New Frontiers in IoT: Networking, Systems, Reliability, and Security Challenges

Saurabh Bagchi, Tarek F. Abdelzaher, Ramesh Govindan, Prashant Shenoy, Akanksha Atrey, Pradipta Ghosh, and Ran Xu

(Invited Paper)

***Abstract*—The field of IoT has blossomed and is positively influencing many application domains. In this article, we bring out the unique challenges this field poses to research in computer systems and networking. The unique challenges arise from the unique characteristics of IoT systems such as the diversity of application domains where they are used and the increasingly demanding protocols they are being called upon to run (such as video and LIDAR processing) on constrained resources (on-node and network). We show how these open challenges can benefit from foundations laid in other areas, such as fifth-generation network cellular protocols, machine learning model reduction, and device–edge–cloud offloading. We then discuss the unique challenges for reliability, security, and privacy posed by IoT systems due to their salient characteristics which include heterogeneity of devices and protocols, dependence on the physical environment, and the close coupling with humans. We again show how open research challenges benefit from the reliability, security, and privacy advancements in other areas. We conclude by providing a vision for a desirable end state for IoT systems.**

***Index Terms*—Foundations, Internet of Things, path forward, reliability and security challenges, systems and networking challenges.**

I. Introduction

IoT IS an interdisciplinary field as evidenced by the breadth of disciplines that contribute techniques to this field. It involves hardware, software, and often humans, with resource constraints on the hardware (cost, complexity, and energy sources) and the software (complexity, compute and memory footprint, and disconnected mode of operation). To focus the discussion, let us lay down a working definition of IoT, and the ways it is different from the allied disciplines of cyber–physical systems, networked control systems, and embedded systems.

> The *Internet of Things* refers to networked devices that interact with their physical surroundings and communicate over wireless networks in social contexts to offer a human-centric application value.

Accordingly, concerns in IoT intersect with cyber–physical systems in that the system may contain embedded components and may include associated control algorithms. However, IoT systems are by definition distributed, putting more emphasis on end-to-end systems challenges, scalability, and network support within the end-to-end application context, as opposed to, say, control systems. Also, IoT systems, by virtue of distribution and scale, are often multipurpose. As such, specific capabilities may be put together dynamically, leading to challenges in composability and integration.

Application Context: IoT application areas fall into three categories.

1) Enhance our spaces, in which humans live (e.g., homes and offices).
2) Empower the devices we use (e.g., appliances and vehicles).
3) Improve the efficiency of production and delivery systems (e.g., agriculture, the power grid, and manufacturing) so as to improve human life and productivity.

An important aspect of these applications is the human in the loop, to generate sensor readings (e.g., crowdsourcing), to validate control decisions, or to act upon the actuation commands.

Challenges and Constraints: There are some key technical challenges that are salient to the IoT domain. There are often constraints on hardware and software that preclude heavy-duty computation (such as expensive asymmetric cryptographic operations) or significant storage overhead (such as a large ensemble of models). There is often a constraint on the wireless networking available to the nodes—it is often low data rate and there may also be periods of disconnected operation, such as due to wireless brownouts or interference from multiple devices operating in a public ISM band. There is often a real-time constraint on the tasks, else financial loss or human

Manuscript received May 14, 2020; accepted June 19, 2020. Date of publication July 8, 2020; date of current version December 11, 2020. This work was supported in part by the National Science Foundation under Grant CNS-1718637 and Grant CNS-1845192, and in part by Army Research Lab under Contract W911NF-17-2-0196 and Contract W911NF-20-2-0026. *(Corresponding author: Saurabh Bagchi.)*

Saurabh Bagchi is with the School of Electrical and Computer Engineering and Department of Computer Science, Purdue University, West Lafayette, IN 47907 USA (e-mail: sbagchi@purdue.edu).

Tarek F. Abdelzaher is with the Department of Computer Science, University of Illinois at Urbana–Champaign, Urbana, IL 61801 USA (e-mail: zaher@cs.uiuc.edu).

Ramesh Govindan and Pradipta Ghosh are with the Department of Computer Science, University of Southern California, Los Angeles, CA 90089 USA (e-mail: ramesh@usc.edu; pradiptg@usc.edu).

Prashant Shenoy is with the College of Information, University of Massachusetts at Amherst, Amherst, MA 01003 USA (e-mail: shenoy@cs.umass.edu).

Akanksha Atrey is with the College of Information and Computer Sciences, University of Massachusetts at Amherst, Amherst, MA 01003 USA (e-mail: aatrey@cs.umass.edu).

Ran Xu is with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: xu943@purdue.edu).

Digital Object Identifier 10.1109/JIOT.2020.3007690

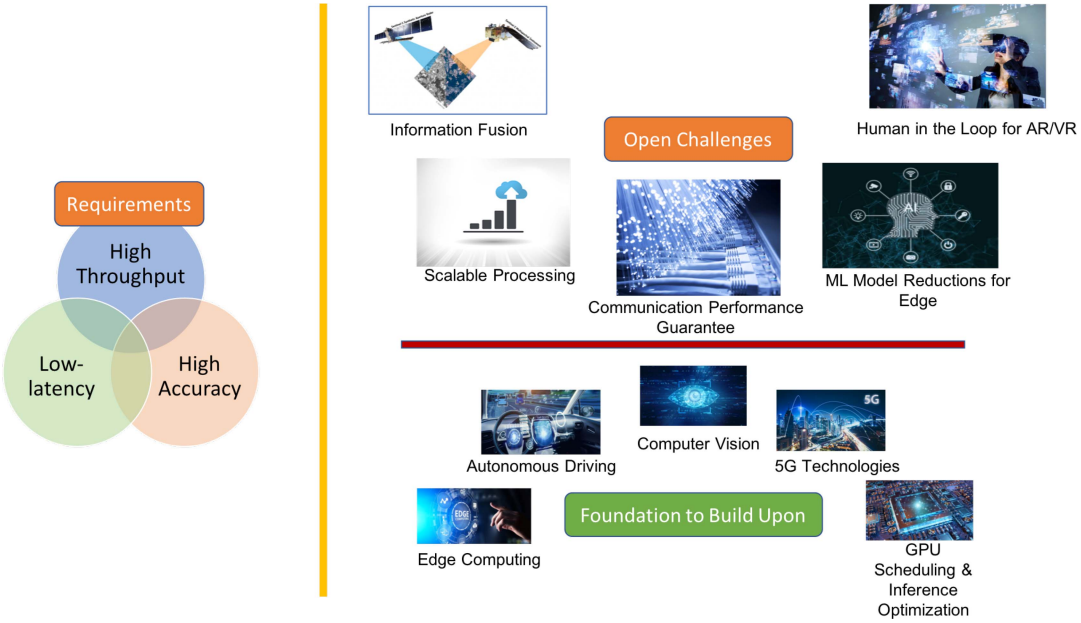

Fig. 1. Overview of the novel systems and networking requirements and challenges in IoT systems and foundations from the current work that we can build upon.

discomfort may occur, such as inefficient electricity use in an industrial setting or an uncomfortable indoor environment. The heterogeneity of devices and the corresponding wireless protocols they support pose challenging engineering problems. For example, one device may have access to trusted hardware such as ARM TrustZone while the majority of devices may not have such hardware; some may speak only ZigBee as a short-range wireless protocol and not have the long-range cellular or LoRa stack, while other nodes may have the capability for long-range communication. Finally, the human in the loop brings forth challenges for the operation (must be simple enough in the parts where human interaction is needed), maintainability (must not require complex or frequent maintenance operations), and safety (must not endanger human users).

In this work, we present the broad open challenges in IoT, from the computer systems and networking aspect and from the reliability, security, and privacy aspects. Within each, we first look at the foundations that we can build on in terms of analytical, architectural, and systems building blocks already available to us.

II. Systems and Networking Challenges

A. Unique Challenges

IoT systems required substantial systems design innovation, primarily because of: 1) the diversity of sensors and actuators with different wireless technologies they use; 2) the variety of indoor and outdoor locations they are deployed in; 3) the unpredictable conditions under which they are deployed, including unpredictability in availability and quality of network connectivity; 4) the interaction of humans in the loop; and 5) the energy and compute power constraints.

In recent years, researchers have explored a wide range of challenges related to IoT networks with small, battery-powered sensors. At present, we can concede that we understand that space well ([124], [134]). The next phase of IoT research will focus on analytics and control using richer sensors that provide various forms of visual information: camera, radar, LiDAR, stereo cameras, etc. These "visual"[1] sensors provide semantically rich information but can require significant processing to extract this information. Equally important, with decreasing cost and form factors, sensors, such as cameras and LiDARs are being deployed densely, even on personal mobile devices. These will enhance the quality of decisions for most applications discussed above. Beyond richer sensors, future IoT systems will include autonomous drones and vehicles that add significant complexity to control and actuation. Constraints imposed by compute and network will be the primary bottlenecks in realizing the full potential of these future IoT systems.

B. Performance Requirements

Before delving into the challenges, we first identify (Fig. 1) some key requirements for an IoT network consisting of sensors (such as cameras and LiDARs), and actuators (such as drones and vehicles).

High Throughput/Frames: While a low-power temperature sensor generates a few bytes of data every minute, a camera or LiDAR can generate data at several hundred Mbps. Their devices are often connected via a wireless interface to a cloud or edge cluster to process the frames. Transmitting raw sensor data may be infeasible given even future wireless standards.

Low Latency: For actuating a drone or a vehicle, an IoT system will need to support ultralow end-to-end latencies on the order of a few milliseconds. The two main sources of latency in the IoT control loop are: 1) processing the sensor data and 2) communication delay.

High Accuracy: Especially for visual sensing, accuracy is an essential performance metric. For example, the accuracy

[1]For simplicity, we call these visual sensors, because they can "see" the environment, albeit in different ways than humans might in some cases.

of sensing and tracking objects can affect vehicular or drone control significantly.

Future IoT systems will need to simultaneously satisfy all of these requirements, and visual sensors, together with near real-time control of vehicles and drones, represent extreme points in the space of requirements across all three dimensions.

C. Research Challenges

We now describe new research challenges that arise as a result of the three requirements mentioned above.

Information Extraction and Fusion: Future IoT networks will include a wide range of sensors in terms of sensing frequency and the amount of data they generate. *Rich sensors*, such as cameras and LiDAR produce significantly more information than a low-end thermostat or an accelerometer. In order to fuse sensors meaningfully, we need to extract only useful and nonredundant information in a timely manner. Combining sensing information from different sensing modalities is an extremely challenging task [56]. The existing systems fuse 2–3 different types of sensors, e.g., camera with LiDAR [34], [67], [111]: but do not generalize to a large number (both in type and count) of heterogeneous rich sensors. The main challenges related to information extraction and fusion in IoT with a large number of rich sensors are as follows.

Data Registration: Incorporating data from different scans or sensors to generate a unified view is often known in the computer vision and autonomous vehicle community as data registration [55]. Data registration deals with properly combining a large number of 3-D point clouds obtained from 3-D sensors, such as stereo cameras and LiDAR where each sensor generates a point cloud with respect to its own frame of reference. 3-D point clouds tend to be very large in size (from hundreds of megabytes to hundreds of gigabytes) and thus cannot be exchanged between devices. On the other hand, such point clouds tend to provide very fine grained and extremely dynamic sensing information. Therefore, the timeliness and accuracy of data registration are important for any IoT applications that rely on the combined point cloud.

Foundations to Build Upon: Data registration has been studied for many years in the context of autonomous vehicles and robotics maneuvers and path planning [15], [75]. Most existing work relies on the fact that the sensors are co-located or located at close proximity (on the vehicle or robot) with significant overlap in the sensing regions. In the context of IoT, roadside 3-D sensors may cover a large area with very small or limited overlap in the sensing area [140]. For such situations, the existing solutions may be inadequate. Recent work has started exploring the problem of data registration in the context of infrastructure (roadside) LiDARs [136], [139]. Nonetheless, these solutions do not address data registration for roadside 3-D sensors at scale and are limited to only 2–3 sensors at maximum. The state-of-the-art methods also lack data registration techniques involving heterogeneous IoT infrastructure sensors, such as LiDAR and stereo cameras.

Detection/Classification/Tracking Techniques: The existing computer vision detection/classification/tracking techniques for camera and LiDAR processing are typically resource-heavy, limited to a small number of devices, and performed on cloud infrastructure [40]. In an IoT environment, significant innovation is required for real-time sensing across multiple devices, such as tracking activity across a large number of overlapping and nonoverlapping cameras [77] or LiDARs.

