The Challenges of Artificial Intelligence in Wireless Networks for the Internet of Things

Exploring Opportunities for Growth

IJAZ AHMAD, SHAHRIAR SHAHABUDDIN, TANESH KUMAR, ERKKI HARJULA, MARCUS MEISEL, MARKKU JUNTTI, THILO SAUTER, and MIKA YLIANTTILA

Digital Object Identifier 10.1109/MIE.2020.2979272
Date of current version: 16 December 2020

The Internet of Things (*IoT*), a term first coined by Ashton in [1], is an extension of network connectivity to physical devices, such as actuators, sensors, and mobile devices, that are enabled to interact and communicate among themselves and can be controlled or monitored remotely. The IoT, hailed as the enabler of the next industrial revolution, will transform how we view, interact with, and use the current physical systems around us. It has already had a major impact on health care, smart homes, manufacturing, commerce, education, and many other key areas of daily life.

CLOUD @SHUTTERSTOCK.COM/SSARO PRAKALUNG.
IOT AND AI @SHUTTERSTOCK.COM/PANCHENKO VLADIMIR

1932-4529/21©2021IEEE

The IoT market is undergoing incredible growth, and the IoT industry is projected to grow tenfold by 2025 [2]. With smart cities in sight, having automated the IoT in various forms, such as unmanned aerial vehicles (UAVs), smart homes, e-health devices, and the context-aware augmented reality (AR) and virtual reality (VR) applications used in daily routines, underlying communication networks must evolve to meet their needs. Communication networks must also support autonomous operations due to continuously changing services, an unprecedented increase in network traffic, and an increasingly complex security threat landscape due to the amalgamation of diverse IoT devices and services. All of these challenges further add into increasing the complexity of network operations.

Artificial intelligence (AI), with its disciplines, i.e., machine learning (ML), is the primary enabler of an autonomous and intelligently operating network. Since the groundbreaking work of Hinton et al. [3] in 2006 on a fast training method for deep neural networks (NNs), there has been a reinvigorated interest in NNs and other ML methods in communication networks [4]. The application of ML in wireless networks has been of immense interest and a plethora of research articles has been published. However, this is not just the first age of AI where it has attracted a huge attention of research community. During the 1970s and 1980s, there was immense, cyclical enthusiasm and optimism about AI, which was followed by periods of *AI winters*, a term coined to explain low interest in AI.

The current era of AI has been bolstered by advanced semiconductor technologies and the advent of cloud and distributed computing. In spite of all these technological advancements, a number of challenges still remain today to successfully deploy AI-based solutions on a competitive basis in wireless networks. Instead of considering AI as an omnipotent solution, a cautious approach and a careful comparison against state-of-the-art solutions is necessary to make AI-based solutions applicable and successful in future communication networks.

For capitalizing on the IoT, which has an increasing number of connected, diverse devices with emerging smart services, autonomous network operations leveraging AI is inevitable. For example, the conglomeration of heterogeneous IoT devices in UAVs, e-health, manufacturing, AR/VR, wearables, and smart homes through communication technologies will make it very difficult to differentiate a security attack from legitimate traffic and may not be practically possible or manageable without using AI [5]. Therefore, autonomous network operations are contemplated to be possible by embedding and using the concepts, technologies, and algorithms of AI in wireless networks. To avoid repeating the definitions of the vast number of types and disciplines of AI, in this article, the term *AI* refers to techniques that are used to 1) gather (raw) data from the network environment; 2) perform computations on it (e.g., for classification, training, and testing); and 3) produce intelligent, actionable information for the network. This may include the required systems of supervised, unsupervised, or semisupervised learning, to name a few.

However, using AI in wireless networks will bring its own challenges, which may not be worth considering in other fields such as machine vision and robotics, but are highly important in communication networks, specifically in the case of the IoT [6]. For instance, gathering raw data for training the system incurs network overhead. Storing raw data requires storage systems, and in big data, big storage systems are required. Similarly, performing computations on the data to extract actionable information requires higher computing resources. If resources are available in high-end servers in centralized cloud systems, latency-critical applications will be challenged by communication latency, besides other factors [7]. In decentralized systems, sharing data and training models or parameters of AI algorithms will not only require higher communication network resources but also open security challenges. Hence, using AI in wireless networks has many challenges that are not counted in most of the research in this direction.

Many state-of-the-art research articles attempt to solve specific challenges using AI in wireless networks while ignoring the resulting challenges arising as a consequence. Therefore, major challenges that result from using AI in wireless networks are discussed in this article, mainly from the point of view of the IoT. The main purpose of highlighting the challenges is twofold; the first of which is to focus research attention on the limitations of AI from the perspective of wireless networks. For example, wireless channels are prone to errors; data distribution can be nonuniform, keeping in mind the possibility of unavailability of data due to various reasons such as jamming attacks; and wireless networks can have limited capacity such as the bandwidth, storage, and computing required for AI.

The second reason is to motivate further research on developing AI-based solutions that are either specific to wireless networks or avoid facing situations where solving one problem creates another in the wireless network infrastructure. For example, learning from the big data generated by the IoT with the help of AI in the edge might yield the required results; however, the required storage and processing might be too costly compared to its benefits. Therefore, how best to avoid pitfalls when using AI in future wireless networks, specifically in the case of the IoT, is the main theme of this article.

Challenges Posed by AI in the Wireless Network Infrastructure

To complement the resource limitations, heterogeneity, and complexity of the IoT on one hand and big data on the other, various concepts of enhanced computing, storage, link, and bandwidth are bundled with the concepts, tools, and algorithms of AI. Therefore, significant research efforts are occurring in this direction, as presented in [8] and [9]. Moreover, new concepts and disciplines of AI in different

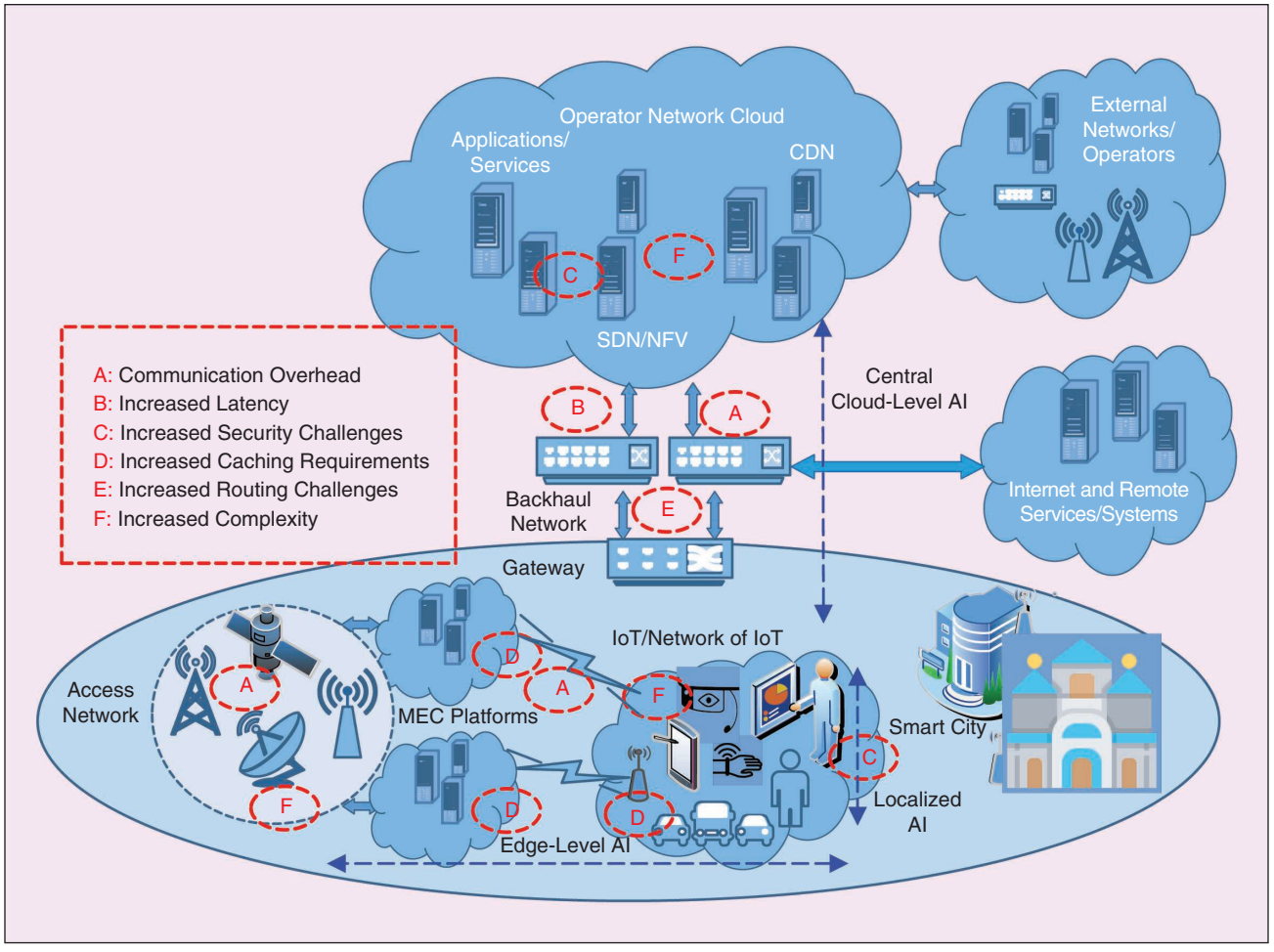

FIGURE 1 – A generic network architecture showing a few key areas of AI in the IoT and its corresponding challenges. SDN: software-defined network; NFV: network function virtualization; MEC: multiaccess edge computing; CDN: content delivery network.

network systems or services are proposed, discussed, and evaluated continuously [10]. Figure 1 presents a generic global network in which AI is used in different segments, including IoT devices, and the network that connects diverse IoT devices. However, a number of challenges will surface when AI is used, but proper consideration is not given to the underlying network architecture and infrastructure. In this section, we discuss the main challenges that will be on the forefront when AI is used in future wireless networks. The most common challenges related to nearly all types of networks, as depicted in red in Figure 1, are described as follows.

