Collaborative Inference for AI-Empowered IoT Devices

Nir Shlezinger and Ivan V. Bajić

ABSTRACT

Artificial intelligence (AI) technologies, and particularly deep learning systems, are traditionally the domain of large-scale cloud servers, which have access to high computational and energy resources. Nonetheless, in Internet-of-Things (IoT) networks, the interface with the real-world is carried out using edge devices; these devices communicate with each other, and are each limited in hardware. The conventional approach to provide AI processing to data collected by edge devices involves sending samples to the cloud, at the cost of latency, communication, connectivity, and privacy concerns. Consequently, recent years have witnessed a growing interest in enabling AI-aided inference on edge devices by leveraging their communication capabilities to establish collaborative inference. This article reviews candidate strategies for facilitating the transition of AI to IoT devices via collaboration. We identify the need to operate in different mobility and connectivity constraints as a motivating factor to consider multiple schemes, which can be roughly divided into methods where inference is done remotely, i.e., on the cloud, and those that infer on the edge. We identify the key characteristics of each strategy in terms of inference accuracy, communication latency, privacy, and connectivity requirements, providing a systematic comparison between existing approaches. We conclude by presenting future research challenges and opportunities arising from the concept of collaborative inference.

INTRODUCTION

The philosophical idea of artificial intelligence (AI), dating back multiple decades, is nowadays evolving into reality. Deep learning is demonstrating unprecedented success in a broad range of applications. The successful combination of deep learning with the expected proliferation of smart edge devices, and particularly Internet of Things (IoT) devices, is expected to bring AI to many aspects of our lives, ranging from intelligent wearable sensors to self-driving vehicle and smart manufacturing systems. We are thus witnessing a growing research and industrial interest in bringing AI to the domain of edge devices.

Deep learning, which is the key enabler technology for AI, relies on highly-parameterized models, trained using massive volumes of data. Consequently, deep learning is traditionally the domain of large-scale cloud servers, which have the computational resources and the ability to aggregate the data required to store, train, and apply deep neural networks (DNNs). However, edge devices do not share these computational and storage resources, and tend to be notably more limited in compute and power compared with powerful centralized servers [1]. This makes the transition of AI from the domain of powerful servers to distributed and computationally-limited edge devices a challenging task.

The challenges associated with using DNNs on IoT edge devices can be divided according to the main machine learning tasks: training and inference. The challenges related to the former stem from the fact that edge devices have access only to a fraction of the data that can be aggregated by centralized AI systems, yet sharing this data with a centralized server may give rise to privacy concerns and limit the ability to train personalized models. Schemes for enabling learning on the edge are widely studied, with arguably the most common approach being federated learning [2], where multiple devices collaborate during training in a centrally orchestrated fashion.

Even when one has access to a trained AI model, having it utilized by IoT devices gives rise to many different challenges. Most notably, DNNs are often comprised of millions and even billions of parameters. Hence, hardware-limited IoT devices may be unable to merely apply such trained DNNs due to storage, energy, and computational considerations. A common practice is to have the edge device communicate its measurements to a powerful centralized server for inference. Yet, this strategy induces delay, gives rise to privacy concerns, imposes a notable burden on the server, and limits AI-aided inference to settings where reliable communications with the server is attainable. These challenges can become limiting factors for emerging applications such as, e.g., AI-empowered wearable devices. Consequently, recent years have witnessed a growing interest in leveraging collaboration for facilitating edge inference, with the proposal of various different techniques [3–5], motivating the unified overview of these methods.

In this article we systematically review candidate approaches for enabling AI-aided inference on IoT devices. While the successful transition of AI to IoT devices is likely to rely on developments in both hardware as well as signal processing and algorithmic techniques, our focus is on the latter. We commence with discussing the diverse use-cases for AI inference, reviewing their associated characteristics and positioning them in the context of the conventional paradigms of edge vs cloud computing [6]. We particularly divide these settings into static scenarios, as arise in, e.g., smart manufacturing systems, and dynamic mobile scenarios, relevant to, e.g., wearable IoT devices and vehicular systems. This division reveals the broad range of requirements and the need for diverse collaborative strategies for AI-aided edge inference.