Foundations to Build Upon: Object detection, classification, and tracking are active areas of research in computer vision [20], [101]. These tasks are usually performed by training a deep neural network (DNN) with a large data set catered toward a particular detection, classification, or tracking task. For illustration, let us consider the task of real-time human activity detection in live camera streams. There exist a large number of monolithic DNN models [49], [108], [121] that extract features from video streams and predict actions of every human appearing in the video. To detect interactions, a class of methods analyze the moving trajectories of objects near a person to predict the interaction between the person and the object [8], [86] while another class of methods opt to train separate DNNs to detect group behavior such as "walk in group" and "stand in queue" ([7] and [10]). While these methods perform well for a small number of detection/classification/tracking tasks (limited by the availability of relevant data sets), they cannot be tailored for any tasks outside the vocabulary and thus are not generalizable for future IoT operations. Even with the existing DNN solutions for specific activities, human intervention is required for analyzing complex activities that involve specific activities being detected by the DNN, e.g., "two men chatting, then exchanging a document, then walking in a group." Moreover, the processing time of the video steams increases exponentially on shared computation resources as the number of tracked objects increases [77]. Additional challenges appear as we scale such detections across multiple heterogeneous devices. To perform tracking across multiple video streams, we need proper synchronization of the images frames, timely processing of image frames, and reidentification of objects/humans across multiple cameras. The existing work has explored the single-camera action detection [108], [121], [142], tracking of people across multiple overlapping cameras [91], [109], [128] and nonoverlapping cameras [29], [102], [118]. However, due to the complexity of the problem, very few researchers have looked into a real-time generic tracking and detection across multiple cameras which is required for future IoT systems [77]. Similar observations can be made for almost all kinds of detection/classification/tracking state of the arts. Looking forward, significant innovation and development are required toward the detection/classification/tracking technique, which are generalizable for a large vocabulary of tasks dealing with a large number of heterogeneous imaging devices. One way to achieve this is to take advantage of the existing DNN solutions and combine them in a semantically meaningful, systematic way as shown in [77].

Compute and Communication Constraints: Processing the data stream from one camera/LiDAR is a challenging task itself. For an IoT network with multiple cameras/LiDARs, the processing time and resource requirements are very high [138]. This calls for innovations at the algorithmic level to process

a large number of sensor data streams with accuracy and processing time similar to the processing of a single sensor data stream. This has to be accomplished on platforms that are not as resource rich as server-class platforms and where the isolation guarantees among applications are weaker. In addition, transmitting multiple sensory data streams to a cloud or edge requires a lot more additional communication bandwidth than supported by a typical shared wireless medium such as WiFi [123].

Foundations to Build Upon: Scalable processing of video and LiDAR data streams is a cutting edge field of research. Both types of data require resource-heavy CNN/DNN to extract the embedded rich information. Future IoT networks will include a large number of heavy sensors such as cameras, LiDAR for smart sensing. While some applications require the processing of combined data (via data registration, explained above), others require concurrent and separate processing of individual data streams on shared compute resources. Often, based on the task query, one might need to run multiple different DNN on the same video stream. Most of the future IoT applications will rely on a chain of DNNs running on edge clusters. Researchers have looked into this problem in the context of live video streaming from multiple cameras [60], [138]. The state-of-the-art live-stream processing systems operate knobs for different performance settings (frame rate and resolution) to maximize the shared server utilization and maintain a minimum quality of service for all task queries.

While downgrading the quality of frames is a potential scalable option, it often results in lowering the accuracy of the DNN/CNN. A more recent class of approaches employ GPU multitenancy scheduling on TensorFlow serving platforms [2] to improve GPU sharing and utilization [58] on the shared edge cloud. Some state-of-the-art techniques also save GPU cycles by caching intermediate results [36], [72], lazily activating DNN [77], and batching the input for higher per-image processing speed on GPU [36]. However, all these solutions work well for a small range of applications for lower frame rate and a small number of concurrent streams ($<$10) and concurrent queries, and cannot scale to large numbers of concurrent streams. This is relevant because we anticipate that future IoT systems will include a large number of concurrent streams, multiple edge clusters, and a large number of DNNs (or chains of DNN) per image stream. To this, we need to remove any redundancy present in the input, DNN, or GPU schedule. Identifying a set of sensing-objective-specific keyframes (instead of processing every frame) is essential and is a promising field of research in this context.

Beyond video, recent work has started exploring the design of vehicular IoT systems that use depth perception sensors. For example, AVR [100] has explored a combination of several techniques, including dynamic object extraction and adaptive transmission of stereo camera point clouds to enable extended vehicular vision. Similarly, CarMap [4] efficiently uploads updates to high-definition maps over a cellular network, using a combination of techniques to produce a lean map representation that does not sacrifice positioning accuracy.

Accuracy Versus Performance Tradeoff: To support communication and processing of multiple data streams from rich sensors (such as cameras and LiDARs) with limited shared resources (bandwidth, GPU, etc.), researchers often adopt techniques to drop data frames (randomly or selectively) by keeping a set of key-frames [123]. Such approaches tend to achieve the performance requirement in terms of throughput (goodput) and runtime at a cost of reduced accuracy. However, to achieve a certain throughput, one needs to achieve accuracy above a certain threshold. Thus, the tradeoff between accuracy and performance requires careful analysis and consideration for designing systems and algorithms for future IoT. Specifically, simple application-agnostic techniques such as frame dropping may not suffice to achieve good accuracy; often, application-specific techniques that leverage problem structure to extract only information essential to the problem [4], [100] can provide orders of magnitude performance improvement while minimally impacting accuracy.

Role of Edge/Cloud Offload and Device Computation: Edge computing [107] will play a central role in future IoT networks. Often the onboard processing power of a camera/LiDAR device is unable to run necessary processing pipelines (deep learning models) to extract the embedded rich information. This calls for offloading the computation either to a cloud with large processing power at the cost of larger unpredictable delays or to a nearby edge device, or a cluster of edge devices, with enough processing power and with lower, more predictable delays. This raises a series of questions: which option should we choose, edge or cloud, or a hybrid? What data to share with the edge/cloud? How to minimize the end-to-end latency of the processing pipeline while reducing the communication overhead? The future also presupposes the possibility of having multiple heterogeneous edge devices, some of which are unmanaged while the rest are managed (by commercial organizations). Unmanaged edge implies that such devices are voluntarily contributed by the public and are unpredictably available.

Of particular interest in this context, is the introduction of machine intelligence into the IoT edge/cloud architecture [130]. IoT will push the boundaries of federated learning motivated by the fact that each individual device may be too resource constrained and by privacy requirements in IoT settings. Neural networks offer a great portable representation, much like a language virtual machine (e.g., Java and Python), that makes it possible to distribute inference algorithms across edge and cloud machines, and control the amount of communication among them. Services might: 1) generate DNN models (from client-supplied training data); 2) help with (automatic) labeling of data sets; and 3) perform model reduction (if needed for caching on the edge device). Generated models might be executed as appropriate on the server or client. This vision poses several challenges.

Model Reduction for IoT Devices: Modern machine intelligence algorithms are heavyweight. To run on a low-end IoT device, solutions are needed to reduce the computational and memory needs of machine inference. Recent work shows that model reductions of orders of magnitude are possible [76], [133]. For example, a device can cache a reduced model that identifies a number of most frequent commands,

leaving the more general (but rarely encountered) identification tasks to the cloud. Models can also be customized to specific hardware. For example, rather than minimizing computational cost, a model that fully utilizes an available GPU will give a better quality/consumption tradeoff [132]. Alternatively, the end device may choose to offload the inference to a server. Communication between the resource-constrained IoT end device and an edge/cloud-processing server will need to be compressed [39]. Autoencoder-like solutions allow asymmetric encoding/decoding where the IoT-device-side encoder (that compresses the data onto a lower dimensional manifold) is lightweight, whereas the decoder (running on an edge server) is more involved. Order-of-magnitude reduction in communication was shown using compressive offloading [131]. On the server, since improvements in result accuracy diminish with increased depth of the neural network, efficiency considerations suggest that once the desired quality is achieved, the service should refrain from executing additional layers. A scheduler may determine how many stages to execute to avoid diminishing returns.

Data Prioritization: A commonly overlooked challenge in IoT-centric machine inference contexts is one of data prioritization. When a human driver observes a scene, they instinctively prioritize regions of higher criticality in the scene (such as a child on the side of the road who might run across at any instant) over regions of lower criticality (such as buildings in the background, fire hydrants, trees, etc.). No such prioritization is done in the current machine learning (ML) software. Rather, some of the heaviest computational operations are performed on all bits of an image in every frame without prioritization. A novel stack is needed that is aware of the importance of different regions in an image. Some examples were proposed in recent literature.

Communication Requirements: IoT networks heavily rely on wireless communication for interdevice communications. State-of-the-art wireless communication technology needs to accommodate for the high throughput and low-delay requirements of future IoT networks involving cameras and LiDARs. Camera or LiDAR data streaming via state-of-the-art wireless communication standards experience many challenges affecting the performance, such as unnecessary retransmission, bandwidth fluctuation due to dynamic channel quality, lack of dedicated channel access due to contention-based MAC protocol, and heterogeneous devices sharing that same medium [57]. The situation is likely to become more adverse by the incorporation of augmented reality (AR) and virtual reality (VR) devices in future IoT networks. A single VR device requires hundreds of megabytes to couple gigabytes of dedicated bandwidth for a reasonable user experience [13]. While data compression and coding techniques [125] can help to reduce the bandwidth requirements, current wireless networking technologies still fall short of fulfilling the bandwidth and performance guarantee requirements. Moreover, in a wireless network with a large number of heterogeneous devices (cameras, LiDARs, etc.) and actuators (drones, autonomous cars), the network needs to have provision for prioritizing certain types of traffic, such as control traffic, as well as maintain fairness and performance guarantees among multiple data streams.

Foundations to Build Upon: The fifth-generation network (5G) is the obvious core wireless technology for the future IoT network as it will allow for a higher data rate (up to tens of GBps) which is orders of magnitude higher than the current wireless technologies [99]. To this, researchers have started to explore 5G-based communication architectures for future IoT [94]. While 5G offers significantly more bandwidth and data rates, we still require technologies to offer precise control of traffic and performance in the shared wireless medium. To meet the performance demands by facilitating the flexible allocation of resources, future IoT networks must make use of recent network virtualization concepts, such as a software-defined network (SDN) [17], [117], network functions virtualization (NFV) [87], and network slicing [5].

In addition, modifications are required in streaming protocols (e.g., video) to reduce unnecessary information and save bandwidth [60], [95], [138]. Conventional streaming protocols (such as RTMP [97] for video) and encoding standards (such as H.265 [57] for video) are tailored toward maximizing user quality of experience (QoS). Such protocols tend to optimize the frame rate and resolutions to avoid unnecessary interruptions and delays. In future IoT networks, the majority of streaming will be tied to analytics where the objective is to maximize the inference accuracy and the performance objectives are different from normal live streaming. For example, in a video analytics application, frame resolution beyond a threshold has a negligible effect on the DNN/CNN-based object detection pipelines and often only a small cropped portion of the frames are used [77], [95]. In addition, sequential frames in a video stream might not have any additional information and can be dropped to save bandwidth without incurring any performance deterioration. Thus, video streaming protocols for future IoT analytics have many parameters to tune for such as frame selection—area cropping (and transmitting only the cropped area), resolution of the image, and compression that are relatively unexplored in the existing video streaming protocols. Similar scope of research lies in other types of streaming applications such as LiDAR data streaming and audio streaming.