Higher Communication Overhead

Using AI to improve the efficiency of IoT devices, the services offered by IoT devices, or the functionality of the underlying network used by the IoT will result in additional communication overhead. The communication overhead caused by AI can be attributed to the very basic operating principles of AI systems. For example, ML systems derive useful information from (large-scale) data that needs to be communicated between the devices running ML algorithms. To illustrate the extra communication costs, consider the learning device in Figure 2 using communication bandwidth and spectrum for observation, communicating the results of the interpreter, and then sharing the action space with other IoT devices in the environment. Thus, the communication costs of learning algorithms in ML can be generically determined by 1) the number of communication rounds required to observe or learn the environment (ML algorithm convergence), 2) the number of channels used per communication round, and 3) the bandwidth or spectrum used per channel in a communication round.

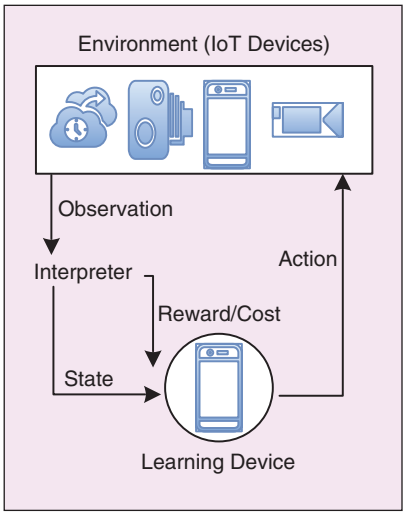

FIGURE 2 – A simplified reinforcement learning system.

In the IoT, the huge amount of diverse data generated by the massive number of connected IoT devices will

require very high memory and processing resources [11]. Thus, the limited capacity of IoT devices will force ML processing and the required storage to other available resources, the most prominent being edge nodes, which are near IoT devices mainly to meet the latency requirements [12]. However, it is highly challenging in most use cases of big data generated by the IoT to fit and process the entire data set in the edge as discussed in [12]. Therefore, two approaches will most likely be used. First, coordinated distributed processing in multiple edge nodes, and second, centralized processing in a pool of larger processing and storage systems, such as centralized cloud systems. In both cases, the communication overhead will increase much higher than anticipated.

For the coordinated distributed processing of ML in multiple edge nodes, distributed ML, called *federated learning*, is proposed [13]. In federated learning, the training data remain distributed over a large number of nodes. A centralized model is trained by the distributed nodes performing computation over their individual data independently [13]. Most distributed ML systems usually contain a group of server nodes that manage global parameters and worker nodes that pull the latest parameters and push the gradients to server nodes for update operation. This process of pulling and pushing the parameters and gradients during the training cause huge network traffic, as demonstrated in [14].

Different ML algorithms for federated learning are evaluated in [15] and reveal astonishing results. For example, when the data distribution is nonuniform, the convergence time of the training model is very high and the accuracy is very low. Therefore, the synchronization of a distributed ML system is critical to accurately update each system to the global model using fresh information. However, synchronization is a high-cost operation that requires significant communication rounds for fresh updates for all of the participating nodes. Heterogeneity in the resource capacity of IoT devices and diversity in data sets will further increase the communication rounds to synchronize ML among all of the participating nodes [15], [16]. In the IoT, as the data sets for learning grow larger, the models will be more complex, and training AI models will increasingly require distributing model-optimization parameters over multiple machines [13]. As a result, using AI in large IoT networks with multiple diverse nodes and heterogeneous links will result in very high communication overhead costs [17].

The Case of Centralized Cloud Systems

Computation and storage costs can be minimized by using centralized cloud-based systems for ML. However, the communication costs of bringing raw data, training model parameters, and later, the outcome of learning algorithms between the cloud and end-user devices will consume high link and bandwidth budgets. Traditional AI algorithms are designed for highly controlled environments such as data centers, where the assumptions are 1) that the data are independent and identically distributed among machines and 2) that high-throughput networks are always available. In wireless networks, both of these assumptions may not always hold true, requiring the frequent resending of data, which leads to further network resource dedication. Similarly, end-to-end security procedures might further increase the network overhead. In summary, for AI-based operations, the continuous gathering of data will be inevitable, the dissemination of decisions must happen, and scaling network resources (e.g., channels and bandwidth, access and backhaul networks, and so on) will pose significant challenges, more so in the case of AI in centralized cloud systems.

Challenges for Latency-Critical IoT Systems

The dynamic nature of future IoT services will require real-time computation, ideally near the users, or otherwise with no observable delays [18]. However, due to low capacity, IoT devices will take considerably longer for AI processing within IoT devices. In the case of processing in the edge, Arjevani and Shamir [16] concluded that many communication rounds will be required and still provide worst-case optimal use in minimum-assumption situations. Simply put, (raw) data acquisition, then data analysis and training, and the continuous feedback loop in ML will introduce much higher delay. Even traditional (non-ML) iterative or feedback systems are facing challenges in terms of computation and link delays to meet the real-time requirements of dynamic services and highly mobile users [19]. Whereas the delay in training ML models, for instance, in streaming applications, will make it very difficult to match the latency requirements [20].

An interesting scenario of vehicle to everything (V2X) using federated learning is evaluated in [21]. The results reveal that in most cases, even near the user scenarios, the federated learning approach incurs higher latency. In latency-critical systems such as V2X communication and telesurgery, an extremely small delay, for instance, in moving the steering wheel or robotic arm, can be catastrophic. In [22], a convolutional NN-based object-inference task was offloaded to the cloud, which leads to a 2–5-s latency. The experiments were run in the United States and China, and the authors concluded that the variability of latency makes the service unreliable. Therefore, the continuous learning and adjustment of systems using the apparatus of AI for such critical operations must first ensure latency.

Furthermore, the limitations in time for the usefulness of data for AI processing and the validity of the outcome of AI mechanisms must be counted. To elaborate on these limitations, consider intrusion-detection systems. The analysis of data for intrusion detection is highly time-sensitive, and if the communication medium introduces delay, the whole process might be rendered useless [23]. Investigating distributed ML systems [14] reveals that network communication consumes more time by an order of magnitude than computation to train ML models. Similarly, the completeness of data required

within the time frame on which to make observations is also crucial. The question holds true for observation- or data-oriented decision-making systems of any kind. Therefore, latency in such systems is as vital as the validity or accuracy of the system. This can be further clarified with the example of object recognition through deep learning, as described in [22]. Images for object-recognition tasks processed locally consume 7 W of energy, and when processed in the cloud it consumed 2 W of energy. However, the latency goes well beyond the constraints of 500 ms. It took between 2–5 s of time when processed in the cloud. Therefore, it is concluded in [22] that for real-time deep learning tasks, the cloud is not yet a viable solution due to its higher latency.

The Case of Industrial Control Systems

Industrial control systems (ICSs) have very complex requirements, such as low latency, high reliability, security, and safety; and 5G seems to be promising from many aspects, as highlighted in [25]. The requirements of ICSs are different from those of other systems and services; for example, the bandwidth requirements for data transmission can be as low as a few bytes, whereas the latency requirements for real-time control messages in production and manufacturing can be as strict as 250 ms [26]. Although ICSs are moving toward distributed automated systems, distributed systems connected through communication networks have a considerable delay mainly due to the interworking architectures from access to core networks. Only the core network in 4G cellular systems introduces a 39-ms delay to contact the gateway toward the Internet [26]. The measurements reported in [26] are in two cells and a low-traffic scenario. During peak hours, the delay can grow further, up to 85 ms. Even though the delay can be minimized, for instance, by localizing various network functions in the edge in 5G, no significant change is made for the traffic reaching outer networks. Hence, adding ML in ICSs will add further delays, which raises serious concerns about the benefits of ML in ICSs.

Challenges in Routing and Network Traffic Control

Even though AI has been proposed for routing, traditional AI/ML techniques such as artificial NNs (ANNs) have evident shortcomings in terms of scalability and computation efficiency when considered for routing [27]. Measuring the benefits of using deep learning-based routing versus the traditional Open Shortest Path First (OSPF) Protocol routing mechanism in [28], the results reveal that OSPF yields the same throughput and average delay when the signaling interval between routers is more than a certain threshold. However, counting the computational and storage resources, straightforward OSPF is a better option for the core and backhaul networks, where changes are less likely compared to the dynamics in access networks. Furthermore, mutating or changing Internet Protocol addresses or packet header fields for either security attacks or preventing security attacks [29] will further challenge the phenomenon of learning and may lead to continuous feedback loops for finding the best route.

