Thirty Years of Machine Learning: The Road to Pareto-Optimal Wireless Networks

Jingjing Wang, Member, IEEE, Chunxiao Jiang, Senior Member, IEEE, Haijun Zhang, Senior Member, IEEE, Yong Ren, Senior Member, IEEE, Kwang-Cheng Chen, Fellow, IEEE, and Lajos Hanzo, Fellow, IEEE

Abstract—Future wireless networks have a substantial potential in terms of supporting a broad range of complex compelling applications both in military and civilian fields, where the users are able to enjoy high-rate, low-latency, low-cost and reliable information services. Achieving this ambitious goal requires new radio techniques for adaptive learning and intelligent decision making because of the complex heterogeneous nature of the network structures and wireless services. Machine learning (ML) algorithms have great success in supporting big data analytics, efficient parameter estimation and interactive decision making. Hence, in this article, we review the thirty-year history of ML by elaborating on supervised learning, unsupervised learning, reinforcement learning and deep learning. Furthermore, we investigate their employment in the compelling applications of wireless networks, including heterogeneous networks (Het-Nets), cognitive radios (CR), Internet of things (IoT), machine to machine networks (M2M), and so on. This article aims for assisting the readers in clarifying the motivation and methodology of the various ML algorithms, so as to invoke them for hitherto unexplored services as well as scenarios of future wireless networks.

Index Terms—Machine learning (ML), future wireless network, deep learning, regression, classification, clustering, network association, resource allocation.

NOMENCLATURE

5G The 5th Generation Mobile Network
AI Artificial Intelligent
AMC Automatic Modulation Classification
ANN Artificial Neural Network

This work is partly supported by the National Natural Science Foundation of China (61922050), the Pre-research Fund of Equipments of Ministry of Education of China (6141A02022615), the Research Fund of China Academy of Space Technology (CoCo-20180605-47), and also partly supported by the National Natural Science Foundation of China (61822104, 61771044), the Fundamental Research Funds for the Central Universities (RC1631, FRF-TP-19-002C1). Dr. Wang would like to acknowledge the financial support of the Shuimu Tsinghua Scholar Program. (Corresponding author: Chunxiao Jiang)

J. Wang and Y. Ren are with the Department of Electronic Engineering, Tsinghua University, Beijing, 100084, China. E-mail: chinaep-hd@gmail.com, reny@tsinghua.edu.cn.

C. Jiang is with the Tsinghua Space Center, Tsinghua University, Beijing, 100084, China. E-mail: jchx@tsinghua.edu.cn.

H. Zhang is with the Institute of Artificial Intelligence, Beijing Advanced Innovation Center for Materials Genome Engineering, Beijing Engineering and Technology Research Center for Convergence Networks and Ubiquitous Services, University of Science and Technology Beijing, Beijing 100083, China. E-mail: hajunzhang@ieee.org.

K.-C. Chen is with the Electrical Engineering Department, University of South Florida, Tampa, FL 33620, USA. Email: kwangcheng@usf.edu.

L. Hanzo is with the School of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, UK. Email: l-h@ecs.soton.ac.uk.

AP Access Point
AWGN Additive White Gaussian Noise
BBU BaseBand processing Unit
BS Base Station
CDF Cumulative Distribution Function
CNN Convolutional Neural Network
CogNet Cognitive Network
CoMP Coordinated Multiple Points
CR Cognitive Radio
C-RAN Cloud Radio Access Network
CRN Cognitive Radio Network
CSI Channel State Information
CSMA/CA Carrier-Sense Multiple Access with Collision Avoidance
CSMA/CD Carrier-Sense Multiple Access with Collision Detection
C-S Mode Client-Server Mode
D2D Device to Device
DBN Deep Belief Network
DNN Deep Neural Network
DQN Deep Q-Network
EA Energy Awareness
EE Energy Efficiency
EH Energy Harvesting
ELP Exponentially-weighted algorithm with Linear Programming
EM Expectation Maximization
eMBB enhanced Mobile Broad Band
ERM Empirical Risk Minimization
EXP3 EXPonential weights for EXPloration and EXPloitation
FANET Flying Ad Hoc Network
FDA Fisher Discriminant Analysis
FDI False Data Injection
FSMC Finite State Markov Channel
GMM Gaussian Mixture Model
HetNet Heterogeneous Network
HMM Hidden Markov Model
ICA Independent Component Analysis
IEEE Institute of Electrical and Electronics Engineers
IoT Internet of Things
ITS Intelligent Transportation System
KNN K-Nearest Neighbors
LED Light Emitting Diode
LOS Line of Sight
LS Least Square
LSTM Long Short Term Memory
I. Introduction

Wireless networks have supported a variety of military services, intelligent transportation, healthcare, etc. To elaborate briefly, next-generation mobile networks are expected to support high data rate communication [1]. As a complement, wireless sensor networks (WSN) support sustained monitoring in unmanned or hostile environments relying on widely dispersed operating sensors [2]. Furthermore, the popular Wi-Fi network provides convenient Internet access for various devices in indoor scenarios [3]. With the rapid proliferation of portable mobile devices and the demand for a high quality of service (QoS) and quality of experience (QoE), future wireless networks will continue to support a broad range of compelling applications, where the users benefit from high-rate, low-latency, low-cost and reliable information services.

A. Motivation

In contrast to the conventional wireless networks, future wireless networks have the following evolutionary tendency [4], [5]:

• **Network Scale**: The network is associated with a tremendous network size including all kinds of entities, each of which has different service capabilities as well as requirements. Furthermore, interactions among these entities result in a diverse variety of traffic, such as text, voice, audio, images, video, etc.

• **Network Structure**: On one hand, the future wireless network tends to have a self-configuring element, where each entity cooperatively completes tasks. This characteristic is termed as “being ad hoc”. On the other hand, the future wireless network is heterogeneous and hierarchical, having different network slices.

• **Network Control**: Future wireless networks facilitate convenient reconfiguration by software-based network management, hence improving network flexibility and efficiency.

Machine learning (ML) was first introduced as a popular technique of realizing artificial intelligence in the late 1950’s [6]. ML algorithms can learn from training data without being explicitly programmed. It is beneficial for classification/regression, prediction, clustering and decision making [7]–[9], whilst relying on the following three basic elements [10]:

• **Model**: Mathematical or signal models are constructed from training data and expert knowledge, in order to statistically describe the characteristics of the given data set. Then again, relying on these trained models, ML can be used for classification, prediction and

1In our paper, network slices are multiple logical networks running on the top of a shared physical network infrastructure and operated by a control center.
decision making. In case the appropriate models are not available, techniques on the feature extraction or knowledge discovery can be developed to achieve the same goal.

- **Strategy**: The criteria used for training mathematical models are called strategies. How to select an appropriate strategy is closely associated with training data. Empirical risk minimization [11] and structural risk minimization [12] constitute a pair of fundamental strategies, where the latter can beneficially avoid the notorious “over-fitting” phenomenon.

- **Algorithm**: Algorithms are constructed to find solutions based on predetermined model and strategy selected, which can be viewed as an optimization process. A powerful algorithm can find a globally optimal solution with high probability at a low computational complexity and storage.

In the last thirty years, ML has been successfully applied to the field of computer vision [13], automatic control [14], bioinformatics [15], etc. Considering the aforementioned characteristics of future wireless networks, data-driven ML is also likely to become a powerful technique of network association for substantially improving the network performance. This is achieved by accurately learning the near-real-time physical operating scenario, which allows them to outperform the traditional model-driven optimization algorithms based on more-restrictive assumptions detailed in [16]. More specifically,

- The wireless data torrance may be conveniently managed by the big data processing capability of ML [17]. For example, the tele-traffic volume generated by on-demand information and entertainment is predicted to substantially increase over the next decade, and an average smart phone may generate as much as 4.4 GB data per month by the year 2020 [18]–[20]. The massive amount of data constitutes a large training set, which can be statistically exploited for data-mining as well as for classification and for prediction with the aid of ML algorithms.

- Future wireless networks require both individual node intelligence and swarm intelligence [21]. Moreover, as for resource allocation and management, we tend to strike a trade-off among numerous factors, such as the capacity, power consumption, latency, complexity, etc. rather than only considering a single-component objective function. Thanks to learning from trial and error experiments, ML is conducive to supporting intelligent multi-objective decision making in the context of multi-agent collaborative network management. Future wireless networks may hence be expected to benefit from intelligent multi-agent systems.

- **Modeling and parameter estimation** play an important role in wireless networks. For instance, in massive multiple-input and multiple-output (MIMO) systems, an accurate estimate of the channel state information (CSI) potentially allows us to approach the system’s capacity. Given that traditional mathematical models often fail to accurately describe the system in typical time-varying scenarios, ML provides an alternative technique of adaptive modeling and parameter estimation relying on learning from the recorded history.

- Future wireless networks are also expected to take into account the human behavior, for example by taking into account the geographic position of access points (AP) in an ultra dense network (UDN), where user-centric designs have been conceived for reducing the cluster-edge effects. By mimicking human intelligence, ML may be deemed to be the most appropriate tool for adapting the network’s structure to the human behavior observed [22], [23].

Next-generation wireless network optimization relying on ML has emerged as an important research topic, so much so that the standard body ITU-T has formed a dedicated focus group to study this subject from 2018 to 2020. When incorporating ML functionalities into the network architecture, there are two fundamental mechanisms of exploiting ML algorithms, namely online and offline ML. The online ML family represents ML functionalities embedded into networking algorithms or protocols. By contrast, offline ML may be executed by a co-located computing facility connected to the corresponding network entities. However, offline ML can also be supported by remote computing facilities.

In recent years, a range of surveys have been conceived on ML paradigms. Some of them focused their scope on a specific wireless scenario, such as WSNs [24], [25], cognitive radio networks (CRN) [26]–[28], Internet of Things (IoT) [29], wireless ad hoc networks (WANET) [30], self-organizing cellular networks [31], etc. Specifically, Alsheikh et al. [24] provided an extensive overview of ML methods applied to WSNs which improved the resource exploitation and prolonged the lifespan of the network. Kulkarni et al. [25] surveyed some common issues of WSNs solved by computational intelligence algorithms, such as data fusion, routing, task scheduling, localization, etc. Moreover, Bkassiny et al. [26] investigated decision-making and feature classification problems solved by both centralized and decentralized learning algorithms in CRN in a non-Markovian environment. Gavrilovska et al. [27] studied the nature of the CRN’s capability of reasoning and learning. Park et al. [29] reviewed a range of learning aided frameworks designed for adapting to the heterogeneous resource-constrained IoT environment. Forster [30] portrayed the advantages of using ML for the data routing problem of WANETs. Furthermore, a detailed literature review of the past fifteen years of ML techniques applied to self-configuration, self-optimization and self-healing, was provided by Klaine et al. [31].

Some of the literature were restricted to a specific application [32]–[38], whilst others considered a single learning technique [39]–[44]. To elaborate, Al-Rawi et al. [32] presented an overview of the features, methods and performance enhancement of learning-assisted routing schemes in the context of distributed wireless networks. Additionally, Fadlullah et al. [33] provided an overview of the state-of-the-art in learning aided network
B. Contributions

In Fig. 1 and Table I to this survey. Wireless networks survey and tutorial papers are contrasted problems. The main contributions of the existing ML aided ML algorithms conceived for various wireless networking al. [44] overviewed the artificial neural networks based of different network layers, including physical layer, data- et al. [43] provided a comprehensive survey of the artificial neural networks based ML algorithms conceived for various wireless networking problems. The main contributions of the existing ML aided wireless networks. Some future research ideas and our conclusions are summarized at a glance in Fig. 2.

C. Organization

The remainder of this article is outlined as follows. In Section II, we provide a brief overview of the history of ML and of the development of wireless networks. In Section III, we introduce a range of typical supervised learning algorithms and highlight their compelling applications in wireless networks. In Section IV, we investigate the family of unsupervised learning algorithms and their related applications. Some popular reinforcement learning algorithms are elaborated on in Section V. Moreover, we present two examples of how these reinforcement learning algorithms can improve the performance of wireless networks. In Section VI, we introduce some typical deep learning algorithms and their applications in wireless networks. Some future research ideas and our conclusions are provided in Section VII. The structure of this treatise is summarized at a glance in Fig. 2.

II. A BRIEF OVERVIEW OF MACHINE LEARNING AND WIRELESS NETWORKS

A. The Thirty-Year Development of Machine Learning

The term “machine learning” was first proposed by Arthur Samuel in 1959 [6], which referred to computer systems having the capability of learning from their large amounts of previous tasks and data, as well as of self-optimizing computer algorithms. Hard-programmed algorithms are difficult to adapt to dynamically fluctuating demands and constantly renewed system states. By contrast, relying on learning from previous experiences, ML aided algorithms are beneficial for scientific decision making and task prediction, which is achieved by constructing a self-adaptation model from sample inputs. To elaborate a little further, as for the concept of “learning”, Tom M.

<table>
<thead>
<tr>
<th>Application-oriented</th>
<th>ML-oriented</th>
</tr>
</thead>
<tbody>
<tr>
<td>[24],[29] resource allocation</td>
<td>[27] capability of learning</td>
</tr>
<tr>
<td>[33] traffic control</td>
<td>[39] unsupervised learning</td>
</tr>
<tr>
<td>[26],[34],[37] traffic classification</td>
<td>[44] artificial neural networks</td>
</tr>
<tr>
<td>[35] intrusion detection</td>
<td>[40] reinforcement learning</td>
</tr>
<tr>
<td>[36] software defined networking</td>
<td>[41],[42],[43] deep learning</td>
</tr>
<tr>
<td>[31],[38] networking techniques</td>
<td></td>
</tr>
</tbody>
</table>
Mitchell [48] provided the widely quoted description: “A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.”

ML began to flourish in the 1990s [8]. Before this era, logic- and knowledge-based schemes, such as inductive logic programming, expert systems, etc. dominated the artificial intelligence scene relying on high-level human-readable symbolic representations of tasks and logic. Thanks to the development of statistics theory and stochastic approximation, ML schemes regained researchers’ attention leading to a range of beneficial probabilistic models. Researchers embarked on creating data-driven programs for analyzing a large amount of data and tried to draw conclusions or to learn from the data. During this era, ML algorithms such as neural networks as well as kernel methods became mature. During the 2000s, researchers gradually renewed their interest in deep learning with the aid of the advances in hardware-based computational capability, which made ML indispensable for supporting a wide range of services and applications.

Given the development of progressive learning techniques [49], at present, the research focus of ML has shifted from “learning being the purpose” to “learning being the method”. Specifically, ML algorithms no longer blindly pursue to imitate the learning capability of human beings, instead they focus more on the task-oriented intelligent data-driven analysis. Nowadays, thanks to the abundance of raw data and to the frequent interaction between exploration and exploitation, ML algorithms have prospered in the fields of computer vision, data mining, intelligent control, etc. Future wireless networks aim for providing ubiquitous information services for users in a variety of scenarios. However, the rapid growth in the number of users and the resulted explosive growth of teletraffic data pushes the limits of network-capacity. As a remedy, ML aided network management and control can be viewed as a corner stone of future wireless networks in view of their limited power, spectrum and cost.

B. Classifying Machine Learning Techniques

1) A General Taxonomy: Again, depending on how training data is used, ML algorithms can be grouped into three categories, i.e. supervised learning, unsupervised learning and reinforcement learning [50], [51]. In the following, we will provide a brief description of the three types of algorithms.

- Supervised Learning: The algorithms are trained on a certain amount of labeled data [45]. Both the input data and its desired label are known to the computer, resulting in a data-label pair. Their goal is to infer a function that maps the input data to the output label relying on the training of sample data-label pairs. Specifically, considering a set of $N$ sample data-label pairs in the form of $\{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$, where $x_n$ is the $n$-th sample input data and $y_n$ represents its label. Let $\mathbb{X} = \{x_1, x_2, \ldots, x_N\}$ denote the input data set and $\mathbb{Y} = \{y_1, y_2, \ldots, y_N\}$ represent...
Fig. 3: A comprehensive survey of ML algorithms.
of the literature, semi-supervised learning is listed as a separate member of supervised learning algorithms. However, in some cases they have been conceived for revealing the underlying features of $N$ input data, namely density estimation [53] as well as feature extraction [54]. To elaborate, density estimation aided methods are characterized by explicitly building statistical models of how the underlying features might create the input. By contrast, feature extraction based techniques aim for directly extracting statistical regularities or even sometimes irregularities from the input data set.

- **Reinforcement Learning:** In contrast to the aforementioned two learning techniques, reinforcement learning algorithms are conceived for decision making by learning from interaction with the environment, which are trained by the data on the basis of trial and error [47], [55]. They neither try to identify a category as supervised learning algorithms do, nor do they aim for finding hidden structures as unsupervised learning algorithms do. Specifically, at each time step, the system or environment is in some state $S$, and the agent selects a legitimate action $A$. The system responds at the next time step by moving into a new state $S'$ with a certain probability influenced both by the specific action chosen as well as by the system’s inherent transitions. Meanwhile, the agent receives a corresponding reward $r(S, A)$ from the system, as time evolves. Reinforcement learning algorithms aim for learning how to map situations $S$ into actions $A$ in order to attain the maximal cumulative weighted reward within the horizon in such a closed-loop fashion.

2) **Deep Learning:** As an important member of the ML family, deep learning has been booming since 2010, because it was found to be capable of handling the soaring growth of training data volume facilitated by the rapid development of computing hardware [56], [57]. Deep learning algorithms rely on a multiple-layer “network” consisting of inter-connected nodes for feature extraction and transformation, which is inspired by the biological nervous system, namely the neural network. Each layer utilizes the output of the previous layer as its input. The term “deep” refers to having multiple layers in the network. Generally, relying on the way the training data is exploited, deep learning algorithms can also be classified into deep supervised learning, deep unsupervised learning as well as deep reinforcement learning [56]. Moreover, some deep learning architectures, such as deep neural networks (DNN) [58], deep belief networks (DBN) [59], recurrent neural networks (RNN) with LSTM units [60] and convolutional neural networks (CNN) [61], have had success in a range of fields including computer vision, natural language processing (NLP), speech recognition, etc. They have also been invoked in compelling applications of wireless networks.

Fig. 3 comprehensively summarizes ML algorithms even including above-mentioned supervised learning, unsupervised learning, reinforcement learning and deep learning algorithms as well as some non-mainstream branches [62]-[124]. Furthermore, Fig. 4 shows the involvement of ML in wireless networks based on the aforementioned four categories. Below we list a variety
of popular learning algorithms and highlight their applications in future wireless networks.

C. Development of Wireless Networks

1) The Birth and Development of Wireless Networks: Just as the terminology implies, wireless networks connect various network nodes via electromagnetic waves. Relying on their coverage, wireless networks can be roughly classified into four categories, such as wireless personal area networks (WPAN) [125], wireless local area networks (WLAN) [126], wireless metropolitan area networks (WMAN) [127] and wireless wide area networks (WWAN) [128]. Correspondingly, a family of networking standards and their variants that cover most of the physical layer specifications have been established by the IEEE 802 Working groups, including the IEEE 802.15 for WPAN, IEEE 802.11 for WLAN, IEEE 802.16 for WMAN and IEEE 802.20 for WWAN standards. Furthermore, when considering the network’s functions, some popular representatives of wireless networks include cellular networks [129], WSNs [130], WANETs [131], wireless body area networks (WBAN) [132], etc.