Humans in the Loop: A key distinguishing feature of IoT systems is the human in the loop. Humans consume the output of IoT systems but may also provide inputs to add reliability and context to IoT systems [92]. While such intervention by humans in IoT systems has its advantages, modeling and analysis of these IoT systems require modeling of human behavior. This is particularly challenging due to the complex physiological, psychological, and behavioral aspects of human beings. Apart from the human modeling aspect, there also exist several system design challenges, such as minimizing human input and coping up with occasional unpredictability and unreliability of human inputs. The set of challenges is even broader in the context of AR/VR applications for future IoT networks. Consider a battlefield IoT setting where relevant roadside camera/LiDARs streams are live-fed to the VR headset of ground troops. The quality of streaming and the switching between

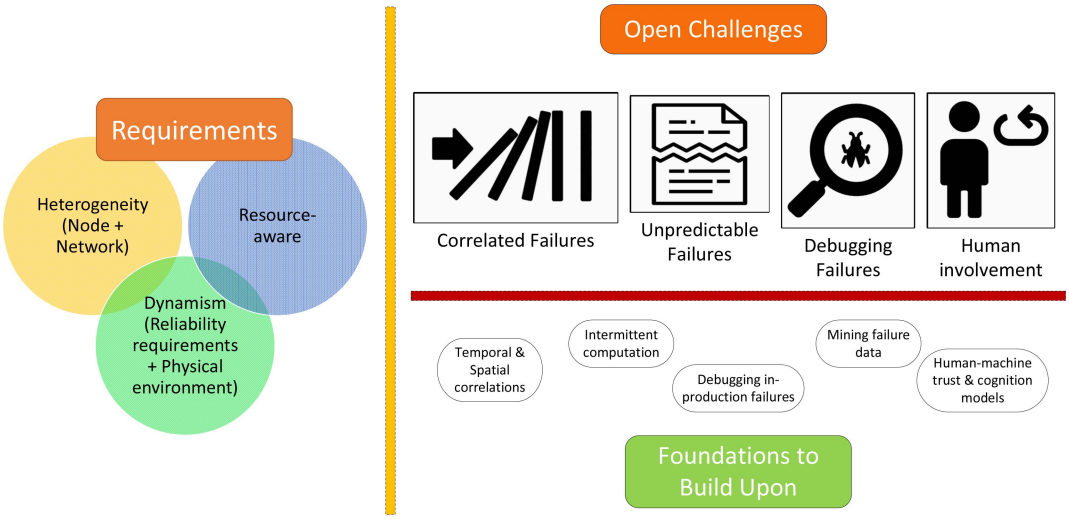

Fig. 2. Overview of the novel reliability requirements and challenges in IoT systems and foundations from the current work that we can build upon.

different infrastructure sensor feeds heavily rely on the soldier input. The main challenge there is to associate the correct infrastructure sensors by comparing the live stream from the infrastructure sensors and the live stream from a head-mounted camera of the soldier. Such a reliable association requires a combination of DNN-based pipelines and inputs from the soldiers and is currently an open area of research.

D. Path Forward

We have to solve the above research challenges through coordinated optimization of the compute and communication that is spread out among a diverse set of resources. This will be helped by the open architecture for IoT that standardizes sensing and actuation and distributed computation. In all our solutions, we have to design for humans as first-order entities interacting with the rest of the system elements.

III. Reliability Challenges

With IoT systems being deployed in critical application areas, including in those where human safety is at stake, reliability is an important and hitherto rather neglected aspect of IoT systems. We discuss the unique requirements and challenges plus the foundations from the current work that we can build upon. These are schematically shown in Fig. 2.

A. Unique Challenges

As IoT systems have become more than playthings and are deployed in applications with moderate to high criticality requirements, they require reliable architecture, operation, and application development. The reliability must address errors in the hardware, the software, interactions with the physical environment, and interactions with the human users. One development is that the systems generate large volumes of data, often at high rates, which put new pressure on the reliability mechanism. The data can be of mixed criticality (i.e., some of it is critical and if not properly handled, lead to user-visible failures, while the rest of it is not) and therefore heterogeneous reliability processing is called for. As introduced earlier, heterogeneity is the first-order feature of our target systems. This heterogeneity also applies to the reliability area, both in design and operation. For example, some devices have the software developed through rigorous software development practices and in programming languages that are safe by design, while some others may have agile software development in unsafe programming languages. Furthermore, due to the runtime instantiation, different devices have different capacity for tolerating errors—some component may be capable of masking errors, while others propagate the errors. Finally, the real-time aspect of the operation implies that reliability measures cannot perturb the timing too much. While there is a mature design and development of reliability for hard real-time systems, our target systems pose new challenges because they are developed much faster (e.g., with little to no formal validation) and they operate in more diverse and uncertain environments. Related to the issue of reliability is the notion of predictable behavior from the system, despite the presence of multiple unpredictable factors—in the IoT platform, in interactions among the platforms, and in interactions between the system and the human users. This is important since the IoT system often has human-in-the-loop or human-on-the-loop (the former means human *has* to be involved in the chain of decision making while the latter makes that optional). Humans have varying degrees of aversion to uncertainty and this underlines the need for this aspect of system operation for IoT systems.

B. Requirements

It is necessary for the reliability protocols to be diverse, in keeping with the heterogeneity of the runtimes where they will execute and heterogeneity of the applications that they are meant to protect. The reliability protocols should be adaptive, to the current state of resources on the device (e.g., a resource-intensive but critical task may start up on the device), the current reliability requirement (e.g., the current data stream being gathered, processed, and communicated to the back end

may be highly critical for some downstream application), and the current state of the physical environment (e.g., a physically hazardous environment may cause correlated failures of multiple devices in spatial proximity).

C. Research Challenges

There are four broad themes in the salient research challenges that face the reliability of IoT systems.

Handling Correlated Failures: This involves dealing with failures that are correlated in space and time. Spatial correlation occurs due to the fact that multiple devices may face similar physical or cyber environments, such as wireless congestion or high-temperature fluctuation. Temporal correlation occurs due to some physical phenomenon spreading with time and affecting devices serially, such as high moisture content causing device failures or the coordinated movement of a large mass of people causing an excessive number of concurrent events.

Handling Unpredictable Failures: A large fraction of failures are unpredictable in any system. This effect is magnified in IoT systems due to several factors. First, the energy resources get drained in an unpredictable manner, say due to environmental conditions for rechargeable solar battery, or unanticipated load leading to high communication activity. Second, an IoT system does not have much headroom when it is deployed, i.e., there is not much safety factor that is built into their deployment. So even mildly aberrant conditions, such as small spikes in load, can cause the system to go into a tailspin leading to failures. Third, there does not exist as good modeling of the failure modes of these systems, as for server-class systems.

Debugging Failures: It is important to enable automated debugging of failures in IoT systems, with the stress on automation due to the fact that the systems are made of a large number of heterogeneous devices, which would stress human cognition for debugging. Automated debugging is challenging because not all execution data can be logged at the devices and not all the logged data can be communicated to a back end for debugging. Furthermore, distributed debugging is often needed, bringing together traces from multiple devices.

Human Considerations: This reliability challenge arises due to the human-in-the-loop (or on-the-loop) in many of these IoT systems. This means different things in different applications and even different deployments for the same application. For example, some human users may be highly reluctant to endure false alarm rates, while some human users may be loathed to look at alarms on small-form factor displays on devices. A typical human-centric form of unreliability arises when human users are distracted while interacting with the systems. The issue of maintainability is inextricably related to this theme, whereby it is important that these systems can be maintained (upgraded, reflashed, reconfigured, etc.) with little to no human intervention, and hardly any expert intervention.

D. Foundations to Build Upon

For each of the above themes, there is a sparse to moderate amount of work that is ongoing. We survey some of the most promising works in each.

First, on the theme of *handling correlated failures*, researchers have developed a rich set of methods to detect faulty sensors and architecture to improve the reliability of IoT systems. The work in [19] uses the insight that with correlated failures, elementary detectors will flag many events almost coincident in time. The authors show that a single-level ML classifier underperforms for many realistic system-level faults while having a two-stage detection (clustering events at the first stage) improves the detection and false-positive rates considerably. To handle space-correlated failures, Bychkovskiy *et al.* [21] presented a two-phase post-deployment calibration technique for large-scale sensors. The key idea is to use the temporal correlation of signals in the co-located sensors and maximize the consistency among the groups of sensor nodes. Balzano and Nowak [12] proposed whether blind calibration approach for sensor networks from weakly correlated sensor readings. On the other hand, focusing on network connectivity, Neumayer and Modiano [90] developed tools to model and analyze geographically correlated network failures. As for temporally correlated failures, Sharma *et al.* [106] proposed time-series analysis-based methods to detect faulty sensors. Jeffery *et al.* [64] presented a framework, called extensible sensor stream processing (ESP), to clean both time and space-correlated sensor data in the pervasive applications. Apart from space and time-correlated failure, Szewczyk *et al.* [114] found that failure of temperature sensors is highly correlated with the failure of the humidity sensors in their lessons from a sensor network expedition. Researchers from the data mining community also provide valuable analytic models for such co-related sensor data. Dong *et al.* [41] considered the dependence between data sources in truth discovery where the conflicting information may come from a large number of sources. Although lots of models have been proposed to clean sensor data, calibrate sensor reading, and detect sensor faults, we have not seen much work that uses the recent ML approaches for failure detection with correlations.

Second, on the theme of *handling unpredictable failures*, a line of solutions have been applied to energy-harvesting IoT devices where failure can happen unpredictably due to energy drain. Some work in this space [79], [122] inserts checkpoints in the code to save state that the application can recover from. Some advanced work [80] does the checkpointing based on available energy. Lightweight approaches are presented in some recent studies. Karimi and Kim [66] presented a new energy scheduling scheme to execute periodic real-time tasks on the intermittently powered embedded devices. Maeng and Lucia [81] also presented the adaptive low-overhead scheduling for intermittent execution. However, the open questions center around how to handle a *larger set* of unpredictable failures in a manner that respects the currently available resources (available storage, energy, etc.).

Third, on the theme of *debugging failures*, most compelling works rely on collecting runtime information and deducing anomalous behavior automatically by mining patterns in the information. Unfortunately, there is a lack of workable solutions for debugging in-production failures. One promising direction is record and replay, whereby execution traces are

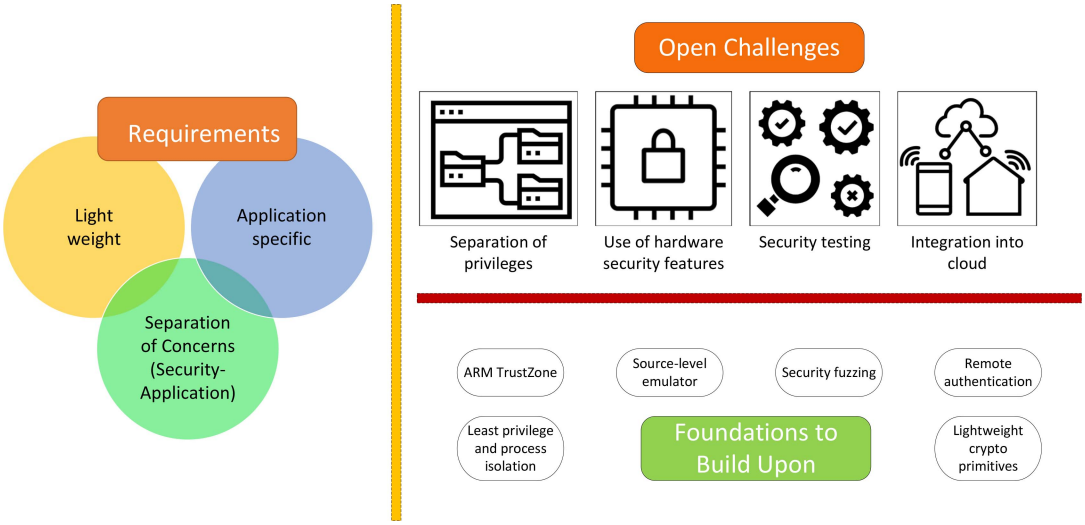

Fig. 3. Overview of the novel security requirements and challenges in IoT systems and foundations from the current work that we can build upon.

recorded on the devices when the system is operational and the traces are somehow brought back to a backend for replaying and debugging. Within the realm of record and replay, our prior work Tardis [116] was the first software-only record and replay system for embedded devices. However, it is only applicable to a single node and does not consider execution on the commonly used microcontrollers (e.g., those which run multithreaded OSes and applications). Our work Aveksha [115] uses extra hardware to record traces from the JTAG port without interfering with the node's execution, but cannot record complete control flows. This and other approaches, such as Minerva [110] and FlashBox [30] that use hardware modifications cannot be deployed to COTS IoT systems. Some software-only efforts, such as TinyTracer [112] and Prius [113], selectively record some events (only control flow for TinyTracer) and therefore cannot enable replay-based debugging. The open questions center around how to provide high fidelity system-level replay, i.e., replay that is able to reproduce both control flow at an instruction level and the state of memory at any point in time for any software module executing on the node. On the broader theme of uncovering patterns in the traces, there needs to be learning algorithms that can learn such patterns from observations in the field. Such solutions will take the place of current, fragile rule-based approaches.

