

Xing et al. [10] propose utilizing 3 low-cost cameras (mounted inside the vehicle cabin) to collect the behavior of drivers during driving. The first camera is mounted in front of the driver to recognize the head pose and eye gaze of drivers. The second one is in the middle of the top of the front window to observe the outside traffic context. The third one is on the

TABLE III
MULTI-TYPE-DEVICE-BASED BEHAVIOR PERCEPTION

Sensing device	EquipPos.	Analysis			Behaviors to perceive	Ref.	Scenario
		Pre-processing	Feature extraction	Behavior classification			
accelerometer, GPS	body	filtering, segmentation	sliding window, mean, variance, energy	DT, K-Means, kNN, etc.	stationary, walking, running, biking	[39]	activity recognition
accelerometer, gyroscope, magnetometer	feet leg ankle chest thigh	filtering	sliding window, mean, standard deviation, variance, minimum, maximum	qualisys motion, capture system	stride, stance, swing, step	[40]	
	ankle	filtering	mean, standard, deviation	random forest	standing, lying	[41]	
	ankle	segmentation	sliding window	kNN, DT, SVM, ANN, CNN	cycling, running, jogging, etc.	[42]	
	arm	filtering, fusion, conversion, etc.	sliding window, mean, variance, kurtosis	SVM, naive Bayes, k-NN, HMM	opening window, drinking, etc.	[43]	
	thigh	trimming	mean, fast fourier transform magnitude, etc.	DT, multilayer perception, kNN, etc.	sitting, walking, jogging, etc.	[44]	
accelerometer, gyroscope	arm hip thigh wrist ankle	filtering	PCA, Karhunen-Loève transform	ANN, k-NN, nearest mean, SVM, etc.	standing, sitting, lying, walking, running, stair climbing, cycling	[45]	
	body	filtering	mean, autoregressive coefficients, etc.	ANN, SVM, DBN	standing, sitting, lying down, etc.	[46]	
	thigh	filtering	pooling layer, convolutional layer	CNN	walking upstairs, sitting, standing, laying	[47]	
	thigh wrist	N/A	time and frequency domains	stacked restricted Boltzmann machine, CNN, CNN-SVM, CNN-kNN	walking, jogging, up-stairs, downstairs, standing, sitting	[48]	
	ankle	normalization	time and frequency domain, entropy	Fisher Linear Discriminant Analysis, etc.	walking, stair-ascending, etc.	[49]	
	wrist	filtering, segmentation	mean, max, absolute values	mRMR , CMIM	brush, clean, cook, eat, etc.	[50]	
	hand	normalization	frequency domain	DSVM	dynamic gestures	[51]	
	ankle	filtering	ime and frequency domains, etc.	k-Mean, GMM, HIER	stair-ascending, walking, fall, etc.	[52]	
	feet shank thigh trunk	filtering, segmentation	comprehensive module	neural network, Bayes network, hidden Markov model, SVM, DT	walking, sitting, and standing still	[53]	
accelerometer, GPS gyroscope,	wrist	sampling	sliding window, mean, variance	ADL, IADL, Neural Network	stand, lean, lying, sit	[54]	health care
accelerometer, altimeter	wrist	WMA	clamping technique	SVM	feeding, brushing teeth, etc.	[55]	
	chest wrist	normalization	MRMR, NMIFS, Clamping, COM, FC	MLP, SVM, RBF	locations, brushing teeth, exercising, feeding, ironing, etc.	[56]	
accelerometer, gyroscope, BPS signals	ankle chest thigh wrist	resampling, filtering	time and frequency domain	kNN	stair ascent, stand, sit, lie down, drink, butter bread, cut food, don/doff shoe, peel carrot, etc.	[57]	
chewing sensor, piezoelectric sensor	neck	N/A	mean	CCS model	swallowing, chewing, talking, yawning, laughing	[58]	
ear-worn, accelerometer	ear	data augmentation	statistical	Gaussian mixture model, Bayes classifier	walking, lying down	[59]	
accelerometer, microphone	body	filtering	classification models	LR, Naive Bayes	voicing, walking, stationary, running	[60]	
cameras, IR sensors	body	filtering	mean, variance, kurtosis, skewness, entropy, zero crossing rate, etc.	KNN, SVM, conditional random rield, MLGL1, random forest, RMTL	sitting, standing, lying, walking, arm movement, kicking, crouching, falling	[36]	
camera, GPS	body	filtering	Lucas-Kanade	kNN, SVM, LogitBoost, HMM	drinking, walking, going upstairs and downstairs	[61]	
gyroscope, pressure sensor, accelerometer	thigh foot	removed spurious values	means, variances	random forest, SMO, LibSVM, naive Bayes, logistic regression, MLP	abduction, extension, sit-2-stand, gait, bipodal bridge	[62]	
gyroscope, accelerometer, magnetometer	body	filtering	frequency domain	machine learning techniques	fall, sleeping	[63]	
accelerometer camera	belt	filtering	average,standard deviation	6MWT	left and right steps	[64]	

sunroof to monitor the hand motion of drivers and the use of turn signals. The authors then develop a novel ensemble bi-directional recurrent neural network (RNN) model with Long Short-Term Memory (LSTM) units to deal with these collected drivers' behavior data and traffic context for learning their intention.

TABLE IV
SCENARIO ACTIVITIES [31]

Activity	Description
1	Sitting down and standing up from an arm chair
2	Sitting down and standing up from a kitchen chair
3	Sitting down and standing up from a toilet seat
4	Walking up and down stairs
5	Sitting down and standing up from a bed
6	Lying down and getting up from a bed
7	Getting in and out of a car seat
8	Walking 10 m

b) Multi-type-device-based behavior perception: This method perceives human behaviors by jointly adopting multiple types of perceptive devices, e.g., the combination of accelerometer and gyroscope. Compared with single-type-device-based behavior perception, this method may identify human behaviors more easily and accurately.

To cope with complex situations in practice and avoid the limitations of a single sensor, Wang et al. [50] propose adopting multiple sensors (e.g., wearable sensors and ambient sensors) for human activity recognition in healthcare. In their study, the authors recruit 21 old people. They wear a multi-sensor device, which includes, e.g., integrating accelerometer, gyroscope, magnetometer, barometer and temperature, on their wrist to collect the data of 17 daily activities including brush, clean, cook, eat, exercise, fall, iron, read, stand, walk, wash dishes, watch, and wipe. Ambient sensors (each being with a passive infrared sensor inside) are installed in rooms to collect the ambient information, e.g., their room-level location information. With these collected data, the authors then adopt four typical mutual-information-based feature selection methods to extract the features of these human activities, and next apply SVM to classify these activities. The study results show that the multiple-sensor approach can achieve high-precision activity recognition.

Bisio et al. [62] propose adopting inertial measurement units (each being a combination of gyroscopes and accelerometers) and pressure sensors to identify lower limbs activities in post-stroke patients for monitoring physical therapy. They collect data via 9 inertial measurement units on thighs and 4 pressure sensors on feet. They then perform the following steps for activity identification: pre-processing (to remove spurious values), windowing (to segment activity sequences), feature extraction, and classification.

2) Online-Behavior Perception: Online behaviors are generally generated by network-application software, e.g., WhatsApp, WeChat, and TikTok or online platforms, e.g., Twitter, Facebook, Weibo. Therefore, we can perceive these online behaviors by monitoring the network applications and online platforms. These online behaviors are shown in Table V and can be classified into two categories: online interaction among users and that between users and platforms.

a) Online interactive behaviors among users: Chen et al. [70] propose a cyber-physical system to describe human behaviors from both online and physical spaces for energy management in smart home. This system monitors social network applications, e.g., Facebook, Twitter, and WeChat to collect users' online interaction information, e.g., appointment time or location between users for understanding social behaviors in the virtual world. It also utilizes wearable sensors (e.g., motion sensors, GPS, and biosensors) to track users' movements, locations, and sensations in the real world. According to this collected information, the system can discover the patterns of power use and cognitively understand the users' behaviors, and further infer users' demands for electricity, optimize the energy scheduling in smart home, as well as respond to users' demands and electricity rates timely. A system prototype is implemented to verify the effectiveness of the proposed system.

In online social networks, cybercriminals may exchange cyberattacks knowledge with each other, while cybersecurity experts may share their criminal-investigation cases and network protection measures. Aslan et al. [73] propose a two-stage approach to extract these online interactive behaviors and identify the related accounts in Twitter, for increasing people's understanding about evolving cyberattacks and cybersecurity. They first extract 3 types of online behavior features as shown in Table VI: profile feature, e.g., # of friends and followers; behavioral feature, e.g., # of tweets and retweets; and content feature, e.g., lexical diversity and prototypical words. They then feed these extracted features to classifiers, e.g., decision tree, random forest and SVM, for automatically identifying cybercriminal accounts and cybersecurity expert accounts. In their experiment, they invoke Twitter APIs to crawl the tweets of 424 accounts (consisting of 212 cyberattacks-discussed accounts and 212 ordinary accounts), with 3,200 tweets per tweet account. Their results show that their approach could achieve an account-detection accuracy of more than 95.

b) Online interactive behaviors between users and system: Gochhayat et al. [75] propose a lightweight context-aware IoT service architecture (LISA) to support IoT push services. LISA can collect real-time online and physical behaviors of a user (e.g., google search and location change) and the past queries (made by other users), then understand the context of the user (e.g., the real-time requirements in current environments), and finally filter and forward the most important and relevant services to the user. They consider a use case of IoT tourist guide to simulate LISA. They then adopt two metrics, namely, precision and recall, to evaluate the performance of LISA. Here, the precision measures the ratio of the extracted relevant services to the total services, while the recall refers to the fraction of the retrieved services in the existing relevant services. Their preliminary experimental results confirmed that LISA could successfully select the most relevant information to the user, thereby reducing unnecessary information greatly. For example, LISA can extract the relevant services from 15000 services with precision up to 0.3 and recall up to 0.8.

Zhang et al. [77] propose an RNN-based approach to predict users' advertisements (ads) clicking behaviors for sponsored search in which search engines such as Google charge the advertisers according to the click-through rate of their ads. They first collect users' online ad-clicking sequential behaviors from a commercial sponsored search system, and extract 3 types of features: advertisement feature, e.g., ad ID, ad

TABLE V
ONLINE-BEHAVIOR PERCEPTION

Online behavior types	Online behaviors	Tool	Ref.	Scenario
Online interaction behaviors between users	Chat on Facebook, Twitter, and WeChat	Smartphone	[70]	Smart home
	Twitter-communication	World Wide Web-enabled devices	[71]	N/A
	Exchange useful information	Handsets	[72]	N/A
	Exchange cyberattacks knowledge	Twitter	[73]	N/A
	Phone calls friends	Smartphone	[60]	Health care
Online interactive behaviors between users and system	Users browsing pages, adding items to cart, buying	e-commerce website	[74]	Smart business
	Search and location	Website	[75]	Smart business
	Browsing,searching, product click, purchase	e-commerce site	[76]	Smart business
	Ad-clicking	Modern commercial web search engines	[77]	Smart business
	Clicked items	e-commerce site	[78]	Smart business
	Click, add-to-favorite, add-to-cart, purchase	e-commerce platforms	[79]	Smart business
	Click	e-commerce site	[80]	Smart business
	Clicking item, viewing item	e-commerce site	[78]	Smart business

TABLE VI
A LIST OF FEATURES [73]

Profile features	Behavioral features	Content features
number of alphabetic characters	number of tweets	lexical diversity
number of numeric characters	number of retweets	Flesch-Kincaid score
number of capitalization	average number of hashtags	SMOG index
number of friends	average number of urls	prototypical words
number of followers	average time between tweets	weirdness score
friends/followers ratio	standard deviation between tweets	tf-idf
use of the avatar picture	fraction of tweets posted in 24 hours	
presence of location		
length of user name		

display position, and ad text relevance with query, user feature, e.g., user ID and query related semantic information, and sequential feature, e.g., time interval since the last ad-clicking, dwell time on the landing page of the last click event, and whether the current ad is the head of sequence. They then adopt RNN to capture the dependency of these sequential behaviors for predicting whether to click an ad for each ad view. Experimental results show that their approach could achieve a higher accuracy of the click prediction than sequence-independent approaches such as Logistic Regression and Neural Networks.

3) Time of Behavior Perception: To perceive a physical or an online behavior, behavior perceptors need time to collect and analyze behavior data. Behavior data can be sampled using sensors, cameras, etc. Sampling frequency (or sampling interval) is an important parameter to perceive a behavior accurately. To fully recover continuous-time signals on behavior data (e.g., blood pressure), the sampling frequency should meet the Nyquist–Shannon sampling theorem [81]. However, if we only obtain statistics of behavior data, we should carefully choose the sampling frequency. Maintaining a high sampling frequency increases the accuracy of statistics estimation but also increases computing load and power consumption, and vice versa. Therefore, we should optimize the sampling frequency to balance the accuracy and cost. Gao et al. [31] adopt multiple sensors to identify human physical activities of older adults with various diseases as shown in Table IV. They first sample behavior data via these sensors to obtain their statistical features (e.g., mean and variance) and then feed these features to machine learning algorithms for behavior perception. They find that a sampling frequency of 20 Hz can achieve a good balance.

In general, people often adopt machine learning algorithms to analyze behavior data. Training a machine learning model is usually time-consuming but applying an already trained model to identify a behavior is usually fast. Depending on particular applications (e.g., real-time interaction or non-real-time applications), people may perform an online or offline analysis of behavior data. For example, Gochhayat et al. [75] construct a tourist guide system to quickly respond to tourists' queries such as bus routes, restaurants, and hotels, based on their real-time online and physical behaviors (e.g., google search

TABLE VII
AN EXAMPLE OF IOB ADDRESS

Behavior Name	Behavior Address				Meaning
	Time	Venue	Type	Mode	
EXER_AER_1	00	00	00	01	Do aerobic exercise at home in morning
EXER_FLE_1	00	00	00	10	Do flexibility exercise at home in morning

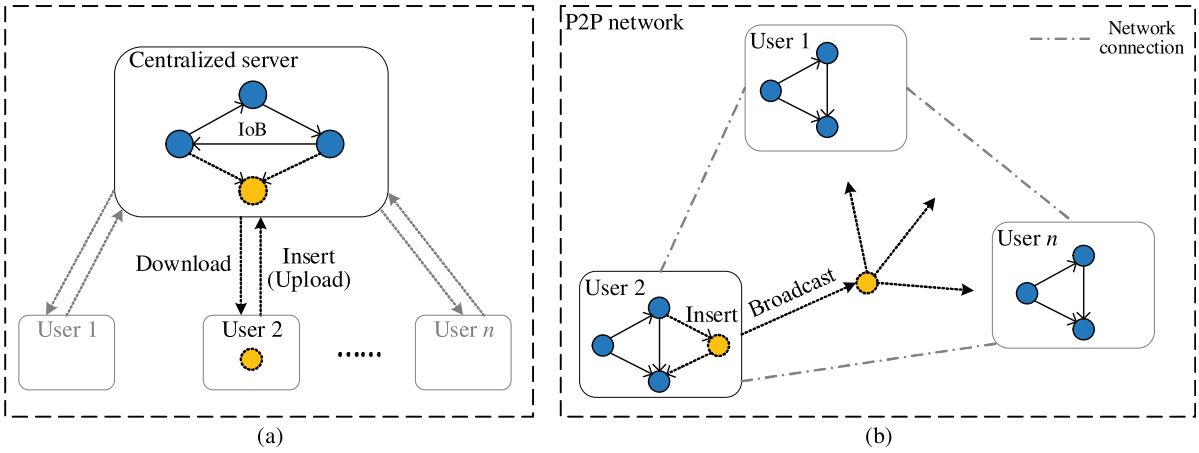


Fig. 7. IoB maintenance: (a) a centralized approach and (b) a decentralized approach.

and location change). This requires the system to perform an online analysis. In contrast, Aslan et al. [73] propose to identify cybercriminal accounts and cybersecurity expert accounts in Twitter, by analyzing Twitter messages. They only need to perform an offline analysis.

B. Behavior Networking

In IoB, we may use a directed cyclic graph to connect human behaviors together, where a vertex represents a behavior and is identified by an IoB address and a directional edge represents a behavior transition. Below, we propose an approach for IoB address representation and IoB maintenance.

1) IoB Address: Nyman proposes borrowing the IPv6 address representation method to code a behavior, because a huge address space (as IPv6 uses 128-bit addresses) is required to identify each of numerous human behaviors uniquely. However, an IPv6 address, which mainly contains network-ID and host-ID fields, is designed to connect routers or hosts, but it cannot be used to represent the semantics and the correlation relationship of behaviors. In this paper, we propose that an IoB address, representing a human behavior, should reflect these semantics and correlation attributes and support behavior search and computing. A comprehensive study of an IoB address representation, which may take into account a professional field, address classification, address length, etc, is beyond the scope of this paper.

With the help of Table VII, we give an example of IoB addresses. In our example, an IoB address contains the occurring time and venue, action type, and action mode of a human behavior (which is defined in specific environments) and is coded with the principle of "the more relevant the behavior, the closer the address code". For example, EXER_AER_1 means the behavior of "do aerobic exercise at home in morning", while EXER_FLE_1 means the behavior of "do flexibility exercise at home in morning". The two behaviors have the same attributes of time, venue and action type, so we use "00 00 00" to represent them. However, the two behaviors have different action modes and hence we use "01" and "10" to represent their different attributes, respectively.

2) IoB Maintenance: In IoB, we often need to insert, delete, modify, and search a behavior. In general, there are two ways to maintain the consistency of these IoB operations: centralized and decentralized IoB maintenance.

Centralized IoB maintenance: This approach adopts a master-slave architecture shown in Fig. 7(a). A user may send the centralized server a request of a new behavior insertion. The latter then validates the request permission of the former and the rationality and correction of the insertion operation, and finally executes the insertion operation after successful validation.

Decentralized IoB maintenance: This approach adopts a decentralized architecture shown in Fig. 7(b). Each user is an equal participant, holds a full copy of the IoB network, and keeps synchronizing it with other users; and all these users compose an underlying IoB P2P network. This approach needs an incentive mechanism to encourage users to update their newly found rational behaviors and also needs a consensus mechanism to reach an agreement on an updated behavior among all users. When users request an update operation (e.g., insert, delete, and modify) of a behavior, they should broadcast this request with a proof of this operation's rationality to the P2P network. If all users can reach an agreement on the proof, they execute this operation in their respective local IoB networks.