Next, we categorize existing and emerging approaches for AI-aided inference into two main strategies:

1. Inference on a central system, either a dedicated edge server or a remote cloud server, with collaboration potentially used to relieve latency, congestion, and privacy concerns;
2. On-device inference, either in a purely decentralized or in a centrally orchestrated manner, where collaboration can improve performance with different levels of compensation for latency, privacy, and flexibility. For each strategy, we first briefly describe conventional non-collaborative inference, highlighting its drawbacks and motivating collaborative methods. These collaborative inference approaches, which are the focus of the article, are then discussed more extensively. We provide comparisons between these approaches, capitalizing on their individual pros and cons in light of the identified families of expected use-cases. We conclude by discussing

Nir Shlezinger is with Ben-Gurion University of the Negev, Beer-Sheva, Israel.
Ivan V. Bajić is with Simon Fraser University, Canada.

Digital Object Identifier: 10.1109/IOTM.001.2200152
AI-EMPOWERED IOT INFERENCE

The term IoT encompasses a broad range of devices that can sense and communicate. Consequently, inference tasks associated with IoT devices span a wide scope of diverse use-cases. Diversity in IoT inference is reflected in multiple aspects, including hardware capabilities, energy resources, and latency tolerance. To pinpoint some of the core challenges and motivate the need for different forms of collaboration, we next focus on diversity that arises from different levels of mobility. To highlight this, we consider two extreme settings, as illustrated in Fig. 1:

- **Static settings**: here, the IoT devices are static, and communicate with a cloud server, possibly with the aid of an intermediate access point or edge server, via relatively reliable links, of fixed capacity. Such settings include, e.g., surveillance cameras in smart cities, biomedical monitors in hospitals, or industrial sensor networks.

- **Dynamic settings**: in many applications, IoT devices are mobile, and are required to operate and infer while traversing through different environments with varying levels of connectivity to the cloud. For instance, wearable devices, ranging from portable bio-medical sensors to and augmented reality cameras in smart cities, biomedical monitors in hospitals, or industrial sensor networks.

The above settings represent the extremes of a spectrum of use-cases varying in mobility, with various cases lying in between. In the static setting, one can reliably process the data on the cloud server; the main challenges here stem from latency and privacy requirements, as well as the storage and computation capacity of the server, which is much larger compared with edge devices, but also has its limits. These can be tackled via collaborative inference, by properly dividing some of the computation between the edge device and the cloud (and/or intermediate servers in the communication network). In the dynamic setting, connectivity to the cloud is not guaranteed, and inference should be carried out on the edge, where one must cope with the limited ability of edge devices to apply highly parameterized DNNs. Collaboration in such cases is feasible between different edge devices, communicating in an adaptive device-to-device fashion, with the aim of improving inference accuracy.

The division into static and dynamic settings highlights the need for collaboration, for both remote and on-device inference. These can be related to the main paradigms of cloud computing, fog computing, and edge computing [6]. The latter infers on the device that acquires the observations, while the former two consider remote inference: in cloud computing, inference is carried out solely on the centralized cloud server, while in fog computing some of the processing is offloaded to intermediate nodes in the communication network, e.g., access points and edge servers. Since we focus on signal processing and algorithmic collaboration approaches, rather than on the design and exploitation of the hierarchical structure of communication networks, we henceforth simplify our categorization into:

1. **Cloud-centric inference**: the inference result is produced remotely regardless of whether it is partitioned between the cloud server and fog nodes
2. **Edge inference**: that is done on the edge device itself

We next discuss collaborative inference based on this categorization, starting with cloud-centric inference, and proceeding with edge inference.