The first wireless network, namely ALOHANET, was developed at the University of Hawaii in 1969 and came into operation in 1971, which for the first time transmitted wireless data packets over a network [133]. The first commercial wireless network was the WaveLAN product family designed by the NCR Corporation in 1986. In 1997, the first IEEE 802.11 protocol was released for WLAN [126]. Afterwards, the emergence and progress of reliable and low-cost Wi-Fi marked the maturity of wireless networking technologies at the end of the 20th century, which facilitated Internet access for a range of Wi-Fi compatible devices including personal computers, smart phones, etc. Future wireless networks aim for providing high-rate, low-latency, full-coverage and low-cost yet reliable information services. Compared to traditional wireless networks connecting humans and their devices, future
wireless networks are expected to interconnect everything under the umbrella of the ‘Internet of Everything’. Fig. 5 demonstrates the development of wireless networks in terms of their milestone techniques.

Wireless networks have evolved from the simple client-server (C-S) mode to the distributed dense multi-layer C-S mode, and finally to the ad hoc peer-to-peer (P2P) mode. The decentralization of network architectures grant more freedom both for the network nodes and their protocols, which requires more sophisticated techniques for supporting efficient and reliable implementations. Furthermore, the soaring growth of both the type and the amount of data provides a promising field of applications for ML algorithms, which are beneficial for self-organized and self-adaptive network architectures.

2) Pareto-Optimal Future Wireless Networks: The challenging real-world optimization problems encountered in future wireless networks have to meet multiple objectives in order to arrive at an attractive solution [134]. In contrast to conventional single-objective optimization where we find the global optimum relying on a single metric, multi-objective optimization aims for finding the globally optimal solution relying on the notion of Pareto optimality [135]. The aim of multi-objective optimization in wireless networks is that of generating a diverse set of Pareto-optimal solutions, where by definition it is only possible to improve any of the metrics considered at the cost of degrading at least one of the others. The collection of Pareto-optimal points is referred to as the Pareto front. Fig. 6 shows a simple Pareto optimization problem having a pair of objective parameters, where the light blue circles represent legitimate operating points, while the dark blue circles denote the approximated Pareto optimal points, which are not dominated by any other solutions. The arrow intimates that the complexity of optimization is gradually increased as we gradually approach the full-search complexity, which defines the ‘true Pareto front’ of Fig. 6.

More specifically, let us briefly reminisce by recalling the past few decades of wireless history. Explicitly, the wireless community has invested decades of research efforts into making near-capacity single-user operation a reality [136], which is however only possible at the cost of an ever-increasing delay, complexity and power consumption. When designing a powerful wireless network, which includes the physical layer, MAC layer, network layer, transmission layer as well as application layer, we face substantial challenges, since we often have to meet conflicting design objectives. Moreover, these conflicting optimization metrics are often coupled with each other. It may also be a substantial challenge to carry out a fair comparison among locally optimal solutions, hence we often strike a tradeoff [137]. For example, in future wireless networks we would like to be more ambitious than ‘only’ optimize the network’s capacity - for delay-sensitive services we would like to reduce the latency and/or reduce the total energy consumption, as well as to improve the system’s reliability and the user’s QoS. By contrast, in wireless sensor networks we may concentrate on optimizing both the connectivity and the network’s life time, just to name a few of these conflicting practical design objectives.

In this context, the family of ML techniques may be viewed as an attractive set of optimization tools for finding Pareto-optimal solutions of multi-objective optimization problems in future wireless networks, which tend to have a large search-space. To expound a little further, it is plausible that every time we incorporate an additional parameter into the objective function, the search-space is expanded and the surface of optimal solutions may exhibit numerous locally optimal solutions. Hence traditional gradient-based techniques routinely fail to find the global optimum. In this context Fig. 7 portrays some popular metrics commonly used in constructing multi-objective optimization problems in future wireless networks.

D. Representative Techniques in Wireless Networks

As shown in Fig. 8, we first of all portray the representative application scenarios and techniques of future wireless networks. In the following, we will briefly introduce a
range of compelling techniques and their development trends in future wireless networks, which is summarized in Fig. 9.

1) From MIMO to Massive MIMO: The MIMO technology relying on multiple antennas in both the transmitter and receiver can be viewed as a breakthrough in terms of multiplying the capacity of a radio link compared to the single-transmit single-receive antenna aided wireless system having a variety of cost, technology and regulatory constraints [138]. Both single-user MIMO (SU-MIMO) and Multi-user MIMO (MU-MIMO) schemes have been proposed. To elaborate, multiple data streams of the same source are sent to a single user in SU-MIMO, while a transmitter simultaneously serves multiple users on the same channel resource in MU-MIMO [139], [140].

Generally, invoking more antennas is beneficial in terms of improving the data rate and/or link reliability [4], [141]–[143]. However, the performance degradation caused by inaccurate CSI and the high computational complexity of channel estimation constitute substantial challenges [144].

2) From D2D, M2M to IoT: In the spirit of direct communication between nearby mobile devices without traversing base stations (BS) or core networks, device-to-device (D2D) communication networks have been widely investigated in recent years, which can be deemed to be important milestones on the road towards self-organization and P2P collaboration. In D2D networks, the same resource slots can be reused both by the D2D links as well as the cellular links, which is capable of substantially improving the network capacity. Moreover, it is potentially beneficial in terms of enhancing the energy efficiency (EE), also reducing the transmission delay and improving the network’s fairness to users [145], [146], which is also closely related to machine-to-machine (M2M) communications. The corresponding massive machine type of communication (mMTC) [147] mode of the 5G network is capable of supporting sensing, transmitting, fusion and processing sensory data. Furthermore, M2M is also capable of supporting the smart home [148], smart grid [149], etc.

In 2015, International Telecommunication Union (ITU) officially defined three application scenarios of 5G network, i.e. enhanced mobile broad band (eMBB), massive machine type communication (mMTC) and ultra reliable low latency communication (uRLLC).
Aiming for “connecting everything”, IoT was first defined for enabling objects to connect and exchange data in 1999 [150]. Furthermore, the IoT allows objects to be sensed and controlled remotely, creating opportunities for direct interaction between the physical world and computer-based virtual systems, which is beneficial in terms of improving operational efficiency and of reducing human intervention. Both WSNs and M2M communications can be viewed as a part of the IoT. Although the IoT faces a range of reliability, robustness and security challenges, there is no doubt that it will make our world ever smarter [151], [152].

3) From UDN to HetNet: In order to meet the demand of supporting massive data traffic, the so-called UDN architecture has been defined where the density of BSs or APs potentially reaches or even exceeds the density of users [153], [154]. The UDN architecture is conducive to increasing the network capacity as well as simultaneously improving the user experience. However, the interference encountered in UDNs tends to be more severe and of higher volatility than that in traditional cellular networks because of the dense deployment of BSs and APs. Hence, the joint consideration of resource allocation, interference management and traffic routing are essential for UDNs [129], [155].

Considering a wide area network scenario, heterogeneous networks (HetNet) are characterized by the employment of multiple types of radio access technologies (RAT) [156]. Upon combining macrocells, microcells, picocells [157] and femtocells [158], [159], HetNets are capable of providing a seamless wireless coverage ranging from outdoor environments to office buildings and even to underground areas by selecting another RAT when a RAT fails, and HetNets can also provide load-balancing in the face of non-uniform spatial distribution of users [160].

4) From DBS to C-RAN: Compared to the traditional BS, which integrates baseband processing units (BBU) and remote radio units (RRU) in a single cabinet, distributed base station (DBS) aided systems separate the BBU as well as the RRU and connects them with optical fiber. The DBS system allows more flexibility in network planning and deployment, where RRUs can be placed a few hundred meters or a few kilometres away for enhancing network’s edge-coverage.

Cloud-radio access networks (C-RAN) can be viewed as an evolution of the aforementioned DBS system, which is a centralized processing and cloud computing aided radio access network architecture [161]. The principle of C-RAN relies on gathering the BBUs from several BSs into a centralized BBU pool, whilst allowing hundreds of RRUs to connect to the centralized BBU pool [162]. Hence, resources can be allocated to each user based on joint dynamic scheduling. By exploiting coordination and virtualization, the spectral efficiency (SE), the system’s flexibility and the load balancing capability are substantially improved. Moreover, the centralized management of resources reduces the cost of the system’s operation and maintenance.

4In some works, RRU is also called remote radio head (RRH)
5) From SDN to NFV: Software-defined networking (SDN) is employed as a programmable network architecture in order to achieve cost-effective dynamic network configuration and monitoring [163], [164]. The SDN philosophy suggests to centralize network intelligence in a single network component by decoupling the control plane and the data plane, which disassociates network control and its forwarding functions. The two planes can communicate with the aid of the OpenFlow protocol\(^5\), and the network resources can be managed logically and efficiently. A SDN connects decentralized users to cloud computing through a “network pipeline” [165] [166].

Relying on IT virtualization techniques, network function virtualization (NFV) transforms the entire set of network node functions into different building blocks, which separates the networking functions from specific hardware blocks [167]. Hence, NFV is eminently suitable for service diversification and promotes the standardization of networking equipment [168]. Explicitly, NFV can be viewed as a beneficial hardware-agnostic design in the application layer of SDN architectures.

6) From EH to EA: Energy harvesting (EH) is an environmentally friendly process, which captures and stores ambient energy, such as solar power, thermal energy, wind energy, etc. for low-power wireless devices [169], especially in WSNs and WBANs, for example.

In future wireless networks, energy optimization is a significant concern motivated by mitigating climate change. However, energy consumption is related to both the network’s throughput and to its entire lifetime with a trade-off between them. As a remedy, instead of only focusing on EH, energy awareness (EA) at every stage of the network’s design and management is the most promising approach to striking a trade-off amongst the conflicting objectives of reducing energy consumption, improving the system’s throughput as well as prolonging its lifetime, especially in energy-constrained networks [170], [171].

7) From CR to CogNet: Cognitive radio (CR) constitutes a technique that allows us to dynamically and efficiently exploit the wireless spectral resources [172]–[175]. By relying on spectrum sensing, CR is capable of achieving dynamic spectrum access and spectrum sharing. Specifically, in the process of spectrum sensing, the secondary user (SU) detects an empty slicer of spectrum, for example, based on energy detection schemes. Then, in the process of spectrum access, power control is invoked by the SU for maximizing its capacity, whilst observing the interference power constraint in order to protect the primary user (PU). As a benefit, CR dynamically and flexibly exploits the scarce wireless spectral resources, hence substantially improving the spectrum efficiency [176].

In contrast to CR techniques, which only deal with the issues of physical-layer spectrum sensing and data link-layer access, cognitive networks (CogNet) are characterized by a cognitive cross-layer process according to their end-to-end goals, where the overall network conditions are monitored, and then decisions are made based on

\(^5\)The OpenFlow protocol is a communication protocol that gives access to the forwarding plane of a switcher or router over the network.

III. SUPERVISED LEARNING IN WIRELESS NETWORKS

Having covered the networking basics, in this section, we will introduce some rudimentary supervised learning algorithms, such as regression, K-nearest neighbors (KNN), support vector machines (SVM) and Bayes classification including their applications in wireless networks. Table II summarizes some of the typical applications of the above-mentioned four supervised learning algorithms in wireless networks.

A. Regression and Its Applications

1) Methods: Regression analysis is capable of estimating the relationships among variables. Relying on modeling the functional relationship between a dependent variable (objective) and one or more independent variables (predictors), regression constitutes a powerful statistical tool of predicting and forecasting a continuous-valued objective given a set of predictors.

In regression analysis, there are three variables, namely the

- Independent variables (predictors): \(X\)
- Dependent variable (objective): \(Y\)
**Other unknown parameters** that affect the estimated value of the dependent variable: \( \varepsilon \)

The regression function \( f \) models the functional \( Y \) vs \( X \) relationship perturbed by \( \varepsilon \), which can be formulated as: \( Y = f(X, \varepsilon) \). Usually, we characterize this relationship in terms of a specific regression function with the aid of its probability distribution. Moreover, the approximation is often modeled as \( E = [Y \mid X] = f(X, \varepsilon) \). When conducting regression analysis, first of all we have to determine the specific form of the regression function \( f \), which relies on both the common knowledge about the dependent vs independent variables as well as on its convenient evaluation. Based on the specific form of regression function, regression analysis methods can be classified as ordinary linear regression [189], logistic regression [190], polynomial regression [191], etc.

In linear regression, the dependent variable is a linear combination of the independent variables or unknown parameters. Let us assume having \( N \) random training samples and \( M \) independent variables, formulated as \( \{y_n, x_{n1}, x_{n2}, \ldots, x_{nM}\}; n = 1, 2, \ldots, N \). Then the linear regression function can be formulated as:

\[
y_n = \varepsilon_0 + \varepsilon_1 x_{n1} + \varepsilon_2 x_{n2} + \cdots + \varepsilon_M x_{nM} + e_n,
\]

(1)

where \( \varepsilon_0 \) is termed as the regression intercept, while \( e_n \) is the error term and \( n = 1, 2, \ldots, N \). Hence, Eq. (1) can be rewritten in the form of a matrix as \( \mathbf{y} = \mathbf{X} \varepsilon + \mathbf{e} \), where \( \mathbf{y} = [y_1, y_2, \ldots, y_N]^T \) is an observation vector of the dependent variable and \( \mathbf{e} = [e_1, e_2, \ldots, e_N]^T \), while \( \varepsilon = [\varepsilon_0, \varepsilon_2, \ldots, \varepsilon_M]^T \) and \( \mathbf{X} \) represents the observation matrix of independent variables, given by:

\[
\mathbf{X} = \begin{bmatrix}
1 & x_{11} & \cdots & x_{1M} \\
1 & x_{21} & \cdots & x_{2M} \\
\vdots & \vdots & \ddots & \vdots \\
1 & x_{N1} & \cdots & x_{NM}
\end{bmatrix}
\]

Linear regression analysis [189] aims for estimating the unknown parameter \( \hat{\varepsilon} \) relying on the least squares (LS) criterion. The corresponding solution can be expressed as:

\[
\hat{\varepsilon} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}.
\]

(2)

By contrast, in logistic regression [190], the dependent variable is binary. In order to facilitate our analysis, in the following we consider the case of a binary dependent variable, for example. The goal of the binary logistic regression is to model the probability of the dependent variable having the value of 0 or 1, given the training samples. To elaborate a little further, let the binary dependent variable \( y \) depend on \( M \) independent variables \( x = [x_1, x_2, \ldots, x_M] \). The conditional distribution of \( y \) under the condition of \( x \) obeys a Bernoulli distribution. Hence, the probability of \( \Pr(y=1 \mid x) \) can be expressed in the form of a standard logistic function\(^6\), also termed as a sigmoid function:

\[
P = \Pr(y = 1 \mid x) = \frac{1}{1 + e^{-g(x)}}
\]

(3)

where \( g(x) = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_M x_M \) and \( \mathbf{w} = [w_0, w_1, \ldots, w_M] \) represents the regression coefficient vector. Similarly, we have:

\[
\Pr(y = 0 \mid x) = 1 - P = \frac{1}{1 + e^{g(x)}}.
\]

(4)

Relying on the aforementioned definitions, we have \( g(x) = \ln \left( \frac{P}{1 - P} \right) \). Hence, for a given dependent variable, the probability of its value being \( y_n \) can be expressed by \( P(y_n) = P(y_n) (1 - P)^{1 - y_n} \). Given a set of training samples \( \{y_n, x_{n1}, x_{n2}, \ldots, x_{nM}\}; n = 1, 2, \ldots, N \), we are capable of estimating the regression coefficient vector \( \mathbf{w} = [w_0, w_1, \ldots, w_M] \) with the aid of the maximum likelihood estimation (MLE) method. Explicitly, logistic regression can be deemed to form a special case of the generalized linear regression family using kernel model.

Furthermore, there exist numerous other useful regression models [191]–[194]. When the dependent variable is a polynomial function of the independent variables, we refer to it as polynomial regression [191], where the best-fit line is a curve. Moreover, ridge regression [192], least absolute shrinkage and selection operator (LASSO) regression [193] and ElasticNet regression [194] are widely

\(^6\)The logistic function is a common “S” shape function, which is the cumulative distribution function (CDF) of the logistic distribution.
applied, when independent variables are of multi-collinear nature and highly correlated. Fig. 10 demonstrates the basic flow of a regression model.

2) Applications: The regression models can be used for estimating, detecting and predicting physical layer radio parameters related to wireless network scenarios. Specifically, Chang et al. [195] proposed a novel regression-aided interference model, which characterized the relationship between the SINR and the packet reception ratio, and evaluated its accuracy relying on the statistics. Based on this model, they constructed an analytic framework for striking a trade-off between the overhead imposed and the accuracy of interference measurement attained. In [196], Umeyashi et al. used regression analysis for formulating a deterministic-stochastic hybrid model for detecting the spectrum usage by PUs, which had a reduced number of parameters and yet maintained a high detection accuracy. In [197], Al Kalaa et al. used logistic regression for estimating the likelihood of Wi-Fi and ZigBee wireless co-existence in the context of medical devices. Furthermore, Xiao et al. [198] constructed a logistic regression-aided physical layer authentication model for detecting spoofing attacks in wireless networks without relying on a known channel model, which exhibited a high detection accuracy, despite its low computational complexity.

The regression models can also be employed for solving both estimation and detection problems in the upper layers of the seven-layer OSI model. For example, Chang et al. derived a regression-based analytical model for the sake of estimating the contention success probability considering heterogeneous sensor-traffic demands, which beneficially improved the channel’s exploitation in IoT [199]. Moreover, in [200], Chen et al. employed a regression model for reconstructing the radio map with the aid of signal strength models for the path planning and UAV-location design in UAV-assisted wireless networks. As a further advance, Lei et al. [201] employed a logistic regression classifier for device-free localization relying on fingerprint signals, which yielded a low localization error.

B. KNN and Its Applications

1) Methods: KNN constitutes a non-parametric instance-based learning method, which can be used both for classification and regression. Proposed by Cover and Hart in 1968, the KNN algorithm is one of the simplest of all ML algorithms. By relying on the distance between the object and training samples in a feature space, the KNN algorithm determines which class of the object belongs to. Specifically, in a classification scenario, an object is categorized into a specific class by a majority vote of its \( K \) nearest neighbors. If \( K = 1 \), the category of the object is the same as that of its nearest neighbor. In this case, it is termed as the one nearest neighbour classifier. By contrast, in a regression scenario, the output value of the object is calculated by the average of the value of its \( K \) nearest neighbors. Fig. 11 shows the illustration of the unweighted KNN mechanism associated with \( K = 4 \).