C. Behavior Computing

Behavior computing involves various computing operations of modeling, analyzing and understanding human behavior

TABLE VIII
ADVANTAGES AND DISADVANTAGES OF FOUR INTENTION-INFERENCE APPROACHES

Approaches	Advantages	Disadvantages
Traditional machine learning	• Its model training is fast. • It requires low computing resources.	• It is difficult to learn and process behavioral sequence data. • It is difficult to infer human intention from complex behavioral networks.
Deep learning	• It can infer human intention from complex behavioral networks. • It can extract information from human behavior sequences (e.g., RNN) and from visual imagery (e.g., CNN). • It can better learn information from layer-crossing features.	• It requires large amounts of training data. • It is sensitive to model parameter settings. • It requires expert experiences to adjust model parameters. • It usually has complicated model structure and requires more time to train.
Inverse reinforcement learning	• It can infer human intention from human behavior data and imitate human behavior. • It can improve model accuracy by long interactions with human beings.	• It is not easy to collect expert data. • It requires high computing resources. • It is difficult to set model hyperparameters.
Cognitive model	• It can infer human intention by commonsense knowledge. • It does not require large amounts of training data.	• Its model is inaccurate and incomplete.

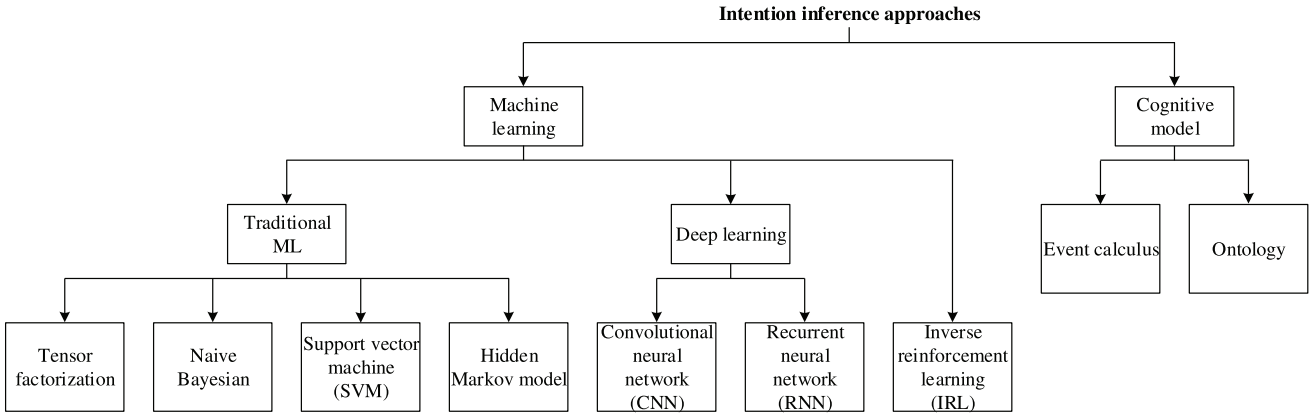


Fig. 8. A taxonomy of the approaches for intention inference.

and their interactions. We propose four forms of behavior computing: intention inference, behavior derivation, behavior programming, and behavior-chain optimization; and present their implications and approaches. We also survey existing approaches to intention inference.

1) Intention Inference: Intention is a mental state that "represents a commitment to carrying out an action or actions in the future" [82]. Intention inference involves inferring human intention according to a series of human behaviors. In general, it is not easy to "guess" human intention by isolated behaviors. In IoB, the behaviors of people are connected to form a network; by following a chain of behaviors, we can infer human intention more easily and accurately. Human intention is hidden in human behaviors and varies with the context (e.g., time-space-dependency and scenario of behavior occurrence), even for the same behavior. Therefore, intention inference should take into account the context.

Intention mining and behavior analysis [83] and [84] are related to intention inference since they aim at analyzing human intention, but they are very different. The studies of the former often adopt an offline analysis approach, in which human behaviors are recorded in text and intention mining is based on these static text data. In contrast, the studies of the latter, which adopts a real-time analysis approach, infer human intention for predicting the next behavior timely by real-time physical behaviors (which are usually tracked through video) or online behaviors (which are usually tracked through software). There are already some surveys on the former. For example, Rashid et al. [83] and Hamroun and Gouider [84] have surveyed the studies that mine human intention from collected social data from Facebook, Twitter, search engines, etc., in terms of intention types, problems addressed, datasets exploited, features selected, approaches applied, and experimental results obtained. Date [85] and Diaz et al. [86] have surveyed the work on intention mining in the information systems engineering field. However, to the best of our knowledge, there is not a survey on the intention inference. Here, we classify existing approaches of inferring human intention timely into four types: (a) traditional-machine-learning-based approach, (b) deep-learning-based approach, which is the current dominant solution, (c) inverse-reinforcement-learning-based approach, which is widely considered as a promising approach for intention inference and definitely worth exploring, and (d) cognitive-model-based approach. Table VIII compares the advantages and disadvantages of these four approaches. Table IX summarizes existing intention inference studies, in terms of inference approach, human behavior, human intention, application domain, and inference accuracy. Fig. 8 hierarchically lists specific algorithms for intention inference.

a) Traditional-machine-learning-based approach: Early studies widely adopted traditional machine learning algorithms for intention inference, including SVM [88], [104], ANN [91], Bayesian network [91], naive Bayes classifier [91] [112],

TABLE IX
INTENTION INFERENCE APPROACHES (①: TRADITIONAL ML, ②: DEEP LEARNING, ③: INVERSE REINFORCEMENT LEARNING, ④: COGNITIVE MODEL)

Ref	Inference approach	Human behavior	Human intention	Application domain	Inference accuracy
[10]	②: EBiLSTM	Physical behavior: head, eye behavior	Driver lane change intention	Smart transportation	96.1%
[11]	②: AT-BiLSTM				93.33%
[87]	①: Sparse Bayesian				N/A
[88]	①: RVM				80%
[89]	④: ACT-R				90%
[90]	④: ACT-R				N/A
[91]	①: ANN,BN,NBC				N/A
[92]	③: Maximum entropy IRL				human likeness: 2.066m
[93]	②: Convolutional LSTM	Physical behavior: head, body, eye behavior	Pedestrian intention to Cross the Road	Smart transportation	79%
[94]	②: RNN				79.1%
[95]	②: T-Pose-LSTM				N/A
[96]	①: Particle Filter				N/A
[97]	①: LDCRF				N/A
[98]	②: CNN				94%
[99]	①: DBN				N/A
[100]	③: IRL				N/A
[8]	④: Event calculus	Physical behavior	Human activity intentions	Smart Healthcare	N/A
[9]	②: GRU and LSTM				GRU:81.7%,LSTM:79.8%
[101]	②: CNN				N/A
[102]	①: HMMR				97%
[103]	④: Ontology				N/A
[104]	①: SVM				86.2%
[105]	②: ARNN and ITFM	Online behavior	User future intention in online system	Recommender systems	N/A
[79]	①: Tensor Factorization				N/A
[106]	②: LSTM				N/A
[78]	②: RNN				N/A
[107]	③: IRL				89.1%
[108]	②: RNN	Physical behavior	Human intention in Human-Robot Cooperation	Human-robot interaction	N/A
[109]	①: Gaussian process				83.8%
[110]	②: ATCRF				75.4%(1s),69.2%(3s)
[111]	④: Semantic representation				85%
[112]	①: NBC				N/A
[113]	②: R-CNN				N/A
[114]	③: IRL				88%

dynamic Bayesian network [99], hidden Markov model [102], tensor decomposition [79], Gaussian process [109], and particle filter [96].

For example, in [91], Lethaus et al. develop a machine-learning-based dynamic driving simulator to infer driving intention (e.g., driver's lane change) according to the gaze behaviors of drivers. Drivers often observe surrounding environments with their eyes before they perform some driving action to achieve their driving intention. Different driving intentions may trigger different gaze patterns. With the driving simulator, the authors first collect drivers' gaze behaviors before and during the driving action, and then input these gaze data to ANNs, Bayesian networks, and naive Bayes classifiers for predicting the occurrence of certain driving action and further inferring the driving intention.

Schulz and Stiefelhagen [97] propose adopting Latent-dynamic Conditional Random Fields to infer pedestrian intention, e.g., whether to cross a road, in advanced video-based driver assistance systems. The proposed scheme is an extension of conventional Conditional Random Fields [115] by adding a layer of hidden latent states. This added layer may well extract an intrinsic structure and feature dependency of human behaviors; for example, in the real world, when crossing a road, a pedestrian often turns his head in the direction of oncoming vehicles to learn about traffic situations. In this study, the authors use the video data of the pedestrian dynamics and situational awareness to train their model. Experiments show that the proposed scheme outperforms SVM and random forests in terms of prediction stability and accuracy in pedestrian intention recognition.

In [109], Wang et al. propose an Intention-Driven Dynamics Model to infer human intentions in human-robot interaction field. It can efficiently represent high-dimensional and noisy observations in a low-dimensional form and find simpler representations of human motions. The authors train their model based on observed human motions. Further, they integrate an efficient approximation inference algorithm in their model to infer human intention from ongoing movements. They verify the feasibility of their model in two scenarios: target inference in robot table tennis as well as action recognition for interactive humanoid robots.

b) Deep-learning-based approach: Deep-Learning-based approaches outperform the traditional-machine-learning-based ones when there is a huge amount of data. Also, they can better learn features and handle complex problems and therefore have been widely used for intention inference.

RNN is a mainstream deep learning model for analyzing sequential data. Unlike traditional ANN, RNN can learn the correlation between sequential data at different time. Hence, it is an effective model for intention inference according to time sequences of behaviors, position, events, etc. [77], [78], [94], [105], [108]. For example, Zhang et al. [77] propose an RNN-based approach to infer whether users click an ad for each ad view. Users' ad clicking behaviors are closely related to those in their past time, e.g., what queries they submitted, what ads they clicked or ignored, and how long they spent on

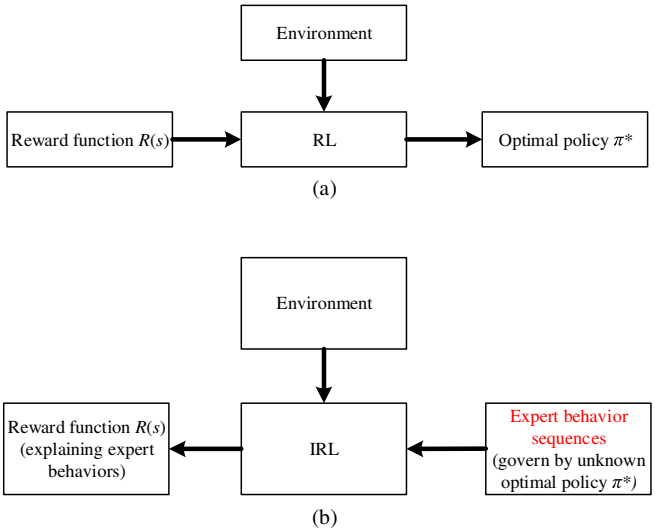


Fig. 9. Frameworks of (a) reinforcement learning (RL) and (b) inverse reinforcement learning (IRL).

the clicked ads' landing pages. Zhang et al. [77] then model the dependency on a user's sequential behaviors with the recurrent structure in RNN for ad-clicking inference. Experimental results show that their approach can significantly improve the accuracy of ad-clicking inference than sequence-independent approaches.

However, RNN has difficulty in learning long-range dependencies, i.e., the correlations at longer time scales. To overcome it, studies [10], [11], [93], [95], [106], [116] have proposed many RNN's variant structures such as long short-term memory (LSTM) and gated recurrent units (GRU). For example, in [10], Xing et al. propose an ensemble bi-directional LSTM model to infer a driver's lane change intention. They input driver behaviors data (e.g., head pose, eye gaze, and body movement) and traffic information (e.g., lane markings, surrounding vehicle positions and digital maps) to train their model for predicting driving actions and inferring driving intention. Experimental results show that the proposed model can infer driving intention with an average accuracy of 96.1%, 0.5 seconds before the driving action starts.

CNN is a class of deep neural networks. It is most commonly used to analyze visual imagery in various domains (e.g., autonomous driving [117], smart home [98], and human-robot interaction [113]), since it can significantly downscale massive data volumes while preserving image properties. For example, in autonomous driving, Abughalieh and Alawneh [98] propose adopting CNN to assist in inferring pedestrian intention of crossing road for reminding drivers of giving their way to pedestrians. They first utilize a publicly available CNN model called YOLO [118] to detect pedestrians from vehicle video, and then construct another CNN model to extract body landmarks (e.g., shoulder, neck and nose) of pedestrians. They next estimate the body orientation of a pedestrian utilizing these body landmarks and the depth information collect by depth-sensing cameras, and finally recognize the road-crossing intention by a sudden change of body orientation toward the road.

c) Inverse-reinforcement-learning-based approach: Inverse reinforcement learning (IRL) [119] has widely been considered as a promising approach that enables machines to understand human behavior and infer human intention [92], [100], [107], [114], [120], [121], [122]. Below, we first introduce the IRL framework and then present examples that apply IRL for intention inference.

Before introducing IRL, we first explain RL [123], [124], [125] briefly. In RL, as shown in Fig. 9(a), an agent learns how to behave in an environment by interacting with the environment. In other words, the agent performs a behavior (or action) on the environment, then observes the state (i.e., the situation in the environment) and receives a reward that signals how good this behavior is, and next decides how to better perform its next behavior. The RL goal is: given a reward function, the agent learns an optimal policy that specifies which behavior should be taken in each state so as to maximize the total rewards. For example, when you drive in a road, the state refers to the position and speed of your car and neighboring cars, the behavior refers to turning the steering wheel, braking, etc., and the reward rates how quickly you reach home safely. Your goal is to learn the best driving behaviors in each driving situation.

In contrast, IRL, as shown Fig. 9(b), reverses the problem of RL. The goal of IRL is: given trajectories or optimal behavior sequences of an expert, which are generated by his optimal but unknown policy, the agent learns the reward function of the expert. In IRL, the reward function is regarded as the most concise, robust, and transferable definition of a task (i.e., learning to drive), which governs desire behaviors of the expert; once obtaining the right reward function, the agent can learn the optimal policy by RL. For example, assume that an agent wants to learn to drive from an experienced driver named Ben. Ben has an optimal policy in her/his mind, which produces her/his driving behavior. Meanwhile s/he has a reward function in her/his mind, which is formed during her/his driving in roads and which rates her/his deriving behaviors. The agent may first learn the reward function from Ben's driving behaviors by IRL and then learn the optimal policy by RL. In IRL, the reward function rates an expert's behaviors and hence implies the understanding of behaviors. The policy specifies which behavior should be taken in each state and hence infers the intention of the expert.

The core task of IRL is to approximate the reward function. Consider an expert and an IRL agent. Below, we first define the following variables and then explain the algorithm that the agent learns the reward function of the expert.

$\pi : (a, s) \rightarrow [0, 1]$ denotes a policy, which is a probability that the agent performs behavior a when it is in state s.

π^* denotes the expert's optimal policy.

π_j denotes the j^{th} policy.

$\mathbf{\Pi} = \{\pi_1, \pi_2, \ldots, \}$ denotes a set of policies used to approximate π^*.

$T^* = \{(s_i^*, s_i^*)|i = 0, 1, 2, \ldots, m\}$ denotes the expert's trajectory, which is produced by π^*.

$T_j = \{(s_i^j, s_i^j)|i = 0, 1, 2, \ldots, m\}$ denotes the j^{th} trajectory which is produced by π_j.

$\boldsymbol{T} = \{T_1, T_2, \ldots, \}_j$ denotes a set of finite trajectories.

$\hat{\cdot}$ denotes the estimated value of $\cdot$.

In IRL, We use a liner approximation $R_{\boldsymbol{\alpha}}(s)$ for the expert's reward function $R(s)$.

$$\begin{aligned} \boldsymbol{\alpha} &\triangleq \langle \alpha_1, \ldots, \alpha_d \rangle \\ R_{\boldsymbol{\alpha}}(s) &\triangleq \alpha_1 \phi_1(s) + \alpha_2 \phi_2(s) + \cdots + \alpha_d \phi_d(s) \approx R(s) \end{aligned} \quad (1)$$

where $\phi_i(s), i = 1, \ldots, d$ are given basis functions, s is an input state, and $\boldsymbol{\alpha}$ is a vector of undetermined coefficients $\{\alpha_i\}$. To estimate the unknown $\boldsymbol{\alpha}$, we resort to the value function. Let $V^{\pi}(s)$ denote a long-term value of state s for a given π. According to the relationship between the value function $V^{\pi}(s)$ and the reward function $R(s)$ as well as (1), we use a liner approximation $V_s^{\pi}(\boldsymbol{\alpha})$ for the value function $V^{\pi}(s)$. From [119], we can express $V_s^{\pi}(\boldsymbol{\alpha})$ as:

$$\begin{aligned} V_s^{\pi}(\boldsymbol{\alpha}) \triangleq \alpha_1 V_1^{\pi}(s) + \alpha_2 V_2^{\pi}(s) \\ + \cdots + \alpha_d V_d^{\pi}(s) \approx V^{\pi}(s). \end{aligned} \quad (2)$$

where the i-th basis function $V_i^{\pi}(s), i = 1, 2, \ldots, d$, represents the value of state s for a given policy π and can be expressed in terms of $\phi_i(s), i = 1, \ldots, d$. We may calculate $V_i^{\pi}(s)$ by the trajectories with initial state s. Next, we initialize $\boldsymbol{\Pi} = \{\pi_1, \ldots, \pi_k\}$ by k random policies and initialize $\boldsymbol{T} = \{T_1, \ldots, T_k\}$ by these random polices. With $\boldsymbol{\Pi}$ and $\boldsymbol{T}$, we are ready to present the IRL algorithm to learn the expert's approximate reward function, $R_{\boldsymbol{\alpha}}(s)$, by following the steps:

1) Express $V_{s_0^*}^{\pi^*}(\boldsymbol{\alpha})$ for π^* according to T^* and (2).
2) Express $\{V_{s_0^j}^{\pi_j}(\boldsymbol{\alpha})\}$ for $\pi_j \in \boldsymbol{\Pi}$ according to T_j and (2).
3) Estimate $\boldsymbol{\alpha}$ by solving the following linear program:

$$\begin{aligned} \hat{\boldsymbol{\alpha}} = \arg\max_{\boldsymbol{\alpha}} \sum_{j=1}^{\|\boldsymbol{\Pi}\|} \left(V_{s_0^*}^{\pi^*}(\boldsymbol{\alpha}) - V_{s_0^j}^{\pi_j}(\boldsymbol{\alpha}) \right), \pi_j \in \boldsymbol{\Pi} \\ \text{s.t.} \quad |\alpha_i| \leq 1, \quad i = 1, \ldots, d \end{aligned}$$

4) Estimate $R_{\hat{\boldsymbol{\alpha}}}(s)$ according to $\boldsymbol{\alpha}$ and (1).
5) Learn a new policy π_{k+1} by RL and $R_{\hat{\boldsymbol{\alpha}}}(s)$. Add π_{k+1} to $\boldsymbol{\Pi}$.
6) Produce a new trajectory T_{k+1} according to π_{k+1}, and add T_{k+1} to $\boldsymbol{T}$.