Fourth, on the theme of *human considerations in reliability*, researchers have developed approaches to create more reliable systems and identify failures when an error occurs. As human operators get involved in the control loop of IoT sensor networks, Gross *et al.* [50] used a tandem human–machine cognition approach to mitigate and avoid cognitive overload situations where false alarms and ambiguity may overwhelm humans. Humans can also affect the connection between smart things. Thus, Guo *et al.* [54] used opportunistic IoT models to enable information forwarding and dissemination within the opportunistic IoT communities that are formed based on the movement of humans. In the context of social IoT applications and services, Truong *et al.* [120] developed a trust service platform with a trust model incorporating both reputation properties and knowledge-based property so that multiple entities can trust each other. On the other hand, to identify potential causes for human failures, Cranor [37] proposed a framework for reasoning about the human in the loop in the secure system. However, the open questions include how to use a unified model to study the human factors in the reliability of IoT systems, considering the large variety of humans, perhaps with ML models.

IV. Security and Privacy Challenges

With the large volume of data generated by sensors and a large number of heterogeneous IoT devices, some embedded in our private secure physical spaces, security and privacy pose new challenges. We discuss each of these individually below.

For context, we should mention that there are some excellent prior works that survey the space of IoT security [6], [46]. However, they take either a subset of the scope of our work [6] or take a different viewpoint [46]. We now give two examples of the first case and one of the second. The work by Alrawi *et al.* [6] creates a useful systematization of home-based IoT devices, so that one can reason in a unified way about their vulnerabilities, attacks, and mitigations. As another example of a smaller scope, consider the work by Celik *et al.* [23], in which they survey program analysis techniques that may be used to improve the security of the commodity IoT application. Now, as an example of the second class, the work in [46] takes as a broad view of the IoT (as we do), touching on consumer-grade, industrial-control systems, and autonomous vehicles. It is focused on comparing and contrasting to the security protocols in conventional computing systems.

A. Security

The unique requirements for security in IoT systems along with the open challenges and foundations to build upon are shown schematically in Fig. 3.

1) Unique Challenges: The proliferation of increasingly connected devices has led to new levels of connectivity and automation in IoT systems. The connectivity has great potential to improve our lives, however, it exposes such systems to network-based attacks on an unprecedented scale. Attacks against IoT devices have already unleashed massive denial-of-service attacks [71], given hackers access to streaming video feeds deep inside the periphery of a corporate IT network [83], taken control of autonomous vehicles [82], and facilitated robbing hotel rooms [84]. Currently, these devices are deployed with no security mitigations against a wide variety of attacks that are commonly expected in server-class systems. For example, data execution prevention (DEP) is a fundamental and widely adopted security primitive in server-class systems, whereby all writeable memory pages are marked as nonexecutable—this is often also referred to as the W⊕X defense [35]. But this relies on special hardware in the CPU [AMD "NX" bit (no-execute), Intel "XD" bit (executed disable)], which is often not present in IoT CPUs. As another example, consider address space layout randomization (ASLR) whereby the memory address layout is randomized from one instance of an application to another instance of the same application [16]. The goal is that ASLR prevents attackers from using the same exploit code effectively against all instantiations of the program containing the same flaw. However, this relies on a certain degree of randomness such that a brute-force attack will take a long time to succeed and such randomness relies on a large memory space [105], which is often not available on our target systems. When security defenses *are* present in IoT systems, mitigations are often implemented in an *ad hoc* manner, relying on the developer to make good security decisions. Therefore, such defenses are easily bypassable, e.g., by writing a single flag value to disable all memory protections [32]. We posit that *as IoT devices become ubiquitous, security must become a first-class principle*.

2) Requirements: Security in our target systems must be able to fit inside the available hardware and software and must not perturb the timing properties significantly, neither increasing significantly the mean execution time or even the variance in it. It must provide a clear separation of concerns between the application development and the security development so that the application developer is not called upon to make subtle security design decisions. This is challenging particularly due to the fact that security configuration here is often *application specific*. For example, an IO register on one system may unlock a lock while on a different system, it may control an LED used for debugging. Clearly, the former is a security-sensitive operation while the latter is not. To balance the two factors, such application-specific requirements should be supported in a manner that does *not* require the developer to make intrusive changes within her application code. Finally, and perhaps most importantly, the security techniques and their instantiations must be easily portable across different systems. Such portability should apply, say, within the same vendor's family of products, e.g., within ARM M-class microcontrollers, despite the presence of a different and heterogeneous set of peripherals from one system to the next.

3) Research Challenges: There are four broad themes in the research challenges that face the security of IoT systems.

1) *Separation of Privileges:* IoT devices no longer focus on a dedicated task but increasingly run multiple independent or loosely related tasks. For example, a single SoC often implements both Bluetooth and WiFi, where neither Bluetooth nor WiFi needs to access the code and data of the other. However, without isolation, a single bug compromises the entire SoC and possibly the entire system (one demonstration was taking over Android smartphones through compromising Broadcom's Wi-Fi SoC [9]). It is important to bring in the notion of least privileges or process isolation to the IoT systems. The first notion refers to the need to grant each software component the minimum privilege needed to complete its functionality, while the second refers to the need to protect the control and the data flow from an unprivileged component affecting a privileged component. The research efforts in this theme need to achieve these while respecting the requirements laid out above. This is a challenge because the overwhelming majority of the existing IoT software is written with the assumption that any software module can access any other software module or hardware block, i.e., there is no notion of separation of privileges. It is complex to first identify the different functional software modules (software in this domain is often deeply tangled) and then it is difficult to figure out what is the right set of privileges to assign to each module. A paramount concern is not to break the existing functionality and thus avoid significant porting costs.
2) *Effective Use of Hardware Security Features:* While high-end trusted hardware solutions such as Intel SGX are typically considered not feasible for large-scale IoT deployments, there are widely used hardware-based trusted execution environments through features such as ARM TrustZone. At a more universal level, most microcontrollers come equipped with a peripheral called the memory protection unit (MPU) that can enforce read, write, and execute permissions on regions of the physical memory. TrustZone is also being pushed down into lower end devices, such as the ARM Cortex-M microcontroller series. The challenge is to use such hardware features efficiently and securely. From an efficiency standpoint, consider that the number of MPU registers is limited, e.g., the latest generation, ARM Cortex v8-M processors, have 13 MPU registers. This means that the security protection granularity has to be appropriately defined at any point in the execution to fit within these many registers. For the TrustZone-based solutions, typically, applications have to be rewritten using the particular API, which imposes a burden, an insurmountable one at times, for adoption. For security consideration, it is important for the solution to be such that it cannot be bypassed by an out-of-band mechanism that simply

disables the use of the security hardware. For example, for MPU protection, it can be disabled simply by writing a 0 to the lowest bit of the `MPU_CTRL` register, which is at a fixed (and therefore, known) memory address. For ARM TrustZone, security challenges arise due to the desire to share the device among multiple applications. It is important to guarantee that the isolation among the applications is preserved even when each makes use of the TrustZone.

3) *Security Testing:* Simply verifying the security guarantees of these IoT systems is often a challenge in the face of black-box software packages. Thus, standard mechanisms for verifying security properties such as symbolic verification cannot be brought to bear on IoT software. Today, developers create and test IoT firmware almost entirely on physical testbeds, typically consisting of development versions of the target devices. However, modern software engineering practices that benefit from a scale, such as test-driven development, continuous integration, or fuzzing, are challenging or impractical due to this hardware dependency [89]. In addition, embedded hardware provides limited introspection capabilities, including extremely limited numbers of breakpoints and watchpoints, significantly restricting the ability to perform dynamic analysis on firmware. Manufacturing best practices dictate stripping out or disabling debugging ports (e.g., JTAG), meaning that many off-the-shelf devices remain entirely opaque. Even if the firmware can be obtained through other means, dynamic analysis remains challenging due to the complex environmental dependencies of the code, such as dependency on the specific version of a garden variety peripheral such as an Ethernet card.
4) *Secure Integration of IoT into Cloud Services:* As there is an increasing drive to integrate IoT devices into cloud services, it is essential from a security standpoint to be able to validate the security properties of the devices, at startup as well as periodically, say before doing any critical operation involving these devices. For this, there are three classes of techniques that need to be developed. First, is *remote authentication* whereby any IoT device being brought online is properly authenticated. This should stay away from using sources of information that are low entropy (or equivalently easily guessed), such as the MAC address (MAC addresses of devices are often allocated based on the vendor and the high-order bits are publicly known). The second class of techniques is remote attestation (RA), which involves verification of the current internal state (i.e., RAM and/or flash) of an untrusted remote hardware platform (an IoT device in our context) by a trusted entity (say, a service running on the cloud on behalf of an end user). RA will allow for devices to be compromised, but a remote verification can uncover the presence of malware or other effects of such compromise. This has to be done in a way that balances the resource usage on the device and the security guarantees (either formal or empirical) that the scheme can provide. The third class and broader techniques relate to the use of *crypto primitives* on these resource-constrained platforms. It is important that the crypto primitives fit within the resource budget of the device, chiefly, memory and energy, but provide rigorously quantified security guarantees. Only then can higher level security protocols that integrate these devices with the cloud be built up. Too often in the past have there been cases of insecure design or implementation of crypto primitives for IoT-class of devices, e.g., WEP for wireless transmissions (insecure design) [22] and car keyless entry (insecure use of crypto keys) [48]. This is a particularly pressing research challenge in this domain because of the ease of eavesdropping on communication, due to the omnidirectional wireless communication channel, and the difficulty of upgrading software (including crypto software) once devices are deployed in the field.

4) Foundations to Build Upon: On the theme of *separation of privileges*, FreeRTOS-MPU provides privilege separation between user tasks and kernel task [103]. However, there is a significant barrier to usability in that the separation has to be carefully and manually programmed in by the application developer. Some other approaches [31], [32], [68] use static and dynamic analysis to enforce separation of privileges between different compartments of IoT software allowing a system owner to enforce the principle of least privileges, which is a bedrock of security. Such approaches break the single application into smaller compartments and enforce data integrity and control-flow integrity between compartments. Specific open questions are how far can the separation be done automatically, what is the relative role of static and dynamic techniques, what is the interplay between performance overhead and security in any compartmentalization, and how does a given design overlay on the available hardware features of the device. It is probably unarguable that we have far to go for the compartmentalization to reach the level of sophistication we have on server-class software and systems and work on all three fronts—programming frameworks, tools for using such frameworks, and runtime environments—will help us get there.

On the theme of *effective use of hardware root of trust*, techniques such as EPOXY [32] and ACES [31] make it impossible for the hardware root of trust to be configured (including bypassed) from any but a small amount of privileged code. For ARM TrustZone compatibility, some solutions present a sophisticated runtime environment that shields the applications from the TrustZone API thus ensuring that legacy applications can be supported [53]. For the secure sharing of the TrustZone, some solutions have been developed that provide secure virtualization and isolation among multiple applications [59]. The broad open question relates to how much application modification is tolerable—if the modification can be templated, then that process can be automated. A second question relates to the efficiency loss due to the intervening layer that tries to support legacy applications. Also, since the runtime has not been scrutinized to the extent that the TrustZone TEE has been, are there security bugs lurking there?

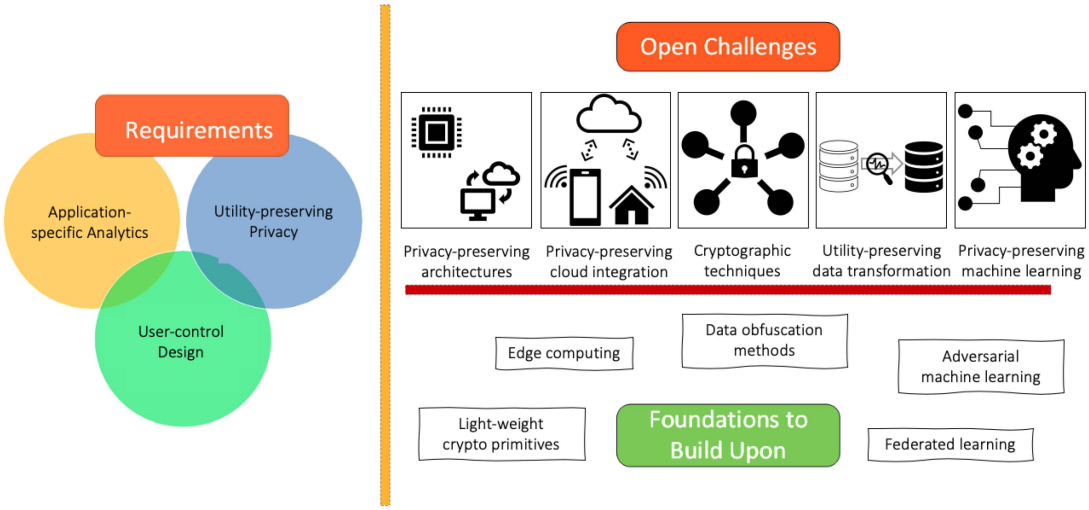

Fig. 4. Overview of the novel privacy requirements and challenges in IoT systems and foundations from the current work that we can build upon.