Let us assume that there are \( N \) training sample pairs of \( \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\} \), where \( y_n \) is the property value or class label of the sample \( x_n, n = 1, 2, \ldots, N \). Typically, we use the Euclidean distance or the Manhattan distance [202] for calculating the similarity between the object \( x \) and the training samples. Let \( x_n = [x_{n1}, x_{n2}, \ldots, x_{nM}] \) contain \( M \) different features. Hence, the Euclidean distance between \( x \) and \( x_n \) can be expressed by:

\[
d_e = \sqrt{\sum_{m=1}^{M} (x_{m} - x_{nm})^2},
\]

while their Manhattan distance is calculated as [202]:

\[
d_m = \sum_{m=1}^{M} |x_{m} - x_{nm}|.
\]

Relying on the associated similarity, the class label or property value of \( x \) can be voted on or first weighted and then voted on by its \( K \) nearest neighbors, which is formulated:

\[
\bar{y} \leftarrow \text{VOTE} \{ K \text{ nearest } (x_k, y_k) \}. \tag{7}
\]

The performance of the KNN algorithm critically depends on the value of \( K \), whilst the best choice of \( K \) hinges upon the training samples. In general, a large \( K \) is conducive to resisting the harmful influence of noise, but it fuzzifies the class boundary between different categories. Fortunately, an appropriate value of \( K \) can be determined by a variety of heuristic techniques based on the true characteristics of the training data set.

2) Applications: In KNN, an object can be classified into a specific category by a majority vote of the object’s neighbours, with the object being assigned to the class that is the most common one among its \( K \) nearest neighbors. Hence, as a kind of simple and efficient classification algorithms, KNN is beneficial in terms of, for example, traffic prediction [203], anomaly detection [204], [205], missing data estimation [206], modulation classification [207], interference elimination [208], etc.

To elaborate, for the sake of capturing the dynamic characteristics of wireless resource demands, Feng et al. constructed a weighted KNN model by learning from a large-scale historical data set generated by cellular operators’ networks, which was used for exploring both the temporal and spatial characteristics of radio resources [203]. In [204], Xie et al. proposed a novel KNN aided online anomaly detection scheme based on hypergrid intuition in the context of WSN applications for overcoming the ‘lazy-learning’ problem [209] especially when the computational resource and the communication cost quantified in terms of bandwidth and energy were constrained. Moreover, in [205], Onireti et al. proposed a KNN based anomaly detection algorithms for improving the outage detection accuracy in dense heterogeneous networks. As for missing data estimation, a KNN assisted missing data estimation algorithm was conceived on the basis of the temporal and spatial correlation feature of sensor data, which jointly utilized the sensor data from multiple neighbor nodes [206]. Furthermore, Aslam et al. [207] combined genetic programming and the KNN
in order to improve the modulation classification accuracy, which can be viewed as a reliable modulation classification scheme for the SU in cognitive radio networks. In [208], the KNN algorithm was used both for extracting the environmental interference imposed by 5G Wi-Fi signals and for reducing the computational complexity and yet improving the performance of indoor localization.

C. SVM and Its Applications

1) Methods:  Being constructed purely by mathematical theory, SVM is another supervised learning model conceived for classification and regression relying on constructing a hyperplane or a set of hyperplanes in a high-dimensional space. The best hyperplane is the one that results in the largest margin amongst the classes. However, the training data set may often be linearly non-separable in a finite dimensional space. To address this issue, SVM is capable of mapping the original space into a higher dimensional space, where the training data set can be more easily discriminated.

Considering a linear binary SVM, for example, there are \( N \) training samples in the form of \( \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\} \), where \( y_n = \pm 1 \) indicates the class label of the point \( x_n \). SVM aims for searching for a hyperplane having the maximum possible separation from the training samples, which best discriminates the two classes of \( x_n \) associated with \( y_n = 1 \) and \( y_n = -1 \). Here, the maximum separation implies having the maximum possible distance between the nearest point and the hyperplane. The hyperplane is
Fig. 11: The illustration of the unweighted KNN mechanism with $K = 4$, for example.

represented by:

$$\omega^T x + b = 0.$$  (8)

Hence, we can quantify the separation of the training sample $(x_n, y_n)$ as:

$$\gamma_n = y_n(\omega^T x_n + b).$$  (9)

Moreover, we assume having the correct classification if $\omega^T x_n + b \geq 0$ when $y_n = 1$, while $\omega^T x_n + b \leq 0$ when $y_n = -1$. Because we have $y_n(\omega^T x_n + b) \geq 0$, a higher separation implies a more reliable classification. Again, the SVM tries to find the optimal hyperplane that maximizes the minimum separation between the training samples and the hyperplane considered. Given a set of linearly separable training samples, after the operation of normalization, the SVM based classification can be formulated as the following optimization problem:

$$\max_{\omega, b} \min_{n=1,\ldots,N} y_n \left( \frac{\omega^T x_n + b}{\|\omega\|} \right)$$

s.t. $y_n(\omega^T x_n + b) \geq \gamma, n = 1, 2, \ldots, N,$

$$\|\omega\| = 1,$$

where we have $\gamma = \min_{n=1,\ldots,N} y_n \left( \frac{\omega^T x_n + b}{\|\omega\|} \right).$

After some further mathematical manipulations, the problem in (10) can be reduced to an optimization problem having a convex quadratic objective function and linear constraints, which can be expressed by:

$$\min_{\omega, b} \frac{1}{2}(\|\omega\|)^2$$

s.t. $y_n(\omega^T x_n + b) \geq 1, n = 1, 2, \ldots, N.$

Problem (11) is a typical convex optimization problem. Taking advantage of Lagrange duality [210], we can obtain the optimal $\omega$ and $b$.

Again, if the training samples are linearly non-separable, SVM is capable of mapping data to a high dimensional feature space with a high probability of being linearly separable. This may result in a non-linear classification or regression in the original space. Fortunately, kernel functions play a critical role in avoiding the “curse of dimensionality” in the above-mentioned dimensionality ascending procedure [211], [212]. To elaborate a little further, given the original input samples $x$, we may be
interested in learning some features \( \phi(x) \). Let us assume \( x_m, x_n \in \mathbb{R}^n \), hence the corresponding kernel function \( K(x_m, x_n) \) is defined as:

\[
K(x_m, x_n) = \phi(x_m)^T \phi(x_n).
\]  

(12)

Fortunately, even though the high dimensional feature mapping \( \phi(x_n) \) may be expensive to calculate, the kernel function calculated relying on their inner product can be easy obtained after some further mathematics manipulations.

There are a variety of alternative kernel functions, such as linear kernel function, polynomial kernel function, radial basis kernel function, neural network kernel function, etc. Furthermore, some regularization methods have been conceived in order to make SVM be less sensitive to outlier points.

The specific choice of the kernel function plays a key role in ML [213], hence we have to beneficially design the kernel function. The construction of kernels can be generally developed by the inner product operations of feature mappings \( \phi(x_n) \) between the input samples over the Hilbert space, whose infinite number of dimensions allow the appropriate representation of big data to exploit their geometric properties. Such a Hilbert space associated with a kernel invoked for producing functions by calculating the inner product of the feature mappings is known as the reproducing kernel Hilbert space (RKHS) [214], and has been applied in diverse learning contents [215], [216]. The RKHS therefore serves a critical foundation in statistical learning theory. Fig. 12 provides a graphical illustration of the kernel-based method.

On the other hand, we may rely on statistical learning theory for appropriately constructing the signal space in order to identify sufficient statistics for reliable signal detection and estimation in statistical communication theory [217]. Inspired by Parzen [214], Kailath observed that RKHS may also be beneficially invoked both for detection and estimation [218] by exploiting the one-to-one relationship between RKHS and finite-variance linear functionals of a random process. Corresponding to the simplest setup of signal detection in additive white Gaussian noise (AWGN) using the Karhunen-Loeve expansion [219], the RKHS representation associated with the noise covariance function is capable of providing an equivalent theoretical framework of statistical communication theory. After a series of efforts inverted into different areas of signal detection and estimation, Kailath and Poor [220] conceived the RKHS approach for the detection of stochastic signals.

2) Applications: As mentioned before, SVM hinges on a mapping that can transform the original training data into a higher dimension, where the events to be classified do become linearly separable. Then it searches for the optimal separating hyperplane for delineating one class from another in this higher dimension considered. As highlighted in Fig. 13, in the spirit of this, SVM aided learning models can be used for detecting and estimating network parameters, for learning and classifying environmental signals and the user’s behavior, as well as for guiding decision making concerning channel selection and anomaly detection, for example [221]–[229].

As for detecting and estimating the network parameters, Feng and Chang [221] constructed a hierarchical SVM (H-SVM) structure for multi-class data estimation. The H-SVM was constructed by a number of levels and each level was composed by a finite number of SVM classifiers. Feng and Chang used their H-SVM model both for estimating the physical locations of nodes in an indoor wireless network and the Gaussian channel’s noise level in a MIMO-aided wireless network. Thanks to its hierarchical structure, the H-SVM was capable of providing an efficient distributed estimation procedure. Furthermore, Tran et al. [222] proposed an SVM model for estimating the geographic location of sensor nodes in WSNs whilst only relying on their connectivity information, more precisely the hop counts. It yielded fast convergence in a distributed manner. The final estimation error can be upper bounded by any small threshold upon relying on a sufficiently large training dataset. Moreover, Sun and Guo [223] conceived a least square-SVM (LS-SVM) algorithm for estimating the user’s position by correlating the time-of-arrival (TOA) of radio frequency signals at the BSs without any detailed knowledge about the base station’s location as well as about the propagation characteristics.

SVM can also be used for learning a user’s behavior and for classifying environmental signals considering the complex spatio-temporal context and the diverse selection of devices. In [224], Donohoo et al. studied the context-aware energy-efficiency improvement options for smart devices. These solutions may become beneficial in terms of configuring their location-specific interface for heterogeneous networks (HetNets) constituted by diverse cells. In [225], by combining the SVM and Fisher discriminant analysis (FDA) Joseph et al. learned the malicious sinking behavior in wireless ad hoc networks for finding the security vulnerabilities and for designing novel intrusion detection scheme. Moreover, features such as delay between data and acknowledgement, number of re-transmits, etc. gleaned from the MAC layer were jointly considered with those from other layers, which constituted a correlated feature set. Furthermore, Pianegiani et al. [226] proposed an SVM-based binary classification solution for classifying acoustic signals emitted by vehicles relying on spectral analysis aided feature extraction, which was beneficial in terms of improving the classification accuracy, despite reducing the implementation complexity.

As for the SVM’s benefit in assisting decision making, in [227], a common control channel selection mechanism was conceived for SU’s during a given frame relying on an SVM-based learning technique proposed for a cognitive radio network, which was capable of implicitly and cooperatively learning the surrounding environment cooperatively in an online way. Moreover, Yang et al. [228] investigated the spoofing attack detection problem based on the spatial correlation of received signal strength gleaned from network nodes, where a cluster-based SVM mechanism was developed for determining the number of attackers. Relying on carefully designed certain training data, the
SVM algorithm employed further improved the accuracy of determining the number of attackers. Rajasegarar et al. [229] also investigated the malicious activity detection issues of WSNs invoking a variety of SVM based algorithms.

D. Bayes Classification and Its Applications

1) Methods: The Bayes classifier, a popular member of the probabilistic classifier family relying on Bayes’ theorem, operates by computing the posterior probability distribution of the objective function values given a set of training samples. As a widely-used classification method, the naive Bayes classifier can be trained for example conditioned on a simple but strong independence assumption in features. Furthermore, the complexity of training a naive Bayes model is linearly proportional to the training set size.

To elaborate a little further, let the vector $\mathbf{x} = [x_1, x_2, \ldots, x_M]$ represent $M$ independent features for a total of $K$ classes $\{y_1, y_2, \ldots, y_K\}$. For each of the $K$ possible class labels $y_k$, we have the conditional probability
of \( p = (y_k|x_1, \ldots, x_M) \). Relying on Bayes’ theorem, we decompose the conditional probability to yield the form of:

\[
p = (y_k|x_1, \ldots, x_M) = \frac{p(y_k)p(x_1, \ldots, x_M|y_k)}{p(x_1, \ldots, x_M)}, \tag{13}
\]

where \( p = (y_k|x_1, \ldots, x_M) \) is the posteriori probability, whilst \( p(y_k) \) is the priori probability of \( y_k \). Given that \( x_i \) is conditionally independent of \( x_j \) for \( i \neq j \), we have:

\[
p = (y_k|x_1, \ldots, x_M) = \frac{p(y_k)}{p(x_1, \ldots, x_M)} \prod_{m=1}^{M} p(x_m|y_k), \tag{14}
\]

where \( p(x_1, \ldots, x_M) \) only depends on \( M \) independent features, which can be viewed as a constant.

The maximum a posteriori probability (MAP) is used as the decision making rule for the naive Bayes classifier. Given a feature vector \( \overline{x} = (x_1, x_2, \ldots, x_M) \), its label \( y \) can be determined according to:

\[
y = \arg \max_{y \in \{y_1, \ldots, y_K\}} p(y_k) \prod_{m=1}^{M} p(x_m|y_k). \tag{15}
\]

Despite idealized simplifying assumptions, naive Bayes classifiers have enjoyed popularity in numerous complex real-world situations, such as outlier detection [230], spam filtering [231], etc.

2) Applications: Based on the Bayes’ theorem, Bayes classifier techniques are particularly applicable to the context where the dimensionality of the input is high. Despite their simplicity, they can often outperform other sophisticated classification methods. As for their applications in wireless networks, in the following, we will elaborate on some typical examples in different wireless scenarios, such as antenna selection, network association, anomaly detection, indoor location and QoE prediction.

Specifically, in [232], He et al. modeled the transmit antenna selection (TAS) problem of MIMO wiretap channels as a multi-classification problem. Then, they used the naive Bayes-based classification scheme to select the optimal antenna for enhancing the physical layer security of the system considered. In contrast to conventional TAS schemes, simulation results showed that the proposed scheme resulted in a reduced feedback overhead at a given secrecy performance. In [233], Abouzar et al. proposed an action-based network association technique for wireless body area networks (WBANs). Relying on the level of received signal strength indicator of the on-body link, the naive Bayes algorithm was employed to recognize the ongoing action, which was beneficial in terms of scheduling the time slot assignment in the context of fixed power allocation on various links by the sink node under a specific data rate constraint. Moreover, Klassen et al. [234] used the naive Bayes classifier for detecting anomaly in ad hoc wireless network involving the black hole attack, the denial of service (DoS) attack and the selective forwarding attack.

Bayes classifier can also be applied to the indoor location estimation. For example, in [235], a probabilistic model was conceived for characterizing the relationship between the received signal strength and location with the aid of the naive Bayes generative learning method, which was used for learning the parameters of an initial probabilistic model, given a limited number of labeled samples. The proposed indoor location estimation method was capable of both reducing the off-line calibration efforts required, whilst maintaining a high location estimation accuracy. Furthermore, as for QoE prediction, in order to evaluate the impact of different networking and channel conditions on the QoE attained in the context of different network services, Charonyktakis et al. [236] proposed a modular algorithm for user-centric QoE prediction. They integrated multiple ML algorithms, including the Gaussian naive Bayes classifier and conceived a nested cross validation protocol for selecting the optimal classifier and its corresponding optimal hyper-parameter value for the sake of accurate QoE prediction.

IV. UNSUPERVISED LEARNING IN WIRELESS NETWORKS

In this section, we will highlight some typical unsupervised learning algorithms, such as K-means clustering [237], expectation-maximization (EM) [238], principal component analysis (PCA) [239] and independent component analysis (ICA) [240] in terms of their methodology and their applications in wireless networks. Table III summarizes some typical applications of the above-mentioned unsupervised learning algorithms in wireless networks.

A. K-Means Clustering and Its Applications

1) Methods: K-means clustering is a distance based clustering method that aims for partitioning \( N \) unlabeled training samples into \( K \) different cohesive clusters, where each sample belongs to one cluster. To elaborate a little further, K-means clustering measures the similarity between two samples in terms of their distance and it has two main steps, namely assigning each training sample to one of \( K \) clusters in terms of the closest distance between the sample and the cluster centroids, and then updating each cluster centroid according to the mean of the samples assigned to it. The whole algorithm is hence implemented by repeatedly carrying out the above-mentioned pair of steps until convergence is achieved.

To elaborate a little further, given a set of samples \( \{x_1, x_2, \ldots, x_N\} \), where \( x_n = [x_{n1}, x_{n2}, \ldots, x_{nM}] \) is a \( M \)-dimensional vector, let \( S = \{s_1, s_2, \ldots, s_K\} \) represent the above-mentioned cluster set, and \( \mu_k \) the mean of the samples in \( s_k \). K-means clustering intends to find an optimal cluster-based segmentation, which solves the following optimization problem:

\[
S^* = \arg \min_{\{s_1, s_2, \ldots, s_K\}} \sum_{k=1}^{K} \sum_{x \in s_k} ||x - \mu_k||^2. \tag{16}
\]

However, problem (16) is a non-deterministic polynomial-time hardness (NP-hard) problem [241]. Fortunately, there are a range of efficient heuristic algorithms, which converge quickly to a local optimum.
Clustering functioning under uncertainty or incomplete information is a common problem in wireless networks, especially in the scenarios associated with numerous small traffic cells, heterogeneous large and small cell structures relying on diverse carrier frequencies, diverse time-varying tele-traffic, etc. First of all, the small cells have to be carefully clustered for avoiding excessive interference using coordinated multi-point transmission. Moreover, the devices and users should be beneficially clustered for the sake of achieving a high energy efficiency, maintaining an optimal access point association, obeying an efficient off-loading policy, and of guaranteeing a high network security. In [243], a mixed integer programming problem was formulated for jointly optimizing both the gateway deployment and the virtual-channel allocation for optical/wireless hybrid networks, where Xia et al. designed an efficient K-means clustering based solution for iteratively solving this problem, which beneficially reduced the delay, as well as improved the network throughput. Moreover, in [244], Hajjar et al. proposed a K-means based relay selection algorithm for creating small cells under the umbrella of an oversailing LTE macro cell within a multi-cell scenario under the constraint of low power clusters. Relying on the proposed relay selection algorithm, the total capacity was increased by reusing the frequency in each low power cluster, which had the benefit of supporting high data rate services. Additionally, Cabria and Gondra [245] proposed a so-called potential-K-means scheme for partitioning data collection sensors into clusters and then for assigning each cluster to a storage center. The proposed K-means solution had the advantage of both balancing the storage center loads and minimizing the total network cost (optimizing the total number of sensors). Parwez et al. [246] invoked both K-means clustering and hierarchical clustering algorithms for their user-activity analysis and user-anomaly detection in a mobile wireless network, which verified genuine identity of users in the face of their dynamic behavior.

One of the popular low-complexity iterative refinement algorithms suitable for K-means clustering is Lloyd’s algorithm [242], which often yields satisfactory performance after a low number of iterations. Specifically, given K initial cluster centroid \( \mu_k, k = 1, \ldots, K \), Lloyd’s algorithm arrives at the final cluster segmentation result by alternating between the following two steps,

- **Step 1:** In the iterative round \( r \), assign each sample to a cluster. For \( n = 1, 2, \ldots, N \) and \( i, k = 1, 2, \ldots, K \), if we have:

  \[
  s_i^{(r)} = \{ \mathbf{x}_n : \| \mathbf{x}_n - \mathbf{\mu}_i^{(r)} \|_2^2 \leq \| \mathbf{x}_n - \mathbf{\mu}_k^{(r)} \|_2^2, \forall k \},
  \]
  \[
  \text{(17)}
  \]

  then we assign the sample \( \mathbf{x}_n \) to the cluster \( s_i \), even if it could potentially be assigned to more than one cluster.