Many research efforts are dedicated to using AI in dynamic networks. Dynamic networks have frequently changing topologies that require the recurrent sharing of information among nodes in the network. An example of dynamic networks are mobile ad hoc networks (MANETs), which are composed of resource-constrained mobile devices. MANETs are formed randomly and spuriously by freely moving nodes. Thus, their routing protocols usually have higher overhead due to the dissemination of topology information as well as sharing information because of transient disruptions during routing protocol convergence [30]. However, their constantly changing topologies lead to the continuous arrival of new information. Such systems behave like a closed-loop system, making it hard for the learning algorithms to converge within the latency constraints.

The Case of Software-Defined Networking

Because traditional network traffic control systems rely heavily on predefined policies hardwired in data plane devices, new solutions such as software-defined networking (SDN) have been sought to minimize manual configurations and enable run-time changes in network policies. SDN splits the network control data planes, centralizes the network control plane, and enables the programmability of network equipment. Thus, SDN enables dynamicity in communication networks, which is required in wireless networks to cope with sudden changes in user behavior, network traffic, and air interfaces. Therefore, the ML-based management of complex network systems and ML-based route selection in SDN, according to the traffic requirements of different applications, have been proposed in [31] and [32], respectively. Hence, AI-based network traffic control in SDN has recently gained research attraction, mainly to cope with the dynamicity of mobile nodes, diverse services, and increasing traffic variations.

Although SDN provides promising solutions to many challenges, it has its own inherent challenges of scalability and security, mainly due to its centralized control architecture [33]. In simple terms, the centralized SDN controllers need to be scalable enough to install flow rules in the entire data plane under its control within latency constraints. In terms of resilience, the work in [34] reveals that it is hard to achieve the carrier-grade requirement of restoration within 50 ms in large OpenFlow networks. However, using AI in SDN will require either adding software modules to the controller or adding an application on top of the control plane. In both situations, controller involvement in the data plane will further increase by consistently feeding information (e.g., flow patterns, flow statistics, or samples of packets) to AI algorithms. Hence, using AI in SDN without giving proper consideration to its inherent limitations will further increase its challenges.

Challenges in Caching

Network caching systems temporarily store data or information near users to minimize redundant network traffic [35]. Traditionally, a router, for example, would cache data that have higher requests or frequently passes through it. However, the explosion of big data from the IoT will significantly challenge the fundamentals of in-network caching. AI-based systems have been proposed to enable the network to learn which data or information to cache [36]. However, using AI within network devices, e.g., routers and switches, will consume the resources meant for storing routing procedures and paths, access control lists, and so forth.

For example, in [37], the authors proposed content caching using deep learning in SDN. Considering the OpenFlow standard of SDN used in their analysis, OpenFlow switches have limited capacity to store unsolicited flows until the controller updates the flow tables, and in some cases, have limited capacity to store flow rules [33]. Furthermore, SDN controllers have serious scalability challenges; therefore, various hierarchical and distributed control plane architectures have been proposed, as described in [33]. Because of these limitations, the authors in [37] suggest sending the prediction output of the deep learning algorithm to the controller so that the controller knows the popularity of the content in the network it manages. The humongous increase in the number, types, and services of the IoT will increase the amounts and varieties of popular content. Hence, using AI algorithms on the content within the network will require a drastic increase in memory size as well as processing capability to meet the requirements of real-time services. Therefore, content caching in the edge, which has its own limitations and challenges, is proposed as follows.

The Case of Edge Resources Versus Data Growth

Partial or full storage and processing in the edge is proposed to deal with varying and massive amounts of data under the constraints of time validity or duration, e.g., for useful information retrieval from raw data and generating actionable information or intelligence. However, the main question (usually ignored) is how much storage and processing will be required? Many evaluations of edge-enabled deep learning, such as those discussed in [12], consider the maximum data size of a task as low as 1 Mb/s and increase the number of edge nodes for processing the data by many numbers at a time (10–90 for 1,000 tasks). Having said that, the user-experienced data rates in 5G are expected to be 1 Gb/s in downlink, 500 Mb/s in uplink, and capacity targets as high as 15 Tbps/km^2 with 250,000 user devices in a square kilometer [38].

Currently, the data size of medium-level operators easily exceeds 100 s of terabytes and will further increase because video traffic (4,000, 8,000, 3D video, and 360° video) will account for roughly 75% of traffic by 2023, according to the GSM alliance. For example, AT&T's network carries more than 200 PB/day. Keeping these facts in mind, the main challenge in the edge is the computation needed for the real-time analysis of raw data generated by end-user devices and the IoT, mainly due to the diversity of applications generating different traffic. Traditional ML, however, requires full access to data sets with centralized computing through ultrafast chipsets, that is, graphics processing units, connected through up to 256-Gb/s connections. Thus, specific processing units, such as tensor processing units [39], are required and will be capable of matching the quantity of data passing through the networks. Keeping such huge amounts of data within the networked devices, or even in edge nodes for AI processing, will be highly challenging.

Security and Privacy Challenges

The application of AI for IoT security has received considerable momentum in recent years. AI is typically used for discovering a pattern in existing data, detecting outliers, predicting values, or feature extraction, which are all very crucial tools for securing IoT devices and network. The main objective of using AI for IoT security is detecting a security breach, which can be divided into three categories, according to [5]: 1) malware, 2) intrusion, and 3) data anomaly detection. For example, in [40], the authors presented a linear support vector machine-based Android malware detection for reliable IoT devices. An example of intrusion detection can be found in [41], where the authors applied a two-tier classification mechanism based on naive Bayes and *K*-nearest neighbor to prevent the intrusion detection of an IoT network. An example of data anomaly detection is presented in [42], where the authors propose using ANNs in an IoT gateway to detect anomalies in the data sent from edge devices. We invite you to peruse [5] to learn more about the AI schemes used for security purposes.

A key question for using AI in the context of IoT security is how best to generate a high-quality training data set containing possible attack types and patterns. A high-quality training data set is essential for the accuracy of AI schemes. A diverse training data set containing information that reflects all the strategies of real-world attacks is required for the successful deployment of AI methods for IoT security. However, due to the large number of devices generating massive volumes of data, a real-time, high-quality data streaming and extraction remains a challenge. In addition, extracting a reliable data set through the collaboration of different devices can also be challenging due to a wide diversity of IoT devices. Most publications on AI for IoT security are applied for high-quality data. For example, the intrusion-detection mechanism of [41] uses NSL-KDD data sets to train and validate the AI scheme. However, NSL-KDD may not be a perfect representative of existing real networks. Due to the amalgamation of a large number of heterogeneous devices in an IoT network, the effect of noise and interference can corrupt a data set. Therefore, AI methods based on high-quality data sets used to secure

Privacy, with all of its potential legal, ethical, and moral considerations, is likely to be exposed in the era of AI-controlled networks.

the IoT are highly infeasible. It should be noted that acquiring a data set for training in the context of IoT security is more difficult than for image or natural language processing.

ML techniques such as supervised, unsupervised, and reinforcement learning-based approaches for IoT authentication, access control, secure offloading, and malware-detection schemes are studied in [43]. The authors conclude that both the supervised and unsupervised learning methods for IoT security have serious challenges of oversampling, a lack of sufficient training data, and bad feature extractions. Supervised learning-based intrusion-detection systems have, sometimes, misdetection rates that cannot be neglected in IoT systems. An *RL*-based system can cause network disaster for IoT systems at the beginning stage of learning, i.e., exploring bad security policies to achieve optimal strategies. The optimal solution in such cases is to have backup security mechanisms to protect IoT systems during the exploration stage of the learning processes [43].

Another key challenge is the inherent security flaws of traditional AI mechanisms. First, adversaries can feed polluted data during training and reduce the performance of AI schemes. This attack is commonly known as a *poisoning attack*. Second, an adversary can feed feasible, new inputs in an attempt to evade detection, which is known as an *evasion attack*. Third, adversaries can create their own AI models by employing a public application programming interface and refine their own model using it as a guide. Therefore, the security of an AI scheme itself needs to be taken into account before using it to secure IoT systems and devices. Several techniques, such as data sanitization, adversaries retraining, and homomorphic encryption, exist to make an AI scheme more secure against its inherent security flaws. The security flaws of AI algorithms and their countermeasures are presented in detail in [44].

In [45], some basic questions are put forward regarding the use of AI in security systems. Comparing the use of AI in other disciplines, Sommer and Paxson [45] state that it is not only harder to use AI for intrusion detection, but also the premise of using AI to find novel attacks does not hold true. The reason is simple: AI algorithms typically use previous experiences or knowledge upon which to build decisions, whereas for novel attacks, the system may not have prior data or information available. Another key question is how best to intervene once an IoT device is discovered to be a part of a distributed denial-of-service attack. Removing the device from the network might not be possible, especially if it is a critical device. Most AI methods just focus on detecting an attack and do not address the mechanism of rectifying this situation.