On the theme of *security testing*, there are several promising directions that attempt to address one or a few of the above-mentioned challenges. The promising line of work here is emulation using the source code were available and binary blobs for other parts [26], [33], [137]. The source code is executed either on the actual hardware or a source-level emulator such as QEMU and the binary blobs are analyzed through mature binary analysis tools, such as IDA, Ghidra, or LibMatch and then rehosted on a standard emulator (thus alleviating the pain point that the actual esoteric version of a peripheral may not be available during testing). Then, all standard software testing techniques can be brought to test the execution on the emulator. These works differ in the layer of the binary at which they do the analysis (high-level libraries versus lower level), the fidelity of the analysis (do they give up when they encounter a binary blob without symbol table or can they perform approximate analysis), and the dependence on hardware-in-the-loop (what kind of hardware do they need to execute). The broad open questions that we need to tackle are how much manual effort is needed in testing IoT software—is a manual understanding of the black-box binary blobs needed or can that be replaced with simple input–output behaviors, do fuzzing and symbolic execution engines need to be equipped with domain-specific constraints. Another broad question is does the fact that testing perturbs the timing of the application change the kinds of bugs that it exposes.

On the theme of *secure integration of IoT devices into cloud-based systems*, Chen *et al.* [28] brought to light questionable security practices with ten IoT vendors for remote binding of IoT devices to cloud services, in the designs of authentication and authorization, including inappropriate use of device IDs, weak device authentication, and weak cloud-side access control. It brings forth a fundamental problem with building authentication from hard-coded device attributes such as device ID. Such attributes may be leaked through device ownership transfer, including device reuse, reselling, stealing, and so forth. One possible approach is to "refresh" these sources of information through remote reprogramming, either periodically or based on critical events (such as a change in location). Such reprogramming can erase the old state or increase the entropy of the variable being relied on for authentication. However, reprogramming has to be done keeping in mind the network constraints of latency and bandwidth and several solutions exist in that space [96]. Several authentication and authorization platforms for IoT have been built [45], [119], which differ in the usability, the granularity of the control, and the kinds of devices they can be run on. Several schemes for RA have been built [44], [61], [93], which differ in what kinds of devices they are targeted at (very low-end TI MSP430 class or higher end ARM R-class), are they hardware based (such as using TrustZone), software based (i.e., based on timing properties), or hybrid, and how formally they have been modeled and verified. The broad open question is how best to combine secure protocols for bringing devices online and remotely managing them including detecting compromise or verifying their integrity. This has to be done while ensuring that any crypto primitive being relied on has enough entropy to be able to withstand cryptanalysis attacks for the required duration of time (the time duration itself may be very application dependent).

B. Privacy

Privacy is another important topic since IoT devices are embedded in our physical spaces including privacy-sensitive locations. The unique challenges for privacy in this domain, along with the open challenges and the foundations to build upon are schematically shown in Fig. 4.

1) Unique Challenges: As IoT devices become more pervasive, they have begun to collect data about our environment, our homes, our health, and many other aspects of our lives. This data may contain sensitive or private information that needs to be safeguarded from the user's perspective. As an example, consider smart voice assistants that listen for spoken commands or smart cameras that continuously view our home environment. Safeguarding the privacy of IoT data raises new challenges that go beyond traditional data privacy challenges.

The issue has become important due to the proliferation of consumer IoT products that range from smart outlets, smart

door locks, thermostats, cameras, fitness bands, voice assistants, object trackers, and many more. Unfortunately, many of these products are designed to provide convenience to users (e.g., remote operation) but often pay little attention to user privacy.

As has been noted earlier, the current generation of consumer IoT products uses a cloud-based architecture where data generated by the device are sent to the cloud for processing [27]. The cloud service can then run any analytics on this data to derive insights for the user. In effect, the user no longer has full control over this data and it is possible for the cloud provider or third parties to mine this data for private information.

2) Requirements: Privacy researchers have proposed data obfuscation as a possible approach to ensure the privacy of user data in the cloud [85], [129]. Data obfuscation involves transforming the data by adding noise to it prior to transmitting it to the cloud. While obfuscation methods can ensure better privacy, they are a blunt instrument. The transformed data reveals nothing after obfuscation and is no longer useful for performing cloud analytics. That is, obfuscation removes all information embedded in the data, both private and nonprivate. Hence, privacy-preserving techniques for IoT data need to carefully consider what type of private and nonprivate information is present in the data and determine how to mask private information without hampering the ability to perform useful analytics on the data. Furthermore, allowing users greater control over their privacy is a key design requirement.

3) Research Challenges:

a) Privacy-preserving architectures: While the current IoT devices rely on a cloud-based architecture, researchers have begun to study new architectures that have better privacy properties [51], [52]. For instance, "cloudless" architectures that process all IoT data on the device itself or an edge device located on customer premises are emerging. These new architectures are becoming feasible due to the rapid hardware advantages that have resulted in specialized chips (e.g., Apple's neural engine [3] and Intel's Movidius VPU [1]) that allow sophisticated computation to be performed on low-end hardware. For instance, such chips have allowed some security cameras to perform face recognition on-device and without sending video data to the cloud. A key advantage of such architectures is that the data are retained by the user and stays on user premises where it is processed locally. Thus, third parties do not have access to the data and can no longer mine it for sensitive information.

b) Privacy-preserving integration into cloud services: The previous section described challenges in the secure integration of IoT into cloud services. Security and privacy are related but distinct challenges. Even with secure cloud integration, IoT services do not necessarily provide privacy. This is because cloud analytics on secure IoT data can still leak private user information.

Consequently, privacy-preserving techniques are needed in addition to security techniques for cloud-based IoT services. Some works have attempted to develop novel cloud-based architectures and integration techniques that preserve IoT privacy [47], [63]. The main challenge is to design techniques that thwart side-channel attacks. Side-channel attacks essentially mine or infer orthogonal information from the original purpose for which the data were collected. For instance, electricity usage data recorded by electric meters are known to reveal occupancy information based on periods of higher usage, a type of side-channel attack [70]. The problem is especially challenging since it is *a priori* unclear what kind of other information may be hidden within the data gathered for a specific purpose. Conventional techniques such as differential privacy often do not apply since we are concerned with masking private information from a single data stream.

c) Cryptographic techniques for IoT privacy: Analogous to crypto-based security methods, researchers have developed cryptographic primitives for ensuring privacy when transmitting data to cloud services [38]. One such approach leverages zero-knowledge cryptography (ZKC), where the IoT device sends a cryptographic proof to the cloud server rather than the raw data [88]. Such a proof, known as zero-knowledge proof, allows the server to verify that the result was derived from valid data. However, each ZK proof is based on a specific query and general methods that allow for a broad set of analytic queries to be performed with ZKC in the IoT context remains an open challenge.

d) Utility-preserving data transformation: An alternate approach to ZKC is to employ intelligent data transformation on the data prior to transmitting it to the cloud. Unlike obfuscation-based methods that leave no useful information in the data, utility-preserving privacy transformation seeks to mask any type of private information in the data while leaving other nonprivate information intact. Doing so allows conventional analytics in the cloud to be performed like before, but prevents side-channel attacks that try to extract private information from the data. Such utility-preserving privacy transformation is more challenging than data obfuscation since they need to selectively mask only information considered to be private. The existing methods such as differential privacy are useful on multiuser data [43]. However, since analyzing single-user streams is more prominent in the IoT context, novel utility-preserving techniques are required.

Utility-preserving privacy also raises an interesting tradeoff between utility and privacy. The more information that is masked in the data, the less useful it becomes (obfuscation can be considered to be an extreme case that masks all information). Thus, it is important to consider user preferences when designing such systems and let the user decide what information to suppress and what to reveal to a cloud service. For instance, Zheng *et al.* [143] used semistructured interviews with smart home owners to understand their reasons for purchasing IoT devices, perceptions of smart home privacy risks, and actions taken to protect their device and data privacy. This is as much of an HCI challenge as a technical one since explaining privacy implications to users to choose the appropriate preferences is a nontrivial issue.

e) Privacy-preserving ML: There has been a growing use of ML in the IoT domain. From using a long short-term memory (LSTM) model on smart watch data to detect medical conditions, such as diabetes and high blood pressure [11] to detecting Distributed Denial-of-Service (DDoS) attacks in IoT

traffic using random forests [42], researchers are employing advanced ML methods to improve the usability of IoT devices.

However, the popularity of ML with IoT data raises a fresh set of privacy concerns. Adversaries with IoT data can employ ML methods to infer private information that may be implicit in the data. For instance, prior work has demonstrated that it is feasible to disaggregate energy usage from smart energy meters into individual components, popularly known as nonintrusive load monitoring (NILM), using a factorial hidden Markov model (FHMM) [14], [69]. This type of disaggregation directly reveals the daily activity patterns of users. Privacy attacks are also possible on trained ML models. Two such popular attacks are membership inference, where an adversary attempts to infer whether a user was part of the training data, and model inversion, where an adversary attempts to infer sensitive features in the training data via model output. Recent work has shown that membership inference attacks can be conducted on aggregate location data from smart services [98]. Model inversion attacks pose a higher concern from an IoT perspective where distributed ML models can be reverse engineered to gain sensitive local information.

These issues raise an overriding question: how can we use ML to improve the usability of IoT devices while preserving privacy? Designing ML models that are privacy-preserving and are robust to model-based attacks in an IoT context is thus a pressing open area of research.

4) Foundations to Build Upon: Edge computing will be a major foundation for future privacy-preserving architectures and privacy-preserving cloud integration. As computation and storage on distributed edge clusters advances, edge-based architectures will gain prominence and cloud integration will involve more aggregate and/or processed data. Lightweight crypto primitives will serve as a foundation for resource-bounded, privacy-preserving cryptography methods for IoT.

Though blunt, data obfuscation methods provide a reasonable foundation toward utility-preserving privacy for cloud-based architectures. Building on data obfuscation methods to mask only private features is an open research direction. Some work has demonstrated success in masking private data in the energy domain [24], [25], [70]. Recent work has employed Metropolis-Hasting statistical sampling to transform data from smart energy meters to suppress private user information while retaining non-private information [18]. Such methods can be used toward developing more general utility-preserving techniques. Recent work in federated learning, an ML technique that allows decentralized training on edge devices, has demonstrated possibilities for privacy-preserving ML in the IoT domain [62], [65], [73], [78], [104], [135], [141]. Federated learning-based methods will gain prominence in analyzing IoT data as edge computing and distributed learning advances. Another promising approach is the combination of traditional differential privacy methods with ML to protect aggregate user data [74], [126], [127].

C. Path Forward

Reliability and security have quickly become important for IoT systems as they are being used in applications where human health or safety or large financial gains/losses are at stake. Privacy is a unique challenge here as IoT systems are embedded in our physical spaces and interact with us through multiple modalities (speech, vision, touch, etc.). We want to make the research and development of these as first-order concerns. Their design and development must be enabled in a way that the application developer does not also have to become an expert in them, but rather clean usable interfaces allow understanding and configuration of these building blocks. In terms of these building blocks, they have to be designed and developed in a way that they are usable, fit within the resource constraints of the devices and the network, and meet the application-specific goals to different and configurable levels of fidelity.

V. Discussion and Take-Aways

IoT systems serve applications that fall under three broad categories: 1) applications that enhance our physical spaces (homes and offices); 2) applications that empower the devices we use (e.g., appliances and vehicles); and 3) applications that enhance the efficiency of production and delivery systems (e.g., food production, manufacturing, and energy delivery). These applications are demanding IoT systems that are *simultaneously* high performing, secure, and reliable. IoT systems are distributed, putting more emphasis on end-to-end systems challenges, scalability, and network support, as opposed to, say, control systems or embedded systems. Also, IoT systems, by virtue of distribution and scale, are often multipurpose and heterogeneous and involve humans in the loop. Each of these leads to unique challenges in systems and networking.