Repeat Steps 2 to 6 until, e.g., a number of iterations is reached or $R_{\hat{\boldsymbol{\alpha}}}(s)$ has changed little iterations.

Once obtaining $R_{\hat{\boldsymbol{\alpha}}}(s)$ that approximates the expert reward $R(s)$, we can estimate $\hat{\pi}^*$ that approximates the optimal policy π^* by RL.

IRL has received growing concerns [92], [100], [107], [114], [120], [121], [122] and has widely been adopted for intention inference. For example, Das and Lavoie [120] use IRL to understand human behavior change in response to feedback on social media. Luceri et al. [107] use IRL to infer incentives that may steer online user behaviors. Rhinehart and Kitani [114] propose an IRL-based algorithm called Discovering Agent Rewards for K-futures Online to forecast long-term goals of a user: what the user will do, where they will go, and what goal they seek, from first-person visual observations of the user's daily behaviors. Huang et al. [92] propose a maximum entropy IRL framework to model driving behaviors using naturalistic human driving data. The framework consists of two models. One is an internal reward function-based driving model that emulates the human's decision-making mechanism. To infer the reward function parameters, this model converts the continuous behavior modeling problem to a discrete one; specifically, it adopts a polynomial trajectory sampler to generate candidate trajectories by considering high-level intentions. Another is an environment model that considers interactive behaviors between the ego and surrounding vehicles to better estimate the generated trajectories. The authors apply the proposed framework to learn personalized reward functions for individual human drivers from the NGSIM highway driving dataset. Their results show that the learned reward functions can explicitly express the preferences of different drivers and interpret their decisions.

d) Cognitive-model-based approach: Cognitive-model-based approaches adopt commonsense-knowledge rules to infer human intention and therefore do not require large amounts of training data as in deep-learning models. However, they tend to be inaccurate and incomplete, and hence were only used in a few scenarios in the past. For example, for smart transportation, Salvucci et al. propose adopting a cognitive model called Adaptive Control of Thought-Rational to facilitate understanding of driving behaviors and infer the lane-changing intention of drivers [89], [90]. For smart healthcare, Kim et al. adopt an event-calculus approach to assist in the independent living of elderly people in residential space. They utilize the commonsense law of inertia to infer the activity intention of an elderly person [8]. OpenCyc is the world's largest and most complete commonsense knowledge base that contains more than 300,000 concepts [126]. Ruan build an OpenCyc-based system for elderly people activity intention recognition [103].

2) Behavior Derivation: Behavior derivation is a function that derives a new behavior from an existing behavior in IoB by utilizing the similarity of behaviors. Let $d : B_1 \rightarrow B_2$, be a behavior derivation function that maps an element x in an existing behavior set B_1 to an element y in a derived behavior set B_2, i.e.,

$$y = d(x), \ x \in B_1 \text{ and } y \in B_2. \quad (3)$$

With the behavior derivation, we can derive a new behavior node in advance, rather than waiting someone to add the new behavior to the IoB later. In this way, we can utilize IoB to provide more timely services.

Consider a simple example of deriving a behavior node "cut the oranges" from an existing node "cut the apples" as shown in Fig. 10. This example looks intuitive for a human, but it is difficult for a computer. In Fig. 10(a), we have a sequence of three behaviors, i.e., "put apples on a chopping board", "pick up a knife", and "cut the apples". In Fig. 10(b), assume that we have a sequence of two behaviors, i.e., "put oranges on a chopping board", and "pick up a knife"; we may safely derive a new behavior, y="cut the oranges", from x="cut the apples", according to the similarity of the preceding two behaviors in Figs. 10(a) and (b), and the similarity of their semantics that apples & oranges are both fruits and are put on a chopping board.

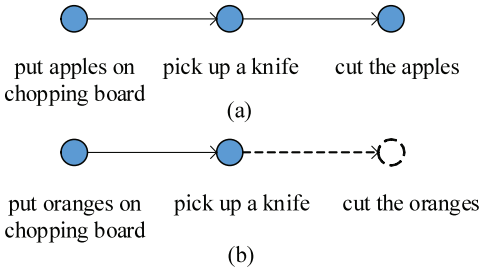


Fig. 10. An example of behavior derivation: (a) an existing behavior sequence of "cut the apples"; (b) the derived behavior sequence of "cut the oranges".

3) Behavior Programming: Behavior programming is a function that constructs a new behavior sequence by combining or shuffling existing behavior sequences in IoB. Let $p : S_1 \rightarrow S_2$, be a behavior programming function that maps an element $\boldsymbol{x}$ in an existing behavior-sequence set B_1 to an element $\boldsymbol{y}$ in a new behavior-sequence set B_2, i.e.,

$$\boldsymbol{y} = p(\boldsymbol{x}), \boldsymbol{x} \in S_1 \text{ and } \boldsymbol{y} \in \mathrm{S}_2. \tag{4}$$

We give the following example to explain how behavior programming can help Bob avoid being late for work. Assume that in the morning, Bob's one typical behavior sequence is

$$\boldsymbol{x} = (\text{take a shower}, \text{ have breakfast}, \text{ read news}, \text{ go to office}).$$

However, one morning, Bob's IoB system predicts that there may be traffic jams. In this case, the behavior programming may transform the behavior sequence $\boldsymbol{x}$ to $\boldsymbol{y}$:

$$\boldsymbol{y} = (\text{take a shower}, \text{ go to office}, \text{ have breakfast}, \text{ read news}).$$

Then, the IoB system suggests that Bob goes to his office right after taking a shower, so as to avoid being late for work.

4) Behavior-Chain Optimization: Behavior-chain optimization is a function of finding the optimal behavior sequence that realizes a user's intention with the lowest cost measured in, e.g., time, and money. Given an initial behavior and an target one in IoB, let l denote one candidate behavior sequence. Let $c_i(l)$ denote the ith type of cost associated with l. Let w_i denote the weight of the ith type of cost. Taking into account multiple types of costs together, we may find the optimal behavior sequence, i.e.,

$$l^* = \underset{l \in \{l_1, \ldots, l_k\}}{\arg\min} \sum_i w_i c_i(l) \tag{5}$$

Now, consider an example of electricity bill payment in Fig. 11. When Bob decides to pay the bill, from the IoB network, he has three options: the first one, l_1, consisting of setting up navigation, driving, parking, queuing and paying; the second one, l_2, consists of searching bank locations, walking to a bank, queuing, and paying; and the third one, l_3, consisting of searching the website, logging in and paying. Assume that we only consider the cost of time. Let $c(l)$ denote the time that Bob spends when it follows the behavior sequence l. We may express the optimal behavior sequence as:

$$l_3 = \underset{l \in \{l_1, l_2, l_3\}}{\arg\min} \; c(l) \tag{6}$$

In this example, l_1, consisting of the maximum number of steps, could consume a lot of time in case of traffic jams; l_2, although consisting of fewer steps than l_1, generally consumes much time for Bob to walk to the bank for the payment; and l_3, consisting of the minimum number of steps, is the most convenient one especially when Bob is familiar with online payment. Therefore, the optimal behavior chain is l_3 as shown in (6).

D. Service Provision

According to the inferred human intention, this layer may invoke appropriate IoT-based hardware/software to provide services for meeting the intention. It includes two aspects: providing services and influencing user behaviors. For example, in intelligent vehicles, when inferring a driver's intention of changing lanes, the driving-assistance system may assist in switching on the turn signal bulb, or warn of not doing so, or remind him of turning his head for observing the traffic situation before changing lanes.

There are different ways to provide service, including automatic execution (in which the default service operations are executed automatically), inquiry (in which the layer may ask the users about whether to execute some service operations), and alerting (in which ringing or vibration may be triggered in case of emergency).

Next we present two types of services: single and multiple-scenario-IoB service. They differ in the complexity of involved service operations.

1) Single-Scenario-IoB Service: In this type of service, the service layer invokes IoT devices of a single scenario, e.g., smart home or smart transportation, to serve users. For example, in smart-home IoB, when obtaining the intention that the home owners would go home after work, this layer may awaken IoT devices (e.g., turning on lights and air conditioners) in their home remotely before they arrive at their home. In this way, the smart-home IoB creates a comfortable home environment. In smart-transportation IoB, when obtaining the intention that pedestrians would cross the road, this layer may play a voice alerting that reminds the driver of giving way to pedestrians.

2) Multiple-Scenario-IoB Service: The service layer invokes IoT devices of multiple scenarios to serve a user. This involves more complex service operations.

Again consider the Go-Home example as shown in Fig. 12. Assume that Bob runs his IoB system in a cloud/edge server. This IoB can be logically divided into smart-office IoB, smart-transportation IoB, smart-home IoB, etc. The server can collect and analyze Bob's behaviors, infer his intention, and instruct his IoT devices in office, vehicle, home, etc. to provide proper services to him.

Assume that Bob is currently at his office. When obtaining Bob's online behavior via his smartphone: set the driving navigation to his home location at 6 pm, the IoB system infers that Bob would drive his car to go home. Then, the IoB system maps Bob's current behavior to "get off work" in his smart-office IoB, and invokes his office's IoT devices, e.g., switching off the air conditioner and PC in the office, closing windows

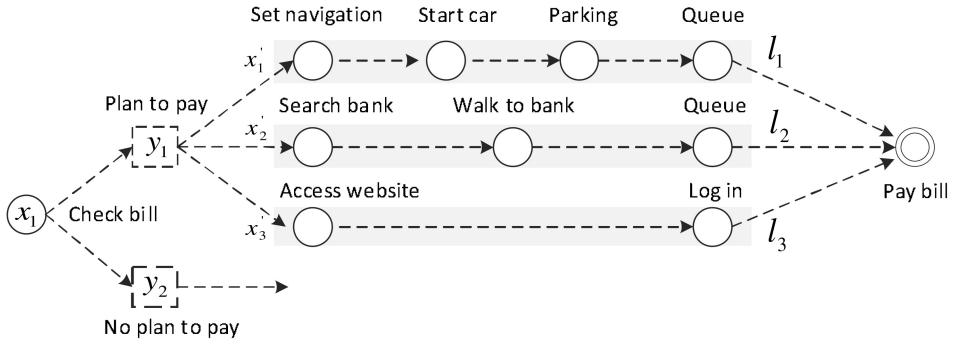


Fig. 11. An example of behavior-chain optimization.

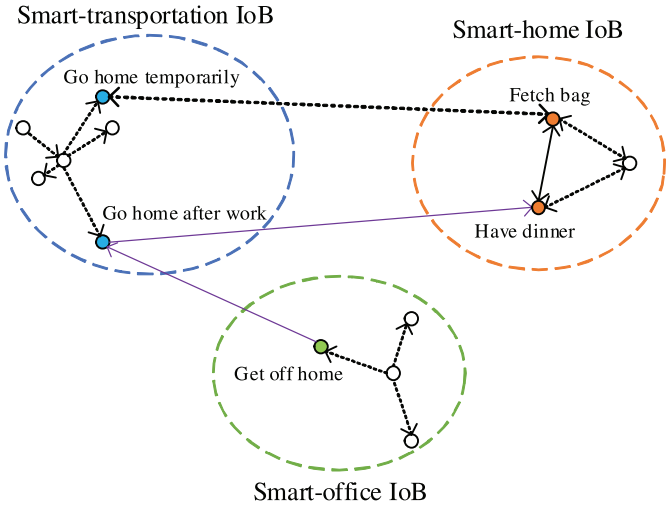


Fig. 12. An example of multiple-scenario-IoB service that involves smart-office IoB, smart-transportation IoB, and smart-home IoB.

and doors, etc. At the same time, the IoB system transfers his current behavior to "drive toward home" in his smart-transportation IoB and invokes his vehicular IoT devices, e.g., starting his car's engine, adjusting the seat & rearview mirror, etc. When Bob reaches his home, the IoB system transfers his current behavior to "stay at home for a long time" in his smart-home IoB and invokes his home's IoT devices, e.g., switching on the air conditioner in home.

E. Security and Privacy

IoB uses behavior data to infer human intention and hence people raise strong concern on the security and privacy of behavior data. Security and privacy have rich implications [127]. For IoB data, security is mainly concerned with the unwanted actions of unauthorized users, while privacy is mainly concerned with hiding or removing the identities of owners of IoB data. There are many methods to promote security and privacy including encryption, laws, and technology regulations and standards (specifying the participants' responsibilities of protecting and controlling access to data). The operations of behavior data involve data aggregation, data transfer, data control, and data sharing. In this section, we present a) a classic encryption method for data aggregation protection, b) technology standards for data transfer and control protection, and c) a decentralized privacy-protection solution to data sharing. For each type of operation, we first present a security and privacy method and then explain how this method is applied to protect behavior data.

1) Data Aggregation: Privacy-preserving data aggregation has been widely studied. We here present a classic data aggregation framework called aggregator-oblivious encryption [128]. This framework allows an untrusted data aggregator to learn desired statistics over the data reported by multiple nodes, without leaking each node's privacy.

Let us describe the following scenario to explain the framework. Assume one untrusted data aggregator and n nodes who periodically send their collected data to the data aggregator. Consider a period. Let $x_i \in \mathcal{D}$ denote the collected data of node i where $1 \leq i \leq n$ and $\mathcal{D}$ is a certain domain. Let $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathcal{D}^n$ denote a data vector from all nodes. Let $f : \mathcal{D}^n \rightarrow \mathcal{O}$ denote a function with which the data aggregate computes some aggregate statistics from some range $\mathcal{O}$.

The framework aims to achieve two goals. One is aggregator-oblivious security. It includes three implications: 1) the aggregator can learn only a noisy statistic which is close to the desired statistic, 2) without knowing the aggregator's capability, one node learns nothing about the collected data, even if several nodes form a coalition against the remaining nodes, and 3) if the aggregator colludes with some nodes, or if parts of the collected data have been leaked, then the aggregator can inevitably learn the statistic of the remaining nodes but no additional information. Another is distributed differential privacy. In the previous differential privacy mechanism, all nodes trust the centralized aggregator who adds noise before publishing the desired statistic. In the distributed differential privacy mechanism, nodes need not trust the centralized aggregator or other nodes. The aggregate statistic revealed is roughly the same whether or not a specific node participates in the system, even when the aggregator may have arbitrary auxiliary information (say, personal knowledge about a specific node), or collude with a small subset of corrupted nodes.

To achieve the two goals, taking the sum statistics as an example, the framework, as shown in Fig. 13, consists of the following three algorithms.

Setup(1^λ)*:* Taking in a security parameter λ, a trusted third-party creates public parameters θ, a private key k_i for each node i $(1 \leq i \leq n)$, and a private key k_0 for the aggregator.

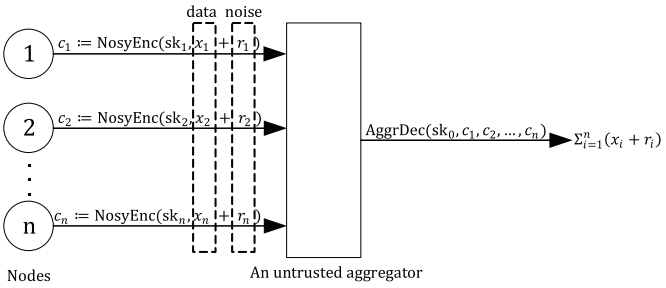


Fig. 13. The framework of the aggregator-oblivious encryption [128].

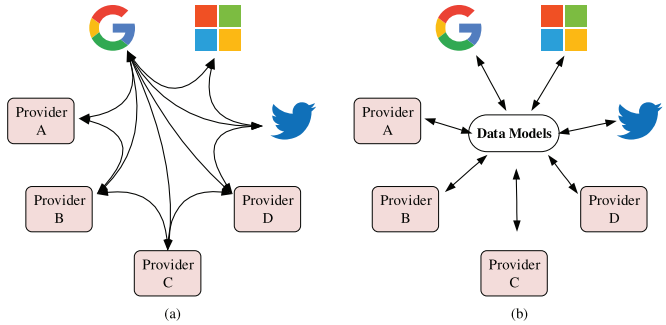


Fig. 14. Data transfer: (a) without the DTP and (b) with the DTP [129].

NoisyEnc(θ, k_i, x_i, r_i)*:* Node i first obtains the noisy version $\hat{x}_i$ of its data x_i with noise r_i, and then encrypts $\hat{x}_i$ using θ and k_i to gets $c_i = \textit{NoisyEnc}(\theta, k_i, x_i, r_i) \triangleq \text{NosyEnc}(\theta, k_i, \hat{x}_i)$.

AggrDec$(\theta, k_0, c_1, c_2, \ldots, c_n)$*.* Taking in θ, k_0, and c_i $(1 \leq i \leq n)$, the aggregator obtains a noisy version $f(\hat{\boldsymbol{x}})$ of the targeted statistics $f(\boldsymbol{x})$, where $f(\hat{\boldsymbol{x}}) = \text{AggrDec}(\theta, k_0, c_1, c_2, \ldots, c_n)$, $\boldsymbol{x} = (x_1, \ldots, x_n)$, and $\hat{\boldsymbol{x}} = (\hat{x}_1, \ldots, \hat{x}_n)$. For example, $f(\hat{\boldsymbol{x}}) = \sum_{i=1}^{n} \hat{x}_i = \sum_{i=1}^{n} x_i + r_i$.

In IoB's application to smart business, search engines such as Google often adopt a sponsored search system to collect users' view time of ads and calculate the mean view time for pushing appropriate ads. We may apply the aggregator-oblivious encryption method in the system to avoid the leakage of individual users' view time. In the method, the client of the system, which is installed in users' computers, invokes the NoisyEnc() function to encode users' view time of ads and then sends it to the server of the system, while the server receives the view time and invokes the AggrDec() function to produce a noisy version of the actual mean view time.

2) Data Transfer: Data Transfer Project (DTP) [129] was launched in 2018 to create "an open-source, service-to-service data portability platform, so that all individuals across the Web could easily move their data between online service providers whenever they want". On the premise of protecting the security and privacy of user data, DTP makes data portability easy and economic by encouraging service providers to adopt canonical interfaces. Assume that service providers need to transfer their data. Fig. 14(a) shows that without DTP, each provider needs to build and maintain their respective application programming interfaces (APIs). In contrast, Fig. 14(b) shows that with DTP, each provider only needs to build and maintain an API that supports the canonical formats (called Data Models) for data transfer.

