Machine Learning: A Catalyst for THz Wireless Networks

Alexandros-Apostolos A. Boulogeorgos[1]*, Edwin Yaqub[2], Marco di Renzo[3], Angeliki Alexiou[1], Rachana Desai[2] and Ralf Klinkenberg[2]

[1]Digital Systems, University of Piraeus, Piraeus, Greece, [2]RapidMiner GmbH, Dortmund, Germany, [3]Universite Paris-Saclay, CNRS, Centrale Supelec, Laboratoire des Signaux et Systémes, Gif-sur-Yvette, France

With the vision to transform the current wireless network into a cyber-physical intelligent platform capable of supporting bandwidth-hungry and latency-constrained applications, both academia and industry turned their attention to the development of artificial intelligence (AI) enabled terahertz (THz) wireless networks. In this article, we list the applications of THz wireless systems in the beyond fifth generation era and discuss their enabling technologies and fundamental challenges that can be formulated as AI problems. These problems are related to physical, medium/multiple access control, radio resource management, network and transport layer. For each of them, we report the AI approaches, which have been recognized as possible solutions in the technical literature, emphasizing their principles and limitations. Finally, we provide an insightful discussion concerning research gaps and possible future directions.

1 INTRODUCTION

As a response to the spectrum scarcity problem that was created due to the aggressive proliferation of wireless devices and quality-of-service (QoS) and quality-of-experience (QoE) hungry services, which are expected to support a broad range of diverse multi-scale and multi-environment applications, sixth-generation (6G) wireless networks adopt higher frequency bands, such as terahertz (THz) that ranges from 0.1 to 10 THz (Boulogeorgos et al., 2018b; Boulogeorgos and Alexiou, 2020c; Boulogeorgos and Alexiou, 2020d; Boulogeorgos et al., 2018a). In more detail and according to the IEEE 802.15.3 days standard (IEEE Standard for Information technology, 2009; IEEE Standard for High Data Rate Wireless Multi-Media Networks, 2017), THz wireless communications are recognized as the pillar technological enabler of a varied set of use cases stretching from in-body nano-scale, to indoor and outdoor wireless personal/local area and fronthaul/backhaul networks. Nano-scale applications require compact transceiver designs and self-organized ad-hoc network topologies. On the other hand, macro-scale applications demand flexibility, sustainability, adaptability in an ever changing heterogeneous environment, and security. Moreover, supporting high data-rate that may reach 1 Tb/s, and energy-efficient massive connectivity are only some of the key demands. To address the aforementioned requirements, artificial intelligence (AI), in combination with novel structures capable of altering the wireless environment, have been regarded as complementary pillars to 6G wireless THz systems.

AI is expected to enable a series of new features in next-generation networks, including, but not limited to, self-aggregation, context awareness, self-configuration as well as opportunistic deployment (Dang et al., 2020). In addition, integrating AI in wireless networks is envisioned to bring a revolutionary transformation of conventional cognitive radio systems into intelligent

platforms by unlocking the full potential of radio signals and exploiting new degrees-of-freedom (DoF) (Letaief et al., 2019; Saad et al., 2020). Identifying this opportunity, a significant amount of researchers turned their eyes on AI-empowered wireless systems and specifically in machine learning (ML) algorithms (see e.g., (Wang J. et al., 2020; Liu Y. et al., 2020; Jiang, 2020; Boulogeorgos et al., 2021a) and references therein). In more detail, in (Wang J. et al., 2020), the authors reviewed the 30-year history of ML highlighting its application in heterogeneous networks, cognitive radios, device-to-device communications and Internet-of-Things (IoT). Likewise, in (Liu Y. et al., 2020), big data analysis is combined with ML in order to predict the requirements of mobile users and enhance the performance of "social network-aware wireless." Moreover, in (Jiang, 2020), a brief survey was conducted that summarizes the basic physical layer (PHY) authentication schemes that employ ML in fifth generation (5G)-based IoT. Finally, in (Boulogeorgos et al., 2021a), the authors provided a systematic and comprehensive review of the ML approaches that was employed to address a number of nano-scale biomedical challenges, including once that refer to molecular and nano-scale THz communications.