There are a host of promising solutions that are being developed and with a growing pace of innovations. These include edge-cloud offload and on-device computation, model reduction and efficient model inferencing, 5G and networking software innovations (such as network function virtualization), and human-in-the-loop design. These need to be targeted to the unique requirements focused to work within the constraints, and provide the appropriate interfaces for the human users.

While developing these solutions, reliability, security, and privacy have to be built into these solutions as first-class primitives. Here also, there are a host of domain-specific challenges. In the area of reliability, promising solutions arise from techniques for dealing with temporally and spatially correlated failures, intermittent computation, debugging in-production failures, discerning failure patterns by mining failure data, and models for human–machine interaction and human cognition. These have to be focused to handle correlated and unpredictable failures (including those due to the close coupling of the devices with the physical environment), debugging large-scale production failures, and reliability bottlenecks due to humans in the loop. In the area of security, several existing solution approaches can be leveraged and many of these are under active development now. These include efficient use of hardware root of trust, enforcement of least privilege and process isolation, remote authentication, lightweight crypto primitives, and security fuzzing for uncovering vulnerabilities. These need to be further developed to reach the goals—allow IoT devices to be securely integrated into the

cloud infrastructure, allow separation of concerns in software development between security and application functionality, and enforce security containment boundaries. These have to be achieved even though hardware features that we rely on in the server world, such as memory management units, are often not present here.

Privacy is a particularly important concern since IoT devices are embedded in our physical spaces including in privacy-sensitive locations. Data obfuscation, on-premises processing of IoT data, and privacy-preserving ML are important building blocks for reaching the goals of privacy. These raise an overriding question: How can we use ML to improve the usability of IoT devices while preserving privacy? Designing ML models that are privacy-preserving and are robust to model-based attacks in an IoT context is thus a pressing open area of research.

In summary, this is an exciting time to be working in IoT, in its systems, network, reliability, security, or privacy areas. We see a slew of energizing technical challenges and a mounting set of compelling solutions, with many more to come in the near future.

ACKNOWLEDGMENT

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

REFERENCES

[1] *Intel Vision Accelerator Design With Intel Movidius Vision Processing Unit (VPU)*. Accessed: May 1, 2020. [Online]. Available: https://software.intel.com/en-us/iot/hardware/vision-accelerator-movidius-vpu#specifications

[2] *TensorFlow Serving*. Accessed: May 1, 2020. [Online]. Available: https://github.com/tensorflow/serving

[3] (Sep. 2017). *Apple's 'Neural Engine' Infuses the Iphone With Ai Smarts*. [Online]. Available: https://www.wired.com/story/apples-neural-engine-infuses-the-iphone-with-ai-smarts/

[4] F. Ahmad, H. Qiu, R. Eells, F. Bai, and R. Govindan, "CarMap-fast 3D feature map updates for automobiles," in *Proc. 17th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, 2020, pp. 1063–1081.

[5] NGMN Alliance. (2016). *Description of Network Slicing Concept*. [Online]. Available: https://www.ngmn.org/publications/description-of-network-slicing-concept.html

[6] O. Alrawi, C. Lever, M. Antonakakis, and F. Monrose, "SoK: Security evaluation of home-based IoT deployments," in *Proc. IEEE Symp. Security Privacy (SP)*, 2019, pp. 1362–1380.

[7] M. R. Amer, P. Lei, and S. Todorovic, "Hirf: Hierarchical random field for collective activity recognition in videos," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 572–585.

[8] B. B. Amor, J. Su, and A. Srivastava, "Action recognition using rate-invariant analysis of skeletal shape trajectories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 1–13, Jan. 2016.

[9] N. Artenstein, "BroaDPWN: Remotely compromising android and IoS via a bug in Broadcom's Wi-Fi chipsets," in *Proc. Black Hat USA*, 2017, pp. 1–28.

[10] T. Bagautdinov, A. Alahi, F. Fleuret, P. Fua, and S. Savarese, "Social scene understanding: End-to-end multi-person action localization and collective activity recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4315–4324.

[11] B. Ballinger *et al.*, "DeepHeart: Semi-supervised sequence learning for cardiovascular risk prediction," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 2079–2086.

[12] L. Balzano and R. Nowak, "Blind calibration of sensor networks," in *Proc. 6th Int. Conf. Inf. Process. Sensor Netw.*, 2007, pp. 79–88.

[13] E. Bastug, M. Bennis, M. Médard, and M. Debbah, "Toward interconnected virtual reality: Opportunities, challenges, and enablers," *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 110–117, Jun. 2017.

[14] N. Batra *et al.*, "NILMTK: An open source toolkit for non-intrusive load monitoring," in *Proc. 5th Int. Conf. Future Energy Syst.*, 2014, pp. 265–276.

[15] B. Bellekens, V. Spruyt, R. Berkvens, and M. Weyn, "A survey of rigid 3D pointcloud registration algorithms," in *Proc. 4th Int. Conf. Ambient Comput. Appl. Services Technol. (AMBIENT)*, Aug. 2014, pp. 8–13.

[16] S. Bhatkar, D. C. DuVarney, and R. Sekar, "Address obfuscation: An efficient approach to combat a broad range of memory error exploits," in *Proc. USENIX Security Symp.*, 2003, pp. 291–301.

[17] N. Bizanis and F. A. Kuipers, "SDN and virtualization solutions for the Internet of Things: A survey," *IEEE Access*, vol. 4, pp. 5591–5606, 2016.

[18] P. Bovornkeeratiroj, S. Iyengar, S. Lee, D. Irwin, and P. Shenoy, "RepEL: A utility-preserving privacy system for IoT-based energy meters," in *Proc. IEEE/ACM Int. Conf. Internet Things Design Implement. (IoTDI)*, 2020, pp. 79–91.

[19] G. Bronevetsky, I. Laguna, B. R. de Supinski, and S. Bagchi, "Automatic fault characterization via abnormality-enhanced classification," in *Proc. IEEE/IFIP Int. Conf. Depend. Syst. Netw. (DSN)*, 2012, pp. 1–12.

[20] A. Brunetti, D. Buongiorno, G. F. Trotta, and V. Bevilacqua, "Computer vision and deep learning techniques for pedestrian detection and tracking: A survey," *Neurocomputing*, vol. 300, pp. 17–33, Jul. 2018.

[21] V. Bychkovskiy, S. Megerian, D. Estrin, and M. Potkonjak, "A collaborative approach to in-place sensor calibration," in *Information Processing in Sensor Networks*. Heidelberg, Germany: Springer, 2003, pp. 301–316.

[22] N. Cam-Winget, R. Housley, D. Wagner, and J. Walker, "Security flaws in 802.11 data link protocols," *Commun. ACM*, vol. 46, no. 5, pp. 35–39, 2003.

[23] Z. B. Celik, E. Fernandes, E. Pauley, G. Tan, and P. McDaniel, "Program analysis of commodity IoT applications for security and privacy: Challenges and opportunities," *ACM Comput. Surveys*, vol. 52, no. 4, pp. 1–30, 2019.

[24] D. Chen, S. Barker, A. Subbaswamy, D. Irwin, and P. Shenoy, "Non-intrusive occupancy monitoring using smart meters," in *Proc. BuildSys*, 2013, pp. 1–8.

[25] D. Chen, D. Irwin, P. Shenoy, and J. Albrecht, "Combined heat and privacy: Preventing occupancy detection from smart meters," in *Proc. PerCom*, Mar. 2014, pp. 208–215.

[26] D. D. Chen, M. Egele, M. Woo, and D. Brumley, "Towards automated dynamic analysis for Linux-based embedded firmware," in *Proc. Netw. Distrib. Syst. Security (NDSS) Symp.*, vol. 16, 2016, pp. 1–16.

[27] D. Chen, P. Bovornkeeratiroj, D. Irwin, and P. Shenoy, "Private memoirs of IoT devices: Safeguarding user privacy in the IoT era," in *Proc. Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2018, pp. 1327–1336.

[28] J. Chen *et al.*, "Your IoTs are (not) mine: On the remote binding between IoT devices and users," in *Proc. 49th Annu. IEEE/IFIP Int. Conf. Depend. Syst. Netw. (DSN)*, 2019, pp. 222–233.

[29] W. Chen, L. Cao, X. Chen, and K. Huang, "An equalized global graph model-based approach for multicamera object tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 11, pp. 2367–2381, Jul. 2017.

[30] S. Choudhuri and T. Givargis, "FlashBox: A system for logging non-deterministic events in deployed embedded systems," in *Proc. ACM Symp. Appl. Comput.*, 2009, pp. 1676–1682.

[31] A. A. Clements, N. S. Almakhdhub, S. Bagchi, and M. Payer, "ACES: Automatic compartments for embedded systems," in *Proc. 27th USENIX Security Symp. (USENIX Sec)*, 2018, pp. 65–82.

[32] A. A. Clements *et al.*, "Protecting bare-metal embedded systems with privilege overlays," in *Proc. IEEE Symp. Security Privacy (SP)*, 2017, pp. 289–303.

[33] A. A Clements *et al.*, "HALucinator: Firmware re-hosting through abstraction layer emulation," in *Proc. 29th USENIX Security Symp. (USENIX Sec)*, 2020, pp. 1–18.

[34] M. Cornick, J. Koechling, B. Stanley, and B. Zhang, "Localizing ground penetrating radar: A step toward robust autonomous ground vehicle localization," *J. Field Robot.*, vol. 33, no. 1, pp. 82–102, 2016.

[35] *Data Execution Prevention*, Microsoft Corporat., Redmond, MA, USA, 2018.

[36] D. Crankshaw, X. Wang, G. Zhou, M. J. Franklin, J. E. Gonzalez, and I. Stoica, "Clipper: A low-latency online prediction serving system," in *Proc. 14th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, 2017, pp. 613–627.

[37] L. F. Cranor, "A framework for reasoning about the human in the loop," in *Proc. UPSEC*, 2008, pp. 1–15.

[38] G. Danezis and B. Livshits, "Towards ensuring client-side computational integrity," in *Proc. ACM Workshop Cloud Comput. Security*, 2011, pp. 125–130.

[39] S. Dey, J. Mondal, and A. Mukherjee, "Offloaded execution of deep learning inference at edge: Challenges and insights," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, 2019, pp. 855–861.

[40] J. Dong, D. Zhuang, Y. Huang, and J. Fu, "Advances in multi-sensor data fusion: Algorithms and applications," *Sensors*, vol. 9, no. 10, pp. 7771–7784, 2009.

[41] X. L. Dong, L. Berti-Equille, and D. Srivastava, "Integrating conflicting data: The role of source dependence," *Proc. VLDB Endown.*, vol. 2, no. 1, pp. 550–561, 2009.

[42] R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning DDoS detection for consumer Internet of Things devices," in *Proc. IEEE Security Privacy Workshops (SPW)*, 2018, pp. 29–35.

[43] C. Dwork, "Differential privacy: A survey of results," in *Proc. Int. Conf. Theory Appl. Models Comput.*, 2008, pp. 1–19.

[44] K. Eldefrawy, G. Tsudik, A. Francillon, and D. Perito, "SMART: Secure and minimal architecture for (establishing dynamic) root of trust," in *Proc. Netw. Distrib. Syst. Security (NDSS) Symp.*, vol. 12, 2012, pp. 1–15.

[45] E. Fernandes, J. Jung, and A. Prakash, "Security analysis of emerging smart home applications," in *Proc. IEEE Symp. Security Privacy (SP)*, 2016, pp. 636–654.

[46] E. Fernandes, A. Rahmati, K. Eykholt, and A. Prakash, "Internet of Things security research: A rehash of old ideas or new intellectual challenges?" *IEEE Security Privacy*, vol. 15, no. 4, pp. 79–84, Aug. 2017.

[47] S. Funke, J. Daubert, A. Wiesmaier, P. Kikiras, and M. Muehlhaeuser, "End-2-end privacy architecture for IoT," in *Proc. IEEE Conf. Commun. Netw. Security (CNS)*, 2015, pp. 705–706.

[48] F. D. Garcia, D. Oswald, T. Kasper, and P. Pavlidès, "Lock it and still lose it on the (in) security of automotive remote keyless entry systems," in *Proc. 25th USENIX Security Symp. (USENIX Sec)*, 2016, pp. 203–220.

[49] D. Gowsikhaa, Manjunath, and S. Abirami, "Suspicious human activity detection from surveillance videos," *Int. J. Internet Distrib. Comput. Syst.*, vol. 2, no. 2, pp. 141–148, 2012.