The Case of Privacy

Privacy, with all of its potential legal, ethical, and moral considerations, is likely to be exposed in the era of AI-controlled networks. For example, in [46], the authors propose harnessing user behavior, social relationships, and other personal attributes from social networks for proactive caching in the edge. Similarly, AI algorithms can themselves leak sensitive information when they are subjected to security attacks [47]. The adversary can perform an inverse operation to attain input data, e.g., patient medical information, user fingerprint, or customer purchase record. Therefore, preserving privacy in the age of AI will be challenging from both the algorithmic security and human-invasion perspectives. There are various approaches now emerging to safeguard the privacy of user data, such as differential privacy models [44] and the encoding and shuffling of algorithms [48]. However, privacy requires more regulatory efforts as well because in most cases, privacy challenges arise on the operator or service provider sides [49].

System Complexity Challenges

Deploying AI in communication networks will further increase the complexity of the system if the implementation is carried out as we see currently: implement case-specific ML to achieve one objective, ignoring other objectives or end-to-end network goals. Hence, one of the main challenges that remain at the forefront is that the research on using AI in wireless networks is optimizing one objective while overlooking other constraints such as latency, link, storage, and processing overhead. For example, increasing spectral efficiency using reinforcement learning is proposed in [50]. The cost of information sharing, storing, and processing while using the proposed mechanism in real-world or large networks is not mentioned.

Looking at the overall network performance or end-to-end network objectives, little attention is paid to a cross-layered approach, as shown in Figure 3, in which AI in one layer could also benefit or help with optimization in another layer. Even more so, the negative effects of using AI in one layer over the performance in other layers is rarely considered. For example, the increased latency in finding the optimal route using ML on scheduling in medium access control and physical layers are not properly investigated. Due to resource (power, storage, and processing) constraints, a massive number of IoT devices will simply transmit data without performing heavy computation, e.g., for compression or encryption. This will require the upper layers to cooperate to adaptively compress or encrypt data. In mobile IoT nodes, the cross-layer interaction, e.g., for channel or topology selection, from the physical to application layers will require the

synchronization of all layers, not only to minimize challenges faced by the IoT but also to facilitate end-to-end communication. To properly elaborate on system design complexity, in the following section, we describe using ML in digital transceiver designs as an example.

The Case of Communication Signal Processing

A communication infrastructure for the IoT consists of a central entity, commonly a base station (BS), to handle the traffic of tens or hundreds of IoT devices. The main challenge for the BS is to enable access of a unknown subset of IoT devices at a given instant. Thus, there is a need for efficient signal processing implementation on the BS side. On the other hand, many IoT devices will not require complex signal processing, as they do not support multiantenna communications or complex channel coding.

Due to the nature of transceiver algorithms, it is difficult to justify the use of ML techniques to replace conventional signal processing algorithms, which are typically designed analytically using statistics, mathematical optimization, and information theory. Algorithms based on such techniques are well established and can provide optimal performance. However, many of such algorithms are based on the assumption of a simple and linear system model [51]. For example, most beamformers or precoders for wireless transmitters are based on the assumption that perfect channel state information (CSI) is available at the transmitter side. However, it is highly unusual to obtain perfect CSI at the transmitter. In this scenario, an ML-based beamformer or precoder can be used, either of which are not dependent on perfect CSI. Thus, the application of AI is perfectly justified for transceiver blocks that are highly nonlinear in nature and where the mathematical model is far from perfection.

There are, however, many suboptimal solutions available for transceiver algorithms, which are suitable for implementation with satisfactory error-rate performance. Suboptimal equalization algorithms, such as zero forcing or minimum mean square error, can reach to a near-optimal level for massive multiple-input, multiple-output (MIMO) systems when the ratio between the number of antennas in a BS and the number of users is relatively large [52]. It is difficult to justify the application of AI techniques when a suboptimal algorithm can provide satisfactory performance. In summary, AI will continue to excel for nonlinear signal processing applications like digital predistortion, which is used to compensate for the nonlinearities of a power amplifier. On the other hand, suboptimal algorithms can provide very good performance with feasible complexity for many applications. As a result, more research is necessary to make AI solutions competitive against those applications.

In most research areas, the processing power required for ML algorithms is no longer a big hindrance due to the advent of cloud and distributed computing. However, the requirements for digital signal processing are significantly more stringent than are traditional applications. Besides, most of the computing required for the physical layer of telecommunications are still carried out by embedded platforms. Some parts of the processing, for example, part of the baseband units, can be transferred to the cloud. In spite of that, a remote radio head (RRU) unit requires highly complex on-site computations, which must be carried out by embedded computing platforms. Thus, the high complexity of sophisticated AI techniques introduces new challenges for RRUs.

We now provide, via a use case, an intuitive discussion on the complexity of NNs and how they fare against traditional signal processing algorithms. In general, NNs require a large number of matrix calculations. An $N_l = 3$ layer fully connected feedforward NN can be represented as

$$\mathbf{y} = f(\mathbf{W}_3 g(\mathbf{W}_2 h(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2) + \mathbf{b}_3), \tag{1}$$

where h, g, and f are different activation functions for different layers [53]. The weight matrices for the layers are represented as $\mathbf{W}_1, \mathbf{W}_2$, and $\mathbf{W}_3$, respectively, and the biases for the layers are $\mathbf{b}_1, \mathbf{b}_2$, and $\mathbf{b}_3$, respectively. The input and output vectors of this network are denoted as $\mathbf{x}$ and $\mathbf{y}$, respectively. It can be seen from the equation that the NN requires three matrix-vector multiplications. The complexity of an $n \times n$ matrix and $n \times 1$ vectors can be denoted by n^2 and

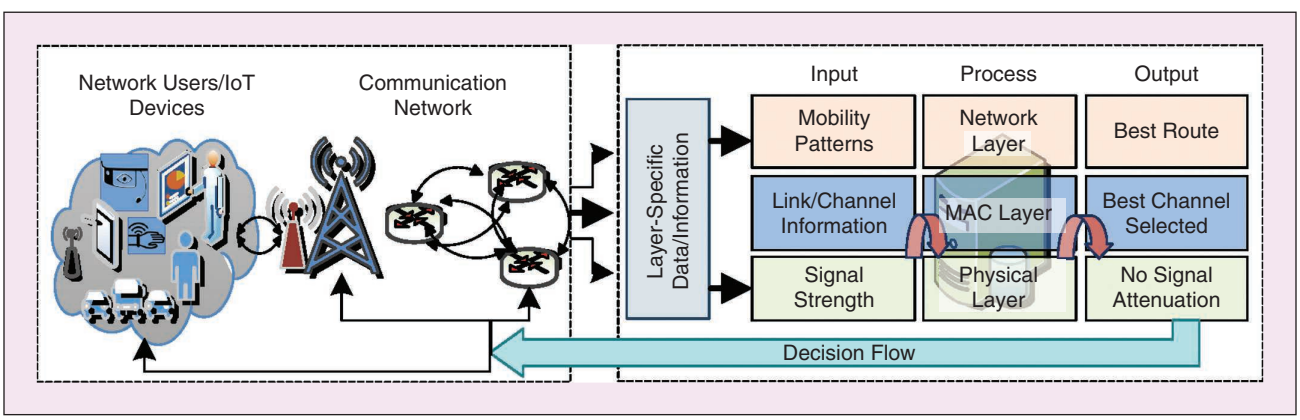

FIGURE 3 – An example of a layerwise AI implementation. MAC: machine-aided cognition.

thus, the NN has an $N_l n^2$ complexity if we consider only the matrix multiplications. Here, we assumed that each layer has n number of neurons to simplify the comparison.

The training or learning process used to determine appropriate weights is a key part of NNs, and the performance of the network is heavily dependent on the methods used for training. The most common training or learning scheme used for an NN is known as *backpropagation*, which follows a gradient descent approach that exploits the chain rule. The backpropagation traverses through the same nodes and layers and thus, the number of multiplications after updating the weights is the same as that of the forward propagation. Therefore, for a three-layer NN, the total number of operations can be expressed as $2N_l n^2$. However, the NN requires a large number of iterations of forward and backward propagation to achieve the required accuracy for the weights and thus, the complexity of the NN training can be expressed as $T(2N_l n^2)$, where T is the number of iterations.

A least-squares solution, which is commonly used in many transceiver operations, requires matrix inversion, which has a complexity of n^3 for traditional applications. For large values of n, the n^3 complexity is higher than the complexity of forward and backward propagation, i.e., $N_l n^2$ [53]. In spite of the difference, the number of operations for a forward-backward pass of an NN and a matrix inversion is still comparable. However, as the value of T is typically very large, for example, in hundreds of thousands or in millions, the time required to train a network is too high and impractical. If we take T into account, the numbers of operations are not significantly higher than least-squares solutions. It should be noted that, once trained, the network can run faster than a traditional least-squares solution. An NN trained for multiantenna symbol detection is proposed in [54]. Even though the network can achieve ML performance, it took two days to train the network to properly function as an MIMO detector. These two days required to train the network can render the ML detector useless for many scenarios.

Roadmap: A Generalized Global AI Architecture

Even though there exists a plethora of research on using AI in communication networks for different use cases, applications, network functions, and segments, little effort has been put into visualizing the holistic network architecture. The major benefit of the holistic network view is to attain end-to-end goals without having situations where achieving one objective leads to a compromise on another. In addition, having a global network view is vital to the efficient utilization of available resources throughout a network. Therefore, a global network architecture using AI is presented in Figure 4.