- **Step 2:** Update the new centroids of the new clusters formulated in the iterative round \( r \) relying on:

  \[
  \mathbf{\mu}_i^{(r+1)} = \frac{1}{|s_i^{(r)}|} \sum_{\mathbf{x}_j \in s_i^{(r)}} \mathbf{x}_j,
  \]
  \[
  \text{(18)}
  \]

  where \( |s_i^{(r)}| \) denotes the number of samples in cluster \( s_i \) in iterative round \( r \).

Convergence is deemed to be obtained when the assignment in Step 1 is stable. Explicitly, reaching convergence means that the clusters formulated in the current round are the same as those formed in the last round. Since this is a heuristic algorithm, there is no guarantee that it can converge to the global optimum. Hence, the result of clustering largely relies on specific choice of the initial clusters and on their centroids.

2) **Applications:** K-means clustering aims for partitioning \( N \) samples into \( K \) clusters. Each sample belongs to the closest cluster. The clustering algorithm proceeds in an iterative manner, where the in-cluster differences are minimized by iteratively updating the cluster centroid, until convergence is achieved.

<table>
<thead>
<tr>
<th>Paper</th>
<th>Application</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[195]</td>
<td>interference estimate</td>
<td>regression</td>
<td>strike a trade-off between the overhead and accuracy of interference measurement</td>
</tr>
<tr>
<td>[196]</td>
<td>spectrum sensing</td>
<td>regression</td>
<td>reduce the number of parameters and maintain a high detection accuracy</td>
</tr>
<tr>
<td>[197]</td>
<td>wireless coexistence</td>
<td>regression</td>
<td>estimate the likelihood of the wireless coexistence of Wi-Fi and ZigBee</td>
</tr>
<tr>
<td>[198]</td>
<td>PHY authentication</td>
<td>regression</td>
<td>do not need the assumption on the accurate known channel model</td>
</tr>
<tr>
<td>[199]</td>
<td>traffic estimation</td>
<td>regression</td>
<td>estimate the contention success probability considering sensors’ heterogeneous traffic demands</td>
</tr>
<tr>
<td>[200]</td>
<td>map reconstruction</td>
<td>regression</td>
<td>reconstruct the wireless radio map for UAV path planning and location design</td>
</tr>
<tr>
<td>[201]</td>
<td>wireless localization</td>
<td>KNN</td>
<td>logistic regression classifier for counteracting the negative influence relying on fingerprint signals</td>
</tr>
<tr>
<td>[202]</td>
<td>traffic prediction</td>
<td>KNN</td>
<td>explore both the temporal and spatial characteristics of radio resources</td>
</tr>
<tr>
<td>[203]</td>
<td>anomaly detection</td>
<td>KNN</td>
<td>rely on the hypergrid intuition in the context of WSN applications</td>
</tr>
<tr>
<td>[204]</td>
<td>missing data estimation</td>
<td>KNN</td>
<td>rely on the temporal and spatial correlation feature of sensor data</td>
</tr>
<tr>
<td>[205]</td>
<td>modulation classification</td>
<td>KNN</td>
<td>combine the genetic programming and KNN for improving the modulation classification accuracy</td>
</tr>
<tr>
<td>[206]</td>
<td>interference elimination</td>
<td>KNN</td>
<td>extract environmental interference from Wi-Fi signal and reduce computational complexity</td>
</tr>
<tr>
<td>[207]</td>
<td>data estimation</td>
<td>SVM</td>
<td>provide an efficient estimation procedure in a distributed manner</td>
</tr>
<tr>
<td>[208]</td>
<td>localization estimation</td>
<td>SVM</td>
<td>yield fast convergence performance and efficiently use the communication resources</td>
</tr>
<tr>
<td>[209]</td>
<td>user location</td>
<td>SVM</td>
<td>without knowledge about base station location and environmental propagation characteristics</td>
</tr>
<tr>
<td>[210]</td>
<td>data prediction</td>
<td>SVM</td>
<td>provide location-specific interface configuration for HetNets</td>
</tr>
<tr>
<td>[211]</td>
<td>behavior learning</td>
<td>SVM</td>
<td>combine both the superior accuracy of SVM and fast convergence speed of FDA</td>
</tr>
<tr>
<td>[212]</td>
<td>signal classification</td>
<td>SVM</td>
<td>classify acoustic signals emitted by vehicles rely on feature extraction</td>
</tr>
<tr>
<td>[213]</td>
<td>channel selection</td>
<td>SVM</td>
<td>propose a control channel selection mechanism for a cognitive radio network</td>
</tr>
<tr>
<td>[214]</td>
<td>attacker counting</td>
<td>SVM</td>
<td>develop a cluster-based SVM mechanism for determining the number of attackers</td>
</tr>
<tr>
<td>[215]</td>
<td>antenna selection</td>
<td>Bayes</td>
<td>enhance the physical layer security relying on Bayes-based optimal antenna selection</td>
</tr>
<tr>
<td>[216]</td>
<td>network association</td>
<td>Bayes</td>
<td>schedule time slot assignment and fixed power allocation under data rate constraint</td>
</tr>
<tr>
<td>[217]</td>
<td>anomaly detection</td>
<td>Bayes</td>
<td>detect anomaly involving black hole attack, DoS attack and selective forwarding attack</td>
</tr>
<tr>
<td>[218]</td>
<td>indoor location</td>
<td>Bayes</td>
<td>characterize the relationship between the received signal strength and location</td>
</tr>
<tr>
<td>[219]</td>
<td>QoE prediction</td>
<td>Bayes</td>
<td>accurate QoE prediction by selecting optimal classifier and optimal hyper-parameter values</td>
</tr>
</tbody>
</table>

**TABLE II: Compelling Applications of Supervised Learning in Wireless Networks**
spatio-temporal activities. Furthermore, El-Khatib [247] designed a $K$-means classifier for selecting the optimal set of features of the MAC layer bearing in mind the specific relevance of each feature, which beneficially improved the accuracy of intrusion detection, despite reducing the learning complexity.

Clustering can also be used in signal detection for the sake of both reducing the detection complexity and for improving the energy efficiency attained. In [248], the $K$-means clustering algorithm was invoked in a blind transceiver, where the training process was completely dispensed within the transmitter for reducing its energy dissipation, since no pilot power was required. Furthermore, Zhao et al. [249] conceived an efficient $K$-means clustering algorithm for optical signal detection in the context of burst-mode data transmission.

**B. EM and Its Applications**

1) **Methods:** The EM algorithm is an iterative method conceived for searching for the maximum likelihood estimate of parameters in a statistical model. Typically, in addition to unknown parameters whose existence has been ascertained, the statistical model also has some latent variables. In this scenario it is an open challenge to derive a closed-form solution, because we are unable to find the derivatives of the likelihood function with respect to all the unknown parameters and latent variables. The iterative EM algorithm consists of two steps, as shown in Fig. 14. During the expectation step (E-step), it calculates the expected value of the log likelihood function conditioned on the given parameters and latent variables, while in the maximization step (M-step), it updates the parameters by maximizing the specific log-likelihood expectation function considered.

More explicitly, upon considering a statistical model with observable variables $X$ and latent variables $Z$, the unknown parameters are represented by $\theta$. The log-likelihood function of the unknown parameters is given by:

$$l(\theta; X, Z) = \log p(X, Z; \theta). \quad (19)$$

Hence, the EM algorithm can be described as follows [238]:

- **E-step:** Calculate the expected value of the log likelihood function under the current estimate of $\bar{\theta}$, i.e.

$$Q(\theta|\bar{\theta}) = E_{Z|X}\{\log p(X, Z; \theta)\}. \quad (20)$$

- **M-step:** Maximize Eq. (20) with respect to $\theta$ for generating an updated estimate of $\bar{\theta}$, which can be formulated as:

$$\bar{\theta} = \arg \max_{\theta} Q(\theta|\bar{\theta}). \quad (21)$$

The EM algorithm plays a critical role in parameter estimation based on many of the popular statistical models, such as the Gaussian mixture model (GMM), hidden Markov model (HMM), etc. which are beneficial both for clustering and prediction.

2) **Applications:** The EM model can be readily invoked for a variety of parameter learning and estimation problems routinely encountered in wireless networks. Specifically, Wen et al. [250] estimated both the channel parameters of the desired links in a target cell and those of the interfering links in the adjacent cells relying on constructing a GMM, which was estimated with the aid of the EM algorithm. Choi et al. [251] modeled the cognitive radio system as a HMM, where the secondary users (SUs) estimated the channel parameters such as the primary user’s (PU) sojourn time, signal strength, etc. based on the standard EM algorithm. Moreover, Asrza et al. [252] also adopted the EM algorithm to jointly estimate the channel unknown frequency domain responses as well as the noise variance and detected the PU’s signal in a cooperative wide-band cognitive system, which was shown to converge to the upper bound solution based on maximum likelihood estimation under the idealized assumption of having perfect channel parameter estimation. Additionally, Zhang et al. [253] proposed an EM aided joint symbol detection and channel estimation algorithm for MIMO-OFDM systems in the presence of frequency selective fading, which provided a distribution-estimate for both the hidden symbol and unknown channel parameters in an iterative manner. Li and Nehorai [254] built an asynchronous state-space model for connecting asynchronous observations with the most likely target state transition in the context of multi-sensor WSNs. Then, they adopted the EM algorithm for jointly estimating the sequential target state as well as the network’s synchronization state under the assumption of knowing the temporal order of sensor clocks. Furthermore, Zhang et al. [255] used a variational EM iterative algorithm to recover the transmitted signals and to identify the active users in a low-activity code division multiple access based M2M communications without the knowledge of the user activity factor. The EM algorithm can also be invoked for target or source localization, which can be viewed as a joint sparse signal recovery and parameter estimation problem [256] [257].

**C. PCA & ICA and Their Applications**

1) **Methods:** Feature reduction and extraction constitutes the preparatory phase of ML, because the initially acquired raw data may contain some irrelevant or redundant features. The feature reduction and extraction processing phase results in a reduced set of features with the aid of feature transforms, conceived for reducing the feature space dimension, by eliminating correlated features. However, given the diverse definitions of “features”, it is only realistic to discuss feature-extraction in specific application-oriented contexts.

PCA and ICA constitute sophisticated dimensionality reduction methods in ML, which are capable of reducing both the computational complexity and the storage requirements.

PCA utilizes an orthogonal transformation for converting a set of potentially correlated features of the training samples into a set of uncorrelated features, which are termed as the “principal components”. The number of
principal components is expected to be lower than the number of the original features of the training samples, which hence provide a more compact representation of the original samples. More explicitly, less principle components can be used for representing the original samples in the transformed domain. In PCA, the first principal component tends to have the largest variance, which indicates that it encapsulates the most information of the original features provided that these features were correlated. Similarly, each succeeding component tends to have the next highest variance. These principal components can be generated by invoking the eigenvectors of the normalized covariance matrix.

Specifically, let us consider \( N \) training samples of \( \{x_1, x_2, \ldots, x_N\} \), where \( x_n = [x_{n1}, x_{n2}, \ldots, x_{nM}]^T \) is composed of \( M \) different features. Let us first pre-process the samples by normalizing their mean and variance. Given a unit vector \( u, x_n^Tu \) can be interpreted as the length of the projection of \( x_n \) onto the direction \( u \). The PCA attempts to maximize the variance of the projections, which is formulated as:

\[
\max_u \frac{1}{N} \sum_{n=1}^{N} (x_n^Tu)^2 = \max_u \left( \frac{1}{N} \sum_{n=1}^{N} (x_n^Tu)^2 \right) \tag{22}
\]

Given the covariance matrix \( \Lambda = \frac{1}{N} \sum_{n=1}^{N} (x_n^T x_n^T) \), the solution of problem (22) is given by the eigenvector of the covariance matrix \( \Lambda \). If we denote the top \( K \) eigenvectors of \( \Lambda \) by \( u_1, u_2, \ldots, u_K \) and \( K < M \), a dimensionality reduction expression of \( x_n \) can be formulated as:

\[
y_n = [u_1, u_2, \ldots, u_K]^T x_n, \tag{23}
\]

where \( u_1, u_2, \ldots, u_K \) are the first \( K \) principle components of the training samples.

By contrast, the ICA attempts to find a new basis for representing original samples that are assumed to be a linear weighted superposition of some unknown latent variables. It aims for decomposing multivariate variables into a set of additive subcomponents, which are non-Gaussian variables and are statistically independent from each other. As for the independent components, also termed as the latent variables, they exhibit the maximum possible “statistical independence”, which can be commonly characterized by either the minimization of their mutual information quantified in terms of the Kullback-Leibler divergence metric and the maximum entropy criterion, or by the maximization of what is termed in parlance as the non-Gaussianity relying on kurtosis and negentropy, for example.

Let us consider the linear noiseless ICA model in a simple example, where the multivariate training variables are denoted by \( x = [x_1, x_2, \ldots, x_N]^T \). Its latent independent component vector is represented by \( s = [s_1, s_2, \ldots, s_M]^T \). Each component of \( x \) can be generated by a linearly weighted sum of independent components, i.e. we have \( x_n = a_{n1}s_1 + a_{n2}s_2 + \cdots + a_{nM}s_M \), where \( a_{nm} \) is the weighting coefficient. The vectorial form of \( x \) can be expressed as:

\[
x = \sum_{m=1}^{M} s_m a_m, \tag{24}
\]

where \( a_m = [a_{1m}, a_{2m}, \ldots, a_{Nm}]^T \). Furthermore, let \( A = (a_1, \ldots, a_M) \). Then the original multivariate training variables can be rewritten as:

\[
x = As, \tag{25}
\]

where the unknown matrix \( A \) is referred to as the mixing matrix. ICA algorithms attempt to estimate both the mixing matrix \( A \) and the independent component vector \( s \) relying on setting up a cost function, which again, either maximizes the non-Gaussianity or minimizes the mutual information. Thus, we can recover the independent component vector by computing \( s = A^{-1}x \), where \( A^{-1} \) is termed as the ‘unmixing’ matrix. Usually, we assume that \( N = M \) and that the mixing matrix \( A \) is a square-shaped
matrix. Moreover, the *apriori* knowledge of the probability distribution of $\mathbf{s}$ is beneficial in terms of formulating the cost function.

2) Applications: As for the application of PCA and ICA in wireless networks, Shi *et al.* [258] utilized PCA to extract the most relevant feature vectors from fine-grained subchannel measurements for improving the localization and tracking accuracy in an indoor location tracking system. Moreover, Morell *et al.* [259] designed an efficient data aggregation method for WSNs based on PCA amalgamated with a non-eigenvector projection basis, while keeping the reconstruction error below a pre-defined threshold. Quer *et al.* [260] exploited PCA for inferring the spatial and temporal features of a range of signals monitored by a WSN. Based on this they recovered the large original data set from a small observation set.

Additionally, Qiu [261] combined ICA with PCA in a smart grid scenario for recovering smart meter data, which were jointly capable of enhancing the transmission efficiency both by avoiding the channel estimation in each frame and by eliminating wide-band interference or jamming signals. A semi-blind received signal detection method based on ICA was proposed by Lei *et al.* [262], which additionally estimated the channel information of a multicell multiuser massive MIMO system. Moreover, Sarperi *et al.* [263] proposed an ICA based blind receiver structure for MIMO OFDM systems, which approached the performance of its idealized counterpart relying on perfect CSI. ICA was also used for digital self-interference cancellation in a full duplex system [264], which relied on a reference signal used for estimating the leakage into the receiver. More explicitly, in full duplex systems the high-power transmit signal leaks into the receiver through a nonlinear leakage path and drowns out the low-power received signal. Hence its cancellation requires at least 120 dB interference rejection. Furthermore, in [265], the Boolean ICA concept was proposed based on the integration of Boolean functions of binary signals for inferring the activities of the underlying latent signal sources. Specifically, it was shown that given $m$ SUs, the activities of up to $(2m - 1)$ PUs can be determined.

V. REINFORCEMENT LEARNING IN WIRELESS NETWORKS

Reinforcement learning deals with an agent interacting with the environment. Three specific aspects of reinforcement learning, multi-arm bandit problem, Markov decision process (MDP) and temporal-difference (TD) learning can be very useful for wireless networks. Then, we explore further on these algorithms of reinforcement learning to wireless networks.

A. Multi-Armed Bandit and Its Applications

1) Methods: The multi-armed bandit technique, also called $K$-armed bandit, models a decision making problem, where an agent is faced with a dilemma of $K$ different actions. After each choice, the agent receives a reward relying on a stationary probability distribution that is associated with its decision. The agent attempts to maximize its expected total reward over a series of decision making rounds relying on a balance striking a trade-off between consulting existing knowledge and acquiring new knowledge when optimizing its decisions.

The action of referring to existing knowledge to make decisions is termed as “exploitation”, while the trial of acquiring new knowledge is referred to as “exploration”. Striking a trade-off between exploration and exploitation is also sought by other reinforcement learning algorithms, where exploitation is the plausible action for maximizing the expected reward within the current round, while exploration may produce a greater reward in the long run.

In a $K$-armed bandit model, $K$ possible actions, $a_1, a_2, \ldots, a_K$, yield different rewards associated with the $K$ unknowns of the problem at hand, which may have different distributions with $K$ mean values of $\mu_1, \mu_2, \ldots, \mu_K$, respectively. The agent iteratively chooses an action $A_i$ at the round $i$ and receives the corresponding reward of $R_i$. Up to the round $i$, the expected reward of an action $a$ can be expressed as $Q_i(a) = E[R_i|A_i = a]$. Upon striking a balance between the exploration and the exploitation, we may arrive at a simple bandit algorithm as follows, for example. In each decision-making round, we greedily opt for the action $A = \text{arg} \max Q_i(a)$ relying on the probability of $(1 - \epsilon)$, whilst riskily embarking on a random action selection based on the probability of $\epsilon$, where $\epsilon$ is the probability of a brave attempt for exploring new knowledge.

In contrast to the above-mentioned $\epsilon$-greedy bandit algorithm, there are also more complex bandit algorithms, such as the gradient aided bandit algorithm, associative-search bandit, non-stationary bandit, etc [47]. Moreover, the multi-armed bandit problem can be extended into a multi-play and multi-armed bandit problem [266], where the reward of each agent depends on others’ actions, and each agent tries to find its optimal decision by predicting the future actions of the other agents relying on previous decision making strategies.