CLOUD-CENTRIC INFERENCE

Based on the division proposed in the previous section, we commence our presentation of collaborative inference approaches with the family of cloud-centric inference schemes. By cloud-centric inference we mean a scenario where the result of the inference procedure is produced in the cloud. This result may then be sent back to the edge, if necessary. Two versions of cloud-centric inference are illustrated in Fig. 2.

**Non-collaborative cloud inference** (Fig. 2a), the entire DNN is deployed in the cloud. The edge device merely captures the data, for example an image, and sends it to the cloud. An advantage of this simple scheme is that it is relatively easy to deploy, especially if the DNN is equipped with advanced hardware-based image/video codecs and communication capabilities, which are more suitable for the cloud. However, there are, however, a number of downsides to this scheme. First, sending the entire dataset to the cloud for inference uses more bits than necessary. It is shown in [9] that, at a given inference accuracy, features from any layer of an arbitrary non-generative DNN are more compressible than its input. Thus, reducing the number of uploaded bits could also reduce the overall latency. Finally, uploading data to the cloud raises privacy concerns. All these issues may be alleviated via collaborative edge-cloud inference, discussed next.

**Collaborative edge-cloud inference**

In this scenario, shown in Fig. 2b, a DNN is partitioned into a front-end (initial few layers) deployed on an edge device, and a back-end (remaining part) residing in the cloud. The edge device computes features that are uploaded to the cloud. As discussed above, this strategy is more bit-efficient, which reduces the communication latency of uploading to the cloud, compared to the non-collaborative approach. The overall inference latency is a combination of this communication latency, and the computation times on the edge device and on the cloud.

Typical plots of the overall inference latency as a function of the upload bitrate for edge inference (where the entire DNN is on the edge device, non-collaborative cloud inference, and collaborative edge-cloud inference are shown in Fig. 3, with actual measurements provided in [10] and references therein. In interpreting these plots, we note that the hardware available in the cloud is faster than that on the edge device, and that uploading intermediate DNN features is more bit-efficient than uploading its input. When the bitrate is very small, communication latency is higher than the computation time even on the edge device, and in this regime, edge inference is the fastest. Since edge inference does not require upload to the cloud, its latency is shown as a constant. At the other extreme, when
the bitrate is large, communication latency becomes negligible compared with the computation time. In this case, non-collaborative cloud inference is the fastest, since the cloud operates faster hardware. In between these extremes, there is an interval of available bitrates where edge-cloud collaborative inference is the fastest.

Finally, the fact that collaborative edge-cloud inference avoids uploading original data may alleviate privacy issues. However, features uploaded to the cloud can still carry some private information, which may be revealed through various attacks. For example, model inversion attacks [11] try to recover the data from the features, essentially inverting the DNN front-end on the edge device. If successful, private information is revealed. There are currently limited defences against such attacks, one being the information-theoretic privacy fan [12], where non-private inference-relevant features are lightly compressed while privacy-revealing features are more heavily compressed. We note that this is a relatively unexplored area, where much future work is required to develop effective solutions, as discussed later.

**EDGE INFERENCE**

The previous section focused on scenarios where the inference result is produced at the cloud. In this section we consider settings where inference is to be carried out on the edge, e.g., by an IoT device. Applying DNNs on IoT devices allows to infer on the same device where the data is collected, rather than having the samples sent to a centralized cloud server. Such DNN-aided edge devices can operate at various connectivity conditions with reduced latency, as well as alleviate privacy issues and facilitate the personalization of AI systems [5].

The core challenge with applying trained DNNs on IoT devices stems from their limited computational resources. Existing strategies for on-device AI-based inference can be divided into non-collaborative and collaborative ones. The former aims at designing compact DNNs that are applicable on hardware-limited devices, as illustrated in Fig. 4a. Collaborative approaches, which are the focus of this article, leverage the ability of IoT devices to communicate with neighbouring peers to enable high performance inference. This is achieved via partitioning of complex computations over multiple devices (Fig 4b), or by forming an ad hoc ensemble (Fig. 4c). In the following we elaborate on these methods.
Non-collaborative Edge Inference

To enable applying trained DNNs on edge devices without having multiple devices collaborate, one typically has to utilize compact DNNs. As deep learning usually employs highly-parameterized models, a key challenge here is to design models that are compact without compromising too much on performance.