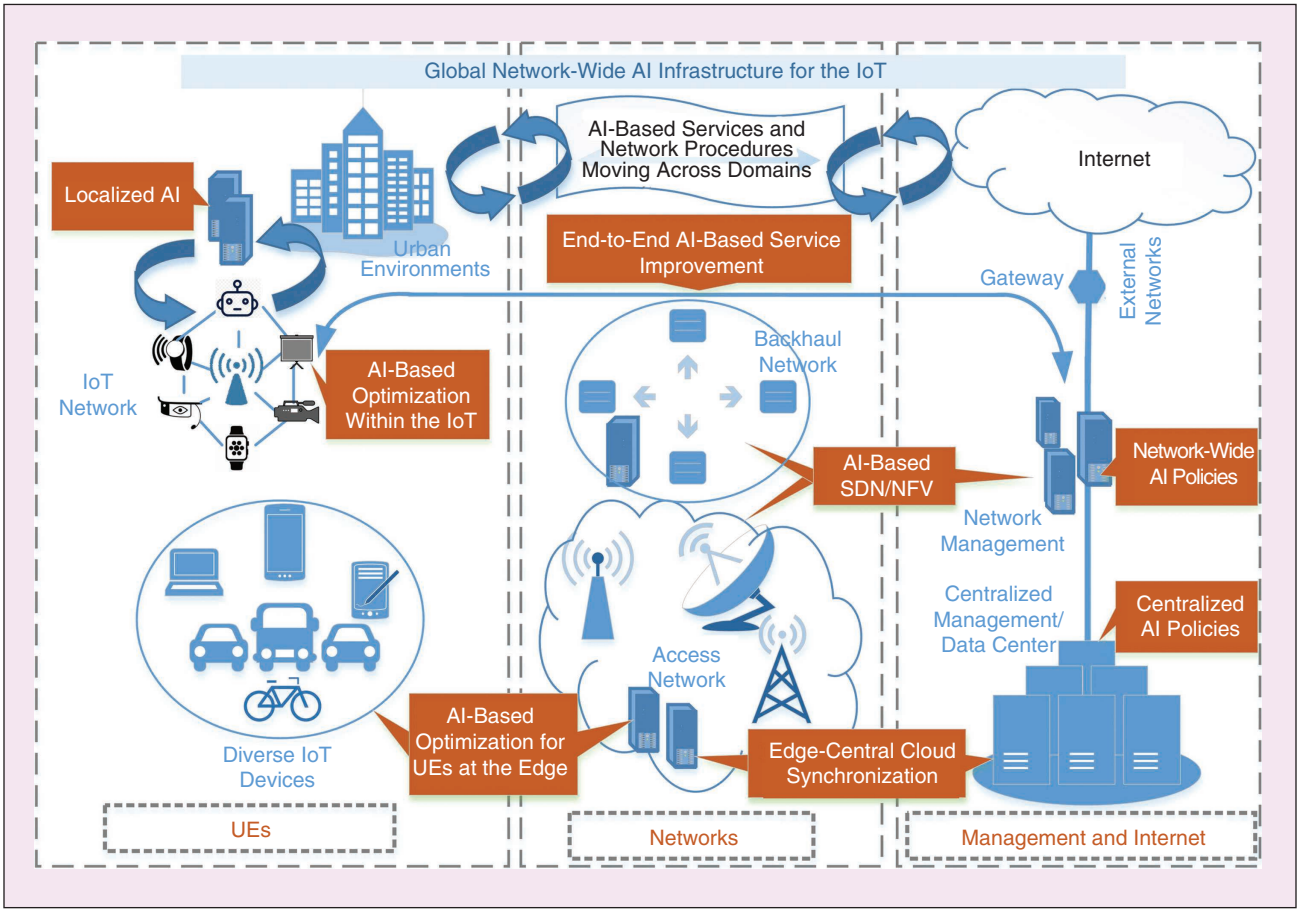

FIGURE 4 – AI operations in an AI-based communication network infrastructure. UEs: user environments.

The three-tier architecture represents autonomous and intelligent network operations leveraging AI in each tier as well as across the three tiers to maintain synchronized AI-based operations in the entire network for different IoT services.

In Figure 4, the user environments comprise end-user devices (IoT devices) and IoT networks that use AI to improve their performance. Due to limitations of IoT devices such as processing and storage, edge [or multiaccess edge computing (MEC)] platforms are used when higher resources for AI operations are needed. Because edge platforms still represent distributed operations with limited capabilities, centralized cloud systems are proposed for two major reasons. First, to maintain global network view, including AI operations to maintain synchronized operations throughout the network. Second, to provide higher resources when the edge platforms fall short of resources. On one hand, the communication network infrastructure using diverse technologies from radio access technologies to the application layer facilitates AI operations throughout the network, i.e., from end-user environments to centralized cloud systems. On the other hand, AI is used in the communication network infrastructure to improve the end-to-end goals of the network. Therefore, it can be seen in Figure 4 that AI is used throughout the network, connecting many IoT networks, edge platforms, and centralized cloud systems. Figure 5 represents the three-tier network architecture, visualizing how, in practice, AI will be used in a large network (see Figure 4).

In Figure 5, the local network represents a network of IoT devices which, in most cases, have very limited resources, e.g., storage, computing, and transceiver capabilities. Edge networks comprise edge nodes, where each node is near an IoT network to meet latency constraints and each has more resources compared to local IoT networks. The centralized cloud serves many edge networks and has higher resources to serve the entire (global) network. The resources available within each network, i.e., the local IoT and edge networks, are visible in the centralized control system, much like the data plane resources that are visible in the SDN control plane. Because the tools and algorithms of AI require data, and a common assumption is that the more there data are, the better the results will be [11], it is highly likely that the needed resources are not available locally or in the edge. In that case, requests must be sent to high-resourced centralized cloud systems to fulfill the requirements of processing and storage. Thus, AI procedures throughout the entire network are carried out in the following three steps:

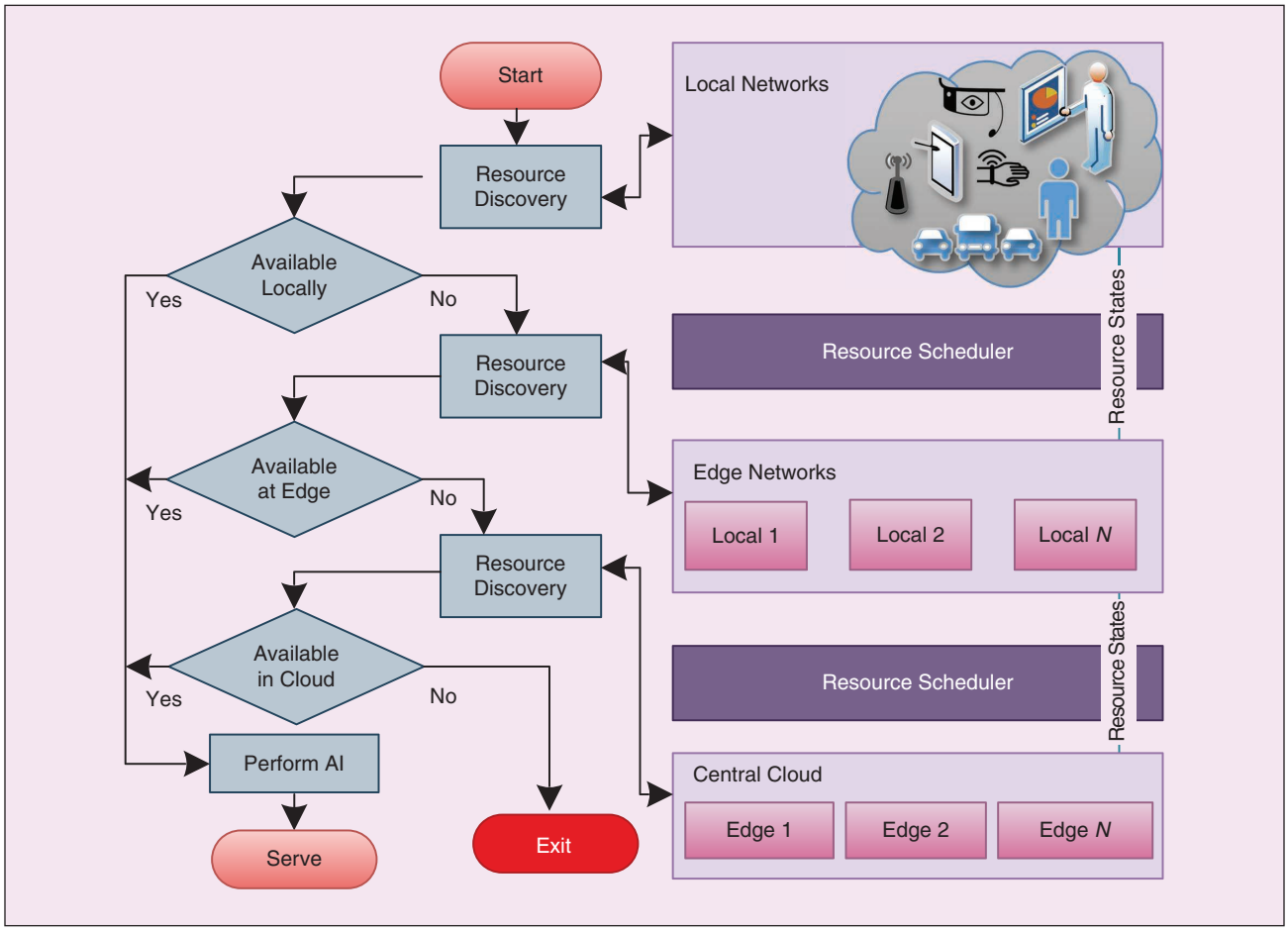

FIGURE 5 – Resource discovery for AI-based operations.