A DTP system consists of three main components:

a) *Data Models:* They are of canonical formats for transferring data between source and destination providers. They include two parts: a file type (e.g., a standard photo format like JPEG) and the additional metadata that helps the destination provider to import the data, e.g., the title, description, and album of photo. Using a common Data Model, service providers may significantly reduce the cost of maintaining and updating their proprietary APIs.

b) *Adapters:* They are methods that convert each provider's proprietary data and authentication formats into Data Models for transferring data. There are two main types of adapters: Data and Authentication ones. The former is used in pairs: an exporter translating the provider's API into the Data Model and an importer translating the Data Model into the provider's API. The latter allows users to authenticate their accounts before transferring data between a pair of Data Adapters.

c) *Task Management Library:* It is a reference implementation on how to utilize the adapters to transfer data between two providers, e.g., adapter calls, secure data storage, rate limiting, failure handling, and individual notifications.

In DTP, transferring data involves multiple parties, e.g., users, hosting entities, providers, and contributors). DTP defines some basic responsibilities (listed in Table X) that these parties should share to achieve security and privacy protection for data transfer. For example:

a) *Data Minimization:* This rule suggests that during the data transfer, the sending providers should provide all needed information only, while the receiving providers should only process and retain the minimum set of data for fulfilling their services.

b) *Token Revocation:* This rule suggests that DTP should revoke the authorization token of a data transfer once the transfer is completed. This ensures that the authorized token is valid only during the data transfer.

In IoB, users may utilize DTP to transfer their behavior data with each other safely and economically via its canonical interfaces. Assume that companies A and B adopt the DTP model to manage data. Then even if Alice adopts company A's system to store her behavior network while Bob adopts company B's system to store his, their data transfer is safe and ease due to the canonical interface of DTP.

3) Centralized Data Control: MyData [130] aims to provide a centralized data control. It is "a human-centered approach in personal data management that combines industry need to data with digital human rights". It is concerned with how to construct a data-management model such that individuals may have more control over their every day's data trails easily. Its core idea is that "we, you and I, should have an easy way to see where data about us goes, specify who can use it, and alter these decisions over time".

MyData evolves from two traditional personal data management models: API ecosystem model in Fig. 15(a) and Platform model in Fig. 15(b). In the former, users may use one individual API (application programming interface) to manage their one individual application (say, shopping or health) and hence

TABLE X
TASKS FOR ACHIEVING SECURITY AND PRIVACY IN DTP [129]

Task	User	Provider-exporter	Provider-importer	Hosting Entity	DTP System
Data Minimization	Select data to transfer	Provide granular controls of what data to export	Discard any data not needed for their service	Configure only appropriate transfer partners	N/A
Rate Limiting	N/A	Implement	N/A	Set reasonable limits to prevent abuse	Support provider-specific rate limiting
User Notification	Receive and review notification of transfer	N/A	N/A	Configure mail sender and delay policy	Send notification, optionally with delay to allow for cancellation
Token Revocation	May need to manually revoke tokens if provider doesn't support automated revocation	Support token revocation	Support token revocation	N/A	Revoke Auth tokens after use (if supported by providers)
Minimal Scopes for auth Tokens	Verify appropriate scopes requested	Implement granular scopes	Implement granular scopes	N/A	Request minimal scopes for each transfer
Data Retention	Transfer of data is not deletion; user should delete source data if desired	Store only data needed to prevent fraud and abuse	Retain only imported data in compliance with privacy policies	Configure system to not retain any identifiable information	Retain no data after transfer completed
Abuse	Protect account credentials (strong passwords, two-factor authentication, etc.)	Implement appropriate fraud and abuse protections on APIs	Implement appropriate fraud and abuse protections on APIs	Implement appropriate fraud and abuse protections on UI	Encrypt data in transit and at rest using ephemeral key; Use solated/dedicated VMs per transfer

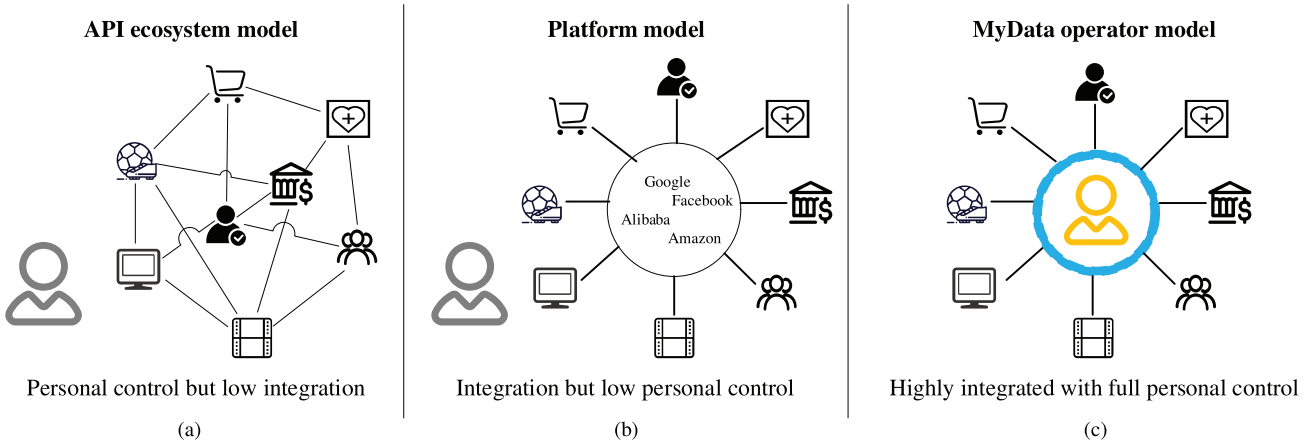


Fig. 15. (a) API ecosystem model: a user uses one individual API to manage his one individual application. (b) Platform model: an organization manages data of multiple sources of each individual person. (c) MyData operator model: create a human centered ecosystem of personal data-management [130].

their generated data wherein. This model enables them to have more personal control over their data, but makes the integration of APIs time-consuming and tedious. In the latter, an organization may manage their data from multiple sources. However, it is primarily designed to support the business models of organizations, rather than enable them to have more control on their data. In contrast, MyData, as shown in Fig. 15(c), aims to create an ecosystem of personal data-management, where the person is at the center of the ecosystem and the organizations are in competition with each other to provide personal data management service. In this way, MyData has merits of the two traditional models: high integration of APIs with full personal control.

In MyData, its personal data ecosystem is composed of actors (who are allowed to use personal data or who provide infrastructure for personal data management and governance), each holding one or more of roles such as Person, Data Source, Data Using Service and Personal Data. For example, as shown in Fig. 16, a job-seeker Bob (acting as the role of Person) may complete his CV information in a recruitment portal (acting as the role of data using services), where he may import his certified data (e.g., his study history) from the national register (acting as the role of a data source). In this example, different actors are connected together to provide services for the job-seeker.

MyData aims at achieving nine core functions: identity management, permission management, service management, value exchange, data model management, personal data transfer, personal data storage, governance support, logging and accountability. The former two functions are involved with security and privacy protection. In particular, identity management handles the authentication and authorization of actors so as to ensure that the right actors have the appropriate access to personal data. Individuals in MyData can have different identities. For example, they can have public, private, or self-sovereign identities. These identities give individuals the control of their digital identities. Permission management enables individual to manage their personal data and to execute

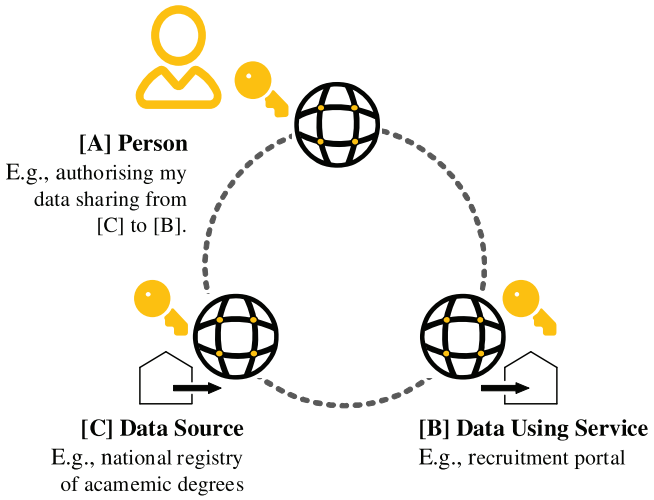


Fig. 16. An example of MyData application [130].

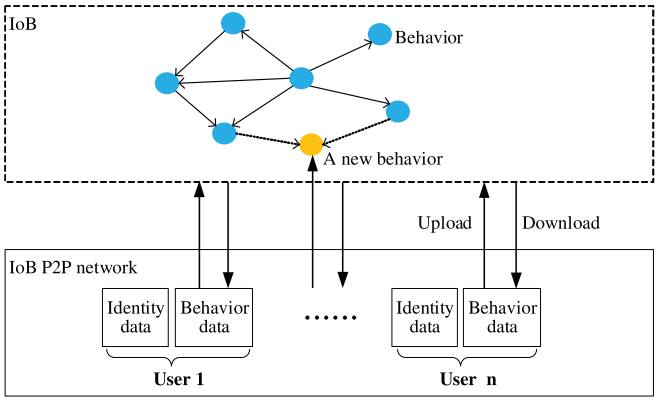


Fig. 17. Proposed decentralized framework that separates behavior data and identity data.

their legal rights. Both functions allow individuals to finely tune their security and privacy settings.

In IoB, users possibly need multiple companies to store their behavior data. If adopting the MyData model, these companies allow them to fully control the use of their personal behavior data with strong security and privacy measures. For example, in the Go-Home example, assume that Bob lets company A store his daily agenda and company B run his IoB edge server, where companies A and B adopt the MyData model to manage data. Then when Bob is driving toward home, he may safely authorize company B to access his daily agenda (managed by company A) for inferring his intention of going home, without worrying about his daily behavior leakage.

4) Decentralized Data Control: The Solid project [131], which is led by Prof. Tim Berners-Lee (inventor of the World Wide Web), aims to create a decentralized system over which individual persons become the owner of their data and controls their data access fully.

The Solid ecosystem is composed of a Solid hosting provider, Pods and Applications. A hosting provider hosts one or more Solid Pods. Individuals may store their data securely in one or more Pods in a decentralized manner. Particularly, Solid supports storing Linked Data, which means that different applications can work with the same data. Consider Ana as an individual using Solid. Her data are linked through her unique ID. She can fully control her Pods and hence her data, e.g., she may decide what data to share, and which individuals and applications can access her data, and may revoke access at any time. Solid applications store and access data in Pods using the Solid protocol, which inherits the client/server communication mode of the HTTP protocol in the Web system. In other words, in Solid communication, Pods act like servers, while Solid applications act like clients. In addition, to increase the flexibility, inter-operation, and robustness of the Solid ecosystem, the Solid protocol further extends and improves HTTP protocol components [131].

The Solid protocol suggests the following security and privacy measures, which are borrowed from those of HTTP protocols [131].

a) The Solid ecosystem relies on the Solid-OIDC specification for authentication and uses WebID-TLS as an additional authentication mechanism.
b) The servers should use access control list (where the Web access control provides an access control list model for Linked Data System to set authorization conditions for HTTP resources), to describe authorization that expresses and determines the access rights of requested resources.
c) The Servers are encouraged to use authorization techniques for preventing unwanted access to resources, rather than depending on the relative obscurity of their resource names.
d) The Servers should use TLS connections to protect the content of the request and response for avoiding eavesdropping and modification by third parties.
e) The Servers should not assume that the user agent is a regular Web browser, even when the request header of the user agent contains familiar values.
f) The Servers should ignore HTTP cookies from untrusted origins.
g) The Servers should reduce potential timing attacks.

In IoB, users may store their behavior data in one or more Solid Pods in a decentralized manner. They may link their behavior data in multiple Solid Pods by using their unique IDs and share their behavior data via links, thereby controlling their data access fully and flexibly. In the Go-Home example, Bob may store his smart-home's and smart-transportation's behavior data in two different Solid Pods. Through links, he can easily allow his multiple friends to access his smart-home's behavior data with different privileges, while allowing them to access his smart-transportation's behavior data with the same privilege.

5) Decentralized Data Sharing: Nyman proposes a rough behavior-identity-separating idea (i.e., separating behavior data and identity data) to provide security and privacy protection for users. Here, we design a blockchain-like decentralized framework [132] to implement the idea.

In our framework, users store their identity data privately, while sharing their behavior data with other users. All users compose a peer-to-peer (P2P) network and maintain together an IoB (consisting of behavior data). We next present an IoB and its underlying P2P network as shown in Fig. 17.

a) *IoB:* An IoB is a directed graph of behaviors, where each vertex represents a behavior and an edge between two vertexes represents a behavior transition.
b) *Underlying P2P network:* The underlying P2P network consists of users' network devices, e.g., PC, smartphone, smart home gateway, and vehicle, from which users may access the IoB. These network devices are equivalent participants and maintain together the IoB, where each device holds a complete or partial copy of the IoB, and keeps synchronizing the IoB with other devices.

Our security and privacy protection measures are as follows. In our framework, a user acts as an uploader or downloader, and is not allowed to remove and modify the behaviors in IoB for preventing from malicious tampering of behavior data. When acting as a downloader, the user keeps the IoB consistent with that of other users. When acting an uploader, the user can add new behaviors to the existing IoB, so that the IoB may finally include all human behaviors. We require that the uploader should provide a verifiable approach to justify their newly uploaded behaviors and all possible relationship (between the uploaded behaviors and the existing behaviors), without a compromise of security and privacy. Two potential mechanisms can be adopted to meet this requirement.

a) *Zero-knowledge proof [133]:* One party (the prover) declares a statement via an identity-hiding credential, while another party (the verifier) can verify the correctness of the statement according to the credential, without knowing the identity of the prover. Our framework can adopt this mechanism such that the uploader of behaviors may act as the prover role and hence it can upload its behaviors, without disclosing its identity.
b) *Anonymous P2P [134]:* Users can communicate with each other anonymously. This mechanism adopts a special routing overlay network, which can hide the physical location of users, to achieve anonymity. Our framework can adopt this mechanism to ensure that the uploader of behaviors uploads its behavior data without disclosing its physical location.

VI. IoB Applications

In this section, we survey IoB-related applications. In particular, we describe their adopted architectures from the angle of our proposed IoB architecture explained in the prior section.

In applications, IoB relies on human intelligence and knowledge and AI technology to provide services in line with human intention and therefore has salient features different from IoT, as summarized in Table I. Particularly, IoB relies on human beings to define their meaningful behaviors and give their societal implications to reflect their intentions. IoB provides fine-grained characteristics of each behavior, e.g., the action type, and action mode of a human behavior, to help infer the context of a behavior occurrence accurately. IoB networks users' meaningful behaviors together and learns the probabilities of behavior transitions; by exploiting the behavior network, we may feed closely related behavior data to AI and better design AI algorithms (e.g., construct AI attention components [135] by employing the behavior-transition probabilities) for predicting a user's next behavior accurately, thereby avoiding prediction blindness seen in existing behavior prediction systems. As a result, IoB has greater potential to do a better job than intelligent IoT.

A. Smart Home

In an IoT-based smart home, smart home devices (e.g., lamps and air-conditioners) with built-in chips or sensors are connected together to help users in monitoring and controlling home attributes (e.g., lighting) and home security (e.g., access control) [136]. These devices passively provide services by users' instructions or preset rules. However, without understanding human intention, these provided services might not meet users' needs well as pointed out in the Go-Home example in Fig. 2.

In contrast, in an IoB-based smart home, the system can perceive users' behaviors, then perform behavior computing (e.g., infer human intention), and next invoke IoT-based hardware/software to provide services more intelligently. Below, we present two IoT-based smart-home systems: cognitive dynamic system [137] and energy management system [70], and explain how they can be implemented in our IoB framework.

Feng et al. [137] integrate IoT and a cognitive dynamic system to enhance the intelligence level of a smart home. The proposed smart home system employs a cognitive perception-action cycle, where its building blocks (e.g., memory and attention) are introduced to perceive users' behaviors via sensors, and enable actuators (e.g., IoT devices) to perform daily tasks. However, this IoT system mechanically performs the predefined rules and provides rule-based ones, rather than intention-based services. For example, at home, when a user sits on a sofa, the system cannot infer whether the user wants to watch TV or takes a rest and therefore cannot provide services in line with his intention. With our proposed IoB architecture, an IoB system can do so, as shown in Fig. 18. IoB interconnects a user's daily life behaviors into a graph, and then calculates transfer probabilities between behaviors to infer his intentions (e.g., watch TV or takes a rest). Based on the inferred intention, the IoB system can provide satisfactory services. Specifically, after perceiving that the user sits on a sofa via cameras and pressure/acoustic sensors (L1), the IoB system maps this behavior to a node in the user's smart-home behavior network about daily living (L2). The system then learns about the user's daily routine (e.g., his agenda and working duration), behavior history, and personal preferences. After that, the system invokes AI technology to calculate the probability that the user wants to watch TV or take a rest, and infers his intention by utilizing the behavior-transition probability in the user's behavior network (L3). Finally, the system may predict and play the users' most favorite TV show if the user wants to watch TV or turns off TV and dims the lights if the user wants to take a rest (L4).

Chen et al. [70] propose a human-centric Smart Home Energy (SHE) management system to minimize the electricity cost while providing intelligence at the "butler" level. The SHE system collects data from physical and online spaces for cognitively understanding users' behaviors and discovering the pattern of power consumption. By utilizing the pattern, the system can dynamically infer the user's electricity requirement

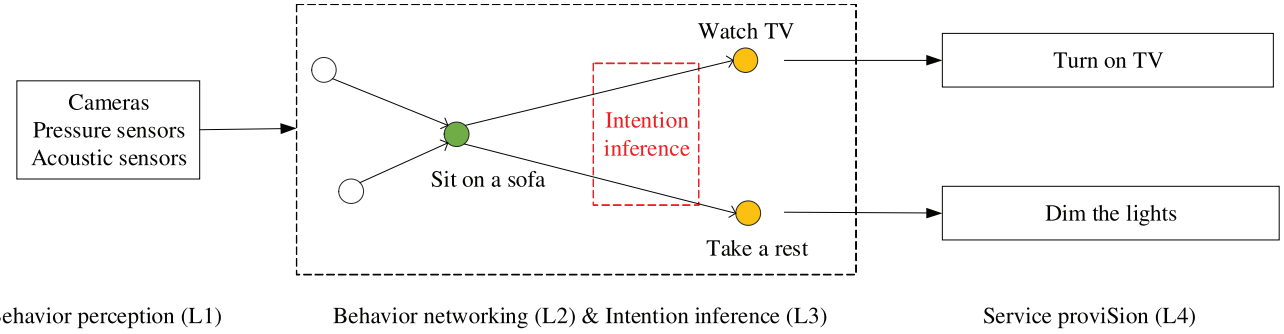


Fig. 18. Proposed IoB architecture for smart home [137].