Various techniques have been proposed for compacting DNNs. The conventional framework deals with scenarios where one has access to a high-performance DNN, and aims at making it more compact. Among the leading approaches for compacting DNNs are knowledge distillation, where a compact DNN is trained to imitate the highly-parameterized pre-trained model; network pruning, which intentionally throws away neurons and/or nullifies weights in the trained model; and network quantization, where weights are coarsely discretized, possibly to a single bit. Methods for realizing these approaches for compacting DNN is still an area of active research, see survey [13].

The above approaches compact a given DNN. Alternatively, one can design an AI model to be light-weight in the first place, rather than starting from a pre-trained large DNN. For instance, the fact that a DNN is to be pruned or quantized can be accounted for in its training. Furthermore, one can prefer architectures, such as convolutional networks with small kernels and shortcuts [13], that are inherently more compact compared with conventional ones. An alternative strategy designs DNN-aided systems that utilize compact networks by incorporating domain knowledge and augmenting classic inference algorithms with trainable models, see survey [14].

Non-collaborative edge-inference strategies focus on a single edge user. As such, they do not exploit the fact that while each device is limited in its hardware, multiple users can confidently collaborate, even in the absence of reliable connectivity to a centralized server. Such collaboration, discussed in the following sections, allows the system to benefit from the joint computational resources of multiple IoT devices.

Computation Partitioning

The ability of edge devices to communicate and collaborate can be harnessed to enable AI-aided inference by partitioning a DNN among multiple devices. Such techniques, coined computation partitioning or offloading [5], are schematically illustrated in Fig. 4b.

The most straight-forward approach divides the DNN by layers (or blocks of layers). In such layer-based collaborative inference, each participating device only applies a subset of the layers of the DNN, and communicates its output features, which are possibly compressed to reduce the overhead, to the specific device that applies the subsequent layers. Layer-based partitioning of a DNN can also be combined with horizontal partitioning, where the computations of each layer are divided among multiple users, i.e., different users apply different neurons of the same layer [3]. The latter is essential when utilizing wide DNNs, in which some layers may be comprised of too many neurons to be applicable on hardware-limited edge devices.

Partitioning a DNN among multiple users allows to jointly form a large network during inference. In the absence of communication errors, it enables AI-aided edge inference without compromising on accuracy compared with cloud-centric inference. Nonetheless, each user cannot infer locally, and must rely on the availability of neighbouring nodes, which have the required DNN partitions. This notably complicates the ability to form an ad hoc collaboration, and typically involves some centralized orchestration. Finally, the repeated communications among the multiple devices, and the potential presence of stragglers due to the heterogeneity of IoT devices, results in possibly increased latency, and requires dedicated optimization of the workload and communication among the devices [3].

Edge Ensembles

The above edge inference approaches either rely solely on local inference (via compact DNNs), or require reliable communica-
The starting point for AI-empowered inference is typically some large DNN, whose accuracy is degraded by compressing it. As such, inferring solely on the cloud, which can host the large uncompressed DNN, is expected to be the most accurate. Techniques that partition the DNN, either via collaborative edge-cloud inference or by computation offloading, may induce some degradation as features being shared often undergo lossy compression. Compacting the DNN is likely to yield the most notable degradation, though its effect can be mitigated by collaboration via edge ensembles.

**Communications**

Non-collaborative edge inference does not entail any communication overhead, as processing is done on-device, and thus offers the least communication latency. Cloud-centric inference may involve notable communication latency due to the need to convey the observed data from the edge to the cloud, though this can be reduced by sharing compressed features via collaborative edge-cloud inference. Among collaborative edge inference schemes, computation partitioning may induce substantial communication latency due to the repeated exchange of features and the need to coordinate the procedure, while edge ensembles entails minimal excessive overhead, as it involves a single round of multi-casting compressed features.