2) Applications: As mentioned before, multi-armed bandit based techniques are capable of dealing with uncertainties in the context of future wireless networks because of limited prior knowledge and the associated resource-thirsty feedback. Moreover, it is beneficial to model the selfishness and the decision conflicts of/among the users during the decision making process. Hence, the multi-armed bandit based algorithms have become powerful tools for rational decision making in wireless networks both for distributed users and APs as well as for the central control center. Specifically, Maghsudi *et al.* [267] proposed a small cell activation scheme relying on the multi-armed bandit philosophy given only limited information about the available energy of the small cell BS as well as the number of users to be served. The overall heterogeneous network’s throughput was improved with the aid of an energy-efficient small cell on-off switching regime controlled by the macro BS, while the inter-interference level was reduced. Another compelling application of the multi-armed bandit regime in the heterogeneous network is
TABLE III: Compelling Applications of Unsupervised Learning in Wireless Networks

<table>
<thead>
<tr>
<th>Paper</th>
<th>Application</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[243]</td>
<td>gateway deployment</td>
<td>$K$-means</td>
<td>reduce delay and improve network throughput for optical/wireless hybrid networks</td>
</tr>
<tr>
<td>[244]</td>
<td>relay selection</td>
<td>$K$-means</td>
<td>create small cells in an LTE macro cell with low power cluster constraint</td>
</tr>
<tr>
<td>[245]</td>
<td>sensor partitioning</td>
<td>$K$-means</td>
<td>balance the load of storage centers and minimize the total network cost</td>
</tr>
<tr>
<td>[246]</td>
<td>anomaly detection</td>
<td>$K$-means</td>
<td>verify spatio-temporal varying users’ genuineness relying on ground truth information</td>
</tr>
<tr>
<td>[247]</td>
<td>intrusion detection</td>
<td>$K$-means</td>
<td>improve intrusion detection accuracy and reduce the learning complexity</td>
</tr>
<tr>
<td>[248]</td>
<td>blind transceiver</td>
<td>$K$-means</td>
<td>not require pilot duration and pilot power for saving energy consumption</td>
</tr>
<tr>
<td>[249]</td>
<td>signal detection</td>
<td>$K$-means</td>
<td>burst-mode data transmission with an unbalanced ratio of bits zero and bits one</td>
</tr>
<tr>
<td>[250]</td>
<td>channel estimation</td>
<td>EM algorithm</td>
<td>construct a GMM to estimate channel parameters in both target cell and adjacent cells</td>
</tr>
<tr>
<td>[251]</td>
<td>PU detection</td>
<td>EM algorithm</td>
<td>SUS estimate PU’s sojourn time and signal strength relying on a HMM model</td>
</tr>
<tr>
<td>[252]</td>
<td>channel state detection</td>
<td>EM algorithm</td>
<td>jointly estimate channel frequency responses, noise variance and PU’s signal</td>
</tr>
<tr>
<td>[253]</td>
<td>symbol detection</td>
<td>EM algorithm</td>
<td>joint symbol detection and channel estimation for MIMO-OFDM systems</td>
</tr>
<tr>
<td>[254]</td>
<td>network state detection</td>
<td>EM algorithm</td>
<td>joint estimate the sequential target state and network synchronization state</td>
</tr>
<tr>
<td>[255]</td>
<td>active user detection</td>
<td>EM algorithm</td>
<td>detect active user for the low-activity CDMA based M2M communications</td>
</tr>
<tr>
<td>[256]</td>
<td>source localization</td>
<td>EM algorithm</td>
<td>formulate localization as a joint sparse signal recovery and parameter estimation problem</td>
</tr>
<tr>
<td>[258]</td>
<td>indoor location</td>
<td>PCA</td>
<td>extract relevant feature vectors from fine-grained subchannel measurements</td>
</tr>
<tr>
<td>[259]</td>
<td>data aggregation</td>
<td>PCA</td>
<td>limit the reconstruction error based on a non-eigenvector projection basis</td>
</tr>
<tr>
<td>[260]</td>
<td>data recovery</td>
<td>PCA</td>
<td>exploit PCA to extract spatial and temporal features of real signals</td>
</tr>
<tr>
<td>[261]</td>
<td>data recovery</td>
<td>ICA &amp; PCA</td>
<td>enhance transmission efficiency by avoiding channel estimation and eliminating jamming signals</td>
</tr>
<tr>
<td>[262]</td>
<td>source localization</td>
<td>ICA</td>
<td>differentiate and decode the received signal, and estimate the channel information</td>
</tr>
<tr>
<td>[263]</td>
<td>blind receiver</td>
<td>ICA</td>
<td>yield an ideal performance close to that with perfect CSI</td>
</tr>
<tr>
<td>[264]</td>
<td>interference cancellation</td>
<td>ICA</td>
<td>digital interference cancellation based on the reference signal from transmitter power amplifier</td>
</tr>
<tr>
<td>[265]</td>
<td>signal detection</td>
<td>ICA</td>
<td>infer the activities of latent signal sources based on the Boolean functions</td>
</tr>
</tbody>
</table>

constituted by the dynamic network selection in the context of uncertain heterogeneous network state information. Wu et al. [268] formulated the optimal network selection problem as a continuous-time multi-armed bandit problem considering diverse traffic types. Moreover, the network access cost function and the QoE reward were defined as the metrics of evaluating the proposed network selection schemes. In [269], given the time-varying and user-dependent fading channels of wireless peer-to-peer (P2P) networks, a multi-armed bandit aided optimal distributed transmitter scheduling policy was conceived for multi-source multimedia transmission, which was beneficial of maximizing the data transmission rate and reducing the related power consumption in the light in terms of the realistic energy constraints of wireless mobile devices. In addition to transmitter scheduling, Maghsudi and Stanczak applied the covariate multi-armed bandit regime [270] for solving the relay selection problem in the wireless network, where the geographical location of relay nodes was assumed to be known by the source node, but no knowledge was assumed about the corresponding fading gains. The proposed covariate multi-armed bandit model is capable of dealing with the exploitation-exploration dilemma of the relay selection process. Lee et al. [271] proposed a $K$-greedy multi-armed bandit based framework for exploiting the gains provided by frequency diversity in Wi-Fi channels. They struck a trade-off between the achievable gain stemming from frequency diversity and the resource consumption imposed by channel estimation and coordination.

Given the open broadcast nature of the wireless channel environment and the access contention mechanism among multi-priority users, multi-armed bandit based techniques have played a special role in cognitive networks [272]–[277]. For example, Zhao et al. [272] formulated a multi-armed restless bandit model for opportunistic multi-channel access, which approached the maximum attainable throughput by accurately predicting which is next idle channel likely to become. In [275], a channel selection scheme was investigated which was capable of adapting to the link quality and hence finding the optimal channel for avoiding interferences and deep fading. Moreover, Gwon et al. [273] and Zhou et al. [277] further considered the choice of access strategy in the presence of both legitimate desired users and jamming cognitive radio nodes, which was resilient to adaptive jamming attacks with different strengths spanning from near no-attack to the full-attack over the entire spectrum. In contrast to only sensing and accessing a single channel, considering the correlated rewards of different arms, a sequential multi-armed bandit regime was conceived by Li et al. [274] for identifying multiple channels to be sensed in a carefully coordinat-ed order. Furthermore, Avner and Mannor [276] studied multi-user coordination in cognitive networks, where each user’s successful channel selection relies on both the channel state as well as on the decisions of the other users.

3) An Example: Visible light communication (VLC) systems have the compelling benefit of a wide unlicensed communication bandwidth as well as innate security in downlink (DL) transmission scenarios, hence they may find their way into the construction of future wireless networks. However, considering the limited coverage and dense deployment of light-emitting diodes (LED), traditional network association strategies are not readily applicable to VLC networks. Hence by exploiting the power of online learning algorithms, in [278], the authors focused their attention on sophisticated multi-LED access point selection strategies conceived for hybrid indoor LiFi-WiFi communication systems with the aid of a multi-armed bandit model. Explicitly, since light-fidelity (LiFi) VLC transmissions are less suitable for uplink (UL) transmissions, a classic WiFi UL was used in this study.

To elaborate, in the indoor VLC system, the communication between the devices and the backbone network relies on the VLC DL as well as on the RF WiFi UL, which hence can be viewed as a hybrid LiFi-WiFi network. In the system model, it is assumed that there are $M$ low-energy LED lamps in the indoor space considered.
Moreover, regardless of their positions, the $N$ mobile devices are capable of accessing any of the $M$ indoor LED lamps and of downloading packets from the Internet via VLC. When a decision round is due, the access control strategy obeys the decision probability distribution of $P = \{p_1, p_2, \ldots, p_M\}$. And it has $\sum_{m=1}^{M} p_m = 1$, where $p_m$ denotes the probability of accessing the $m$th LED lamp. Furthermore, the service time of each LED lamp obeys the negative exponential distribution with a departure rate $\zeta$, while the interval between system access requests, in the same way, obeys the negative exponential distribution with an arrival rate $\lambda$. The VLC DL channel is characterized by a diffuse link, where the light beam is radiated within a certain angle. Thus, the indoor VLC channel can be modelled by combining the line of sight (LOS) path (Fig. 15 (a)) as well as a single one-hop reflected path (Fig. 15 (b)).

The expectation of the accumulated reward gap function is defined as the metric for characterizing the performance of our AP selection scheme, which represents the difference between the maximum theoretical reward and the actually acquired reward after sequential decision making experiments relying on the system’s decision probability distribution, which is formulated as:

$$R_P(K) = \max_{t_1, t_2, \ldots, t_K} \left( \sum_{k=1}^{K} Q(i_k, t_k) - \sum_{k=1}^{K} Q(a_k, t_k) \right)$$

(26)

where $Q(i_k, t_k)$ denotes the user rate associated with the $k$th decision round in terms of the access decision $i_k$ at the instant $t_k$, with $a_k$ being the actual access decision.

Furthermore, in [278] a pair of multi-armed bandit learning techniques, i.e. the ‘exponential weights for exploration and exploitation’ (EXP3) as well as the ‘exponentially-weighted algorithm with linear programming’ (ELP), were advocated for updating the AP-assignment decision probability distribution of each AP at each time instant for the sake of improving the link throughput based on the probability distribution $R_P(K)$ of (26). More explicitly, in contrast to the trial-and-error EXP3 algorithm, the ELP based AP selection algorithm was constructed for taking into account both the partially observed conditions of the APs as well as the network topology.

The theoretical upper bound of the expected value of the accumulated reward gap function of the EXP3- and ELP-based multi-armed bandit learning algorithms was also derived in [278]. In Fig. 16 and Fig. 17, the normalized throughput of the selected VLC links and of the whole system relying on the EXP3-based, ELP-based as well as on random LED AP selection schemes was compared. By contrast, the random selection scheme granted an identical decision probability of accessing any of the $M$ LEDs, namely $1/M$, for each lamp at each decision-making time instant. It was assumed that the negative exponential departure probability of each downloading service was $\zeta = 0.2$. Moreover, the initial state of the number of downloading services supported by each lamp was randomly chosen between $[1, 30]$. Upon increasing the number of decision rounds $K$, the EXP3- and ELP-based selection schemes had a higher accumulated normalized throughput than random selection. Furthermore, relying on more neighbor observation information as well as by exploiting the connection of the LED lamps, the ELP-based AP-selection scheme was shown to outperform that based on EXP3.
B. MDP & POMDP and Their Applications

1) Methods: The classic Markov decision process (MDP) [279] constitutes a framework of making decisions in the context of a discrete-time stochastic environment of Markov state transitions, which provides the decision maker with the optimal actions to opt for at each state. It has been used in a wide range of disciplines, especially in automatic control [280]. The goal of the decision maker, generally speaking, is to maximize the cumulative reward received over a long run and to find the corresponding optimal policy \( \pi^* \) which represents a mapping from each state to the specific probabilities of choosing each legitimate action.

In an MDP model, the system’s state transition follows the Markovian property, where the system’s response at time epoch \((t+1)\) depends exclusively on the current state and on the agent’s action at time epoch \(t\). Mathematically, at time epoch \(t\), the system is in a certain state \(S\), where the agent selects a legitimate action \(A\) that is available in the state \(S\). As a result, the system then acts at the next time epoch \((t+1)\) by moving into a new state \(S'\) relying on the system’s state transition probability of \(p(S'|S,A)\). At the same time, the decision maker receives the corresponding reward \(r(S,A)\). The associated value function is then defined for quantifying how well the agent carries out its action over a long run commencing from the initial state \(S_0\), which can be formulated as:

\[
v_\pi = E_\pi \left[ \sum_{t=0}^{\infty} \gamma^t r_{t+1} | S_t \right],
\]

where \(\gamma\) represents the discount factor and the mapping \(\pi(A|S)\) represents the probability of opting for action \(A\) in the state \(S\). Hence, the optimal policy \(\pi^*\) can be formulated by maximizing the value function considered, i.e. we have \(\pi^* = \arg\max_{\pi} v_\pi\). The maximization of the value function can be reformulated as an iterative equation with the aid of Bellman’s optimality theorem [281], which is given by:

\[
\pi^*(A | S) = \arg\max_A v^*(S) = \arg\max_A \sum_{S'} p(S' \mid S, A) \left[r(S, A) + \gamma v^*(S')\right].
\]

By contrast, as an extension of MDP, the partially observable Markov decision process (POMDP) only relies on partial knowledge about the hidden Markov system which is eminently suitable for scenarios, where the agent cannot directly observe the underlying system’s state transitions. Hence, the agent has to constitute belief states and the associated belief transition function by relying on a set of observations instead of the real system states. In a nutshell, the POMDP framework can be formulated as a quintuple of \(\langle S, B, A, B, r \rangle\), i.e.

- **System’s State S:** The system’s state \(S\) represents the system’s legitimate state;
- **Belief State b:** The belief state \(b\) benchmarks the degree of the similarity between each of the system’s legitimate state \(S\) and the state estimated by the agent;
- **Action A:** The action \(A\) denotes the specific action that can be selected in the given state;
- **Belief Transition Function B:** The belief transition function \(B(b' \mid b, A)\) represents the probability of the belief state traversing from \(b\) to \(b'\) conditioned on selecting action \(A\);
- **Reward Function r:** The reward function \(r(b, A)\) quantifies the immediate reward received by performing the selected action.

Similarly, the optimal policy \(\pi^*\) can be obtained by solving the optimization problem of:

\[
\pi^*(A | b) = \arg\max_A v^*(b) = \arg\max_A \sum_{b'} B(b' \mid b, A) \left[r(b, A) + \gamma v^*(b')\right].
\]

2) Applications: As another important decision-making tools, which is different from the multi-armed bandit solutions, MDP/POMDP should firstly model the environment relying on either fully or partially observed knowledge. To elaborate a little further, Massey et al. [282] proposed an MDP based downlink service scheduling policy for wireless service providers. Considering the time-sensitive nature of wireless tele-traffic patterns, their proposed scheduling policy was capable of maximizing the expected reward for the wireless service provider in the context of a multiplicity of services. In [283], Tang et al. resorted to the MDP approach for enhancing a basic node-misconduct detection method, where a novel reward-penalty function was defined as a function of both correct and wrong decisions. The resultant adaptive node-misconduct detector maximized this reward-penalty function in diverse network states. Moreover, Kong et al. conceived a discrete-time MDP (DTMDP) aided mechanism [284] for dynamically activating and deactivating certain resources of the BS in the context of time-varying network traffic. More explicitly, at each decision round, the DTMDP had the option of activating a new resource module, deactivating the currently active resource module and no operation. The proposed switching mechanism reduced the power consumption, i.e. improved the energy efficiency at the BS.

As a further development, relying on the POMDP paradigm, Tseng et al. [285] designed a cell selection scheme for improving the network’s capacity, where the full cell loading status was not observable. Hence, it predicted the unavailable cell loading information from set of non-serving base stations and then took actions for improving the various performance metrics, including the system’s capacity, the handover time as well as the mobility management as a whole. Moreover, the belief state was defined for representing the state uncertainty in terms of the statistical probability of a cell’s specific loading state. The simulation results of Tseng et al. [285] showed that their solution outperformed the conventional signal-strength aided and load-balancing based methods. In order to save the energy of sensors, Fei et al. [286] proposed a POMDP aided K-sensor scheduling policy, which guaranteed the sensors’ high-quality coverage and reduced
the total energy consumption. Similarly, by striking a trade-off between the detection performance and energy consumption, Zois et al. [287] designed a POMDP aided sensor node selection scheme for WBANs by maximizing the system’s lifetime as well as optimizing the physical state detection accuracy. The main goal of the sensor node selection was to devise a schedule under which the sensors alternated between the active state and the dormant state relying on the specific network activity. Upon relying on the decentralized POMDP (DEC-POMDP), Pajarinen et al. [288] proposed a MAC solution, which promptly adapted both to the spatial and temporal opportunities facilitated by the wireless network dynamics, which yielded an increased throughput and reduced latency compared to the traditional carrier-sense multiple access relying on conventional collision avoidance (CSMA/CA) methods. Here, the POMDP tackled the uncertainty both in the environment’s evolution and in the associated inaccurate observations. Thanks to the cross layer optimization employed, more information can be gleaned from the lower layers for enhanced network condition estimation. Then, Xie et al. [289] used the POMDP model for solving the frame size selection problem of the ubiquitous transmission control protocol (TCP) with the objective of improving the total estimated throughput by striking a tradeoff between the contention probability and back-off time based on the current network condition. Furthermore, Michelusi and Mitra [290] conceived a cross-layer framework for jointly optimizing the spectrum-sensing and access processes of cognitive wireless networks with the objective of maximizing the throughput of the SU under a strict constraint on the maximal performance degradation imposed on the PU. Furthermore, the high complexity of the POMDP formulation was mitigated by a low-dimensional belief representation, which was achieved by minimizing the Kullback-Leibler divergence defined in [291].

3) An Example: As shown in Fig. 18, the ‘super-WiFi’ network concept has been originally proposed for nationwide Internet access in the USA. However, the traditional mains power supply is not necessarily ubiquitous in this large-scale wireless network. Furthermore, the non-uniform geographic distribution of both the BSs and of the tele-traffic requires carefully considered user-association. Relying on the rapidly developing energy harvesting techniques, in [292], a POMDP-based access point selection strategies was conceived for an energy harvesting aided super-WiFi network.

It was assumed that both the battery states as well as the user access states were completely observable. However, in practice the solar radiation intensity changes over time in a year, as influenced by the weather conditions. Furthermore, the radiation sensors have a limited sampling rate, which makes it hard to simultaneously record the solar radiation intensity and to accurately estimate the system’s battery state. Fortunately, relying on historical solar radiation observation data provided by the University of Queensland, Australia [293], in a short period of time, say, within an hour, the real-time harvested solar power can be modeled as 

\[ P_H = \theta + \kappa, \]

where \( \theta \) is constant for an hour, while \( \kappa \) is a small perturbation. Moreover, multiple factors, such as the effective irradiation area, the clouds’ distribution, the sensors’ operating status, etc. may independently affect the harvested power. Relying on the central-limit theorem, the perturbation \( \kappa \) can be regarded as being Gaussian distributed. Hence, the distribution of \( P_H \) can be written as 

\[ P_H \sim N(\theta, \sigma^2), \]

where \( \theta \) and \( \sigma^2 \) can be learned from the harvested data set.

Moreover, a queue-based user-association state model as well as a dynamic battery state model was established. Hence, the system’s state having \( K \) APs is constituted by both the user-association states as well as by the battery states. Let \( U = (U_1, U_2, ..., U_K) \) denote the user-association states, while \( B = (B_1, B_2, ..., B_K) \) represent the AP battery states, where \( U_k \leq U_M \) and \( B_k \leq B_M - 1 \). Furthermore, the super-WiFi system state can be written as a \( 2K \)-element vector \( S = (U_1, B_1, U_2, B_2, ..., U_K, B_K) \), which includes both the \( K \) APs’ user-association states and the \( K \) APs’ battery states. Assuming the independence of each AP’s two sub-states, the system’s state transition probability can be expressed as:

\[ P(S' | S) = \prod_{k=1}^{K} \Phi_k(U'_k|U_k, A)\Psi_k(B'_k|B_k, A), \quad (30) \]

where \( A \) represents the users’ actions in terms of which available APs they request association with.