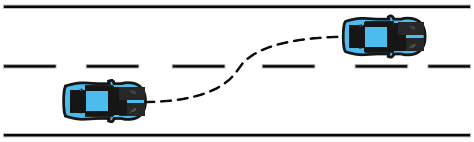


Fig. 19. An example of lange change.

and optimize the energy use. For example, SHE may intelligently start some devices (e.g., a washing machine) at an appropriate working time, e.g., at midnight when the electricity load and price are low. With our proposed IoB architecture, we may better implement SHE. In our system, we may use online platforms and wearable sensors to perceive users' behaviors, e.g., getting off work, and driving car after work and playing basketball, more intelligently and punctually (L1). These behaviors are then mapped to nodes in a smart-home IoB (L2). Next, our behavior-computing component infers users' intention (L3), e.g., go home on time or later after doing sports, and finally our service-providing component invokes IoT devices at appropriate time (L4). For example, in case that the user goes home later, the IoB system may turn on the air-conditioner to cool down rooms later than usual, prepare more hot water for a bath, run a washing machine at midnight, etc. In this way, the IoB system can provide more energy-saving living space and butler-style services.

B. Smart Transportation

In smart transportation, advanced driver-assistance systems are widely used to assist in driving and increase car and road safety. Below, we describe two known systems: Lane Change System [10] and Pedestrian Crossing Prediction System [98], and present how to leverage our IoB architecture to better implement them.

Xing et al. [10] propose a lane change intention inference (LCII) system. It works as follows: a) Collecting driver behavior data using cameras, such as head pose, eye gaze direction; b) Processing driver behaviors data and generate a sequence of behavior data; c) Using an ensemble learning-based bi-directional LSTM (EBiLSTM) model to predict driving actions, e.g., lane-keeping, left and right lane changing, left lane changing as shown in Fig. 19; and d) Generating assistant and collaborative control strategies to improve driving security.

With our proposed IoB architecture, we can better implement such a system, as shown in Fig. 20. Instead of generating a behavior sequence from behavior data directly, IoB networks all driving behaviors and then generates behavior sequences that are the most related to the current behavior for intention inference, according to the behavior transition probability in the behavior network. Further, IoB defines and exploits fine-grained driving behaviors (e.g., by recording and using drivers' behaviors associated specific road sections) to improve the intention-inference accuracy. Consider an example that Ben often drives from city X to city Y. Our IoB system records his histories of fine-grained driving behaviors, where the properties of a behavior may include his driving speed, driving habits, head pose, frequency of lane changing, the traffic and weather conditions, road section number from X to Y, etc. Then the system constructs his behavior network by these driving-behavior behaviors. Assume that one day, Ben is driving from X to Y. Perceiving that he turns his head at some road section (L1), the system first maps this behavior to one node of his deriving-behavior network (L2), and then predicts his next behavior (L3): lane changing or keeping. To do so, the system searches his behavior histories that are closely related to this road section, which is one property of his head-turning behavior. From his behavior histories, the system learns that at this road section, the user usually drives at a speed of 70 kilometers per hour in normal conditions and often turns his head to observe roadside signs. Exploiting such information and other factors (e.g., traffic and weather conditions, the vehicles before and after his car), the system may predict his next behavior accurately. For example, under normal conditions, if the user is currently driving at a speed of 50 kilometers per hour and there is a car before his car, with high probability, his intention of turning head is to observe traffic conditions and he plans to change to the left lane next. If he is currently driving at a speed of 70 kilometers per hour, his intention of turning head is to observe roadside signs and he keeps the same lane. Finally, based on the intention inference, the system may provide timely services (L4), e.g., turning signals or keeping steering wheel.

Abughalieh and Alawneh [98] develop a system to remind a driver of a pedestrian intending to cross the road. This system builds a CNN model to predict pedestrian orientation by detecting the human body's features (i.e., shoulders, neck, and face) and uses a depth-sensing camera (a higher accuracy camera) to estimate the distance between the pedestrian and the vehicle. With our proposed IoB architecture, we may better implement the system. Our IoB system invokes the depth-sensing cameras to perceive pedestrians' behaviors (such as

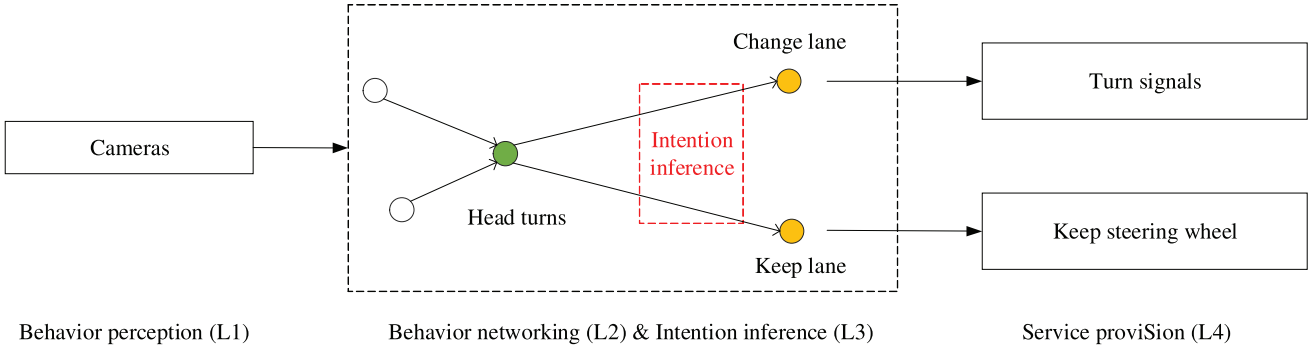


Fig. 20. Proposed IoB architecture for smart transportation [10].

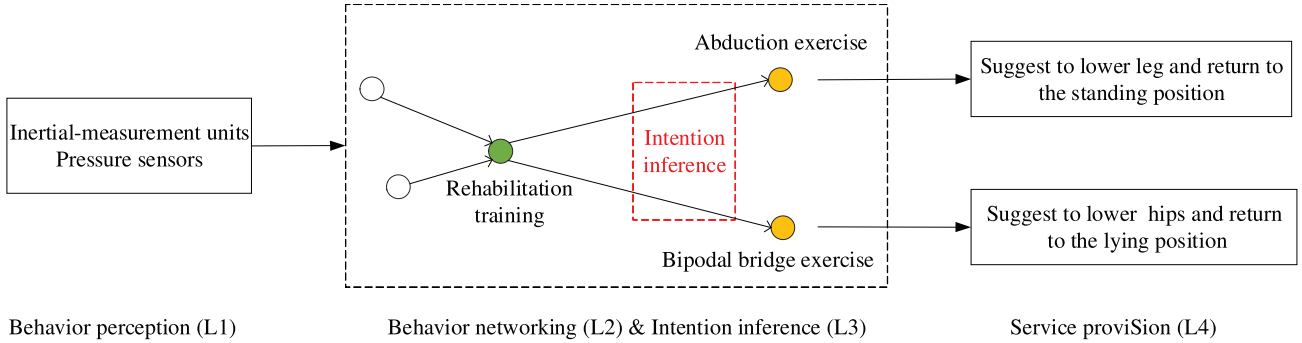


Fig. 21. Proposed IoB architecture for smart healthcare [62].

stopping walking, walking in front of the car and changing the orientation) (L1). Then, it maps these behaviors to nodes in a smart transportation IoB (L2). Next, our intention-inference component infers the pedestrian's intention (e.g., intending to cross the road) (L3) and finally our service-providing component plays a voice warning to the driver or assist the driver in braking the car (L4).

C. Smart Healthcare

In IoT-based smart healthcare, wearable sensor and smart medical devices can collect patients' health data and then send them to doctors or healthcare professionals for evaluating their health statuses and providing advice for further treatment.

In contrast, in IoB-based smart healthcare, the system is able to perceive patients' behaviors, infer their intentions and provide more timely assistance and treatment. Next we mainly present two IoT-based smart-healthcare systems: SmartPants [62] and City4Age [138], [139], [140], [141], and explain how they can be implemented in our IoB framework.

Bisio et al. [62] propose SmartPants, an IoT-based eHealth system for the rehabilitation of lower limbs of post-stroke patients. This system adopts inertial measurement units (each being a combination of gyroscopes and accelerometers) and pressure sensors to identify the lower-limbs, activities of post-stroke patients, then recognizes their current exercise, and next sends this useful information to a remote server for healthcare professionals' evaluation of the recovery and assistance to them. With our proposed IoB architecture, we may better implement SmartPants, as shown in Fig. 21. Instead of only recording patients' activities and healthcare data (e.g., blood pressure) as in SmartPants, IoB networks these exercise behaviors together and guides their next exercise behaviors (e.g., standing and lying) based on the network, and further infers their next exercise intention and provides timely services. Assume that a patient named Bob is performing rehabilitation training. By using inertial measurement units on his thighs and pressure sensors on his feet, our IoB system first perceives the patient's behavior data (e.g., standing, lying, moving leg laterally with respect to the body and raising the hips) (L1). Then the system maps the current behavior to one node in his smart-healthcare behavior network (L2), and identifies his current ongoing exercise, e.g., abduction or bipodal bridge in this example. Next, taking into account Bob's physical condition, exercise records (e.g., exercise duration) and rehab progress, the system may adopt LSTM to infer his next exercise intention (L3), e.g., continuing current exercise or switching to another exercise program. Finally, based on the intention inference, the system may provide timely suggestions (L4), e.g., lowering his leg and returning to the standing position. Whenever needed, our IoB system can contact healthcare experts as well.

City4Age [138], [139], [140], [141] aims to create a framework for the elderly population that enhances the early risk detection of frailty and mild cognitive impairments, and provides personalized intervention. The project has developed a monitoring-detection-intervention cycle framework, where a monitoring module collects a large amount of data from wearable and mobile devices and sensors. These data are used for analyzing elderly people's behaviors, and identifying geriatric factors leading to frailty and mild cognitive impairments. Finally, the personalized assistance module can help the elderly population to improve their daily life and promote positive behavior changes. With our proposed IoB architecture,

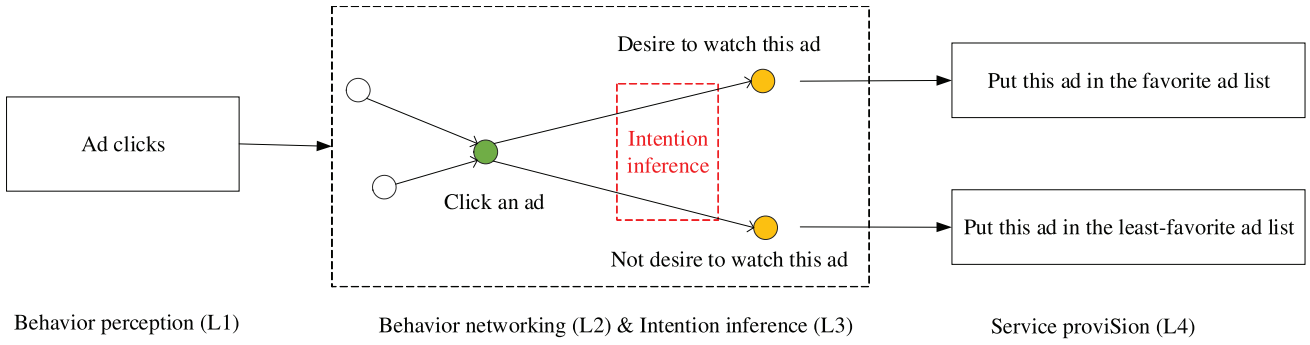


Fig. 22. Proposed IoB architecture for smart business [77].

we can better implement City4Age. Our system can invoke IoT infrastructures to collect elderly users' behavior data and health data (L1). Based on these data, the behavior perception layer perceives a user's behavior, and the behavior networking layer maps the perceived behavior to a node in IoB (L2). After that, the behavior computing layer analyzes a user's health status and disease risk, and optimizes their behavior path (L3). Finally, the system provides behavior suggestions (e.g., reminding users to exercise), and sends their health status to their physician and authorized relatives/friends (L4).

Besides the above smart healthcare systems, Yao et al. [36] propose a Web-based system to track users' continuous daily activities, detect their abnormal behavior, and send an alarm to the caregivers whenever necessary. Hussain et al. [37] propose a baby behavior monitoring system that employs control charts to analyze the baby behavior. This system represents a baby motion using points on the control chart. Upon detecting an abnormal behavior, this system sends an alarm to parents or medical care personnel. Sood and Mahajan [142] propose an IoT-fog-based healthcare system to continuously monitor blood pressure and other health parameters. They use ANN to predict a user's risk level of hypertension. Like the above analysis, with our proposed IoB architecture, we may better implement the above systems.

D. Smart Business

In smart business, companies market their products by analyzing consumers' purchasing behaviors and mining their purchasing preferences. Among smart business studies [77], [79], we mainly describe a sponsored search system [77] and then present how to leverage our IoB architecture to better implement the system.

Zhang et al. [77] propose a sponsored search system that uses RNN to predict the click-through rate of ads. Sponsored search is a major business model for commercial Web search engines where the number of times (a user clicks on an ad) is directly related to the commercial value. In order to maximize the revenue of sponsored search and maintain good user experiences, Zhang et al. [77] use users' online sequential behaviors on ads (e.g., query submission, ads clicking or ignoring, and time spent on clicked ads pages) to train their RNN for the click prediction. With our proposed IoB architecture, we can better implement such a system, as shown in Fig. 22. Instead of adopting a sequential structure of search behaviors, recording coarse-grained behaviors (e.g., ad-clicking), and applying AI only for click prediction, IoB adopts a graph structure to network search behaviors together, records fine-grained behaviors (e.g., the location of ad-clicking), and applies AI and the behavior chains from the IoB graph to infer users' intention. In this way, IoB is used to improve the click-prediction accuracy. For example, besides watching time of ads, the location of ad-clicking (e.g., the center and the right-top corner of an ad) is also an important property of ads clicking behaviors. If a user intends to watch an ad, he usually clicks the center of this ad. If not, he clicks the close button on the right-top corner (or other "close" signs' positions) of the ad; however, sometimes he might click other locations of the right-top corner by mistake, instead of the close button. Our IoB system networks such fine-grained behaviors and trains machine learning algorithms (e.g., RNN) by using closely related behaviors. When perceiving a user's current ad-clicking behavior (L1), the system maps it to a node in the network of ad-clicking behaviors (L2). Next, the system may infer users' intention (e.g., desire or not desire to watch some type of ads) according to the trained RNN (L3). Finally, the system puts the ad to the favorite or least-favorite ad list (L4). In this way, the system may push the ads that the user is most interested in and is the most likely to click, and maximize the revenue of sponsored search.

Li et al. [78] propose an online shopping system that uses a neural attentive recommendation machine to model users' sequential behaviors and predict their purchasing purpose. Yin et al. [79] propose a scalable probabilistic tensor factorization model to predict the semantics (e.g., click, add-to-favorite, add-to-cart, and purchase) of user behaviors in e-commerce platforms. Sakar et al. [80] propose a real-time online shopper behavior analysis system to predict users' purchasing intention. Wu et al. [106] propose a novel RNN-based recommender system to predict users' perception and appreciation changes of movies. Like the above analysis, with our proposed IoB architecture, we should better implement the above mentioned systems.

E. Human–Robot Interaction

In human-robot interaction, robots need to perceive human behaviors and understand human intentions so as to interact with human beings, without injuring the latter. With our proposed IoB framework, we may construct a network of a robot's behaviors, besides a network of human being's behaviors. In this way, a robot may not only better perceive

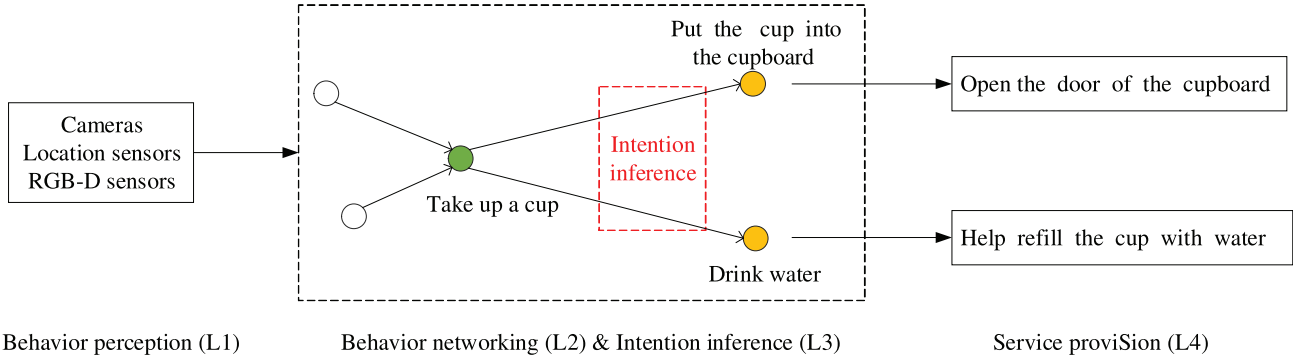


Fig. 23. Proposed IoB architecture for human–robot interaction [110].

human being's behavior but also make its behavior predictable. Among human-robot interaction studies [110], [111], [112], we describe a robotic response technology [110] and a human-robot collaboration technology [143], and then present how to leverage our IoB architecture to better implement them.

Koppula and Saxena [110] propose a reactive robotic response technology that detects the behavior of humans, and then provides necessary assistance. With our proposed IoB architecture, we may better implement this technology, as shown in Fig. 23. Instead of providing a triggered service responding to their current behaviors, IoB networks human behaviors, infers their intention according to the behavior chains in the network and their behavior context, and finally provides the service in line with their intention. Consider an example that a robot services a user. Assume that our IoB system runs in the robot. In this example, our IoB system continually collects a user's behavior data via the robot's sensors (e.g., cameras), then feeds these collected data to a machine learning algorithm (e.g., CNN) for identifying the user's current behavior(L1), e.g., taking up a cup, and finally map the identified behavior to one node in the user's IoB behavior network (L2). Next, to exploit the behavior network, the system searches the user's behavior history, checks his/her previous behavior to infer his/her intention (L3), and provides services in line with his/her intention (L4). For example, the system may infer that the user Ben would intend to place the cup into the cupboard if he has already drunk water just now, and drink water if he has worked for a long time without drinking or has just finished exercising. Finally, the system instructs the robot to service Ben, e.g., open the door of the cupboard. In this way, a robot can provide more attentive services.