**Privacy**

The data used for inference may contain private information. Thus, sharing it over the communication network, as in cloud inference, does not preserve privacy. In all collaborative schemes, one can enhance privacy by sharing extracted features rather than the data itself, though this requires dedicated crafting of the features. Clearly, inferring locally with a compact DNN is the most privacy-preserving.

**Connectivity**

Cloud-centric inference requires reliable connectivity between the IoT device and the cloud, where non-collaborative cloud inference needs high-throughput links for sharing the observations. Edge inference can typically be robust to limited connectivity and

| Method                  | Collaborate     | Accuracy                        | Communications | Privacy                        | Connectivity                  |
|------------------------|-----------------|---------------------------------|----------------|-------------------------------|------------------------------|
| Cloud-centric inference | Cloud inference| Highest — usage of large DNNs   | High — due to  | None — data shared over      | Requires reliable             |
|                        |                 |                                 | communications observations | communications network      | high-throughput link to server|
| Collaborate            | Edge-cloud      | High — usage of large DNNs, possible distortion | Medium — sharing of compressed features | Partial — shared features that can be crafted to enhance privacy | Requires reliable link server |
|                        | inference       | due to feature compression       |                |                               |                              |
| Edge inference         | Compact networks| Typically degraded due to network compacting | Minimal — no communications | Fully private — no data sharing | Invariant                    |
| Computation partitioning| Multiple edge devices | High — usage of large DNNs, possible distortion | High — due to repeated device-to-device communications | Partial — shared features that can be crafted to enhance privacy | Requires reliable links with specific edge devices |
| Edge ensembles         | Multiple edge devices | Adaptive — degraded in low connectivity, increases when collaboration is feasible | Low — multi-casting of compressed features via device-to-device links | Partial — shared features that can be crafted to enhance privacy | Fully adaptive — operable in different connectivity levels |

**Table 1.** Qualitative comparison between the considered approaches for IoT AI-empowered inference.

The approaches detailed earlier differ in their properties, and are each suitable for different scenarios. Broadly speaking, cloud-centric inference is geared towards scenarios with reliable connectivity, while edge inference is most suitable in mobile settings. To provide a meaningful comparison, we focus on four key figures-of-merit — inference accuracy, communication latency, privacy, and connectivity requirements.
applicable in mobile settings, though computation partitioning still requires reliable communications between a (possibly fixed) set of users that jointly possess all the partitions of the DNN.

**SUMMARY**

The comparison detailed above is summarized in Table 1. Based on this qualitative comparison, one can identify different IoT settings for which each approach is attractive. For instance, collaborative edge-cloud inference is highly suitable for fixed surveillance and monitoring devices, which are often static and have reliable connectivity, yet their data often high dimensional and is likely to be associated with privacy considerations. Computation partitioning is suitable when devices operate in the same environment, such as industrial IoT sensors, where orchestration and advance DNN partitioning is feasible. Edge ensembles are most relevant for highly mobile settings, e.g., vehicular systems, which operate in diverse environments and where collaboration is established ad hoc.

**FUTURE RESEARCH DIRECTIONS**

Collaborative inference bears the potential of paving the way to a smooth transition of AI from the domain of powerful centralized servers, into a multitude of easily accessible and portable devices. Accordingly, it gives rise to numerous research directions that should be further explored in order to realize the potential of these methods. Here, we discuss a few representative topics, providing an initial characterization of some relevant future directions. These selected topics encompass a broad range of scientific disciplines, considering theoretical studies, algorithmic aspects, and system design.