Since the requesting users only have partial knowledge of the entire super-WiFi system’s state, relying on the above definitions and hypotheses, we construct the POMDP decision-making model in terms of a quintuple of \( (S, b, A, B, r) \) as mentioned above. The POMDP formulation can be reduced to a belief MDP with the aid of the belief state vector. Therefore, the expected reward of the system relying on strategy \( \Pi \) after an infinite number...
of time slots can be written as:

\[ V^\Pi(S|S^0) = \mathbb{E}\left( \sum_{t=0}^{\infty} \gamma^t r_t(S^t, \Pi(S^t)) | b(S^t), S^0 \right), \]  

where \( S^0 \) is the initial system state, while \( b(S) \) is the belief state vector reflecting the grade of similarity between the current estimated state and the legitimate system state \( S \). Moreover, \( r \) is the immediate reward of the system and \( \gamma \) represents the discount rate. Then, the optimal strategy can be constructed by invoking dynamic programming aided iterative algorithms for maximizing the expected reward function.

Bearing in mind the large values of \( K, U_M, B_M \), as well as the users’ rapidly fluctuating arrival rate \( \lambda \) and departure rate \( \mu \), obtaining the optimal POMDP solution may face the curse of dimension disaster. In order to reduce the computational complexity, a suboptimal algorithm was proposed in [292]. Explicitly, Algorithm 2 of [292] aimed for maximizing the expectation of the system’s energy function, which was defined as:

\[ H[b(S)] = \sum_{S \in \mathcal{S}} b(S) \sum_{i=1}^{K} \min_{\mathcal{H}} \{ E_{iR} + E_{iH} - E_{iC}, E_{\max} \}, \]

where \( E_{iR} \) represents the residual energy of AP \( i \), while \( E_{iH} \) is its energy harvested under the assumption that the harvested power level remains quasi-static during the information transmission interval and \( E_{iC} \) denotes the energy consumption. Finally, \( E_{\max} \) is the capacity of the AP’s battery.

The efficiency of the AP selection algorithms proposed in [292] was compared in terms of the system’s access efficiency defined as \( \xi = N_S/T_S \), where \( N_S \) is the total number of successful access attempts during the entire simulation time \( T_S \). In Fig. 19 and Fig. 20, multiple APs (\( K = 2 \)) are considered with the maximum number of admitted users being \( U_M = 1 \), while having a maximum number of battery states given by \( B_M = 3 \). Moreover, the departure rate is \( \mu = 0.05 \). We may conclude from Fig. 19 that a highly loaded system makes the carrier-sense multiple access with collision detection (CSMA/CD) method almost useless, when the users’ arrival rate reaches a certain value. As shown in Fig. 20, where \( \lambda = 0.4 \), the system’s access efficiency recorded for all the AP selection algorithms only increases with the solar radiation intensity in a relatively small range. However, the performance of the CSMA/CD, CSMA/CA, as well as of the random selection algorithm remains unchanged, regardless of the increase in solar radiation intensity. Moreover, the suboptimal Algorithm 2 of [292] is capable of outperforming the POMDP method at a strong solar radiation intensity, which may be deemed to be the result of the approximations and hypotheses inherent in the POMDP model.

7Strictly speaking, the CSMA/CD and CSMA/CA in this paper are different from the Ethernet’s data link layer protocols. Here, both of them represent the access control mechanisms. We use the same acronym CSMA/CD and CSMA/CA for convenience.

Fig. 19: The system access efficiency versus users’ arrival rate for different access schemes (\( K = 2, U_M = 1, B_M = 3 \)). ©IEEE

Fig. 20: The system access efficiency versus solar radiation intensity for different access schemes (\( K = 2, U_M = 1, B_M = 3 \)). ©IEEE

C. Temporal Difference Learning and Its Applications

1) Methods: Temporal-difference (TD) learning is a model-free reinforcement learning method, which is capable of directly gleaning knowledge from raw experience without a model of the environment or receiving delayed reward, which can be typically viewed as a combination of Monte Carlo methods and of dynamic programming. More specifically, it samples the environment like the Monte Carlo methods, and then updates the corresponding parameters relying on current estimates like dynamic programming does. By contrast, TD learning operates in an on-line fashion by relying on the result of a single time step, rather than waiting for the final outcome until the end of an episode of the Monte Carlo method. Moreover, it has an advantage over the dynamic programming methods since it does not require a model of the state transition probabilities as shown in Fig. 21. TD learning can be readily invoked for finding an optimal action policy for any finite MDP associated with an unknown system model. Fig. 21 shows the difference between the MDP, POMDP
and TD learning.

A pair of popular representatives of the TD learning family are constituted by the Q-learning and by the “state-action-reward-state-action” (SARSA) technique, which interacts with the environment and updates the state-action value function, namely the Q-function, based on the action it takes. In contrast to SARSA, Q-learning updates the Q-function relying on the maximum reward provided by one of its available actions. Specifically, the update of the Q-function in SARSA can be formulated as [55]:

\[ Q(S, A) \leftarrow (1 - \alpha)Q(S, A) + \alpha [r + \gamma Q(S', A')] \]  

while in Q-learning, the update of the Q-function can be cast as [88]:

\[ Q(S, A) \leftarrow (1 - \alpha)Q(S, A) + \alpha \left[ r + \gamma \max_{A' \in \mathcal{A}} Q(S', A') \right] \]  

where \( S \) represents the system’s state and \( A \) is the action selected by the agent, whilst \( \mathcal{A} \) represents the available set of actions. Moreover, \( \alpha \) is the update weighting coefficient and \( \gamma \) denotes the discount factor. As for the convergence analysis, SARSA is capable of converging with probability 1 to an optimal policy as well as to an optimal state-action value function, provided that all the state-action pairs are visited a sufficiently high number of times. However, because of the independence of making an action and that of updating the Q-function, Q-learning has no delayed reward as TD-learning, which tends to facilitate an earlier convergence than SARSA [47].

2) Applications: As a benefit of being free from modeling the environment, TD learning is capable of providing competent decisions even in unknown environments. Table IV summarizes a variety of compelling applications found in wireless networks for both SARSA and Q-learning along with their brief description.

VI. DEEP LEARNING IN WIRELESS NETWORKS

A. Deep Artificial Neural Networks and Their Applications

1) Methods: Artificial neural networks [304] constitute a set of algorithms conceived by imitating the interaction amongst neurons in the human brain, which are designed to extract features for clustering and classification tasks.

In a common artificial neural network (ANN) model [305], the input of each artificial neuron is a real-valued signal, where the output of each artificial neuron is subjected to by some non-linear functions, namely the so-called activation functions. This is formulated as:

\[ y_k = \psi(u_k + b_k), \]

where

\[ u_k = \sum_{i=1}^{m} w_{ki}x_i. \]

Here, \( x_i \) and \( y_k \) represent the input signal and output signal, respectively, while \( w_{ki} \) is the associated weight and \( b_k \) is the bias. Moreover, \( \psi(\bullet) \) is the activation function. Artificial neurons and their connections typically use a weighting factor for adjusting the “speed” of the learning process. Moreover, artificial neurons are organized in the form of layers. Different layers perform different transformations of their inputs. Basically, the input signals travel from the first layer to the last layer, possibly via multiple hidden layers.

The general deep neural network (DNN) is characterized by multiple hidden layers between the input and output layers as shown in Fig. 22 (a), which is capable of modeling complex relationships of the processed data with the aid of multiple non-linear transformations. In a DNN, the provision of extra layers facilitates the composition of features from lower layers, which is beneficial in terms of more accurately modeling complex data than a ‘shallow’ network having a single hidden layer. In comparison to ‘shallow’ networks, DNNs are capable of attaining a performance, despite requiring less training. Furthermore, DNNs may be viewed as a type of feed-forward network, where the processed data flows in the direction from the input layer to the output layer without looping back. The forward propagation algorithms and backward propagation algorithms represent a step-by-step problem solving process. Specifically, the forward propagation algorithms apply the carefully trained weight matrices and bias vectors for carrying out the associated linear and
TABLE IV: Compelling Applications of the SARSA Algorithm and Q-learning in Wireless Networks

<table>
<thead>
<tr>
<th>Paper</th>
<th>Method</th>
<th>Scenario</th>
<th>Application &amp; Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[294]</td>
<td>reduced-state SARSA</td>
<td>cellular network</td>
<td>dynamic channel allocation considering both mobile traffic and call handoffs.</td>
</tr>
<tr>
<td>[295]</td>
<td>on-policy SARSA</td>
<td>CR network</td>
<td>distributed multiagent sensing policy relying on local interactions among SU</td>
</tr>
<tr>
<td>[296]</td>
<td>on-policy SARSA</td>
<td>MANET</td>
<td>energy-aware reactive routing protocol for maximizing network lifetime</td>
</tr>
<tr>
<td>[297]</td>
<td>on-policy SARSA</td>
<td>HetNet</td>
<td>resource management for maximizing resource utilization and guaranteeing QoS</td>
</tr>
<tr>
<td>[298]</td>
<td>approximate SARSA</td>
<td>P2P network</td>
<td>energy harvesting aided power allocation policy for maximizing the throughput</td>
</tr>
<tr>
<td>[299]</td>
<td>Q-learning</td>
<td>WBAN</td>
<td>power control scheme to mitigate interference and to improve throughput</td>
</tr>
<tr>
<td>[300]</td>
<td>Q-learning</td>
<td>OFDM system</td>
<td>adaptive modulation and coding not relying on off-line training from PHY</td>
</tr>
<tr>
<td>[301]</td>
<td>Q-learning</td>
<td>cooperative network</td>
<td>efficient relay selection scheme meeting the symbol error rate requirement</td>
</tr>
<tr>
<td>[302]</td>
<td>decentralized Q-learning</td>
<td>CR network</td>
<td>aggregated interference control without introducing signaling overhead</td>
</tr>
<tr>
<td>[303]</td>
<td>convergent Q-learning</td>
<td>WSN</td>
<td>sensors’ sleep scheduling scheme for minimizing the tracking error</td>
</tr>
</tbody>
</table>

activation algorithms. By contrast, the backward propagation algorithms, which are widely used in the industrial field define a so-called loss function for quantifying the difference between the output produced by the training samples’ and the real output. Then, by minimizing the loss or error we can optimize the connection weights of the neural network. Although there are impressive applications of DNN, improving their convergence behavior requires further research.

By contrast, in a recurrent neural network (RNN) a neuron in one layer is capable of connecting to the neurons of previous layers. Therefore, a RNN is capable of exploiting the dynamic temporal information hidden in a time sequence and it beneficially exploits its “memory” inherited from the previous layers for processing the future inputs, as shown in Fig. 22 (b). RNNs have the advantage of requiring reduced training and impose a reduced computational complexity, because fewer tensor operations have to be calculated. Popular algorithms used for training the RNN include the real time recurrent learning technique of [306], the causal recursive backpropagation algorithm of [307] and the backpropagation through time algorithm of [308], just to name a few.

The convolutional neural network (CNN) constitutes a popular class of feed-forward deep artificial neural networks, which only require modest preprocessing. They are capable of reducing the number of parameters step-by-step as we move from the input layer to the hidden layers. As seen in Fig. 22 (c), a basic CNN architecture is composed of an input layer, an output layer as well as multiple hidden layers, which are often referred to as convolutional layers, pooling layers and fully connected layers. More particularly, the convolutional layers carry out the convolution operation, also termed as the cross-correlation operation, which generates a multi-dimensional feature map relying on a number of so-called filters. The CNN has been successfully used both in image and video recognition [309], in natural language processing [310], in recommender systems [311], etc. Fig. 22 contrasts the basic architecture of DNN, RNN and CNN, respectively.

2) Applications: In this subsection, we will consider the benefits of deep artificial neural network algorithms in a variety of wireless networking scenarios. As mentioned before, deep artificial neural networks are capable of capturing the non-linear and often dynamically varying relationship between the inputs and outputs. Hence they have a powerful prediction, inference and data analysis capability by exploiting the vast amount of data generated both by the environment and by the users. As for learning from the environment, we are able to harness DNNs trained by the data gleaned over the air for the sake of channel estimation [312], interference identification [313], localization [314]–[318], etc. By contrast, with regard to learning from the users or devices, DNN algorithms can also be used for predicting the users’ behaviors, such as their content interests [319], mobility patterns [320], etc. in order to beneficially design the dynamic content caching of BSs and to efficiently allocate wireless resources, for example.

Traditional signal processing approaches supported by statistics and information theory in communication systems substantially rely on accurate and tractable mathematical models. Unfortunately, however, practical communication systems may have a range of imperfections and non-linear factors, which are difficult to model mathematically. Given that DNN algorithms do not require a tractable model, they are capable of remedying the imperfections in the physical layer by learning both from the environment and from previous inputs relying on a specific hardware configuration. To elaborate, Ye et al. [321] proposed a DNN aided channel estimation method for learning the wireless channel characteristics, such as the nonlinear distortion, interference and frequency selectivity. The DNN aided channel estimation method was shown to be more robust than traditional methods, especially in the context of having fewer training pilots, in the absence of cyclic prefix, as well as in the face of nonlinear clipping noise. Apart from estimating the channel characteristics, DNNs can also be used for classifying modulated signals in the physical layer. Rajendran et al. [321] conceived a date-driven automatic modulation classification (AMC) scheme hinging on the long short term memory (LSTM) aided RNN, which captured the time domain (TD) amplitude and phase information of modulation schemes carried in the training data without expert knowledge. Their simulations showed that the novel AMC had an average classification accuracy of about 90% in the context of time-varying SNR ranging from 0dB to 20dB. As for signal detection, Farsad and Goldsmith [322] developed a deep learning aided signal detector, where the transmitted signal can be efficiently estimated from its corrupted version observed at the receiver. The detector was trained relying on known transmitted signals, but without any knowledge of the underlying wireless channel model and estimated the likelihood of each symbol, which was beneficial for carrying out soft decision error correction afterwards. In the application of
interference identification, Schmidt et al. [313] proposed a 64-feature-map assisted CNN based wireless interference identification scheme. The CNN model learned the relevant features through self-optimization during the GPU based training process, which was first designed in [323]. By carefully considering the realistic capability of wireless sensors, the model relied on the time- and frequency-limited sensing snapshots having the duration of 12.8 $\mu$s as well as the bandwidth of 10MHz. The proposed CNN based wireless interference identifier was shown to have a higher identification accuracy than the state-of-the-art schemes in the context of low SNRs, such as $-5$dB, for example.

Furthermore, we can use DNNs for modelling the entire physical layer of a communication system without any classic components such as source coding, channel coding, modulation, equalization, etc. In [324], O’Shea et al. used a DNN to represent a simple communication system with one transmitter and one receiver that can be trained as a so-called auto-encoder without knowing the accurate channel model. Moreover, a CNN algorithm was conceived for modulation classification based on both sampled radio frequency time-series data and expert knowledge integrated by radio transformer networks (RTN). Additionally, O’Shea et al. [325] extended the DNN aided auto-encoder to a single user MIMO communication scenario, where the physical layer encoding and decoding processes were jointly optimized as a single end-to-end self-learning task. Their simulation results showed that the auto-encoder based system outperformed the classic space time block code (STBC) at 15dB SNR. Furthermore, Dörner et al. [326] also developed a DNN based prototype system solely composed of two unsynchronized off-the-shelf software-defined radios (SDR). This prototype system was capable of mitigating the current restriction on short block lengths.

DNNs also play a critical role in supporting a variety of compelling upper layer applications, such as traffic prediction [327], packet routing [328] and control [329], traffic offloading [330], resource allocation [331], attack detection [332], just to name a few. For instance, Wang et al. [327] presented a hybrid deep learning aided structure for spatio-temporal traffic modeling and prediction in cellular networks by mining information from the China Mobile dataset. It used a novel deep learning aided auto-encoder for modeling the spatial features of wireless traffic, while using LSTM units for temporal modeling. Additionally, Kato et al. [328] proposed a supervised DNN aided traffic routing scheme, which outperformed the classic open shortest path first (OSPF) scheme in terms of requiring a lower overhead, whilst maintaining a higher throughput and lower delay. By contrast, a real-time deep CNN based traffic control mechanism learning from previous network anomalies was conceived by Tang et al. [329], which substantially reduced the average delay and packet loss rate. Hence, deep learning aided traffic control may indeed constitute a potential candidate for gradually replacing traditionally routing protocols in future wireless networks. Furthermore, Li et al. [330] integrated both the DNN structure and the edge computing technique into the multimedia IoT, which was able to improve the efficiency of multimedia processing. Sun et al. [331] treated the power control problem in interference-limited wireless networks as a ‘black box’. They proposed an ‘almost-real-time’ power control algorithm relying on a DNN structure trained by simulated data. In comparison to traditional mathematical tools, the approximation error of the DNN aided algorithm is closely related to the depth of the DNN considered. As for network security issues, for example, He et al. [332] constructed a conditional deep belief network (CDBN) for the real-time detection of malicious false data injection (FDI) attacks in the smart grid, which was trained by historical measurement data. The simulations conducted using the IEEE 118-bus test system and the IEEE 300-bus test system showed that the CDBN aided FDI detection scheme was resilient to the environmental noise and had a higher detection accuracy than its SVM aided counterparts.

As a successful example of learning from the environment, DNNs are beneficial in terms of extracting electromagnetic fingerprint information from the wireless channel for indoor localization. In [314]–[316], Wang et al. proposed a DNN having three hidden-layer for training the calibrated CSI phase data, where the fingerprint information was represented by the DNN’s weights. Their experimental results showed that the DNN aided localization scheme performed well in different propagation environments, including an empty living room, and a laboratory in the presence of mobile users. In [317], Wang et al. proposed a deep learning method for supporting device-free wireless localization and activity recognition relying on learning from the wireless signals around the

---

**Fig. 22:** The basic architecture of DNN, RNN and CNN.
target, where a sparse auto-encoder network was used for automatically learning the discriminative features of wireless signals. Furthermore, a softmax-regression-based framework [333] was formulated for the location and activity recognition based on merged features. Moreover, in [318], Zhang et al. constructed a four-layer DNN for extracting reliable high level features from massive Wi-Fi data, which was pre-trained by the stacked denoising auto-encoder. Additionally, an HMM aided high-accuracy localization algorithm was proposed for smoothing the estimate variation. Their experimental results showed a substantial localization accuracy improvement in the context of a widely fluctuating wireless signal strength.

With regard to learning from users or devices, Ouyang et al. [320] conceived a CNN-aided online learning architecture for understanding human mobility patterns relying on analyzing continuous mobile data streams. Al-Molegi et al. [334] integrated both the spatial features gleaned from GPS data and the temporal features extracted from the associated time stamps for predicting human mobility based on a RNN. Moreover, Song et al. [335] proposed an intelligent deep LSTM RNN based system for predicting both human mobility and the specific transportation mode in a large-scale transportation network, which was beneficial in terms of providing accurate traffic control for intelligent transportation systems (ITS). Additionally, a mobility prediction technique relying on a complex extreme learning machine (CELM) was developed by Ghoui et al. [336] in order to jointly optimize both the bandwidth and the power MANETs. In [337], both the multi-layer perception and RNN models were employed by Agarwal et al. for characterizing the activity of primary users in CR networks, where three different traffic distributions, namely Poisson traffic, interrupted Poisson traffic and self-similar traffic were used for training the related models.

Table V lists a range of typical applications of DNNs along with a brief description.