[50] K. C. Gross, K. Baclawski, E. S. Chan, D. Gawlick, A. Ghoneimy, and Z. H. Liu, "A supervisory control loop with prognostics for human-in-the-loop decision support and control applications," in *Proc. IEEE Conf. Cogn. Comput. Aspects Situation Manag. (CogSIMA)*, 2017, pp. 1–7.

[51] M. Großmann and C. Ioannidis, "Cloudless computing—A vision to become reality," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, 2020, pp. 372–377.

[52] M. Großmann, C. Ioannidis, and D. T. Le, "Applicability of serverless computing in fog computing environments for IoT scenarios," in *Proc. IEEE/ACM Int. Conf. Utility Cloud Comput. Companion*, 2019, pp. 29–34.

[53] L. Guan *et al.*, "TrustShadow: Secure execution of unmodified applications with ARM TrustZone," in *Proc. 15th Annu. Int. Conf. Mobile Syst. Appl. Services (Mobisys)*, 2017, pp. 488–501.

[54] B. Guo, D. Zhang, Z. Wang, Z. Yu, and X. Zhou, "Opportunistic IoT: Exploring the harmonious interaction between human and the Internet of Things," *J. Netw. Comput. Appl.*, vol. 36, pp. 1531–1539, Nov. 2013.

[55] A. Habib, M. Ghanma, M. Morgan, and R. Al-Ruzouq, "Photogrammetric and Lidar data registration using linear features," *Photogr. Eng. Remote Sens.*, vol. 71, no. 6, pp. 699–707, 2005.

[56] D. L. Hall and J. Llinas, "An introduction to multisensor data fusion," *Proc. IEEE*, vol. 85, no. 1, pp. 6–23, 1997.

[57] Y.-M. Hsiao, J.-F. Lee, J.-S. Chen, and Y.-S. Chu, "H.264 video transmissions over wireless networks: Challenges and solutions," *Comput. Commun.*, vol. 34, no. 14, pp. 1661–1672, 2011.

[58] Y. Hu, S. Rallapalli, B. Ko, and R. Govindan, "Olympian: Scheduling gpu usage in a deep neural network model serving system," in *Proc. 19th Int. Middleware Conf.*, 2018, pp. 53–65.

[59] Z. Hua, J. Gu, Y. Xia, H. Chen, B. Zang, and H. Guan, "VTZ: Virtualizing arm trustzone," in *Proc. 26th USENIX Security Symp. (USENIX Sec)*, 2017, pp. 541–556.

[60] C.-C. Hung *et al.*, "VideoEdge: Processing camera streams using hierarchical clusters," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, 2018, pp. 115–131.

[61] A. Ibrahim, A.-R. Sadeghi, and S. Zeitouni, "SEED: Secure non-interactive attestation for embedded devices," in *Proc. 10th ACM Conf. Security Privacy Wireless Mobile Netw.*, 2017, pp. 64–74.

[62] A. Imteaj and M. H. Amini, "Distributed sensing using smart end-user devices: Pathway to federated learning for autonomous IoT," in *Proc. IEEE Int. Conf. Comput. Sci. Comput. Intell. (CSCI)*, 2019, pp. 1156–1161.

[63] P. P. Jayaraman, X. Yang, A. Yavari, D. Georgakopoulos, and X. Yi, "Privacy preserving Internet of Things: From privacy techniques to a blueprint architecture and efficient implementation," *Future Gener. Comput. Syst.*, vol. 76, pp. 540–549, Nov. 2017.

[64] S. R. Jeffery, G. Alonso, M. J. Franklin, W. Hong, and J. Widom, "Declarative support for sensor data cleaning," in *Proc. Int. Conf. Pervasive Comput.*, 2006, pp. 83–100.

[65] L. Jiang, X. Lou, R. Tan, and J. Zhao, "Differentially private collaborative learning for the IoT edge," in *Proc. EWSN*, 2019, pp. 341–346.

[66] M. Karimi and H. Kim, "Energy scheduling for task execution on intermittently-powered devices," in *Proc. 9th Embedded Oper. Syst. Workshop*, 2019, pp. 1–6.

[67] T. Kato, Y. Ninomiya, and I. Masaki, "An obstacle detection method by fusion of radar and motion stereo," *IEEE Trans. Intell. Transp. Syst.*, vol. 3, no. 3, pp. 182–188, Sep. 2002.

[68] C. H. Kim *et al.*, "Securing real-time microcontroller systems through customized memory view switching," in *Proc. Netw. Distrib. Syst. Security (NDSS) Symp.*, 2018, pp. 1–15.

[69] J. Kolter and M. Johnson, "REDD: A public data set for energy disaggregation research," in *Proc. SustKDD*, 2011, pp. 1–6.

[70] J. Koo, X. Lin, and S. Bagchi, "RL-BLH: Learning-based battery control for cost savings and privacy preservation for smart meters," in *Proc. 47th Annu. IEEE/IFIP Int. Conf. Depend. Syst. Netw. (DSN)*, 2017, pp. 519–530.

[71] B. Krebs. *DDoS on Dyn Imapcts Twitter, Spotify, Reddit*. Accessed: May 1, 2020. [Online]. Available: https://krebsonsecurity.com/2016/10/ddos-on-dyn-impacts-twitter-spotify-reddit/comment-page-2/

[72] Y. Lee *et al.*, "PRETZEL: Opening the black box of machine learning prediction serving systems," in *Proc. 13th USENIX Symp. Oper. Syst. Design Implement. (OSDI)*, 2018, pp. 611–626.

[73] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," 2019. [Online]. Available: arXiv:1908.07873.

[74] J. Liu, C. Zhang, and Y. Fang, "EPIC: A differential privacy framework to defend smart homes against Internet traffic analysis," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1206–1217, Apr. 2018.

[75] S. Liu *et al.*, "A linear feature-based approach for the registration of unmanned aerial vehicle remotely-sensed images and airborne Lidar data," *Remote Sens.*, vol. 8, no. 2, p. 82, 2016.

[76] S. Liu, Y. Lin, Z. Zhou, K. Nan, H. Liu, and J. Du, "On-demand deep model compression for mobile devices: A usage-driven model selection framework," in *Proc. 16th Annu. Int. Conf. Mobile Syst. Appl. Services*, 2018, pp. 389–400.

[77] X. Liu, P. Ghosh, O. Ulutan, B. S. Manjunath, K. Chan, and R. Govindan, "CAESAR: Cross-camera complex activity recognition," in *Proc. 17th Conf. Embedded Netw. Sensor Syst.*, 2019, pp. 232–244.

[78] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Blockchain and federated learning for privacy-preserved data sharing in industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 4177–4186, Jun. 2020.

[79] K. Maeng, A. Colin, and B. Lucia, "ALPACA: Intermittent execution without checkpoints," in *Proc. ACM Program. Lang.*, vol. 1, 2017, pp. 1–30.

[80] K. Maeng and B. Lucia, "Adaptive dynamic checkpointing for safe efficient intermittent computing," in *Proc. 13th USENIX Symp. Oper. Syst. Design Implement. (OSDI)*, 2018, pp. 129–144.

[81] K. Maeng and B. Lucia, "Adaptive low-overhead scheduling for periodic and reactive intermittent execution," in *Proc. 41st ACM SIGPLAN Conf. Program. Lang. Design Implement. (PLDI)*, 2020, pp. 1005–1021.

[82] *Hacked Driverless Cars Could Cause Collisions and Gridlock in Cities*, Forbes Mag., Jersey City, NJ, USA, Mar. 2019.

[83] *An Elaborate Hack Shows How Much Damage IoT Bugs Can Do*, Wired Mag., San Francisco, CA, USA, Apr. 2018.

[84] *Hackers Built a 'Master Key' for Millions of Hotel Rooms*, ZDNet Mag., Seoul, South Korea, Apr. 2018.

[85] S. McLaughlin, P. McDaniel, and W. Aiello, "Protecting consumer privacy from electric load monitoring," in *Proc. CCS*, Oct. 2011, pp. 87–98.

[86] P. Mettes and C. G. M. Snoek, "Spatial-aware object embeddings for zero-shot localization and classification of actions," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4443–4452.

[87] R. Mijumbi, J. Serrat, J.-L. Gorricho, S. Latré, M. Charalambides, and D. Lopez, "Management and orchestration challenges in network functions virtualization," *IEEE Commun. Mag.*, vol. 54, no. 1, pp. 98–105, Jan. 2016.

[88] A. Molina-Markham, P. Shenoy, K. Fu, E. Cecchet, and D. Irwin, "Private memoirs of a smart meter," in *Proc. BuildSys*, 2010, pp. 1–6.

[89] M. Muench, J. Stijohann, F. Kargl, A. Francillon, and D. Balzarotti, "What you corrupt is not what you crash: Challenges in fuzzing embedded devices," in *Proc. Netw. Distrib. Syst. Security (NDSS) Symp.*, 2018, pp. 1–15.

[90] S. Neumayer and E. Modiano, "Network reliability with geographically correlated failures," in *Proc. IEEE INFOCOM*, 2010, pp. 1–9.

[91] K. Nithin and F. Brémond, "Globality–locality-based consistent discriminant feature ensemble for multicamera tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 3, pp. 431–440, Mar. 2017.

[92] D. S. Nunes, P. Zhang, and J. S. Silva, "A survey on human-in-the-loop applications towards an Internet of all," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 2, pp. 944–965, 2nd Quart., 2015.

[93] I. D. O. Nunes, K. Eldefrawy, N. Rattanavipanon, M. Steiner, and G. Tsudik, "VRASED: A verified hardware/software co-design for remote attestation," in *Proc. 28th USENIX Security Symp.*, 2019, pp. 1429–1446.

[94] T. Ojanperä, J. Mäkelä, O. Mämmelä, M. Majanen, and O. Martikainen, "Use cases and communications architecture for 5G-enabled road safety services," in *Proc. IEEE Eur. Conf. Netw. Commun. (EuCNC)*, 20118, pp. 335–340.

[95] C. Pakha, A. Chowdhery, and J. Jiang, "Reinventing video streaming for distributed vision analytics," in *Proc. 10th USENIX Workshop Hot Topics Cloud Comput. (HotCloud)*, 2018, p. 1.

[96] R. K. Panta, S. Bagchi, and S. P. Midkiff, "Efficient incremental code update for sensor networks," *ACM Trans. Sensor Netw.*, vol. 7, no. 4, pp. 1–32, 2011.

[97] H. Parmar and M. Thornburgh, *Adobe's Real Time Messaging Protocol*, Adobe Syst. Inc., San Jose, CA, USA, 2012.

[98] A. Pyrgelis, C. Troncoso, and E. De Cristofaro, "Knock knock, who's there? Membership inference on aggregate location data," in *Proc. Netw. Distrib. Syst. Security (NDSS) Symp.*, 2018, pp. 1–15.

[99] Y. Qi, M. Hunukumbure, M. Nekovee, J. Lorca, and V. Sgardoni, "Quantifying data rate and bandwidth requirements for immersive 5G experience," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC)*, 2016, pp. 455–461.

[100] H. Qiu, F. Ahmad, F. Bai, M. Gruteser, and R. Govindan, "AVR: Augmented vehicular reality," in *Proc. 16th Annu. Int. Conf. Mobile Syst. Appl. Services (Mobisys)*, 2018, pp. 81–95.

[101] K. Ragland and P Tharcis, "A survey on object detection, classification and tracking methods," *Int. J. Eng. Res. Technol*, vol. 3, no. 11, pp. 622–628, 2014.

[102] E. Ristani and C. Tomasi, "Features for multi-target multi-camera tracking and re-identification," 2018. [Online]. Available: arXiv:1803.10859.

[103] Free RTOS. *The FreeRTOS Kernel MPU Support*. Accessed: May 1, 2020. [Online]. Available: https://www.freertos.org/FreeRTOS-MPU-memory-protection-unit.html

[104] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-IID data," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Nov. 1, 2019, doi: 10.1109/TNNLS.2019.2944481.

[105] H. Shacham, M. Page, B. Pfaff, E.-J. Goh, N. Modadugu, and D. Boneh, "On the effectiveness of address-space randomization," in *Proc. 11th ACM Conf. Comput. Commun. Security*, 2004, pp. 298–307.

[106] A.B. Sharma, L. Golubchik, and R. Govindan, "Sensor faults: Detection methods and prevalence in real-world datasets," *ACM Trans. Sensor Netw.*, vol. 6, no. 3, pp. 1–39, 2010.