**PRIVACY GUARANTEES**

Collaboration during inference leads to privacy considerations, as it involves sharing of data samples that may contain private information. This gives rise to the algorithmic question of how to enhance privacy in collaborative inference without notably degrading accuracy and latency. A core challenge here follows since measuring privacy, which is essential to design, evaluate, and compare such methods, is not trivial for inference tasks. A widely-accepted concept is differential privacy, which deals with guaranteed obscuring of individual samples in large data sets, and may thus be less suitable for inference based on a single data sample. An alternative notion of privacy is information-theoretic privacy, which represents the statistical dependence between the shared and private features. The latter was considered for license plate privacy in traffic analysis [12] which focused on collaborative edge-cloud settings. Developing privacy enhancements methods that are based on meaningful privacy measures for inference tasks is thus critical for allowing collaborative inference that is free of privacy concerns.

**HYBRID COLLABORATION DESIGN**

The division into cloud-centric and edge inference is motivated by the categorization of static and mobile IoT settings discussed earlier. However, one can also expect situations involving both static and mobile users. For instance, inference carried out by a mobile autonomous vehicle that can ad hoc communicate with both neighbouring vehicles and road-side units that are wired to the network infrastructure. Such scenarios motivate the research question of when to infer by sending data to the cloud, and when to collaborate with neighbouring users. This study can lead to hybrid collaboration strategies that adaptively benefit from collaboration among multiple edge devices, as in edge ensembles, while leveraging possible connectivity with a centralized server, as in collaborative edge-cloud inference.

**OVER-THE-AIR COMPUTATIONS**

Recent years have witnessed a growing interest in over-the-air computation techniques for reducing the communication latency when learning on the edge [2]. These techniques are most relevant when the edge users communicate over a shared wireless channel, such that a desired joint computation can be achieved by non-orthogonal synchronized communications with suitable precoding. This motivates the research question of how to employ over-the-air computations for edge inference. As initially explored in [15], such schemes are likely to be naturally suitable for edge ensembles, with the promise of reducing latency and possibly enhancing privacy (as the channel noise is now added to the shared features and decisions).

**JOINT HARDWARE-ALGORITHMIC DESIGN**

While we focus on algorithmic aspects of AI-empowered edge inference, using DNNs on IoT devices will also involve hardware developments. The fact that DNN-aided edge inference requires novelty in both system hardware and collaboration algorithms gives rise to the question of how to jointly design the hardware along with the collaborative inference algorithm. Algorithmic approaches can guide hardware design, or alternatively, the characteristics of hardware accelerators can reveal unique requirements for efficient collaborative inference. Such joint designs constitute an important research direction as they can alleviate some of the power and latency drawbacks associated with directly applying DNNs on hardware-limited edge devices, by properly exploiting the available computation and communication resources along with how they are utilized.

**DIVERSE EDGE MODELS**

Collaboration via edge ensembles requires having diverse models among the users. For edge ensembles to benefit from collaboration of many devices, they must possess sufficiently diverse models. Achieving diversity is key to such collaboration, motivating the question of how to boost diversity for collaborative inference. While different methods for training and deploying DNNs on the edge can yield diverse models, their ability to facilitate collaboration has not yet been investigated. For instance, when training is done on the edge, diversity typically emerges as the training data differs between devices. Alternatively, diversity can be obtained when compacting a trained DNN by, e.g., having each small model trained with a different initialization, or by compressing a large network into different compressed realizations using stochastic quantization. These ideas require further exploration to understand their effect on collaborative inference and what technique is suitable for which scenario.

**CONCLUSIONS**

This article reviewed approaches for collaborative AI-aided IoT inference. We categorized existing methods into two main strategies — cloud-centric inference and edge inference — and
highlighted their main characteristics. By harnessing collaboration, we showed that one can improve upon inferring solely on the cloud and/or the edge in accuracy, communication latency, privacy, and adaptivity. We discussed research directions that can unveil the potential of collaborative inference in bringing AI to the edge.