1) *Local resource discovery*: Before initializing an AI procedure, a local resource-discovery procedure is carried out. If the resources, such as storage and computing, are available locally, the process is carried out within the local IoT network.
2) *Edge resource discovery*: If the local IoT network resources are not enough, the resource-discovery procedure in the edge layer will be carried out. If the resources in the edge are enough and are available for the AI procedure, edge resources will be allocated and all of the processing will happen in the edge layer. It is important to note that link and bandwidth resources will also count for sending the (possibly raw) data, training parameters, and decisions back and forth between the local IoT network and the edge nodes.
3) *Central cloud resource discovery*: If the edge resources fall short, the resource-discovery and allocation procedure will be carried out in the central cloud. Thus, AI processing will happen in the centralized cloud, and even more network resource will be consumed in this case.

Such globally optimized network architecture will have the potential to avoid many of the challenges described in the previous section. For example, localized IoT-based AI processing yields benefits such as low bandwidth consumption and meets the requirements of latency, as evaluated for the wireless sensor networks in [55]. The edge nodes are involved for two major reasons: First, the resources in the IoT or in local networks are not capable (fall short) of performing the tasks, and second, latency constraints do not allow for performing the tasks in centralized cloud systems, as discussed in [12]. Yet, edge nodes have challenges in terms of resources, mainly due to the humongous growth of data as well as distributed ML being required for distributed services that require the global aggregation of, for instance, data and learning parameters [56].

As many IoT environments are dynamic and evolving, learning models are bound to evolve, which can also create challenges in synchronizing and monitoring multiple edge nodes [57]. Therefore, a dynamic architecture capable of synchronizing multiple edge nodes through a centralized control and monitoring system is required, as depicted in Figure 4. In the proposed architecture, the concepts of intelligent service decoupling for enabling the mobility of AI systems (e.g., an AI system running as a virtual function) between multiple edge nodes and between edge nodes and centralized monitoring and control systems is visualized. Intelligent AI services for the IoT can be decoupled much like network function virtualization with the unique requirements of the IoT and AI, as presented in [58]. AI system or service mobility, along with synchronizing the needed AI processing among multiple edge nodes [56], can be achieved through the novel technological development of communication and computing technologies such as SDN and MEC, as explained with examples in [59]. The hierarchical architecture is beneficial in terms of proactive caching without draining local and edge resources or compromising the latency constraints, as discussed in [60].

TABLE 1 – A SUMMARY OF CHALLENGES IN WIRELESS NETWORK INFRASTRUCTURE LEVERAGING AI FOR THE IoT.

	IOT ENVIRONMENT				
CHALLENGES*	IoT DEVICE	LOCALIZED AI	EDGE-LEVEL AI	CLOUD-LEVEL AI	REFERENCES
Communication overhead	Low	Low	Medium	High	[11], [15]–[17]
End-to-end latency	Low	Low	Medium	High	[19]–[22]
Security challenges	Low	High	High	High	[45], [62], and [63]
Caching and memory	High	High	Medium	Low	[23] and [39]
Network traffic control	High	High	Medium	Low	[27], [28], and [30]
System complexity	High	High	Medium	Low	[51] and [54]

*Challenge levels or severity represented by low, medium, and high.

Discussion and Future Research Directions

AI, with its many disciplines, tools, and algorithms, will play an important role in efficiently utilizing available network resources for the IoT through autonomous network operations. However, deploying AI mechanisms requires proper investigation of the resulting consequences in terms of different performance indicators. Furthermore, the effects of using AI in one service over the other and in one network segment for function over the other must also be properly investigated. For example, the gathering and processing of raw data and dissemination of the resulting information or decisions of AI can increase communication overhead, causing network congestion that induces delays in different network functions, such as routing or access control.

Therefore, it is highly important to investigate the resulting challenges in the underlying network infrastructure due to integration of the mechanisms of AI in communication network infrastructures, which will be used by the IoT. The major challenges discussed throughout this article are summarized in Table 1 and include the most important references. The challenges are presented with respect to using AI within IoT devices and within a localized network of the IoT, edge-level AI that runs AI procedures in the edge nodes, and centralized cloud-level AI in which high-performance cloud infrastructures are used for AI processing. The challenges are classified according to different levels (low, medium, and high) to give an insight into their severity based on the references.

The certain measures taken according to their contexts, network

infrastructure, and available resources can help us use the mechanisms of AI more effectively. Comparing the requirements, for instance, the time sensitivity of applications versus the benefits of using the disciplines of AI either in the local IoT network or in the centralized cloud systems, might provide better conclusions. For example, latency-critical applications need to use the concepts of service migration (AI processing) from the central cloud to the edge or to local IoT gateways. In this case, AI must be bundled with efficient service-migration techniques and slice elasticity to increase or decrease resources (e.g., in the edge nodes) accordingly. Therefore, the holistic view of the global network infrastructure and its available resources will be highly beneficial. However, further research is necessary in the following communication network-specific areas to reap the full benefits of AI.

AI-Based Approaches Within Bandwidth, Spectrum, and Latency Constraints

It is foreseeable that the number of end-user devices will grow exponentially in future wireless networks, have different traffic patterns, and be prone to security challenges [61]. Hence, how best to efficiently use the existing allocated bandwidth and spectrum resources while not compromising on the required data rates, the quality of service, and the quality of experience will be a huge challenge. AI-based approaches, which can predict traffic growth and flash traffic, proactively move services between edge and centralized cloud systems and dynamically allocated resources and will definitely yield better results. However, more research is needed to develop AI mechanisms that can be trained quickly and effectively with less data to consume less bandwidth or spectrum resources.

AI-Based Security Approaches for AI-Based Security Challenges

Conventionally, using AI for improving network security is highly researched; however, on the contrary, e.g., using AI for security attacks on network entities, must also be investigated. Security attacks leveraging AI can be more challenging to detect or stop, as demonstrated in [62] and described in [63]. Similarly, AI needs data, and data needs privacy. Therefore, AI-based approaches used to secure resources and data from AI-based security threats and privacy issues represent interesting challenges that need further research.

Meeting the Caching Requirements in Times of Big Data

Using the tools of big data analytics requires resources such as storage, computing, and link capacities. However, determining early enough whether the data can be counted as big data will lead to a better selection of technologies for the purpose, as described in [23]. Now that resources near the users, i.e., edge clouds, are gaining a foothold in communication networks, scaling the resources up for big data will, eventually, not pose major challenges. However, further research is required to enable using AI on big data near the data sources within the resource constraints.

Network Abstraction to Cope With Complexity

Abstracting the underlying network infrastructure from services that will use it will simplify the network to be used for any kind of service. A granular, event-driven control of network elements through high-level policies and the avoidance of low-level configurations has been enabled by SDN [24]. The same mechanisms, i.e., functional split and abstraction, have been proposed for radio access technologies; however, further research is required on the MIMO side. Abstraction in the IoT has driven research and industry interests as well, as seen from the Pelion IoT platform. Leveraging AI in the same fashion, AI as a service whenever and wherever needed in a network, will result in the same benefits without increasing the overall complexity of the system, which needs further research.

Conclusion

AI has gained research momentum in wireless networks to cope with the increasingly complex nature of diverse IoT devices and services. However, most of the state-of-the-art research takes the concepts of AI from other mature technologies, such as robotics and computer vision, and uses them as-is it to solve different complex challenges faced by IoT devices and services as well as the underlying network serving the IoT. Such a right-away use of the concepts of AI in the wireless network infrastructure gives rise to many challenges.

In this article, the main challenges were highlighted with the potential solutions and open research issues that need further research. The main objective of this article is to drive attention for future research toward a wireless, network-specific design of the concepts, tools, algorithms, and even disciplines of AI for communication of the IoT. Furthermore, the generic requirements of an IoT wireless network were highlighted to elaborate on need and integration points of the concepts of AI into the wireless network infrastructure used by the IoT. The challenges arising in each integration point of AI and wireless networks were discussed. As a roadmap, a generalized conceptual framework was suggested, which could solve most of the challenges using novel technological concepts employed for network programmability, global network resource visibility, and the granular control of network and AI functions.

Acknowledgment

Ijaz Ahmad is the corresponding author.

Biographies

Ijaz Ahmad (ijaz.ahmad@vtt.fi) earned his Ph.D. degree in wireless communications from the University of Oulu, Finland. He is currently a research scientist at VTT Technical Research Centre of Finland, Espoo, Finland. He has received several awards, including the Nokia Foundation Scholarship, the Tauno Tönning and Jorma Ollila grant awards, and two IEEE Best Paper awards. His research interests include 5G, 6G, 5G security, the Internet of Things, and the application of machine

learning in wireless networks. He is a Member of IEEE.