[107] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Jun. 2016.

[108] Z. Shou *et al.*, "Online detection of action start in untrimmed, streaming videos," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 534–551.

[109] F. Solera, S. Calderara, E. Ristani, C. Tomasi, and R. Cucchiara, "Tracking social groups within and across camera," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 3, pp. 441–453, Mar. 2017.

[110] P. Sommer and B. Kusy, "MINERVA: Distributed tracing and debugging in wireless sensor networks," in *Proc. 11th ACM Conf. Embedded Netw. Sensor Syst. (Sensys)*, 2013, pp. 1–14.

[111] H. Song, W. Choi, and H. Kim, "Robust vision-based relative-localization approach using an RGB-depth camera and Lidar sensor fusion," *IEEE Trans. Ind. Electron.*, vol. 63, no. 6, pp. 3725–3736, Jun. 2016.

[112] V. Sundaram, P. Eugster, and X. Zhang, "Demo abstract: Diagnostic tracing of wireless sensor networks with TinyTracer," in *Proc. 10th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw.*, 2011, pp. 145–146.

[113] V. Sundaram, P. Eugster, and X. Zhang, "PRIUS: Generic hybrid trace compression for wireless sensor networks," in *Proc. 10th ACM Conf. Embedded Netw. Sensor Syst.*, 2012, pp. 183–196.

[114] R. Szewczyk, J. Polastre, A. Mainwaring, and D. Culler, "Lessons from a sensor network expedition," in *Proc. Eur. Workshop Wireless Sensor Netw.*, 2004, pp. 307–322.

[115] M. Tancreti, M. S. Hossain, S. Bagchi, and V. Raghunathan, "AVEKSHA: A hardware–software approach for non-intrusive tracing and profiling of wireless embedded systems," in *Proc. 9th ACM Conf. Embedded Netw. Sensor Syst. (Sensys)*, 2011, pp. 288–301.

[116] M. Tancreti, V. Sundaram, S. Bagchi, and P. Eugster, "TARDIS: Software-only system-level record and replay in wireless sensor networks," in *Proc. 14th Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, 2015, pp. 286–297.

[117] S. K. Tayyaba, M. A. Shah, O. A. Khan, and A. W. Ahmed, "Software defined network (SDN) based Internet of Things (IoT) a road ahead," in *Proc. Int. Conf. Future Netw. Distrib. Syst.*, 2017, pp. 1–8.

[118] Y. T. Tesfaye, E. Zemene, A. Prati, M. Pelillo, and M. Shah, "Multi-target tracking in multiple non-overlapping cameras using constrained dominant sets," 2017. [Online]. Available: arXiv:1706.06196.

[119] Y. Tian *et al.*, "SmartAuth: User-centered authorization for the Internet of Things," in *Proc. 26th USENIX Security Symp. (USENIX Security)*, 2017, pp. 361–378.

[120] N. B. Truong, T.-W. Um, and G. M. Lee, "A reputation and knowledge based trust service platform for trustworthy social Internet of Things," in *Proc. Innov. Clouds Internet Netw. (ICIN)*, 2016, pp. 1–8.

[121] O. Ulutan, S. Rallapalli, C. Torres, M. Srivatsa, and B. S. Manjunath, "Actor conditioned attention maps for video action detection," in *Proc. IEEE IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, 2020, pp. 1–10.

[122] J. Van Der Woude and M. Hicks, "Intermittent computation without hardware support or programmer intervention," in *Proc. 12th USENIX Symp. Oper. Syst. Design Implement. (OSDI)*, 2016, pp. 17–32.

[123] J. Wang *et al.*, "Bandwidth-efficient live video analytics for drones via edge computing," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, 2018, pp. 159–173.

[124] Q. Wang, Y. Zhu, and L. Cheng, "Reprogramming wireless sensor networks: challenges and approaches," *IEEE Netw.*, vol. 20, no. 3, pp. 48–55, May/Jun. 2006.

[125] T. Wark, P. Corke, J. Karlsson, P. Sikka, and P. Valencia, "Real-time image streaming over a low-bandwidth wireless camera network," in *Proc. 3rd Int. Conf. Intell. Sensors Sensor Netw. Inf.*, 2007, pp. 113–118.

[126] J. Xiong *et al.*, "Enhancing privacy and availability for data clustering in intelligent electrical service of IoT. *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1530–1540, Apr. 2019.

[127] C. Xu, J. Ren, D. Zhang, and Y. Zhang, "Distilling at the edge: A local differential privacy obfuscation framework for IoT data analytics," *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 20–25, Aug. 2018.

[128] Y. Xu, X. Liu, L. Qin, and S.-C. Zhu, "Cross-view people tracking by scene-centered spatio-temporal parsing," in *Proc. AAAI*, 2017, pp. 4299–4305.

[129] W. Yang, N. Li, Y. Qi, W. Qardaji, S. McLaughlin, and P. McDaniel, "Minimizing private data disclosures in the smart grid," in *Proc. ACM Conf. Comput. Commun. Security*, 2012, pp. 415–427.

[130] S. Yao *et al.*, "Eugene: Towards deep intelligence as a service," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2019, pp. 1630–1640.

[131] S. Yao, T. Wang, J. Li, and T. Abdelzaher, "Stardust: A deep learning serving system in IoT: Demo abstract," in *Proc. 17th Conf. Embedded Netw. Sensor Syst.*, 2019, pp. 402–403.

[132] S. Yao *et al.*, "FastDeepIoT: Towards understanding and optimizing neural network execution time on mobile and embedded devices," in *Proc. 16th ACM Conf. Embedded Netw. Sensor Syst.*, 2018, pp. 278–291.

[133] S. Yao, Y. Zhao, A. Zhang, L. Su, and T. Abdelzaher, "DeepIoT: Compressing deep neural network structures for sensing systems with a compressor-critic framework," in *Proc. 15th ACM Conf. Embedded Netw. Sensor Syst.*, 2017, pp. 1–14.

[134] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Comput. Netw.*, vol. 52, no. 12, pp. 2292–2330, 2008.

[135] T. Yu *et al.*, "Learning context-aware policies from multiple smart homes via federated multi-task learning," in *Proc. IEEE/ACM Int. Conf. Internet Things Design Implement. (IoTDI)*, 2020, pp. 104–115.
[136] R. Yue, H. Xu, J. Wu, R. Sun, and C. Yuan, "Data registration with ground points for roadside Lidar sensors," *Remote Sens.*, vol. 11, no. 11, p. 1354, 2019.
[137] J. Zaddach, L. Bruno, A. Francillon, and D. Balzarotti, "AVATAR: A framework to support dynamic security analysis of embedded systems' firmwares," in *Proc. Netw. Distrib. Syst. Security (NDSS) Symp.*, vol. 14, 2014, pp. 1–16.
[138] H. Zhang *et al.*, "Live video analytics at scale with approximation and delay-tolerance," in *Proc. 14th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, 2017, pp. 377–392.
[139] J. Zhao, "Exploring the fundamentals of using infrastructure-based LiDAR sensors to develop connected intersections," Civil Eng., Texas Tech Univ., Lubbock, TX, USA, Ph.D. dissertation, 2019.
[140] J. Zhao, H. Xu, H. Liu, J. Wu, Y. Zheng, and D. Wu, "Detection and tracking of pedestrians and vehicles using roadside Lidar sensors," *Transp. Res. C Emerg. Technol.*, vol. 100, pp. 68–87, Mar. 2019.
[141] Y. Zhao, J. Zhao, L. Jiang, R. Tan, and D. Niyato, "Mobile edge computing, blockchain and reputation-based crowdsourcing IoT federated learning: A secure, decentralized and privacy-preserving system," 2019. [Online]. Available: arXiv:1906.10893.
[142] Y. Zhao *et al.*, "Temporal action detection with structured segment networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2914–2923.
[143] S. Zheng, N. Apthorpe, M. Chetty, and N. Feamster, "User perceptions of smart home IoT privacy," in *Proc. ACM Human Comput. Interact.*, vol. 2, 2018, pp. 1–20.

Saurabh Bagchi received the bachelor's degree in computer science and engineering from the Indian Institute of Technology, Kharagpur, India, and the M.S. and Ph.D. degrees in computer science from the University of Illinois at Urbana–Champaign, Champaign, IL, USA.

He is a Professor with the School of Electrical and Computer Engineering and the Department of Computer Science, Purdue University, West Lafayette, IN, USA, where he has been the Founding Director of a university-wide resilience center, called CRISP. He is a co-lead on the WHIN-SMART Center with Purdue University for IoT and data analytics. His research interest is in dependable computing and distributed systems. He is proudest of the 21 Ph.D. students and 50 Masters thesis students who have graduated from his research group and who are in various stages of building wonderful careers in industry or academia. In his group, he and his students have way too much fun building and breaking real systems.

Prof. Bagchi was elected to the IEEE Computer Society Board of Governors for the 2017–2019 term and re-elected in 2019.

Tarek F. Abdelzaher received the Ph.D. degree in computer science from the University of Michigan at Ann Arbor, Ann Arbor, MI, USA, in 1999.

He is currently a Professor and a Willett Faculty Scholar with the Department of Computer Science, University of Illinois at Urbana–Champaign, Urbana, IL, USA. He has authored/coauthored more than 300 refereed publications in real-time computing, cyber–physical systems, IoT, and social applications. His research interests lie broadly in understanding and influencing performance and temporal properties of networked embedded, social, and software systems in the face of increasing complexity, distribution, and degree of interaction with an external physical environment.

Prof. Abdelzaher is a recipient of the IEEE Outstanding Technical Achievement and Leadership Award in Real-Time Systems in 2012, the Xerox Award for Faculty Research in 2011, as well as several best paper awards. He serves as an Editor-in-Chief of the *Journal of Real-Time Systems*, and as an Associate Editor of multiple journals, including the IEEE TRANSACTIONS ON MOBILE COMPUTING, the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, the *ACM Transaction on Sensor Networks*, the *ACM Transactions on Internet of Things*, the *ACM Transactions on Internet Technology*, and *Ad Hoc Networks*. He chaired (as a Program or General Chair) several conferences in his area, including RTAS, RTSS, IPSN, Sensys, DCoSS, ICDCS, and ICAC. He is a Fellow of ACM.

Ramesh Govindan received the B.Tech. degree from the Indian Institute of Technology at Madras, Chennai, India, and the M.S. and Ph.D. degrees from the University of California at Berkeley, Berkeley, CA, USA.

He is the Northrop Grumman Chair of engineering and a Professor of computer science and electrical engineering with the University of Southern California, Los Angeles, CA, USA. His research interests include routing and measurements in large Internet, networked sensing systems, and mobile computing systems.

Prashant Shenoy received the B.Tech. degree in computer science and engineering from the Indian Institute of Technology Bombay, Mumbai, India, in 1993, and the M.S. and Ph.D. degrees in computer sciences from the University of Texas at Austin, Austin, TX, USA, in 1994 and 1998, respectively.

He is currently a Professor with the College of Information and Computer Sciences, University of Massachusetts at Amherst, Amherst, MA, USA. His current research focuses on distributed systems and networking.

Prof. Shenoy is a Fellow of ACM and AAAS.

Akanksha Atrey received the B.S. degree in mathematics and computer science from the State University of New York at Albany, Albany, NY, USA, in 2016. She is currently pursuing the Ph.D. degree with the College of Information and Computer Sciences, University of Massachusetts at Amherst, Amherst, MA, USA.

She is a member of the Laboratory for Advanced System Software, where she is advised by Prof. P. Shenoy. She is particularly interested in characterizing, evaluating, and building privacy-preserving systems and models. Prior to joining UMass, she was a Software Engineer with IBM, Armonk, NY, USA, where she worked on the IBM z/OS mainframe. Her research interests lie at the intersection of machine learning, distributed systems, and privacy.

Pradipta Ghosh received the Ph.D. degree in electrical engineering from the Ming Hsieh Department of Electrical Engineering, University of Southern California, Los Angeles, CA, USA, in 2018.

He is a Postdoctoral Research Scholar with the Department of Computer Science, University of Southern California, where he is also an MHI Institute Scholar and a USC Provost Fellow. His research interests are in wireless sensor networks, wireless robotics networks, vehicular networks, and cloud computing.

Dr. Ghosh is also a member of ACM research communities.

Ran Xu is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA.

His research interests include computer vision, machine learning, and system optimizations on mobile and embedding systems with approximation techniques.