### B. Deep Reinforcement Learning and Its Applications

1) Methods: The deep reinforcement learning technique is constituted by the integration of the aforementioned DNNs and reinforcement learning. Explicitly, in deep reinforcement learning methods, DNNs are used for approximating certain components of reinforcement learning, including the state transition function, reward function, value function and the policy. These components can be viewed as a function of the weights in these DNNs, which can be updated with the aid of the classic stochastic gradient descent.

In particular, the deep Q-Network (DQN) constitutes the first deep reinforcement learning solution, which was proposed by Mnih et al. in 2015 [58], which avoids the instability of the reinforcement learning algorithm, which may even become divergent when its action-value function is approximated relying on a non-linear function. To elaborate a little further, DQN stabilizes the training process of the action-value function approximation by relying on experience replay. Furthermore, DQN only requires modest domain knowledge. The deep Q-learning algorithm in DQN is a variant of the classical Q-learning algorithms, which is integrated with the deep CNN model, where the convolutional filters seen in Fig. 22 (c) are used for representing the effects of receptive fields. One of the outputs of the deep CNN involved yields the specific value of the Q-function for a possible action. Beyond the DQN, substantial efforts have also been invested in improving the performance and stability, as exemplified by the double DQN [342] and the dueling DQN [343]. Thanks to the powerful feature representation capability of DNNs and of the reinforcement learning algorithms, DQN performs well in a range of compelling applications as exemplified by the AlphaGo, which is the first super-human program to defeat a professional human chess-player.

2) Applications: Deep reinforcement learning is eminently suitable for supporting the interaction in autonomous systems in terms of a higher level understanding of the visual world, which can be readily applied to a diverse analytically intractable problems in future wireless networks.

Given the intrinsic advantages of the reinforcement learning in environment in interactive decision making, it may play a significant role in the field of control decision [344], [345]. Specifically, Zhang et al. [344] proposed a model-free UAV trajectory control scheme relying on deep reinforcement learning for data collection in smart cities, where a powerful deep CNN was used for extracting the necessary features, while a DQN model was used for decision making. Given the sensing region and the related tasks, this algorithm supported efficient route planning for both the UAVs and mobile charging stations involved. In [345], a deep reinforcement learning aided communication-based train control system was conceived by Zhu et al. which jointly optimized the communication handoff strategy and the control functions, while reducing the energy consumption. Real channel measurements and real-time train position information were used for training the DQN model, which resulted in optimal communication and control decisions.

Furthermore, the resource allocation problems of wireless networks, such as energy scheduling, traffic scheduling, caching decisions, user association, etc. can be efficiently solved by deep reinforcement learning at a low computational complexity [153], [346]–[351]. For example, Zhang et al. [153] proposed a deep Q-learning model for system’s dynamic energy scheduling, which relied on the amalgamated stacked auto-encoder and Q-learning model. More specifically, the stacked auto-encoder was used for learning the state-action value function of each strategy in any of the available system states. Moreover, Xu et al. [346] proposed a deep reinforcement learning framework for power-efficient resource allocation in CRANs, which optimized the expected and cumulative long term power consumption, including the transmit power consumption, the sleep/active transition power consumption as well as the RRU’s power consumption. A two-step deep reinforcement learning aided decision making scheme was conceived, where the learning agent first decides on activating/deactivating the sleeping mode of
each RRU, and then determines the optimal beamformer’s power allocation. As for traffic scheduling, Zhu et al. [347] designed a stacked auto-encoder assisted deep learning model for packet transmission planning in the face of multiple contending channels in cognitive IoT networks, which aimed for maximizing the system’s throughput. In this architecture, MDP was used for modelling the system states. Given the large state-action space of the system, the stacked auto-encoder was used for constructing the mapping between the state and the action for accelerating the process of optimization. Furthermore, a deep Q-learning algorithm was conceived for designing both the cache allocation and the transmission rate in content-centric IoT networks for the sake of maximizing the long-term QoE [348], where He et al. considered both the networking cost as well as the users’ mean opinion score. In [349] and [350], He et al. proposed a DQN based user scheduling scheme for a cache-enabled opportunistic interference alignment (IA) assisted wireless network in the context of realistic time-varying channels formulated as a finite-state Markov model. More specifically, the DQN was constructed by relying on a sophisticated action-value function for the sake of reducing the computational complexity. Their simulation results demonstrated that the DQN aided IA assisted user scheduling was beneficial in terms of substantially improving the network’s throughput vs energy efficiency trade-off. To elaborate a little further, He et al. [351] utilized deep reinforcement learning for constructing their resource allocation policy relying on a joint optimization problem, which considered the programmable networking, information-centric caching as well as mobile edge computing in the context of connected vehicular scenarios. Moreover, the ε-greedy policy was utilized for striking an attractive trade-off between exploration and exploitation.

In order to curb the potentially excessive computational complexity resulting from having a large state space and to deal with its partial observability in cognitive radio networks, Naparstek and Cohen developed a distributed dynamic spectrum access scheme relying on deep multi-user reinforcement learning, where each user maps his/her current state to spectrum access actions with the aid of a DQN for the sake of maximizing the network’s utility which was achieved without any message exchanges [352]. Additionally, Wang et al. [353] proposed an adaptive DQN algorithm for dynamic multichannel access, which was capable of achieving a near-optimal performance outperforming the Myopic policy [354] and the Whittle’s Index-based heuristic algorithms [355] in complex scenarios.

Table VI lists some typical applications of deep reinforcement learning in wireless networks.

VII. FUTURE RESEARCH AND CONCLUSIONS

A. Future Machine Learning Aided Network Applications

It is anticipated that the next-generation network infrastructure will be migrating to a service-oriented architecture by taking advantage of both NFV and SDN for supporting eMBB, mMTC, and uRLLC along with attractive social media and augmented reality (AR) aided entertainment, smart manufacturing, smart energy, e-health and autonomous driving.

Given that ML can be readily executed both by agents, as well as by edge- and cloud-computing, it is eminently suitable for supporting sophisticated networking functionalities. We can categorize ML-based networking into:

- **ML-Aware Networking**: The networking entities are aware of the availability of ML functionalities and optionally might exploit advantages of ML, for example, when the user load escalates.
- **On-/Off-line ML-Aided Networking**: The network functionalities must resort to invoking ML, for example, because the search-space of joint scheduling

---

<table>
<thead>
<tr>
<th>Paper</th>
<th>Application</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[314]</td>
<td>channel estimation</td>
<td>DNN</td>
<td>learn nonlinear distortion, interference and frequency selectivity of wireless channels</td>
</tr>
<tr>
<td>[321]</td>
<td>modulation classification</td>
<td>RNN</td>
<td>capture amplitude and phrase information without expert knowledge</td>
</tr>
<tr>
<td>[322]</td>
<td>signal detection</td>
<td>DNN</td>
<td>transmit signal detection from noisy and corrupted signals without underlying CSI</td>
</tr>
<tr>
<td>[331]</td>
<td>interference identification</td>
<td>CNN</td>
<td>learn features through self-optimization during the GPU based training process</td>
</tr>
<tr>
<td>[325]</td>
<td>PHY representation</td>
<td>DNN</td>
<td>represent single user MIMO system relying on DNN aided auto-encoder</td>
</tr>
<tr>
<td>[326]</td>
<td>software-defined radio</td>
<td>DNN</td>
<td>be capable of easing the current restriction on short block lengths</td>
</tr>
<tr>
<td>[327]</td>
<td>traffic prediction</td>
<td>DNN</td>
<td>deep auto-encoder and LSTM for modeling spatial and temporal features</td>
</tr>
<tr>
<td>[328]</td>
<td>packet routing</td>
<td>DNN</td>
<td>traffic routing scheme with little signal overhead, large throughput and small delay</td>
</tr>
<tr>
<td>[329]</td>
<td>traffic control</td>
<td>CNN</td>
<td>consider previous network abnormalities, lower average delay and packet loss rate</td>
</tr>
<tr>
<td>[330]</td>
<td>traffic offloading</td>
<td>DNN</td>
<td>integrate both DNN structure and edge computing technique into multimedia IoT</td>
</tr>
<tr>
<td>[331]</td>
<td>power control</td>
<td>DNN</td>
<td>an almost-real-time power control algorithm in interference-limited wireless networks</td>
</tr>
<tr>
<td>[332]</td>
<td>network security</td>
<td>DNN</td>
<td>a real-time detection of malicious false data injection attack in smart grid</td>
</tr>
<tr>
<td>[333]</td>
<td>indoor localization</td>
<td>DNN</td>
<td>device-free wireless localization and recognition by learning from ambient wireless signals</td>
</tr>
<tr>
<td>[334]</td>
<td>mobility prediction</td>
<td>CNN</td>
<td>learn human mobility pattern relying on analyzing continuous mobile data stream</td>
</tr>
<tr>
<td>[335]</td>
<td>transportation mode</td>
<td>RNN</td>
<td>predict both human mobility and transportation mode for large-scale transport networks</td>
</tr>
<tr>
<td>[336]</td>
<td>activity prediction</td>
<td>RNN</td>
<td>characterise primary users’ activity in CR with different traffic distribution</td>
</tr>
<tr>
<td>[337]</td>
<td>traffic prediction</td>
<td>CNN</td>
<td>traffic modeling and prediction based on a convolutional long short-term memory network</td>
</tr>
<tr>
<td>[338]</td>
<td>resource allocation</td>
<td>DNN</td>
<td>universal DNN framework for a number of common wireless resource allocation problems</td>
</tr>
<tr>
<td>[339]</td>
<td>PHY representation</td>
<td>CNN</td>
<td>modulation, equalization, and demodulation with a convolutional autoencoder</td>
</tr>
<tr>
<td>[340]</td>
<td>link scheduling</td>
<td>CNN</td>
<td>optimal scheduling of interfering links in dense wireless networks</td>
</tr>
</tbody>
</table>

---

8The Myopic policy is one that simply optimizes the average immediate reward. It is called ‘myopic’ in the sense that it merely considers the single criterion, while it has the advantage of being easy to implement.

9The Whittle’s Index-based algorithm is one of index heuristic algorithms which is designed to solve a problem in a more efficient way than traditional methods often used for solving NP-hard problems. More explicitly, the Whittle’s Index policy is a low-complexity heuristic policy.
and resource-allocation is excessive for full search. Depending on the specific application, this may rely either on off-line ML or on on-line ML.

- **On-line ML-Aided Networking**: The network functionalities must resort to on-line ML, for example, anticipatory mobility management requires on-line ML.

To facilitate prompt ML-aided network control, new tele-traffic samples may potentially be included for near-instantaneous learning according to the following use cases:

- In contrast to traditional networking, near-instantaneous mobility trajectory inference relies on on-line datasets for training the ML module.
- Context information relying on location information inference or on user experience as a function of latency, jitter, packet dropping probability can be taken into account.
- Planning and policy relying on ML functionalities for improving a certain work flow, an agent’s specific policy and/or rewards, along with the extracted features or learning models may be exploited, where the agent may be constituted either by a network entity or a smart UE.

To expound a little further, ML can be used for diverse application scenarios, such as:

- **Channel State Estimation**: Having accurate CSI is critical for all air-interface techniques, which has been inferred/estimated with the aid of deep learning.
- **User Behavior Inference**: User behavior may be characterized for example in terms of their mobility patterns, while supporting high-quality network management and mobility management with the aid of big data analysis relying on ML.
- **Traffic Prediction**: Network intelligence and user mobility patterns can be used for predicting the wired/wireless network traffic in support of efficient network/radio resource allocation.
- **Network Security**: ML is also capable of enhancing the network security, whilst detecting attacks and intrusions.
- **Anticipatory Networking Mechanism**: Online learning can be used to develop new predictive networking mechanism. For example, anticipatory mobility management (AMM) using Naive Bayesian and recursive belief update was conceived for ultra-low latency wireless networking [359], [360].

Based on the aforementioned future network architectural features, we list a range of research ideas on promising applications of ML in future wireless networks.

- **UAV-aided networking**: Given the agility of UAV nodes as well as the bursty and often unpredictable nature of terrestrial wireless traffic, ML models can be used both for predicting the traffic demand and for adaptively adjusting the UAVs’ location.
- **mMTC and uRLLC network**: While wireless networks have primarily served communications among individuals, in the era of the IoT, wireless networking also supports myriads of machines and intelligent devices. In this era a pair of 5G operational modes - namely mMTC and uRLLC - are expected to play key roles [361]. ML is capable of enhancing conventional networks designed for mMTC, for example by invoking reinforcement learning to appropriately select the access points of MTC [362]. The uRLLC mode of operation constitutes a rather young technological territory, which can be jointly designed with mMTC [363]. To reduce the network’s latency from hundreds of milliseconds as experienced in state-of-the-art mobile communications to the desirable range of just a few milliseconds, ML is capable of supporting so-called anticipatory mobility management, which integrates the naive Bayesian classification of the previously used APs and geographical regression for the predictive analysis of data. Another example of disruptive technical trend is to investigate how wireless networking impacts on the smart agents operated by ML.
- **Narrow band IoT (NB-IoT)**: NB-IoT allows a large number of low-power devices to connect to the cellular network, where the devices require long-term Internet access and dense wireless coverage. ML algorithms are capable of supporting intelligent resource allocation, optimal AP deployment and efficient access.
- **Socially-aware wireless networking**: The operation of socially-aware wireless networks relies on a variety of social attributes, where ML schemes are beneficial in terms of facilitating feature extraction, social group-of-interest formation, classification and prediction of

<table>
<thead>
<tr>
<th>Paper</th>
<th>Scenario</th>
<th>Application</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[344]</td>
<td>UAV network</td>
<td>trajectory control</td>
<td>a model-free UAV trajectory control scheme in smart cities relying on DQN</td>
</tr>
<tr>
<td>[345]</td>
<td>ITS</td>
<td>train control</td>
<td>jointly optimize the communication handoff strategy and control performances</td>
</tr>
<tr>
<td>[353]</td>
<td>energy-aware network</td>
<td>energy scheduling</td>
<td>associate the stacked auto-encoder and the deep Q-learning model</td>
</tr>
<tr>
<td>[346]</td>
<td>CRAN</td>
<td>power allocation</td>
<td>decide RRU’s sleeping mode and the optimal beamformer’s power allocation</td>
</tr>
<tr>
<td>[347]</td>
<td>cognitive IoT</td>
<td>traffic scheduling</td>
<td>construct the mapping between states and actions relying on stacked auto-encoder</td>
</tr>
<tr>
<td>[348]</td>
<td>content-centric IoT</td>
<td>cache allocation</td>
<td>jointly design cache allocation and transmission rate for maximizing long-term QoE</td>
</tr>
<tr>
<td>[350]</td>
<td>IA network</td>
<td>user scheduling</td>
<td>obtain the action-value function relying on DQN for lowering complexity</td>
</tr>
<tr>
<td>[351]</td>
<td>vehicular network</td>
<td>resource allocation</td>
<td>consider programmable SDN, information-centric caching and mobile edge computing</td>
</tr>
<tr>
<td>[352]</td>
<td>CR network</td>
<td>spectrum access</td>
<td>distributed spectrum access for maximizing network utility without message exchanges</td>
</tr>
<tr>
<td>[353]</td>
<td>CR network</td>
<td>multichannel access</td>
<td>adaptive DQN aided multichannel access yielding a near-optimal performance</td>
</tr>
<tr>
<td>[356]</td>
<td>MEC network</td>
<td>resource allocation</td>
<td>DQN assisted cache, computation and power resources joint association</td>
</tr>
<tr>
<td>[357]</td>
<td>CR network</td>
<td>resource allocation</td>
<td>energy efficiency optimization for distributed cooperative spectrum sensing</td>
</tr>
<tr>
<td>[358]</td>
<td>vehicular network</td>
<td>resource allocation</td>
<td>edge computing and caching resources dynamic association for IoV</td>
</tr>
</tbody>
</table>
these social attributes, such as human mobility, social relations, behavior preference, etc.

- **Wireless Virtual reality (VR) networks**: VR networks facilitate for the users to experience and interact with immersive environments, which requires flawless audio and video data processing capability. ML algorithms have the potential of circumventing the conception of complex joint source- and channel-coding schemes by further developing the auto-encoder principles.

- **Network integration, representation and design**: ML may provide an alternative for network representation, where we can integrate each classic communication-theoretic blocks including source- and channel-encoding, modulation, demodulation, decoding, etc. into a "black box". By simply learning from and processing previous input and output signals, the receivers become capable of adaptively understanding the operational mechanism of the "black box" considered.

- **Wireless network tomography**: State-of-the-art wireless networks support a vast number of nodes, such as those of the IoT, where the provision of global information is practically impossible for each node. Hence a new class of problems arises in the context of distributed wireless networks, which is related to the acquisition of network-related information. Classic network tomography [364] defines the problem as: \( \mathbf{Y} = \mathbf{AX} \), where \( \mathbf{X} \) is a \( J \)-dimensional vector of the network's dynamically fluctuating parameters, such as the link delay or traffic activity, \( \mathbf{Y} \) is the \( I \)-dimensional vector of measurements and \( \mathbf{A} \) is known as the routing matrix that is likely to define a rank-deficient scenario in network tomography. Such inverse matrix problems of inference may be solved by EM or Markov Chain Monte Carlo methods, and they are mathematically similar to the regression problems of ML. Network tomography can be applied to the inference of a wireless network’s status and the subsequent optimization of the network’s operation [365]. Further generalization using statistical inference and singular value decomposition for statistically characterizing and enhancing the radio resource utilization relies on accurate spectrum sensing in cognitive radio networks [366]. Another interesting application is to infer activities beyond sight (i.e. see-through walls) by applying variance-based radio tomography, Tikhonov least-squares regularization, and Kalman tracking [367].

- **Large-scale Pareto-optimization**: In the above-mentioned optimization problems the designer is routinely faced by having to strike a trade-off amongst conflicting factors, as exemplified by the ubiquitous bandwidth- vs. power-efficiency trade-off in wireless communications. However, there are numerous other trade-offs to be taken into account, such as for example the data-security vs. integrity trade-off or the BER vs. delay trade-off, as well as the BER vs. complexity trade-off, just to mention a few. Hence it is desirable to find all the Pareto-optimal operating points of future wireless systems, which jointly constitute the optimal Pareto-front of multi-component optimization problems [134]. Again, none of the solutions on the optimal Pareto-front may be improved without degrading at least one of the parameters. Naturally, as we mentioned, the optimization search-space is extended every time, when a new parameter is incorporated. Hence ML constitutes a promising tool for the fertile ground of multi-component Pareto-optimization of a plethora of systems.

**B. Future Wireless Networking Techniques for Multi-Agent Systems Using Machine Learning**

In addition to applying ML in wireless networking, the question of how to design a wireless network for example for multi-robot systems or for multi-agent systems arises.

In this context, Legg and Hutter [368] gave an informal definition of machine intelligence: "Intelligence measures an agent’s ability to achieve goals in a wide range of environments.” Distributed artificial intelligence has been brought into the lime-light of artificial intelligence research over three decades ago [369], which has a pair of popular sub-disciplines: distributed problem solving and multi-agent systems. Distributed problem solving typically decomposes tasks into several not completely independent sub-problems that can be executed on different processors and then synthesizes a solution. On the other hand, multi-agent systems are typically constituted by robots, having certain goals and actions in challenging operating environments. Although the communication supporting the actions of agents has been studied for a while, under idealized simplifying assumptions, the features of realistic wireless communications and networking have not been taken into consideration [370]–[372].