Shahriar Shahabuddin (shahriar.shahabuddin@nokia.com) earned his Ph.D. degree in wireless communications from the University of Oulu, Finland, in 2019. He is currently a system-on-chip specialist with Nokia in Finland. He earned his M.Sc. degree with distinction. He was the recipient of the Best Masters Thesis Award from the Department of Communications Engineering, University of Oulu, Finland. He has also received the Nokia Foundation Scholarship, the University of Oulu Scholarship Foundation Grant, and the Taunu Tonningen Foundation Grant. His research interest is machine learning applications for wireless communications.

Tanesh Kumar (tanesh.kumar@oulu.fi) earned his M.Sc. degree in computer science from South Asian University, India, in 2014. Currently, he is working as a doctoral researcher at the Centre for Wireless Communications, the University of Oulu, Oulu, Finland. He is working on various national and international research projects related to security and privacy aspects for future Internet of Things (IoT) wireless systems. His research interests include IoT security and privacy, blockchain, edge/fog computing, and Industry 4.0. He is a Student Member of IEEE.

Erkki Harjula (erkki.harjula@oulu.fi) earned his D.Sc. degree in communications engineering from the University of Oulu, Oulu, Finland, in 2016, where he is currently an assistant professor (tenure track) with the Centre for Wireless Communications research group. His research interests include edge computing, Internet of Things, health-care information and communications technology, distributed ledger technologies, artificial intelligence and machine learning. He also has experience as a research project manager. He is a Member of IEEE.

Marcus Meisel (marcus.meisel@tuwien.ac.at) earned his M.Sc. degree in software engineering and Internet computing from TU Wien, Austria. He is currently the product manager for cybersecurity at Sprecher Automation, Linz, Austria. Heading the Energy&IT Group, his responsibilities have included hiring, signing for the funding acquired, and leading research projects in the areas of distributed communication technologies, the Internet of Things, Industry 4.0, general artificial intelligence, and smart energy systems. He is a Member of IEEE.

Markku Juntti (markku.juntti@oulu.fi) earned his D.Sc. (EE) degree from the University of Oulu, Finland, in 1997. Since 2000, he has been a professor of communications engineering at the Centre for Wireless Communications, the University of Oulu, Oulu, Finland, where he also serves as the head of the Radio Technologies Research Unit. He is also an adjunct professor at Rice University, Houston, Texas, where he was a visiting scholar from 1994 to 1995. His research interests include signal processing for wireless networks as well as communication and information theory. He is a Fellow of IEEE.

Thilo Sauter (thilo.sauter@tuwien.ac.at) earned his Ph.D. degree in electrical engineering from TU Wien, Austria, in 1999, where he is currently a professor of automation technology. He was the founding director of the Department for Integrated Sensor Systems at Danube University Krems, Wiener Neustadt, Austria. He is a senior administrative committee member of the IEEE Industrial Electronics Society. His research interests include smart sensors and automation networks with a focus on real-time, security, interconnection, and integration issues. He is a Fellow of IEEE.

Mika Ylianttila (mika.ylianttila@oulu.fi) earned his doctoral degree in communications engineering in 2005 from the University of Oulu, Oulu, Finland, where he is a full-time associate professor (tenure track) at the Centre for Wireless Communications (CWC). He serves as editor of *Wireless Networks* journal and leads the Network Security and Softwarization research group at CWC Networks and Systems research unit. His research interests include edge computing, network security, network virtualization, and software-defined networking. He is a Senior Member of IEEE.

References

[1] K. Ashton, "That "Internet of Things" thing," *RFID J.*, vol. 22, no. 7, pp. 97–114, 2009.

[2] K. L. Lueth, "State of the IoT 2018: Number of IoT devices now at 7B–Market accelerating," IoT Analytics, 2018. https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/#:~:text=The%20number%20of%20connected%20devices,laptops%20or%20fixed%20line%20phones

[3] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006. doi: 10.1162/neco.2006.18.7.1527.

[4] M. G. Kibria, K. Nguyen, G. P. Villardi, O. Zhao, K. Ishizu, and F. Kojima, "Big data analytics, machine learning and artificial intelligence in next-generation wireless networks," *IEEE Access*, vol. 6, pp. 32,328–3,338, May 2018. doi: 10.1109/ACCESS.2018.2837692.

[5] M. Moh and R. Raju, "Machine learning techniques for security of internet of things (IoT) and fog computing systems," in *Proc. 2018 Int. Conf. High Performance Comput. Simul. (HPCS)*, pp. 709–715. doi: 10.1109/HPCS.2018.00116.

[6] T. Park, N. Abuzainab, and W. Saad, "Learning how to communicate in the Internet of Things: Finite resources and heterogeneity," *IEEE Access*, vol. 4, pp. 7063–7073, Nov. 2016. doi: 10.1109/ACCESS.2016.2615643.

[7] J. Park, S. Samarakoon, M. Bennis, and M. Debbah, "Wireless network intelligence at the edge," *Proc. IEEE*, vol. 107, no. 11, pp. 2204–2239, Nov. 2019. doi: 10.1109/JPROC.2019.2941458.

[8] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep learning for IoT big data and streaming analytics: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2923–2960, Fourthquarter 2018. doi: 10.1109/COMST.2018.2844341.

[9] O. B. Sezer, E. Dogdu, and A. M. Ozbayoglu, "Context-aware computing, learning, and big data in Internet of Things: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 1–27, Feb. 2018. doi: 10.1109/JIOT.2017.2773600.

[10] C. Jiang, H. Zhang, Y. Ren, Z. Han, K. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Commun.*, vol. 24, no. 2, pp. 98–105, Apr. 2017. doi: 10.1109/MWC.2016.1500356WC.

[11] A. L'heureux, K. Grolinger, H. F. Elyamany, and M. A. M. Capretz, "Machine learning with big data: Challenges and approaches," *IEEE Access*, vol. 5, pp. 7776–7797, Apr. 2017. doi: 10.1109/ACCESS.2017.2696365.

[12] H. Li, K. Ota, and M. Dong, "Learning IoT in edge: Deep learning for the Internet of Things with edge computing," *IEEE Netw.*, vol. 32, no. 1, pp. 96–101, Jan. 2018. doi: 10.1109/MNET.2018.1700202.

[13] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," 2016, arXiv:1610.05492.

[14] P. Sun, Y. Wen, T. N. B. Duong, and S. Yan, "Timed dataflow: Reducing communication overhead for distributed machine learning systems," in *Proc. IEEE 22nd Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Dec. 2016, pp. 1110–1117. doi: 10.1109/ICPADS.2016.0146.

[15] S. Wang et al., "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1–16, 2019. doi: 10.1109/JSAC.2019.2904348.

[16] Y. Arjevani and O. Shamir, "Communication complexity of distributed convex learning and optimization," in *Proc. 28th Int. Conf. Neural Inform. Process. Syst. (NIPS'15)*, vol. 1. Cambridge, MA: MIT Press, 2015, pp. 1756–1764.

[17] Z. M. Fadlullah et al., "On intelligent traffic control for large scale heterogeneous networks: A value matrix based deep learning approach," *IEEE Commun. Lett.*, vol. 22, no. 12, pp. 2479–2482, 2018. doi: 10.1109/LCOMM.2018.2875431.

[18] I. Ahmad et al., "Towards gadget-free internet services: A roadmap of the naked world," *Telematics Inform.*, vol. 35, no. 1, pp. 82–92, Apr. 2018. doi: 10.1016/j.tele.2017.09.020.

[19] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training deep neural networks for wireless resource management," in *Proc. IEEE 18th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, July 2017, pp. 1–6. doi: 10.1109/SPAWC.2017.8227766.

[20] C. Augenstein, N. Spangenberg, and B. Franczyk, "Applying machine learning to big data streams: An overview of challenges," in *Proc. IEEE 4th Int. Conf. Soft Comput. Mach. Intell. (ISCMI)*, Nov. 2017, pp. 25–29. doi: 10.1109/ISCMI.2017.8279592.

[21] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, "Federated learning for ultra-reliable low-latency V2V communications," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–7. doi: 10.1109/GLOCOM.2018.8647927.

[22] J. Tang, D. Sun, S. Liu, and J.-L. Gaudiot, "Enabling deep learning on IoT devices," *Computer*, vol. 50, no. 10, pp. 92–96, 2017. doi: 10.1109/MC.2017.3641648.

[23] S. Suthaharan, "Big data classification: Problems and challenges in network intrusion prediction with machine learning," *SIGMETRICS Perform. Eval. Rev.*, vol. 41, no. 4, pp. 70–73, Apr. 2014. doi: 10.1145/2627534.2627557.

[24] B. A. A. Nunes, M. Mendonca, X. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 3, pp. 1617–1634, Third 2014. doi: 10.1109/SURV.2014.012214.00180.

[25] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0," *IEEE Ind. Electron. Mag.*, vol. 11, no. 1, pp. 17–27, Mar. 2017. doi: 10.1109/MIE.2017.2649104.

[26] P. Schulz et al., "Latency critical IoT applications in 5G: Perspective on the design of radio interface and network architecture," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 70–78, Feb. 2017. doi: 10.1109/MCOM.2017.1600435CM.

[27] N. Kato et al., "The deep learning vision for heterogeneous network traffic control: Proposal, challenges, and future perspective," *IEEE Wireless Commun.*, vol. 24, no. 3, pp. 146–153, June 2017. doi: 10.1109/MWC.2016.1600317WC.