Casalino et al. (2018) propose a human-robot collaboration technology that predicts a human operator's action in industrial scenarios through a recursive Bayesian classifier and wearable vibrotactile feedback. In this study, a robot monitors an operator's gaze, detects the tool that the operator is using, and predicts the operator's intentions by their proposed classifier. The operator's hand is equipped with a vibrotactile ring to provide feedback in handling collaborative tasks between the operator and the robot. With our proposed IoB architecture, we can better implement this technology. In our technology, we can construct an IoB networking the operator's all actions. The robot monitors and detects the operator's current action, and maps it to an action node in IoB, and then infers the operator's next action according to the associated actions in IoB, Bayesian classifier, and wearable vibrotactile feedback. In this way, our action prediction should be more highly accurate.

VII. Open Research Issues

A. Behavior Perception

Behavior perception is the first step to construct an IoB network of all humans. To date, the behavior perception studies are limited to a few areas only. To perceive and collect human behaviors in other areas, there exist the following challenges.

Challenge 1: How to better exploit existing behavior perception techniques economically in other areas?

In recent years, behavior perception in elderly healthcare has extensively been studied and greatly facilitates the lives of the elderly [56], [144], [145]. Currently, there are three famous healthcare systems for human activity recognition (HAR): WSHAR-Wearable sensor-based HAR, ASHAR-Ambient sensor-based HAR, and HSHAR-Hybrid sensory-based HAR [146], each having respective advantages and disadvantages.

a) WSHAR recognizes human activities by mining the information data of wearable sensors. It is miniature-sized, low-cost, flexibly worn on human body, and can capture motion-related information, but cannot combine different sensor modalities for activity recognition.
b) ASHAR identifies human activities by vision-based devices that are fixed in environments. It may accurately and directly collect human activity information, but often leads to privacy issues, is expensive, and works in a limited space only.
c) HSHAR is a hybrid system of WSHAR and SHAR. It may capture rich human activity information by utilizing different sensor modalities, but has a complex structure, is costly, and requires data fusion and synchronization mechanisms when collecting and processing collected data.

The success of behavior perception systems in assisting the elderly has inspired us to perceive human behaviors in much wider areas including smart buildings, smart cities, and smart businesses, smart healthcare, smart home and autonomous driving assistance. Exploitation of existing behavior perception techniques in other areas should take into account multiple factors, e.g., space, perception accuracy, robustness,

user preference, cost, intrusiveness, and privacy. For example, in autonomous driving assistance, because of space limit and high perception accuracy requirements, we recommend using ASHAR to collect drivers' behavior data and traffic information. In a smart home, we have no space limit but have privacy concerns, and, therefore, we may deploy different types of sensors in different rooms. In this case, we recommend using HSHAR to capture rich human activity information, while protecting privacy.

Challenge 2: How to standardize all behavior names?

The standardization of behavior names is to make the same behavior have one expression only even in different areas, e.g., smart buildings, smart cities, and smart business. It can eliminate ambiguity and redundancy, as well as avoid naming inconsistency. For example, in order to describe the behavior of "upstairs", if there is no uniform naming convention, some may use "up-stairs" [48], and the others may use "stair-ascending" [49]. In addition, in different areas, the accuracy requirements of behavior naming are also different. For example, to name the behavior "walking", there are more precise descriptions: "walking fast", "walking slowly" or "walking at a constant speed". Therefore, it is desirable to have a uniform naming convention.

So far, no one has proposed a method for standardized naming of behaviors. We believe that a good naming method should be concise and can express information accurately. A feasible naming method should at least include such information: application area, behavior type, behavior mode, behavior name, etc.

Challenge 3: How to describe and record mental behaviors?

A general model should be built to represent any possible human behavior, physical or mental, in a digital form [147]. Compared to physical behaviors, mental behaviors are harder to detect, collect and record.

Some mental behaviors are manifested through facial expressions and physical behaviors, while others do not. Mental behaviors, which can be inferred through facial expressions (e.g., joy, anger, and sadness,), are usually very rich and fleeting and hence are difficult to capture directly through sensing devices. However, those that are hidden behind facial expressions and physical behaviors, are almost impossible to capture via physical devices.

To collect and record mental behaviors, there are three potential solutions. For those that can be expressed through physical behaviors, we utilize vision-based devices to record people's facial expressions and physical behaviors, and use AI algorithms to infer such behaviors. For those that do not exhibit in any obvious physical activities, we may use a brain-computer interface to collect people's mental activity signal data and further analyze their mental behaviors. The third solution is to record mental behaviors manually, say, via notebook and pen, google sheets, etc.

Challenge 4: How to reasonably place sensor devices?

A wide variety of sensing devices can be used to recognize and collect human behavior. When deploying and placing these sensing devices, we should take into account multiple factors: safety, privacy protection, human comfort, collection accuracy, etc.

For example, when placing a wearable sensing device, it should be placed in a position where it does not injure the wearer or interfere with his or her movement, and where it can accurately recognize the wearer's behavior. One feasible option is to place the sensing device on a relatively inactive body part (e.g., arms, thighs). This reduces the risk of injury to the wearer, does not interfere with the wearer's daily activities, and ensures the accuracy of behavior data collection.

B. Behavior Networking

Currently, behavior networking is conceptual. We face the following fundamental challenges.

Challenge 1: How to define an IoB address?

In IoB, we aim at interconnecting behaviors together for intention inference, behavior derivation, etc., which involve understanding the semantics of behaviors. To achieve this goal, the core problem is to define an IoB address, i.e., a unique identity (ID), for a behavior. In contrast, in Internet and IoT, an IP address is originally defined to uniquely identify a device for routing data and is not applicable for IoB, since it mainly consists of a network ID and device ID for routing data and does not provide information to understand the data content itself. So far, except Nyman who suggested using IP addresses for IoB, this paper gives an IoB address structure example in Section V-B1. In our opinion, an IoB address should at least satisfy the following three requirements.

a) *Expressing the semantics of a behavior:* This is the most salient feature different from an IP address. IoB was conceived to understand human intention for providing better service. Therefore, it is indispensable that the address structure should include fields to help understand the semantics of behaviors. For example, in Table VII, an address structure may include a field called *behavior mode* to indicate the context of behaviors.
b) *Supporting huge behavior space:* It expects that an IoB network all human behaviors. Therefore, we should use enough bits to represent all behaviors.
c) *Supporting behavior networking:* For example, to make the behavior network concise and compact, we may code a behavior with the principle of "the more relevant the behavior, the closer the address code," as explained in Section V-B1.

Challenge 2: How to quickly and automatically assign an address to a new behavior?

In IP networks, the Dynamic Host Configuration Protocol (DHCP) is designed to automatically assign an IP address to a device. With DHCP, an address is randomly chosen for a newly joined device but it may be reallocated to another device some time later after the device exits. Therefore, DHCP should not be applicable for IoB, because we hope to assign each behavior a permanent address for global search.

Challenge 3: How to interconnect behaviors?

In the process of constructing an IoB network, one person may upload behaviors and their relationships, but may have different behavior sequences even for the same set of behaviors. For example, in Fig. 24, people may get up, eat breakfast and go to work; sometimes they may get up, go to their companies

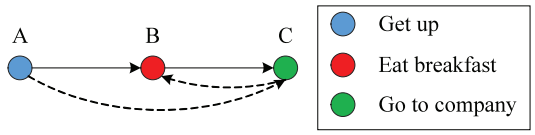


Fig. 24. Different behaviors sequences.

and then eat breakfast at office. Imagine that different people may have different behaviors and behavior sequences. It will a great challenge to interconnect all behaviors.

C. Behavior Computing

Behavior computing includes all computing operations involving human behaviors for understanding human intention behind behaviors. We focus on the proposed four forms of behavior computing, i.e., intention inference, behavior derivation, behavior programming, and behavior-chain optimization, to specify the challenges ahead of us.

Challenge1: How to understand the long-term intentions of users?

User's intention generally includes both short-term and long-term intentions. To date, there have been some studies on short-term intentions, e.g., HMM [102]. However, there is no relevant research on the long-term intention of users. In general, it is difficult to understand human's long-term intention, because it involves understanding a series of behaviors.

a) One behavior may imply multiple intentions, whereas one intention may exhibit in different behaviors. For example, in different situations, one behavior "waving a hand" may imply the intention "to say hello to a friend" or "to say goodbye to a friend". To express the intention "to say hello to a friend", the exhibited behaviors may be "to hug the friend" or "to nod and walk away."
b) Long-term intention may directly or indirectly associate with some behaviors of the behavior history. Finding the associated behaviors with the long-term intention is not easy.

Challenge 2: How to automatically verify whether the derived behavior is reasonable?

The derived behaviors are generated according to existing behaviors. Their reasonableness should be automatically verified, e.g., they can be conducted by people. For example, the derived behavior "cut the oranges" from "cut the apples" is reasonable; but the derived behavior "cut the bowls" is unreasonable. A potential verification solution is to adopt a generative adversarial network [148], [149]. Generative adversarial network is composed of a generative network and a discriminant network. Typically, a generative network can be used to derive new behaviors from existing behaviors, whereas a discriminant network can be used to verify the reasonableness of the derived behaviors.

Challenge 3: How to construct meaningful behavior sequences through behavior programming?

Behavior programming constructs new behavior sequences by combining/shuffling existing behaviors in a certain order. A new behavior sequence should be meaningful, i.e., it should be subject-related, context-sensitive, and operational. For example, According to the three behaviors: "wear pants", "wear shoes", and "go out", the behavior sequence, "wear pants→wear shoes→go out", is reasonable, but the behavior sequences, "wear shoes→wear pants→go out" and "go out→wear shoes→wear pants", might be meaningless. It is a challenge to design behavior programming schemes that produce meaningful behavior sequences.

Challenge 4: How to define criteria to rate behavior paths?

When rating the quality of a behavior chain, we may use its length (i.e., the number of behaviors in it), or the time of traversing it, etc. Different criteria tend to lead to different results. In general, we should synthetically consider multiple factors including chain length, consumed time, cost, difficulty, etc. Further, each individual has a different preference for each factor in different situations, and thus the criteria should vary with each individual.

Challenge 5: How to find the optimal behavior chain?

In order to find the optimal behavior chain that achieves a user's intention, a behavior chain search algorithm needs to be designed. The algorithm should be lightweight by considering users' preference and privacy.

D. Service Provision

IoB is in its infancy and there is currently no research on its service provision. In general, we face the following challenges.

Challenge 1: How to define a common service interface to invoke various IoT devices?

IoB provides services to users by invoking IoT devices. However, different companies currently have different invoking interfaces. For example, in smart home, Amazon's AWS IoT [150], Apple's HomeKit [151], and Microsoft's Azure [152] have their respective interfaces and only support their respective IoT devices. This calls for an international standard to define an interface layer or convention that enables IoB to invoke different vendors' IoT devices.

Challenge 2: How to measure the quality of experience (QoE) of IoB users?

Users' QoE feedback is very conducive to improving IoB's service. However, there is currently no study on IoB's QoE. One main reason is that QoE is subjective and different people have different ratings. For example, in a smart home, if IoB makes air conditioning and hot water ready when a home user gets home, some people may enjoy this precise and amazing service, but others may think that this service is a waste of resources. Therefore, IoB may provide users with personalized services, i.e., learning different users' preferences, measuring their respective QoE, and providing user-desired services.

E. Security and Privacy

The security and privacy of users' behavior data are vital for IoB. IoB involves handling sensitive individual behavior data and hence its security and privacy are especially vital. We face the following Challenges.

Challenge 1: How to protect online behavior privacy in IoB?

The online behaviors of IoB users often expose their personal identification data. For example, user-to-user online communications via WeChat, user-system interactions when watching videos, and user-APP interactions when shopping. On one hand, we may protect users' online behavior privacy according to global industry standards. For example, MyData,

a personal data-management model, advocates that all service providers should operate in accordance with a unified standard, which considers industry needs and digital human rights. With MyData, users complete control over their everyday's data trails while maintaining their privacy. On the other hand, we may adopt Trusted Execution Environment, such as Intel SGX, ARM Trust Zoon [153]) to protect the privacy of online behavior computing, besides encryption technologies for ensuring the privacy of online data transmission. In this way, we may safeguard the confidentiality and privacy of users' online behaviors when they are stored, transmitted, and computed.

Challenge 2: How to prevent IoT-IoB interdependence attack?

IoB is an IoT-based behavior network. According to a user's behaviors perceived by IoT devices, IoB infers this user's intention, and then provides better services. For example, in smart transportation, IoB may use cameras to collect driver behaviors, suggests good driving behaviors and finally improves driving safety. Therefore, IoB and IoT devices are inextricably linked and interdependent. As a result, attackers may conduct IoT attacks to attack the associated IoB indirectly. In the above driving example, an attacker may attack IoT devices so as to prohibit reminding a driver of his/her dangerous driving behaviors, which may cause disastrous consequences. Therefore, preventing IoT-IoB interdependence attack is significantly important. To this end, we may adopt mechanisms to protect the IoT-IoB interface for preventing the interdependence attack in advance. Other related approaches [154], [155], [156] can be considered.

VIII. Conclusion

In Dr. Göte Nyman's conceived IoB, we can collect all human behaviors and build a huge behavior network. Using the network, we can mine an individual's behavior profile and infer his/her intention, and further provide more intelligent services to meet his/her needs. This inevitably incurs strong security and privacy concerns. Ten years later since IoB was proposed, Nyman's forward-looking IoB idea has gradually been recognized and listed as one of Gartner's nine strategic technology trends that would influence 40% of people worldwide. Nyman's IoB ideas, which are primitive and conceptual, appeared in his blogs from 2012 to 2021. Based on these ideas, in this paper, we present our insights and opinions. We formally define the concept to IoB and propose a 5-layer IoB architecture. Then, we provide an in-depth analysis of the heart of IoB—behavior networking and computing, next investigate potential IoB applications, and finally discuss open research challenges. To date, IoB is still in its infancy and is expected to play a growing role in meeting human needs more intelligently and promoting desired behaviors of humans. This study aims at providing a timely, systematic and deep introduction to IoB for its further research and applications [157], [158], [159], [160], [161].

References

[1] "IoT growth demands rethink of long-term storage strategies, says IDC." IDC. Jul. 2020. Accessed: Jul. 16, 2021. [Online]. Available: https://www.idc.com/getdoc.jsp?containerId=prAP46737220

[2] "DIKW pyramid." Wikipedia. Nov. 2021. Accessed: Jan. 5, 2022. [Online]. Available: https://en.wikipedia.org/wiki/DIKW_pyramid

[3] C. Kidd. "What is the Internet of behaviors? IoB explained." Dec. 2019. Accessed: Jan. 5, 2022. [Online]. Available: https://www.bmc.com/blogs/iob-internet-of-behavior

[4] G. Nyman. "Internet of behaviors (IoB)." Mar. 2012. Accessed: Jul. 16, 2021. [Online]. Available: https://gotepoem.wordpress.com/2012/03/16/internet-of-behaviors-ib/

[5] C. Gilbert. "The Internet of behaviour promises more personalised and context-sensitive assistive technology." Jan. 2021. Accessed: Jul. 16, 2021. [Online]. Available: https://www.linkedin.com/pulse/internet-behaviour-promises-more-personalised-clive-gilbert

[6] K. Panetta. "Gartner top strategic predictions for 2020 and beyond." Oct. 2019. Accessed: Jul. 16, 2021. [Online]. Available: https://www.gartner.com/smarterwithgartner/gartner-top-strategic-predictions-for-2020-and-beyond/

[7] K. Panetta. "Gartner top strategic technology trends for 2021." Oct. 2020. Accessed: Jul. 16, 2021. [Online]. Available: https://www.gartner.com/smarterwithgartner/gartner-top-strategic-technology-trends-for-2021/

[8] J.-M. Kim, M.-J. Jeon, H.-K. Park, S.-H. Bae, S.-H. Bang, and Y.-T. Park, "An approach for recognition of human's daily living patterns using intention ontology and event calculus," *Expert Syst. Appl.*, vol. 132, pp. 256–270, Oct. 2019.

[9] H. Zhu, H. Chen, and R. Brown, "A sequence-to-sequence model-based deep learning approach for recognizing activity of daily living for senior care," *J. Biomed. Inform.*, vol. 84, pp. 148–158, Aug. 2018.

[10] Y. Xing, C. Lv, H. Wang, D. Cao, and E. Velenis, "An ensemble deep learning approach for driver lane change intention inference," *Transp. Res. C, Emerg. Technol.*, vol. 115, Jun. 2020, Art. no. 102615.

[11] Y. Guo, H. Zhang, C. Wang, Q. Sun, and W. Li, "Driver lane change intention recognition in the connected environment," *Physica A, Stat. Mech. Appl.*, vol. 575, Aug. 2021, Art. no. 126057.

[12] G. Nyman. "Gote Nyman's (gotepoem) blog." Accessed: Oct. 14, 2021. [Online]. Available: https://gotepoem.wordpress.com/

[13] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of Things: Challenges, opportunities, and directions," *IEEE Trans. Ind. Informat.*, vol. 14, no. 11, pp. 4724–4734, Nov. 2018.

[14] Y. Zhang, Z. Guo, J. Lv, and Y. Liu, "A framework for smart production-logistics systems based on CPS and industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 14, no. 9, pp. 4019–4032, Sep. 2018.

[15] G. Li et al., "GT-chain: A fair blockchain for intelligent industrial IoT applications," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 5, pp. 3244–3257, Sep./Oct. 2022.

[16] B. B. Sinha and R. Dhanalakshmi, "Recent advancements and challenges of Internet of Things in smart agriculture: A survey," *Future Gener. Comput. Syst.*, vol. 126, pp. 169–184, Jan. 2022.

[17] M. S. Farooq, S. Riaz, A. Abid, K. Abid, and M. A. Naeem, "A survey on the role of IoT in agriculture for the implementation of smart farming," *IEEE Access*, vol. 7, pp. 156237–156271, 2019.

[18] O. Friha, M. A. Ferrag, L. Shu, L. Maglaras, and X. Wang, "Internet of Things for the future of smart agriculture: A comprehensive survey of emerging technologies," *IEEE/CAA J. Automatica Sinica*, vol. 8, no. 4, pp. 718–752, Apr. 2021.

[19] M. S. U. Chowdury et al., "IoT based real-time river water quality monitoring system," *Procedia Comput. Sci.*, vol. 155, pp. 161–168, Sep. 2019.