REFERENCES
[1] J. Chen and X. Ran, “Deep Learning with Edge Computing: A Review,” Proc. IEEE, vol. 107, no. 8, 2019, pp. 1655–74.
[2] T. Gafni et al., “Federated Learning: A Signal Processing Perspective,” IEEE Signal Proc. Mag., vol. 39, no. 3, 2022, pp. 14–41.
[3] Y. Huang et al., “Toward Decentralized and Collaborative Deep Learning Inference for Intelligent IoT Devices,” IEEE Network, vol. 36, no. 1, 2022, pp. 59–68.
[4] M. Malka et al., “Decentralized Low-Latency Collaborative Inference via Ensembles on the Edge,” arXiv preprint arXiv:2206.03165, 2022.
[5] Y. Mao et al., “A Survey on Mobile Edge Computing: The Communication Perspective,” IEEE Commun. Surveys & Tuts., vol. 19, no. 4, 2017, pp. 2322–58.
[6] R. Mahnud, R. Kotagiri, and R. Buyya, “Fog Computing: A Taxonomy, Survey and Future Directions,” Internet of Everything, Springer, 2018, pp. 103–30.
[7] I. V. Bajić, “Latent Space Inpainting for Loss-Resilient Collaborative Object Detection,” Proc. IEEE Int’l Conf. Communications (ICC), 2021.
[8] M. Jankowski, D. Gündüz, and K. Mikołajczyk, “Wireless Image Retrieval at the edge,” IEEE JSAC, vol. 39, no. 1, 2020, pp. 89–100.
[9] H. Choi and I. V. Bajić, “Scalable Image Coding for Humans and Machines,” IEEE Trans. Image Proc., vol. 31, 2022, pp. 2739–54.
[10] M. Ulhaq and I. V. Bajić, “Shared Mobile-Cloud Inference for Collaborative Intelligence,” arXiv preprint arXiv:2002.00157, 2019, demo at NeurIPS 19.
[11] Z. He, T. Zhang, and R. B. Lee, “Model Inversion Attacks Against Collaborative Inference,” Proc. 35th Annual Computer Security Applications Conf., 2019, p. 148–62.
[12] S. R. Alvar, K. Uyanik, and I. V. Bajić, “License Plate Privacy in Collaborative Visual Analysis of Traffic Scenes,” Proc. IEEE MIPR, 2022, pp. 300–305.
[13] K. Zhang et al., “Compacting Deep Neural Networks for Internet of Things: Methods and Applications,” IEEE Internet Things J., vol. 8, no. 15, pp. 11 935–11 959, 2021.
[14] N. Shlezinger et al., “Model-Based Deep Learning,” arXiv preprint arXiv:2012.08405, 2020.
[15] S. F. Yilmaz, B. Hasircioglu, and D. Gunduz, “Over-the-Air Ensemble Inference with Model Privacy,” arXiv preprint arXiv:2202.03129, 2022.

BIographies
NIR SHLEZINGER (nirshl@bgu.ac.il) is an assistant professor in the School of Electrical and Computer Engineering in Ben-Gurion University, Israel. He received his B.Sc., M.Sc., and Ph.D. degrees in 2011, 2013, and 2017, respectively, from Ben-Gurion University, Israel, all in electrical and computer engineering. From 2017 to 2019 he was a postdoctoral researcher in the Technion, and from 2019 to 2020 he was a postdoctoral researcher in Weizmann Institute of Science, where he was awarded the FGS prize for outstanding achievements in postdoctoral research. His research interests lie in the intersection of signal processing, machine learning, communications, and information theory.

IVAN V. BAJIĆ (ibajic@ensc.sfu.ca) is a Professor of Engineering Science at Simon Fraser University in Canada. His current research interests are in multimedia signal processing and compression, machine learning, and collaborative intelligence. His work has received awards at ICME 2012, ICIP 2019, MMSP 2022, and other recognitions (e.g., best paper finalist, top n%) at Asilomar, MMSP, ICIP, ICME, and CVPR. He served on the editorial boards of IEEE Transactions on Multimedia, IEEE Signal Processing Magazine, and Signal Processing: Image Communication, and is currently serving as a Senior Area Editor of IEEE Signal Processing Letters and the Chair of the IEEE Technical Committee on Multimedia Signal Processing.