[28] B. Mao et al., "Routing or computing? The paradigm shift towards intelligent computer network packet transmission based on deep learning," *IEEE Trans. Comput.*, vol. 66, no. 11, pp. 1946–1960, Nov. 2017. doi: 10.1109/TC.2017.2709742.

[29] J. H. Jafarian, E. Al-Shaer, and Q. Duan, "An effective address mutation approach for disrupting reconnaissance attacks," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 12, pp. 2562–2577, Dec. 2015. doi: 10.1109/TIFS.2015.2467358.

[30] M. Suchara, D. Xu, R. Doverspike, D. Johnson, and J. Rexford, "Network architecture for joint failure recovery and traffic engineering," in *Proc. ACM SIGMETRICS Joint Int. Conf. Measurement Modeling Comput. Syst., ser. SIGMETRICS '11*. New York: ACM, 2011, pp. 97–108. doi: 10.1145/1993744.1993756.

[31] M. Zorzi, A. Zanella, A. Testolin, M. D. F. D. Grazia, and M. Zorzi, "Cognition-based networks: A new perspective on network optimization using learning and distributed intelligence," *IEEE Access*, vol. 3, pp. 1512–1530, Aug. 2015. doi: 10.1109/ACCESS.2015.2471178.

[32] S. T. V. Pasca, S. S. P. Kodali, and K. Kataoka, "AMPS: Application aware multipath flow routing using machine learning in SDN," in *Proc. 23rd Nat. Conf. Commun. (NCC)*, Mar. 2017, pp. 1–6. doi: 10.1109/NCC.2017.8077095.

[33] I. Ahmad, S. Namal, M. Ylianttila, and A. Gurtov, "Security in software defined networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2317–2346, Fourthquarter 2015. doi: 10.1109/COMST.2015.2474118.

[34] D. Staessens, S. Sharma, D. Colle, M. Pickavet, and P. Demeester, "Software defined networking: Meeting carrier grade requirements," in *Proc. 18th IEEE Workshop Local Metropolitan Area Netw. (LANMAN)*, Oct. 2011, pp. 1–6. doi: 10.1109/LANMAN.2011.6076935.

[35] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. M. Leung, "Cache in the air: Exploiting content caching and delivery techniques for 5G systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 131–139, Feb. 2014. doi: 10.1109/MCOM.2014.6736753.

[36] Z. Chang, L. Lei, Z. Zhou, S. Mao, and T. Ristaniemi, "Learn to cache: Machine learning for network edge caching in the big data era," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 28–35, June 2018. doi: 10.1109/MWC.2018.1700317.

[37] W. Liu, J. Zhang, Z. Liang, L. Peng, and J. Cai, "Content popularity prediction and caching for ICN: A deep learning approach with SDN," *IEEE Access*, vol. 6, pp. 5075–5089, 2018. doi: 10.1109/ACCESS.2017.2781716.

[38] T. Norp, "5G service requirements," 3rd Generation Partnership Project (3GPP), Sophia Antipolis, France, Feb. 27, 2017. [Online]. Available: https://www.3gpp.org/news-events/1831-sa1_5g

[39] N. P. Jouppi et al., "In-datacenter performance analysis of a tensor processing unit," in *Proc. ACM/IEEE 44th Annu. Int. Symp. Comput. Archit. (ISCA)*, June 2017, pp. 1–12. doi: 10.1145/3079856.3080246.

[40] H.-S. Ham, H.-H. Kim, M.-S. Kim, and M.-J. Choi, "Linear SVM-based android malware detection for reliable IoT services," *J. Appl. Math.*, vol. 2014, Sept. 2014, Art. no. 594501. doi: 10.1155/2014/594501.

[41] H. H. Pajouh, R. Javidan, R. Khayami, A. Dehghantanha, and K. R. Choo, "A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in IoT backbone networks," *IEEE Trans. Emerg. Topics Comput.*, vol. 7, no. 2, pp. 314–323, Apr. 2019. doi: 10.1109/TETC.2016.2633228.

[42] J. Canedo and A. Skjellum, "Using machine learning to secure IoT systems," in *Proc. 2016 14th Annu. Conf. Privacy, Security Trust (PST)*, pp. 219–222. doi: 10.1109/PST.2016.7906930.

[43] L. Xiao, X. Wan, X. Lu, Y. Zhang, and D. Wu, "IoT Security techniques based on machine learning: How do IoT devices use ai to enhance security?" *IEEE Signal Process. Mag.*, vol. 35, no. 5, pp. 41–49, Sept. 2018. doi: 10.1109/MSP.2018.2825478.

[44] Z. Guan, L. Bian, T. Shang, and J. Liu, "When machine learning meets security issues: A survey," in *Proc. 2018 IEEE Int. Conf. Intell. Safety Robot. (ISR)*, pp. 158–165. doi: 10.1109/IISR.2018.8535799.

[45] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *Proc. IEEE Symp. Security Privacy*, May 2010, pp. 305–316. doi: 10.1109/SP.2010.25.

[46] E. Bastug, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5G wireless networks," *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 82–89, Aug. 2014. doi: 10.1109/MCOM.2014.6871674.

[47] A. Holzinger, P. Kieseberg, E. Weippl, and A. M. Tjoa, "Current advances, trends and challenges of machine learning and knowledge extraction: From machine learning to explainable AI," in *Machine Learning and Knowledge Extraction*, A. Holzinger, P. Kieseberg, A. M. Tjoa, and E. Weippl, Eds. Cham: Springer-Verlag, 2018, pp. 1–8.

[48] A. Bittau et al., "Prochlo: Strong privacy for analytics in the crowd," in *Proc. 26th Symp. Oper. Syst. Principles, ser. SOSP '17*. New York: ACM, 2017, pp. 441–459. doi: 10.1145/3132747.3132769.

[49] I. Ahmad, S. Shahabuddin, T. Kumar, J. Okwuibe, A. Gurtov, and M. Ylianttila, "Security for 5G and beyond," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3682–3722, Fourthquarter 2019. doi: 10.1109/COMST.2019.2916180.

[50] M. Bennis, S. M. Perlaza, P. Blasco, Z. Han, and H. V. Poor, "Self-organization in small cell networks: A reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 12, no. 7, pp. 3202–3212, July 2013. doi: 10.1109/TWC.2013.060513.120959.

[51] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017. doi: 10.1109/TCCN.2017.2758370.

[52] S. Shahabuddin, M. Juntti, and C. Studer, "ADMM-based infinity norm detection for large MU-MIMO: Algorithm and VLSI architecture," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2017, pp. 1–4. doi: 10.1109/ISCAS.2017.8050311.

[53] H. B. Demuth, M. H. Beale, O. De Jess, and M. T. Hagan, *Neural Network Design*. Martin Hagan, 2014.

[54] N. Samuel, T. Diskin, and A. Wiesel, "Deep MIMO detection," in *Proc. IEEE 18th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, July 2017, pp. 1–5. doi: 10.1109/SPAWC.2017.8227772.

[55] J. W. Branch, C. Giannella, B. Szymanski, R. Wolff, and H. Kargupta, "In-network outlier detection in wireless sensor networks," *Knowl. Inform. Syst.*, vol. 34, no. 1, pp. 23–54, Jan. 2013. doi: 10.1007/s10115-011-0474-5.

[56] S. Wang et al., "When edge meets learning: Adaptive control for resource-constrained distributed machine learning," in *Proc. IEEE INFOCOM 2018 - IEEE Conf. Comput. Commun.*, Apr. 2018, pp. 63–71. doi: 10.1109/INFOCOM.2018.8486403.

[57] J. A. C. Soto, M. Jentsch, D. Preuveneers, and E. Ilie-Zudor, "CEML: Mixing and moving complex event processing and machine learning to the edge of the network for IoT applications," in *Proc. 6th Int. Conf. Internet Things, ser. IoT'16*. New York: ACM, 2016, pp. 103–110. doi: 10.1145/2991561.2991575.

[58] E. Ramos and R. Morabito, "Intelligence stratum for IoT. Architecture Requirements and Functions," 2019, arXiv:1908.08921.

[59] A. C. Baktir, A. Ozgovde, and C. Ersoy, "How can edge computing benefit from software-defined networking: A survey, use cases, and future directions," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2359–2391, Fourthquarter 2017. doi: 10.1109/COMST.2017.2717482.

[60] J. Kwak, Y. Kim, L. B. Le, and S. Chong, "Hybrid content caching in 5G wireless networks: Cloud versus edge caching," *IEEE Trans. Wireless Commun.*, vol. 17, no. 5, pp. 3030–3045, May 2018. doi: 10.1109/TWC.2018.2805893.

[61] M. Frustaci, P. Pace, G. Aloi, and G. Fortino, "Evaluating critical security issues of the IoT world: Present and future challenges," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2483–2495, Aug. 2018. doi: 10.1109/JIOT.2017.2767291.

[62] M. Abadi and D. G. Andersen, "Learning to Protect Communications with Adversarial Neural Cryptography," 2016. [Online]. Available: http://arxiv.org/abs/1610.06918

[63] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. D. Tygar, "Adversarial machine learning," in *Proc. 4th ACM Workshop Security Artif. Intell., ser. AISec '11*, New York: ACM, 2011, pp. 43–58.