[20] R. Arridha, S. Sukaridhoto, D. Pramadihanto, and N. Funabiki, "Classification extension based on IoT-big data analytic for smart environment monitoring and analytic in real-time system," *Int. J. Space-Based Situated Comput.*, vol. 7, no. 2, pp. 82–93, 2017.

[21] H. Mokrani, R. Lounas, M. T. Bennai, D. E. Salhi, and R. Djerbi, "Air quality monitoring using IoT: A survey," in *Proc. IEEE Int. Conf. Smart Internet Things (SmartIoT)*, 2019, pp. 127–134.

[22] L. Nóbrega, A. Tavares, A. Cardoso, and P. Gonçalves, "Animal monitoring based on IoT technologies," in *Proc. IoT Vertical Topical Summit Agr.-Tuscany (IOT Tuscany)*, 2018, pp. 1–5.

[23] F. Maroto-Molina et al., "A low-cost IoT-based system to monitor the location of a whole herd," *Sensors*, vol. 19, no. 10, p. 2298, 2019.

[24] L. Nóbrega, P. Gonçalves, M. Antunes, and D. Corujo, "Assessing sheep behavior through low-power microcontrollers in smart agriculture scenarios," *Comput. Electron. Agr.*, vol. 173, Jun. 2020, Art. no. 105444.

[25] Z. He and X. Bai, "A wearable wireless body area network for human activity recognition," in *Proc. 6th Int. Conf. Ubiquitous Future Netw. (ICUFN)*, 2014, pp. 115–119.

[26] A. G. Bonomi, A. H. Goris, B. Yin, and K. R. Westerterp, "Detection of type, duration, and intensity of physical activity using an accelerometer," *Med. Sci. Sports Exercise*, vol. 41, no. 9, pp. 1770–1777, 2009.

[27] F. Chamroukhi, S. Mohammed, D. Trabelsi, L. Oukhellou, and Y. Amirat, "Joint segmentation of multivariate time series with hidden process regression for human activity recognition," *Neurocomputing*, vol. 120, pp. 633–644, Nov. 2013.

[28] O. Banos, M. Damas, H. Pomares, F. Rojas, B. Delgado-Marquez, and O. Valenzuela, "Human activity recognition based on a sensor weighting hierarchical classifier," *Soft Comput.*, vol. 17, no. 2, pp. 333–343, 2013.

[29] L. Wang, T. Gu, H. Xie, X. Tao, J. Lu, and Y. Huang, "A wearable RFID system for real-time activity recognition using radio patterns," in *Proc. Int. Conf. Mobile Ubiquitous Syst. Comput., Netw., Services*, 2013, pp. 370–383.

[30] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, "Activity recognition from accelerometer data," in *Proc. AAAI Conf.*, vol. 5, 2005, pp. 1541–1546.

[31] L. Gao, A. Bourke, and J. Nelson, "Evaluation of accelerometer based multi-sensor versus single-sensor activity recognition systems," *Med. Eng. Phys.*, vol. 36, no. 6, pp. 779–785, 2014.

[32] T. G. Pavey, N. D. Gilson, S. R. Gomersall, B. Clark, and S. G. Trost, "Field evaluation of a random forest activity classifier for wrist-worn accelerometer data," *J. Sci. Med. Sport*, vol. 20, no. 1, pp. 75–80, 2017.

[33] D. Rodriguez-Martin, A. Sama, C. Perez-Lopez, A. Catala, J. Cabestany, and A. Rodriguez-Molinero, "SVM-based posture identification with a single waist-located triaxial accelerometer," *Expert Syst. Appl.*, vol. 40, no. 18, pp. 7203–7211, 2013.

[34] L. Atallah, B. Lo, R. King, and G.-Z. Yang, "Sensor placement for activity detection using wearable accelerometers," in *Proc. Int. Conf. Body Sens. Netw.*, 2010, pp. 24–29.

[35] Y. Zheng, W.-K. Wong, X. Guan, and S. Trost, "Physical activity recognition from accelerometer data using a multi-scale ensemble method," in *Proc. IAAI*, 2013, pp. 1575–1581.

[36] L. Yao et al., "WITS: An IoT-endowed computational framework for activity recognition in personalized smart homes," *Computing*, vol. 100, no. 4, pp. 369–385, 2018.

[37] T. Hussain, K. Muhammad, S. Khan, A. Ullah, M. Y. Lee, and S. W. Baik, "Intelligent baby behavior monitoring using embedded vision in IoT for smart healthcare centers," *J. Artif. Intell. Syst.*, vol. 1, no. 1, pp. 110–124, 2019.

[38] Y. Xing et al., "Driver lane change intention inference for intelligent vehicles: Framework, survey, and challenges," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 4377–4390, May 2019.

[39] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, "Using mobile phones to determine transportation modes," *ACM Trans. Sens. Netw.*, vol. 6, no. 2, pp. 1–27, 2010.

[40] A. R. Anwary, H. Yu, and M. Vassallo, "An automatic gait feature extraction method for identifying gait asymmetry using wearable sensors," *Sensors*, vol. 18, no. 2, p. 676, 2018.

[41] H. Gjoreski and M. Gams, "Activity/posture recognition using wearable sensors placed on different body locations," in *Proc. Signal Image Process. Appl.*, 2011, Art. no. 716724.

[42] J. Suto, S. Oniga, C. Lung, and I. Orha, "Recognition rate difference between real-time and offline human activity recognition," in *Proc. Int. Conf. Internet Things Global Community (IoTGC)*, 2017, pp. 1–6.

[43] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Comput. Surveys*, vol. 46, no. 3, pp. 1–33, 2014.

[44] W. Wu, S. Dasgupta, E. E. Ramirez, C. Peterson, and G. J. Norman, "Classification accuracies of physical activities using smartphone motion sensors," *J. Med. Internet Res.*, vol. 14, no. 5, p. e130, 2012.

[45] A. Mannini and A. M. Sabatini, "Machine learning methods for classifying human physical activity from on-body accelerometers," *Sensors*, vol. 10, no. 2, pp. 1154–1175, 2010.

[46] M. M. Hassan, M. Z. Uddin, A. Mohamed, and A. Almogren, "A robust human activity recognition system using smartphone sensors and deep learning," *Future Gener. Comput. Syst.*, vol. 81, pp. 307–313, Apr. 2018.

[47] C. A. Ronao and S.-B. Cho, "Deep convolutional neural networks for human activity recognition with smartphone sensors," in *Proc. Int. Conf. Neural Inf. Process.*, 2015, pp. 46–53.

[48] S. Sani, S. Massie, N. Wiratunga, and K. Cooper, "Learning deep and shallow features for human activity recognition," in *Proc. Int. Conf. Knowl. Sci. Eng. Manage.*, 2017, pp. 469–482.

[49] X. Xi, M. Tang, S. M. Miran, and Z. Luo, "Evaluation of feature extraction and recognition for activity monitoring and fall detection based on wearable sEMG sensors," *Sensors*, vol. 17, no. 6, p. 1229, 2017.

[50] Y. Wang, S. Cang, and H. Yu, "A data fusion-based hybrid sensory system for older people's daily activity and daily routine recognition," *IEEE Sensors J.*, vol. 18, no. 16, pp. 6874–6888, Aug. 2018.

[51] A. S. Kundu, O. Mazumder, P. K. Lenka, and S. Bhaumik, "Hand gesture recognition based omnidirectional wheelchair control using IMU and EMG sensors," *J. Intell. Robot. Syst.*, vol. 91, no. 3, pp. 529–541, 2018.

[52] Y. Kwon, K. Kang, and C. Bae, "Unsupervised learning for human activity recognition using smartphone sensors," *Expert Syst. Appl.*, vol. 41, no. 14, pp. 6067–6074, 2014.

[53] M. Bahrepour, N. Meratnia, Z. Taghikhaki, and P. J. M. Havinga, "Sensor fusion-based activity recognition for Parkinson patients," in *Sensor Fusion—Foundation and Applications*. London, U.K.: IntechOpen, 2011, pp. 171–190.

[54] P. Vepakomma, D. De, S. K. Das, and S. Bhansali, "A-Wristocracy: Deep learning on wrist-worn sensing for recognition of user complex activities," in *Proc. IEEE 12th Int. Conf. Wearable Implantable Body Sens. Netw. (BSN)*, 2015, pp. 1–6.

[55] S. Chernbumroong, S. Cang, A. Atkins, and H. Yu, "Elderly activities recognition and classification for applications in assisted living," *Expert Syst. Appl.*, vol. 40, no. 5, pp. 1662–1674, 2013.

[56] S. Chernbumroong, S. Cang, and H. Yu, "A practical multi-sensor activity recognition system for home-based care," *Decis. Support Syst.*, vol. 66, pp. 61–70, Oct. 2014.

[57] A. Moncada-Torres, K. Leuenberger, R. Gonzenbach, A. Luft, and R. Gassert, "Activity classification based on inertial and barometric pressure sensors at different anatomical locations," *Physiol. Meas.*, vol. 35, no. 7, p. 1245, 2014.

[58] J. M. Fontana et al., "Energy intake estimation from counts of chews and swallows," *Appetite*, vol. 85, pp. 14–21, Feb. 2015.

[59] J. Pansiot, D. Stoyanov, D. McIlwraith, B. P. Lo, and G.-Z. Yang, "Ambient and wearable sensor fusion for activity recognition in healthcare monitoring systems," in *Proc. 4th Int. Workshop Wearable Implantable Body Sens. Netw. (BSN)*, 2007, pp. 208–212.

[60] N. D. Lane et al., "BeWell: A smartphone application to monitor, model and promote wellbeing," in *Proc. 5th Int. ICST Conf. Pervasive Comput. Technol. Healthcare*, vol. 10, 2011, pp. 1–8.

[61] K. Zhan, F. Ramos, and S. Faux, "Activity recognition from a wearable camera," in *Proc. 12th Int. Conf. Control Autom. Robot. Vis. (ICARCV)*, 2012, pp. 365–370.

[62] I. Bisio, C. Garibotto, F. Lavagetto, and A. Sciarrone, "When eHealth meets IoT: A smart wireless system for post-stroke home rehabilitation," *IEEE Wireless Commun.*, vol. 26, no. 6, pp. 24–29, Dec. 2019.

[63] O. Debauche, S. Mahmoudi, P. Manneback, and A. Assila, "Fog IoT for health: A new architecture for patients and elderly monitoring," *Procedia Comput. Sci.*, vol. 160, pp. 289–297, Nov. 2019.

[64] N. A. Capela, E. D. Lemaire, and N. Baddour, "Novel algorithm for a smartphone-based 6-minute walk test application: Algorithm, application development, and evaluation," *J. Neuroeng. Rehabil.*, vol. 12, no. 1, pp. 1–13, 2015.

[65] A. M. Khan, Y.-K. Lee, S. Y. Lee, and T.-S. Kim, "A triaxial accelerometer-based physical-activity recognition via augmented-signal features and a hierarchical recognizer," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 5, pp. 1166–1172, Sep. 2010.

[66] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *Proc. Int. Conf. Pervasive Comput.*, 2004, pp. 1–17.

[67] S. J. Preece, J. Y. Goulermas, L. P. Kenney, and D. Howard, "A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data," *IEEE Trans. Biomed. Eng.*, vol. 56, no. 3, pp. 871–879, Mar. 2009.

[68] N. Kern, B. Schiele, and A. Schmidt, "Multi-sensor activity context detection for wearable computing," in *Proc. Eur. Symp. Ambient Intell.*, 2003, pp. 220–232.

[69] C. Doukas and I. Maglogiannis, "Advanced patient or elder fall detection based on movement and sound data," in *Proc. 2nd Int. Conf. Pervasive Comput. Technol. Healthcare*, 2008, pp. 103–107.

[70] S. Chen et al., "Butler, not servant: A human-centric smart home energy management system," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 27–33, Feb. 2017.

[71] A. M. Ortiz, D. Hussein, S. Park, S. N. Han, and N. Crespi, "The cluster between Internet of Things and social networks: Review and research challenges," *IEEE Internet Things J.*, vol. 1, no. 3, pp. 206–215, Jun. 2014.

[72] L. Atzori, A. Iera, and G. Morabito, "SIoT: Giving a social structure to the Internet of Things," *IEEE Commun. Lett.*, vol. 15, no. 11, pp. 1193–1195, Nov. 2011.

[73] Ç. B. Aslan, R. B. Sağlam, and S. Li, "Automatic detection of cyber security related accounts on online social networks: Twitter as an example," in *Proc. 9th Int. Conf. Social Media Soc.*, 2018, pp. 236–240.

[74] B. Requena, G. Cassani, J. Tagliabue, C. Greco, and L. Lacasa, "Shopper intent prediction from clickstream e-commerce data with minimal browsing information," *Sci. Rep.*, vol. 10, no. 1, pp. 1–23, 2020.

[75] S. P. Gochhayat et al., "LISA: Lightweight context-aware IoT service architecture," *J. Cleaner Prod.*, vol. 212, pp. 1345–1356, Mar. 2019.

[76] Y. S. Kim, B.-J. Yum, J. Song, and S. M. Kim, "Development of a recommender system based on navigational and behavioral patterns of customers in e-commerce sites," *Expert Syst. Appl.*, vol. 28, no. 2, pp. 381–393, 2005.

[77] Y. Zhang et al., "Sequential click prediction for sponsored search with recurrent neural networks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 28, 2014, pp. 1–7.

[78] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma, "Neural attentive session-based recommendation," in *Proc. ACM Conf. Inf. Knowl. Manage.*, 2017, pp. 1419–1428.

[79] H. Yin, H. Chen, X. Sun, H. Wang, Y. Wang, and Q. V. H. Nguyen, "SPTF: A scalable probabilistic tensor factorization model for semantic-aware behavior prediction," in *Proc. IEEE Int. Conf. Data Min. (ICDM)*, 2017, pp. 585–594.

[80] C. O. Sakar, S. O. Polat, M. Katircioglu, and Y. Kastro, "Real-time prediction of online shoppers' purchasing intention using multilayer perceptron and LSTM recurrent neural networks," *Neural Comput. Appl.*, vol. 31, no. 10, pp. 6893–6908, 2019.

[81] T. M. Cover, *Elements of Information Theory*. Somerset, NJ, USA: Wiley, 1999.

[82] M. Bratman, *Intention, Plans, and Practical Reason*, vol. 10. Cambridge, MA, USA: Harvard Univ. Press, 1987.

[83] A. Rashid, M. S. Farooq, A. Abid, T. Umer, A. K. Bashir, and Y. B. Zikria, "Social media intention mining for sustainable information systems: Categories, taxonomy, datasets and challenges," *Complex Intell. Syst.*, vol. 2021, pp. 1–27, Apr. 2021.

[84] M. Hamroun and M. S. Gouider, "A survey on intention analysis: Successful approaches and open challenges," *J. Intell. Inf. Syst.*, vol. 55, no. 3, pp. 423–443, 2020.

[85] S. S. Date, "A comprehensive review on intents, intention mining and intention classification," *Int. J. Sci. Res*, vol. 9, no. 11, pp. 16–20, 2020.

[86] O. E. Diaz, M. G. Perez, and J. E. Lascano, "Literature review about intention mining in information systems," *J. Comput. Inf. Syst.*, vol. 61, no. 4, pp. 295–304, 2021.

[87] J. C. McCall, D. P. Wipf, M. M. Trivedi, and B. D. Rao, "Lane change intent analysis using robust operators and sparse Bayesian learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 3, pp. 431–440, Sep. 2007.

[88] B. Morris, A. Doshi, and M. Trivedi, "Lane change intent prediction for driver assistance: On-road design and evaluation," in *Proc. IEEE Intell. Veh. Symp. (IV)*, 2011, pp. 895–901.

[89] D. D. Salvucci and A. Liu, "The time course of a lane change: Driver control and eye-movement behavior," *Transp. Res. F, Traffic Psychol. Behav.*, vol. 5, no. 2, pp. 123–132, 2002.

[90] D. D. Salvucci, "Modeling driver behavior in a cognitive architecture," *Human Factors*, vol. 48, no. 2, pp. 362–380, 2006.

[91] F. Lethaus, M. R. Baumann, F. Köster, and K. Lemmer, "A comparison of selected simple supervised learning algorithms to predict driver intent based on gaze data," *Neurocomputing*, vol. 121, pp. 108–130, Dec. 2013.

[92] Z. Huang, J. Wu, and C. Lv, "Driving behavior modeling using naturalistic human driving data with inverse reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 10239–10251, Aug. 2022.

[93] A. Rasouli, I. Kotseruba, T. Kunic, and J. K. Tsotsos, "PIE: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 6262–6271.

[94] B. Liu et al., "Spatiotemporal relationship reasoning for pedestrian intent prediction," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 3485–3492, Apr. 2020.

[95] L. Sun, Z. Yan, S. M. Mellado, M. Hanheide, and T. Duckett, "3DOF pedestrian trajectory prediction learned from long-term autonomous mobile robot deployment data," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2018, pp. 5942–5948.

[96] F. Flohr, M. Dumitru-Guzu, J. F. Kooij, and D. M. Gavrila, "Joint probabilistic pedestrian head and body orientation estimation," in *Proc. IEEE Intell. Veh. Symp. Process.*, 2014, pp. 617–622.

[97] A. T. Schulz and R. Stiefelhagen, "Pedestrian intention recognition using latent-dynamic conditional random fields," in *Proc. IEEE Intell. Veh. Symp. (IV)*, 2015, pp. 622–627.

[98] K. M. Abughalieh and S. G. Alawneh, "Predicting pedestrian intention to cross the road," *IEEE Access*, vol. 8, pp. 72558–72569, 2020.

[99] H. Wu, L. Wang, S. Zheng, Q. Xu, and J. Wang, "Crossing-road pedestrian trajectory prediction based on intention and behavior identification," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, 2020, pp. 1–6.

[100] K. Saleh, M. Hossny, and S. Nahavandi, "Long-term recurrent predictive model for intent prediction of pedestrians via inverse reinforcement learning," in *Proc. Digit. Image Comput. Techn. Appl. (DICTA)*, 2018, pp. 1–8.

[101] J. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, 2015, pp. 3995–4001.

[102] K. Safi, S. Mohammed, F. Attal, M. Khalil, and Y. Amirat, "Recognition of different daily living activities using hidden Markov model regression," in *Proc. 3rd Middle East Conf. Biomed. Eng. (MECBME)*, 2016, pp. 16–19.

[103] W. Ruan, "Unobtrusive human localization and activity recognition for supporting independent living of the elderly," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, 2016, pp. 1–3.

[104] A. Fleury, M. Vacher, and N. Noury, "SVM-based multimodal classification of activities of daily living in health smart homes: Sensors, algorithms, and first experimental results," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 2, pp. 274–283, Mar. 2010.