An interesting study disseminated in [373] has considered the collective behavior of autonomous vehicles moving across Manhattan in New York city. Each autonomous vehicle acted as an intelligent agent using reinforcement learning. In contrast to human-to-human personal communication, the reward map and policy of another autonomous vehicle within the interaction range would be useful information to exchange. In this wireless networking scenario, the freshness of such information is critical hence ultra-low-latency wireless networking is preferred. In fact, for collaborative robots relying on their own machine learning algorithms while aiming for achieving a common goal, ultra-low-latency wireless networking is of paramount importance. Given the plethora of open research problems in networked multi-agent systems, such as their network topology, this is a compelling research area.

**C. Conclusions**

In summary, we have reviewed the thirty-year history of ML. We surveyed a range of typical algorithms and models conceived for supervised learning, unsupervised learning,
reinforcement learning as well as deep learning, respectively. Furthermore, we have highlighted the development tendency of wireless network techniques and a variety of representative scenarios for future wireless networks as seen in Fig. 5 and Fig. 8. We also have provided a case-by-case description of numerous compelling applications relying on ML algorithms in wireless networks as shown in Table VII, followed by a pair of detailed application examples relying on our recent research results. In comparison with state-of-the-art survey papers seen in Fig. 1, our paper overviews all the four popular kinds of learning schemes and their applications in future wireless networks, which has a full scope of how ML algorithms bear fruits in the past decades in wireless networks. Indeed, powerful ML methods are poised to occupy an important position in tackling intractable and hitherto uncharted scenarios and applications in wireless networks.

APPENDIX

The applications of ML in wireless networks are summarized in Table VII.

REFERENCES


<table>
<thead>
<tr>
<th>Scenario</th>
<th>Application</th>
<th>Learning Technique</th>
<th>Specific Method</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Air ad hoc network</td>
<td>packet routing</td>
<td>reinforcement learning</td>
<td>on-policy SARSA</td>
<td>[296]</td>
</tr>
<tr>
<td>body area network</td>
<td>power control</td>
<td>reinforcement learning</td>
<td>Q-learning</td>
<td>[299]</td>
</tr>
<tr>
<td>cellular network</td>
<td>dynamic channel allocation</td>
<td>reinforcement learning</td>
<td>reduced-state SARSA</td>
<td>[294]</td>
</tr>
<tr>
<td>cellular network</td>
<td>traffic prediction</td>
<td>deep learning</td>
<td>DNN</td>
<td>[327]</td>
</tr>
<tr>
<td>cellular network</td>
<td>relay selection</td>
<td>unsupervised learning</td>
<td>K-means</td>
<td>[244]</td>
</tr>
<tr>
<td>cellular network</td>
<td>interference alignment</td>
<td>deep learning</td>
<td>DQN</td>
<td>[358]</td>
</tr>
<tr>
<td>centralizer CR network</td>
<td>channel selection</td>
<td>supervised learning</td>
<td>SVM</td>
<td>[227]</td>
</tr>
<tr>
<td>cognitive femtocell network</td>
<td>interference control</td>
<td>reinforcement learning</td>
<td>centralized Q-learning</td>
<td>[302]</td>
</tr>
<tr>
<td>CR ad hoc network</td>
<td>signal detection</td>
<td>unsupervised learning</td>
<td>ICA</td>
<td>[261]</td>
</tr>
<tr>
<td>CR network</td>
<td>activity prediction</td>
<td>deep learning</td>
<td>Q-learning</td>
<td>[381]</td>
</tr>
<tr>
<td>CR network</td>
<td>spectrum access</td>
<td>deep learning</td>
<td>deep reinforcement learning</td>
<td>[352]</td>
</tr>
<tr>
<td>CR network</td>
<td>channel access</td>
<td>deep learning</td>
<td>DQN</td>
<td>[353]</td>
</tr>
<tr>
<td>CR network</td>
<td>energy scheduling</td>
<td>deep learning</td>
<td>deep reinforcement learning</td>
<td>[357]</td>
</tr>
<tr>
<td>CR network</td>
<td>PHY representation</td>
<td>deep learning</td>
<td>INN</td>
<td>[324]</td>
</tr>
<tr>
<td>CRAN network</td>
<td>power allocation</td>
<td>deep learning</td>
<td>deep reinforcement learning</td>
<td>[346]</td>
</tr>
<tr>
<td>dense cellular network</td>
<td>link scheduling</td>
<td>deep learning</td>
<td>CNN</td>
<td>[362]</td>
</tr>
<tr>
<td>energy-aware network</td>
<td>energy scheduling</td>
<td>deep learning</td>
<td>DQN</td>
<td>[183]</td>
</tr>
<tr>
<td>heterogeneous network</td>
<td>resource allocation</td>
<td>reinforcement learning</td>
<td>on-policy SARSA</td>
<td>[297]</td>
</tr>
<tr>
<td>heterogeneous network</td>
<td>packet routing</td>
<td>unsupervised learning</td>
<td>ICA</td>
<td>[265]</td>
</tr>
<tr>
<td>heterogeneous network</td>
<td>data estimation</td>
<td>supervised learning</td>
<td>H-SVM, SVM</td>
<td>[221]</td>
</tr>
<tr>
<td>heterogeneous network</td>
<td>gateway deployment</td>
<td>unsupervised learning</td>
<td>K-means</td>
<td>[233]</td>
</tr>
<tr>
<td>IoT network</td>
<td>traffic scheduling</td>
<td>deep learning</td>
<td>deep Q-learning</td>
<td>[341]</td>
</tr>
<tr>
<td>IoT network</td>
<td>cache allocation</td>
<td>deep learning</td>
<td>deep reinforcement learning</td>
<td>[348]</td>
</tr>
<tr>
<td>IoT network</td>
<td>traffic estimation</td>
<td>supervised learning</td>
<td>regression</td>
<td>[199]</td>
</tr>
<tr>
<td>IoT network</td>
<td>traffic offloading</td>
<td>deep learning</td>
<td>INN</td>
<td>[357]</td>
</tr>
<tr>
<td>M2M network</td>
<td>user detection</td>
<td>unsupervised learning</td>
<td>EM algorithm, Bayes method</td>
<td>[255]</td>
</tr>
<tr>
<td>MANET+ network</td>
<td>behavior learning</td>
<td>supervised learning</td>
<td>SVM</td>
<td>[225]</td>
</tr>
<tr>
<td>MANET+ network</td>
<td>anomaly detection</td>
<td>unsupervised learning</td>
<td>K-means</td>
<td>[249]</td>
</tr>
<tr>
<td>MEC network</td>
<td>resource allocation</td>
<td>deep learning</td>
<td>DQN</td>
<td>[356]</td>
</tr>
<tr>
<td>MIMO</td>
<td>blind transceiver</td>
<td>unsupervised learning</td>
<td>K-means</td>
<td>[258]</td>
</tr>
<tr>
<td>MIMO</td>
<td>channel estimation</td>
<td>unsupervised learning</td>
<td>EM algorithm, Bayes method</td>
<td>[259]</td>
</tr>
<tr>
<td>MIMO</td>
<td>symbol detection</td>
<td>unsupervised learning</td>
<td>EM algorithm, Bayes method</td>
<td>[258]</td>
</tr>
<tr>
<td>MIMO</td>
<td>channel estimation</td>
<td>unsupervised learning</td>
<td>ICA</td>
<td>[262]</td>
</tr>
<tr>
<td>MIMO</td>
<td>blind receiver</td>
<td>unsupervised learning</td>
<td>ICA</td>
<td>[263]</td>
</tr>
<tr>
<td>MIMO</td>
<td>PHY representation</td>
<td>deep learning</td>
<td>INN</td>
<td>[344]</td>
</tr>
<tr>
<td>MIMO</td>
<td>modulation and coding</td>
<td>reinforcement learning</td>
<td>Q-learning</td>
<td>[300]</td>
</tr>
<tr>
<td>mobile computing network</td>
<td>user location</td>
<td>supervised learning</td>
<td>SVM</td>
<td>[223]</td>
</tr>
<tr>
<td>multihop M2M network</td>
<td>primary user detection</td>
<td>unsupervised learning</td>
<td>EM algorithm</td>
<td>[241]</td>
</tr>
<tr>
<td>multiantenna CR network</td>
<td>channel state detection</td>
<td>unsupervised learning</td>
<td>EM algorithm</td>
<td>[252]</td>
</tr>
<tr>
<td>OFDM network</td>
<td>channel estimation</td>
<td>deep learning</td>
<td>DNN</td>
<td>[312]</td>
</tr>
<tr>
<td>P2P network</td>
<td>power allocation</td>
<td>reinforcement learning</td>
<td>SARSA</td>
<td>[259]</td>
</tr>
<tr>
<td>smart grid network</td>
<td>network security</td>
<td>deep learning</td>
<td>IHN</td>
<td>[332]</td>
</tr>
<tr>
<td>software-defined network</td>
<td>resource allocation</td>
<td>deep learning</td>
<td>INN</td>
<td>[326]</td>
</tr>
<tr>
<td>UAV network</td>
<td>trajectory control</td>
<td>deep learning</td>
<td>DQN</td>
<td>[344]</td>
</tr>
<tr>
<td>vehicular network</td>
<td>traffic prediction</td>
<td>deep learning</td>
<td>deep reinforcement learning</td>
<td>[345]</td>
</tr>
<tr>
<td>vehicular network</td>
<td>traffic prediction</td>
<td>supervised learning</td>
<td>IKN</td>
<td>[205]</td>
</tr>
<tr>
<td>vehicular network</td>
<td>resource allocation</td>
<td>deep learning</td>
<td>deep reinforcement learning</td>
<td>[351]</td>
</tr>
<tr>
<td>vehicular network</td>
<td>traffic prediction</td>
<td>supervised learning</td>
<td>KNN</td>
<td>[205]</td>
</tr>
<tr>
<td>vehicular network</td>
<td>resource allocation</td>
<td>deep learning</td>
<td>deep reinforcement learning</td>
<td>[351]</td>
</tr>
<tr>
<td>vehicular network</td>
<td>interference elimination</td>
<td>supervised learning</td>
<td>KNN</td>
<td>[208]</td>
</tr>
<tr>
<td>Wi-Fi network</td>
<td>indoor localization</td>
<td>deep learning</td>
<td>DNN</td>
<td>[314]</td>
</tr>
<tr>
<td>Wi-Fi network</td>
<td>attacker counting</td>
<td>supervised learning</td>
<td>SVM</td>
<td>[228]</td>
</tr>
<tr>
<td>Wi-Fi network</td>
<td>indoor location</td>
<td>unsupervised learning</td>
<td>PCA</td>
<td>[258]</td>
</tr>
<tr>
<td>Wi-Fi network</td>
<td>wireless coexistence</td>
<td>supervised learning</td>
<td>logistic regression</td>
<td>[191]</td>
</tr>
<tr>
<td>wireless communication</td>
<td>signal detection</td>
<td>unsupervised learning</td>
<td>K-means</td>
<td>[249]</td>
</tr>
<tr>
<td>wireless communication</td>
<td>modulation classification</td>
<td>supervised learning</td>
<td>IKN</td>
<td>[205]</td>
</tr>
<tr>
<td>wireless communication</td>
<td>interference cancellation</td>
<td>unsupervised learning</td>
<td>ICA</td>
<td>[264]</td>
</tr>
<tr>
<td>wireless communication</td>
<td>modulation classification</td>
<td>deep learning</td>
<td>IKN, CNN</td>
<td>[321]</td>
</tr>
<tr>
<td>wireless communication</td>
<td>signal detection</td>
<td>deep learning</td>
<td>IKN</td>
<td>[321]</td>
</tr>
<tr>
<td>wireless communication</td>
<td>interference identification</td>
<td>deep learning</td>
<td>CNN</td>
<td>[313]</td>
</tr>
<tr>
<td>wireless communication</td>
<td>mobility prediction</td>
<td>deep learning</td>
<td>CNN</td>
<td>[320]</td>
</tr>
<tr>
<td>wireless communication</td>
<td>mobility prediction</td>
<td>deep learning</td>
<td>CNN</td>
<td>[334]</td>
</tr>
<tr>
<td>wireless communication</td>
<td>transportation mode</td>
<td>deep learning</td>
<td>IKN, DNN</td>
<td>[327]</td>
</tr>
<tr>
<td>wireless communication</td>
<td>PHY representation</td>
<td>deep learning</td>
<td>CNN</td>
<td>[340]</td>
</tr>
<tr>
<td>wireless mesh networks</td>
<td>traffic control</td>
<td>deep learning</td>
<td>CNN</td>
<td>[329]</td>
</tr>
<tr>
<td>wireless network</td>
<td>antenna selection</td>
<td>supervised learning</td>
<td>Bayes method, SVM</td>
<td>[232]</td>
</tr>
<tr>
<td>wireless network</td>
<td>QoS prediction</td>
<td>supervised learning</td>
<td>Bayes, SVM</td>
<td>[236]</td>
</tr>
<tr>
<td>wireless network</td>
<td>intrusion detection</td>
<td>unsupervised learning</td>
<td>K-means</td>
<td>[247]</td>
</tr>
<tr>
<td>wireless network</td>
<td>power control</td>
<td>deep learning</td>
<td>DNN</td>
<td>[331]</td>
</tr>
<tr>
<td>wireless network</td>
<td>resource allocation</td>
<td>deep learning</td>
<td>INN</td>
<td>[339]</td>
</tr>
<tr>
<td>wireless sensor network</td>
<td>sensor partitioning</td>
<td>unsupervised learning</td>
<td>K-means</td>
<td>[255]</td>
</tr>
<tr>
<td>wireless sensor network</td>
<td>interference estimate</td>
<td>supervised learning</td>
<td>regression</td>
<td>[195]</td>
</tr>
<tr>
<td>wireless sensor network</td>
<td>spectrum sensing</td>
<td>supervised learning</td>
<td>regression</td>
<td>[190]</td>
</tr>
<tr>
<td>wireless sensor network</td>
<td>PHY authentication</td>
<td>supervised learning</td>
<td>logistic regression</td>
<td>[198]</td>
</tr>
<tr>
<td>wireless network</td>
<td>map reconstruction</td>
<td>supervised learning</td>
<td>regression</td>
<td>[200]</td>
</tr>
<tr>
<td>wireless network</td>
<td>wireless incarcitation</td>
<td>supervised learning</td>
<td>regression, SVM</td>
<td>[192]</td>
</tr>
<tr>
<td>wireless sensor network</td>
<td>anomaly detection</td>
<td>supervised learning</td>
<td>KNN</td>
<td>[208]</td>
</tr>
<tr>
<td>wireless sensor network</td>
<td>missing data estimation</td>
<td>supervised learning</td>
<td>KNN</td>
<td>[208]</td>
</tr>
<tr>
<td>wireless sensor network</td>
<td>localization estimation</td>
<td>supervised learning</td>
<td>SVM</td>
<td>[208]</td>
</tr>
<tr>
<td>wireless sensor network</td>
<td>signal classification</td>
<td>supervised learning</td>
<td>SVM, DNN</td>
<td>[226]</td>
</tr>
<tr>
<td>wireless sensor network</td>
<td>network association</td>
<td>supervised learning</td>
<td>Bayes</td>
<td>[233]</td>
</tr>
<tr>
<td>wireless sensor network</td>
<td>network state detection</td>
<td>unsupervised learning</td>
<td>EM algorithm</td>
<td>[254]</td>
</tr>
<tr>
<td>wireless sensor network</td>
<td>source localization</td>
<td>unsupervised learning</td>
<td>EM algorithm</td>
<td>[254]</td>
</tr>
<tr>
<td>wireless sensor network</td>
<td>data aggregation</td>
<td>unsupervised learning</td>
<td>PCA</td>
<td>[259]</td>
</tr>
<tr>
<td>wireless sensor network</td>
<td>data recovery</td>
<td>unsupervised learning</td>
<td>PCA</td>
<td>[260]</td>
</tr>
<tr>
<td>wireless sensor network</td>
<td>steering control</td>
<td>reinforcement learning</td>
<td>Q-learning</td>
<td>[301]</td>
</tr>
<tr>
<td>WLAN</td>
<td>indoor localization</td>
<td>supervised learning</td>
<td>Bayes method, SVM</td>
<td>[235]</td>
</tr>
<tr>
<td>WLAN</td>
<td>anomaly detection</td>
<td>supervised learning</td>
<td>naive Bayes method</td>
<td>[234]</td>
</tr>
</tbody>
</table>


Yong Ren (M'11-SM'16) received his B.S., M.S and Ph.D. degrees in electronic engineering from Harbin Institute of Technology, China, in 1984, 1987, and 1994, respectively. He worked as a post doctor at Department of Electronics Engineering, Tsinghua University, China from 1995 to 1997. Now he is a professor of Department of Electronics Engineering and the director of the Complexity Engineered Systems Lab in Tsinghua University. He holds more than 300 technical papers in the behavior of computer network, P2P network and cognitive networks. He has served as a reviewer of IEICE Transactions on Communications, Digital Signal Processing, Chinese Physics Letters, Chinese Journal of Electronics, Chinese Journal of Computer Science and Technology, Chinese Journal of Aeronautics and so on. His current research interests include complex systems theory and its applications to the optimization and information sharing of the Internet, Internet of Things and ubiquitous network, cognitive networks and Cyber-Physical Systems.

Kwang-Cheng Chen (F’07) received the B.S. degree from the National Taiwan University in 1983, and the M.S. and Ph.D. degrees from the University of Maryland, College Park, MD, USA, in 1987 and 1989, all in electrical engineering. From 1987 to 1998, he was with SSE, COMSAT, IBM Thomas J. Watson Research Center, and National Tsing Hua University, working on mobile communications and networks. From 1998 to 2016, he was a Distinguished Professor with the National Taiwan University, Taipei, Taiwan. He also served as the Director of the Graduate Institute of Communication Engineering, the Director of the Communication Research Center, and the Associate Dean for Academic Affairs with the College of Electrical Engineering and Computer Science, from 2009 to 2015. Since 2016, he has been a Professor of electrical engineering with the University of South Florida, Tampa, FL, USA. His recent research interests include wireless networks, artificial intelligence and machine learning, IoT and CPS, social networks, and cybersecurity. He has been actively involved in the organization of various IEEE conferences as the General/TPC Chair/Co-Chair, and has served in editorships with a few IEEE journals. He also actively participates in and has contributed essential technology to various IEEE 802, Bluetooth, LTE and LTE-A, 5G-NR, and ITU-T FG ML5G wireless standards. He has received a number of awards, including the 2011 IEEE COMSOC WTC Recognition Award, the 2014 IEEE Jack Neubauer Memorial Award, and the 2014 IEEE COMSOC AP Outstanding Paper Award.

Lajos Hanzo (http://www-mobile.ecs.soton.ac.uk, https://en.wikipedia.org/wiki/Lajos_Hanzo) FREng, FIEEE, FIET, Fellow of EURASIP, DSc received his Master degree in electronics in 1976 and his doctorate in 1983. He holds an honorary doctorate by the Technical University of Budapest (2009) and by the University of Edinburgh (2015). He is a Foreign Member of the Hungarian Academy of Sciences and a former Editor-in-Chief of the IEEE Press. He has served as Governor of both IEEE ComSoc and of VTS. He has published 1900+ contributions at IEEE Xplore, 18 Wiley-IEEE Press books and has helped the fast-track career of 119 PhD students.