[105] T. Chen, H. Yin, H. Chen, R. Yan, Q. V. H. Nguyen, and X. Li, "Air: Attentional intention-aware recommender systems," in *Proc. IEEE 35th Int. Conf. Data Eng. (ICDE)*, 2019, pp. 304–315.

[106] C.-Y. Wu, A. Ahmed, A. Beutel, A. J. Smola, and H. Jing, "Recurrent recommender networks," in *Proc. 10th ACM Int. Conf. Web Search Data Min.*, 2017, pp. 495–503.

[107] L. Luceri, S. Giordano, and E. Ferrara, "Detecting troll behavior via inverse reinforcement learning: A case study of Russian trolls in the 2016 U.S. election," in *Proc. Int. AAAI Conf. Web Social Media*, vol. 14, 2020, pp. 417–427.

[108] P. Schydlo, M. Rakovic, L. Jamone, and J. Santos-Victor, "Anticipation in human–robot cooperation: A recurrent neural network approach for multiple action sequences prediction," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2018, pp. 5909–5914.

[109] Z. Wang, M. P. Deisenroth, H. B. Amor, D. Vogt, B. Schölkopf, and J. Peters, "Probabilistic modeling of human movements for intention inference," in *Proc. 8th Robot. Sci. Syst.*, 2012, pp. 1–8.

[110] H. S. Koppula and A. Saxena, "Anticipating human activities using object affordances for reactive robotic response," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 14–29, Jan. 2016.

[111] K. Ramirez-Amaro, M. Beetz, and G. Cheng, "Understanding the intention of human activities through semantic perception: Observation, understanding and execution on a humanoid robot," *Adv. Robot.*, vol. 29, no. 5, pp. 345–362, 2015.

[112] R. Liu and X. Zhang, "Fuzzy context-specific intention inference for robotic caregiving," *Int. J. Adv. Robot. Syst.*, vol. 13, no. 5, 2016, Art. no. 1729881416662780.

[113] S.-C. Hsu, Y.-W. Wang, and C.-L. Huang, "Human object identification for human–robot interaction by using fast R-CNN," in *Proc. 2nd IEEE Int. Conf. Robot. Comput. (IRC)*, 2018, pp. 201–204.

[114] N. Rhinehart and K. M. Kitani, "First-person activity forecasting with online inverse reinforcement learning," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 3696–3705.

[115] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. ICML*, 2001, pp. 282–289.

[116] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[117] P. M. Kebria, A. Khosravi, S. M. Salaken, and S. Nahavandi, "Deep imitation learning for autonomous vehicles based on convolutional neural networks," *IEEE/CAA J. Automatica Sinica*, vol. 7, no. 1, pp. 82–95, Jan. 2020.

[118] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788.

[119] A. Y. Ng and S. Russell, "Algorithms for inverse reinforcement learning," in *Proc. ICML*, vol. 1, 2000, p. 2.

[120] S. Das and A. Lavoie, "The effects of feedback on human behavior in social media: An inverse reinforcement learning model," in *Proc. Int. Conf. Auton. Agents Multi-Agent Syst.*, 2014, pp. 653–660.

[121] A. Likmeta, A. M. Metelli, G. Ramponi, A. Tirinzoni, M. Giuliani, and M. Restelli, "Dealing with multiple experts and non-stationarity in inverse reinforcement learning: An application to real-life problems," *Mach. Learn.*, vol. 110, no. 9, pp. 2541–2576, 2021.

[122] S. J. Lee and Z. Popović, "Learning behavior styles with inverse reinforcement learning," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 1–7, 2010.

[123] Z. Zhang, Z. Mo, Y. Chen, and J. Huang, "Reinforcement learning behavioral control for nonlinear autonomous system," *IEEE/CAA J. Automatica Sinica*, vol. 9, no. 9, pp. 1561–1573, Sep. 2022.

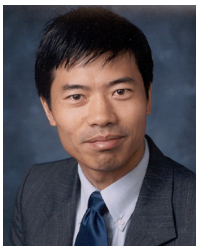
[124] J. Wang, Q. Zhang, and D. Zhao, "Highway lane change decision-making via attention-based deep reinforcement learning," *IEEE/CAA J. Automatica Sinica*, vol. 9, no. 3, pp. 567–569, Mar. 2022.

[125] Z. Cao, C. Lin, M. Zhou, and R. Huang, "Scheduling semiconductor testing facility by using cuckoo search algorithm with reinforcement learning and surrogate modeling," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 2, pp. 825–837, Apr. 2019.
[126] D. B. Lenat, "CYC: A large-scale investment in knowledge infrastructure," *Commun. ACM*, vol. 38, no. 11, pp. 33–38, 1995.
[127] G. Almashaqbeh and R. Solomon, "SoK: Privacy-preserving computing in the blockchain era," in *Proc. IEEE 7th Eur. Symp. Security Privacy (EuroSP)*, 2022, pp. 124–139.
[128] E. Shi, T.-H. H. Chan, E. G. Rieffel, R. Chow, and D. X. Song, "Privacy-preserving aggregation of time-series data," in *Proc. Netw. Distrib. Syst. Security Symp.*, 2011, pp. 1–17.
[129] "Data transfer project overview and fundamentals." Accessed: Jul. 20, 2018. [Online]. Available: https://datatransferproject.dev/dtp-overview.pdf
[130] "MyData: An introduction to human-centric use of personal data." Accessed: Jul. 8, 2020. [Online]. Available: https://mydata.org/wp-content/uploads/sites/5/2020/08/mydata-white-paper-english-2020.pdf
[131] "Solid: Your data, your choice. Advancing Web standards to empower people." Accessed: Jul. 16, 2021. [Online]. Available: https://solidproject.org/
[132] P. Zhang and M. Zhou, "Security and trust in blockchains: Architecture, key technologies, and open issues," *IEEE Trans. Comput. Social Syst.*, vol. 7, no. 3, pp. 790–801, Jun. 2020.
[133] M. Blum, P. Feldman, and S. Micali, "Non-interactive zero-knowledge and its applications," in *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*. New York, NY, USA: Assoc. Comput. Mach., 2019, pp. 329–349.
[134] "Anonymous P2P." Wikipedia. Dec. 2021. Accessed: Jan. 5, 2022. [Online]. Available: https://en.wikipedia.org/wiki/Anonymous_P2P
[135] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–15.
[136] "Home automation." Wikipedia. Jun. 2021. Accessed: Jul. 16, 2021. [Online]. Available: https://en.wikipedia.org/wiki/Home_automation
[137] S. Feng, P. Setoodeh, and S. Haykin, "Smart home: Cognitive interactive people-centric Internet of Things," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 34–39, Feb. 2017.
[138] R. Mulero et al., "An IoT-aware approach for elderly-friendly cities," *IEEE Access*, vol. 6, pp. 7941–7957, 2018.
[139] P. Paolini, N. D. Blas, S. Copelli, and F. Mercalli, "City4Age: Smart cities for health prevention," in *Proc. IEEE Int. Smart Cities Conf. (ISC2)*, 2016, pp. 1–4.
[140] P. Abril-Jiménez, J. R. Lacal, S. de los Ríos Pérez, M. Páramo, J. B. M. Colomer, and M. T. A. Waldmeyer, "Ageing-friendly cities for assessing older adults' decline: IoT-based system for continuous monitoring of frailty risks using smart city infrastructure," *Aging Clin. Exp. Res.*, vol. 32, no. 4, pp. 663–671, 2020.
[141] A. Almeida, A. Fiore, L. Mainetti, R. Mulero, L. Patrono, and P. Rametta, "An IoT-aware architecture for collecting and managing data related to elderly behavior," *Wireless Commun. Mobile Comput.*, vol. 2017, pp. 1–17, Dec. 2017.
[142] S. K. Sood and I. Mahajan, "IoT-fog-based healthcare framework to identify and control hypertension attack," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1920–1927, Apr. 2019.
[143] A. Casalino, C. Messeri, M. Pozzi, A. M. Zanchettin, P. Rocco, and D. Prattichizzo, "Operator awareness in human–robot collaboration through wearable vibrotactile feedback," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 4289–4296, Oct. 2018.
[144] T. Diethe, N. Twomey, M. Kull, P. Flach, and I. Craddock, "Probabilistic sensor fusion for ambient assisted living," 2017, *arXiv:1702.01209*.
[145] A. Jalal, S. Kamal, and D. Kim, "A depth video-based human detection and activity recognition using multi-features and embedded hidden Markov models for health care monitoring systems," *Int. J. Interactive Multimedia Artif. Intell.*, vol. 4, no. 4, pp. 52–62, 2017.
[146] Y. Wang, S. Cang, and H. Yu, "A survey on wearable sensor modality centred human activity recognition in health care," *Expert Syst. Appl.*, vol. 137, pp. 167–190, Dec. 2019.
[147] G. Nyman. "Internet of Behaviors (IoB) in good company—Future behavior markets | Göte Nyman's (gotepoem) Blog." Jan. 2021. Accessed: Jul. 15, 2021. [Online]. Available: https://gotepoem.wordpress.com/2021/01/13/internet-of-behaviors-iob-in-good-company/
[148] H. Han, W. Ma, M. Zhou, Q. Guo, and A. Abusorrah, "A novel semi-supervised learning approach to pedestrian reidentification," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 3042–3052, Feb. 2021.
[149] T. Zhang, J. Wang, and M. Q.-H. Meng, "Generative adversarial network based heuristics for sampling-based path planning," *IEEE/CAA J. Automatica Sinica*, vol. 9, no. 1, pp. 64–74, Jan. 2022.
[150] "Amazon Alexa voice AI | Alexa developer official site." Amazon. 2021. Accessed: Jul. 16, 2021. [Online]. Available: https://developer.amazon.com/en-US/alexa
[151] "HomeKit—All accessories." Apple. 2021. Accessed: Jul. 16, 2021. [Online]. Available: https://www.apple.com/shop/accessories/all/homekit
[152] "Cloud computing services: Microsoft Azure." Microsoft. Jul. 2021. Accessed: Jul. 16, 2021. [Online]. Available: https://azure.microsoft.com/en-us/
[153] "Trusted execution environment." Wikipedia. Jul. 2021. Accessed: Jul. 16, 2021. [Online]. Available: https://en.wikipedia.org/wiki/Trusted_execution_environment
[154] W. Duo, M. Zhou, and A. Abusorrah, "A survey of cyber attacks on cyber physical systems: Recent advances and challenges," *IEEE/CAA J. Automatica Sinica*, vol. 9, no. 5, pp. 784–800, May 2022.
[155] D. You, S. Wang, M. Zhou, and C. Seatzu, "Supervisory control of Petri Nets in the presence of replacement attacks," *IEEE Trans. Autom. Control*, vol. 67, no. 3, pp. 1466–1473, Mar. 2022.
[156] X. Wang et al., "Privacy-preserving content dissemination for vehicular social networks: Challenges and solutions," *IEEE Commun. Surv. Tut.*, vol. 21, no. 2, 2nd Quart., 2019.
[157] M. Ghahramani, Y. Qiao, M. Zhou, A. O. Hagan, and J. Sweeney, "AI-based modeling and data-driven evaluation for smart manufacturing processes," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 4, pp. 1026–1037, Jul. 2020.
[158] Q. Fang et al., "Process monitoring, diagnosis and control of additive manufacturing," *IEEE Trans. Automat. Sci. Eng.*, early access, Nov. 2022, doi: 10.1109/TASE.2022.3215258.
[159] X. Ren, Z. Li, M. Zhou, and Y. Hu, "Human intention-aware motion planning and adaptive fuzzy control for a collaborative robot with flexible joints," *IEEE Trans. Fuzzy Syst.*, early access, Nov. 2022, doi: 10.1109/TFUZZ.2022.3225660.
[160] Q. Li, R. Gravina, Y. Li, S. H. Alsamhi, F. Sun, and G. Fortino, "Multi-user activity recognition: Challenges and opportunities," *Inf. Fusion*, vol. 63, pp. 121–135, 2020.
[161] R. Yang, Z. Ding, C. Jiang, and M. Zhou, "Modeling and analysis of three properties of mobile interactive systems based on variable Petri nets," *IEEE Trans. Automat. Sci. Eng.*, early access, Sep. 2022, doi: 10.1109/TASE.2022.3206999.

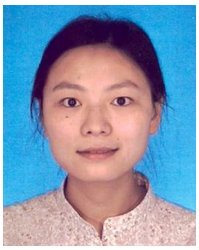
Qinglin Zhao (Senior Member, IEEE) received the B.S. degree in mathematics education from the Hubei University, Wuhan, China, in 1998, the M.S. degree in applied mathematics from the Huazhong University of Science and Technology, Wuhan, in 2001, and the Ph.D. degree in computer architecture from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2005. From May 2005 to August 2009, he worked as a Postdoctoral Researcher with The Chinese University of Hong Kong and the Hong Kong University of Science and Technology. Since September 2009, he has been with the School of Computer Science and Engineering, Macau University of Science and Technology, where he is currently a Professor. He has published more than 80 peer-reviewed papers, including 20 IEEE Transactions papers, held more than 30 patents, including eight U.S. patents. His research interests include blockchain and decentralization computing, machine learning, Internet of Things, wireless communications and networking, cloud/fog computing, and software-defined wireless networking. He received the BOC Excellent Research Award of Macau University of Science and Technology in 2011 and 2015.

Guangcheng Li received the M.S. degree in computer and information systems from the Macau University of Science and Technology, Macau, in 2018, where he is currently pursuing the Ph.D. degree in computer technology and application. His current research interests include blockchain and decentralization computing, machine learning and its applications, cloud/edge computing, and Internet of Things.

Jincheng Cai received the M.S. degree in computer technology from Shantou University, Shantou, China, in 2018. He is currently pursuing the Ph.D. degree in artificial intelligence with the Macau University of Science and Technology, Macau, China. His research interests are in blockchain, machine learning, and Internet of Things.



MengChu Zhou (Fellow, IEEE) received the B.S. degree in control engineering from the Nanjing University of Science and Technology, Nanjing, China, in 1983, the M.S. degree in automatic control from the Beijing Institute of Technology, Beijing, China, in 1986, and the Ph.D. degree in computer and systems engineering from Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990. He joined the New Jersey Institute of Technology (NJIT), Newark, NJ, USA, in 1990. He is a Distinguished Professor of Electrical and Computer Engineering and the Director of Discrete-Event Systems Laboratory. He is also with the School of Information and Electronic Engineering, Zhejiang Gongshang University, Hangzhou, China. He has led or participated in over 60 research and education projects with total budget over 12M, funded by the National Science Foundation, Department of Defense, NIST, New Jersey Science and Technology Commission, and industry. He has over 1000 publications, including 14 books, over 700 journal papers (over 600 in IEEE TRANSACTIONS), and 32 book chapters. He holds 31 patents and several pending ones. His recently coauthored books include *Sustainable Manufacturing Systems: An Energy Perspective* (Hoboken, NJ, USA: IEEE Press/Wiley, 2022 with L. Li) and *Supervisory Control and Scheduling of Resource Allocation Systems: Reachability Graph Perspective* (Hoboken, NJ, USA: IEEE Press/Wiley, 2020 with B. Huang). His research interests are in intelligent automation, machine leaning, Petri nets, robotics, Internet of Things, big data, cloud/edge computing, transportation, and energy systems. He is a recipient of the Excellence in Research Prize and Medal from NJIT, the Humboldt Research Award for U.S. Senior Scientists from the Alexander von Humboldt Foundation, and the Franklin V. Taylor Memorial Award and the Norbert Wiener Award from IEEE SMC Society, the Computer-Integrated Manufacturing University-Lead Award from the Society of Manufacturing Engineers, the Distinguished Service Award from the IEEE Robotics and Automation Society, and the Edison Patent Award from the Research and Development Council of New Jersey. He has been among most highly cited scholars since 2012 and ranked top one in the field of engineering worldwide in 2012 by Web of Science. He is the Founding Editor of IEEE Press Book Series on Systems Science and Engineering. He served as the Editor-in-Chief for IEEE/CAA JOURNAL OF AUTOMATICA SINICA, an Associate Editor for IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, and IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, and the Editor for IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING. He served as the Guest Editor for many journals, including IEEE INTERNET OF THINGS JOURNAL, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, and IEEE TRANSACTIONS ON SEMICONDUCTOR MANUFACTURING. He is currently an Associate Editor of *Research*, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE INTERNET OF THINGS JOURNAL, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, and *Frontiers of Information Technology & Electronic Engineering*. He is the Founding Chair/Co-Chair of Technical Committee on AI-Based Smart Manufacturing Systems and Technical Committee on Humanized Crowd Computing of IEEE Systems, Man, and Cybernetics Society, Technical Committee on Semiconductor Manufacturing Automation, and Technical Committee on Digital Manufacturing and Human-Centered Automation of IEEE Robotics and Automation Society. He is the Chair of Fellow Evaluation Committee of Chinese Association of Automation, and a member of IEEE Fellow Evaluation Committee for IEEE Systems, Man, and Cybernetics Society and IEEE Robotics and Automation Society. He is also a member of IEEE TAB Periodicals Committee and Periodicals Review and Advisory Committee. He was the General Chair of IEEE Conference on Automation Science and Engineering, Washington, DC, USA, 23–26 August, 2008, the General Co-Chair of 2003 IEEE International Conference on System, Man and Cybernetics (SMC), Washington, DC, USA, 5–8 October, 2003 and 2019 IEEE International Conference on SMC, Bari, Italy, 6–9 October, 2019, the Founding General Co-Chair of 2004 IEEE International Conference on Networking, Sensing and Control, Taipei, 21–23 March, 2004, and the General Chair of 2006 IEEE International Conference on Networking, Sensing and Control, Ft. Lauderdale, FL, USA, 23–25 April, 2006. He was the Program Chair of 2010 IEEE International Conference on Mechatronics and Automation, Xi'an, China, 4–7 August, 2010, 1998 and 2001 IEEE International Conference on SMC, and 1997 IEEE International Conference on Emerging Technologies and Factory Automation. He is a Life Member of the Chinese Association for Science and Technology-USA and served as its President in 1999. He is a Fellow of the International Federation of Automatic Control, the American Association for the Advancement of Science, the Chinese Association of Automation, and the National Academy of Inventors.



Li Feng received the M.S. degree in operation research from the Department of Mathematics, University of Hong Kong, Hong Kong, in 2007, and the Ph.D. degree in electronic information technology from the School of Computer Science and Engineering, Macau University of Science and Technology, Macau, China, in 2013, where she is currently an Associate Professor. Her current research interests include wireless and mobile networks, power saving, software defined networking, and performance analysis.