The methodologies, which are presented in the aforementioned contributions, are tightly connected to the communication technology characteristics and building blocks of radio and microwave communication systems and networks, which are inherently different from higher frequency bands that can support higher data rates, and propagation environments with conventional and unconventional structures, like RIS. Additionally, in order to quantify the AI-approaches efficiency, new key performance indicators (KPIs) need to be defined. Motivated by this, this article focuses on reporting the role of AI in THz wireless networks. In particular, we first identify the THz wireless systems particularities that require the adoption of AI. Building upon this, we present a brief survey that summarizes the contributions in this area and focus on indicative AI approaches that are expected to play an important role in different layers of the THz wireless networks. Finally, possible future research directions are provided.

The structure of the rest of the paper is as follows. **Section 2** discusses the role of ML in THz wireless systems and networks. In particular, a systematic review is conducted concerning the ML algorithms that are used to solve PHY, medium access control (MAC), radio resource management (RRM), network and transport layer related problems. Moreover, in **Section 2**, the ML techniques that was employed or have the potential to be adopted to THz wireless systems and networks are documented. In **Section 3**, the aforementioned ML algorithms are reviewed and a methodology to select a suitable ML algorithm is presented. Likewise, in **Section 4**, ML deployment strategies are discussed, while, in **Section 5**, future research directions are reported.

Notations

In this paper, matrices are denoted in bold, capital letters, while vectors in bold, lower case letters. The base-10 logarithm of x is given by $\log(x)$. Additionally, $\frac{\partial f(x_1,...,x_N)}{\partial x_i}$ stands for the partial derivative of $\partial f(x_1, \ldots, x_N)$ with respect to x_i, with $i \in [1, N]$. The operator $\max(x_1, x_2, \ldots, x_N)$ yields the numerically largest of the x_i, while $\min(x_1, x_2, \ldots, x_N)$ returns the numerically smallest of

TABLE 1 | Nomenclature.

3D	Three dimensional
6G	Sixth-generation
A3C	Asynchronous actor critic algorithm
ADC	Analog-to-digital converter
AI	Artificial intelligence
AMR	Automatic modulation recognition
AoA	Angle-of-arrival
AP	Access point
B5G	Beyond fifth generation
BER	Bit error rate
BS	Basestation
CNN	Convolutional neural network
DCA	Direct conversion architectures
DDRL	Distributed deep reinforcement learning
DLCS	Deep learning compressed sensing
DoF	Degrees-of-freedom
DNN	Deep neural network
DSP	Digital signal processing
EM	Expectation maximization
ETF	Extract, Transform, Load
FG-AN	Focus Group on Autonomous Networks
GAN	Generative adversarial network
GPML	Gaussian process based machine learning
IEEE	Institute of electrical and electronic engineering
IoT	Internet-of-Things
IoV	Internet-of-vehicles
kNN	k-nearest neighbor
KPI	Key performance indicator
LOS	Line-of-sight
LSTM	Long short term memory
MAC	Medium access control
MIMO	Multiple-input multiple-output
MISO	Multiple-input single-output
ML	Machine learning
mmW	Millimeter wave
MU	Multi-user
NN	Neural network
NOMA	Non-orthogonal multiple access
OSI	Open system interconnection
PHY	Physical layer
PSO	Particle swarm optimization
RIS	Reconfigurable intelligent surface
RNN	Recurrent neural networks
RRB	Radio resource block
RRM	Radio resource management
RX	Receiver
SGD	Stochastic gradient descent
SOMP	Simultaneous orthogonal match pursuit
SVM	Support vector machine
THz	Terahertz
TX	Transmitter
UAV	Unmanned areal vehicle
UE	User equipment
Q-CNN	Convolutional neural network with quantized weights
QoE	Quality-of-experience
QoS	Quality-of-service
V2I	Vehicle-to-infrastructure

the x_i. Moreover, $\sqrt{x}$ represents the square root of x. The index of the value of $\mathbf{x}$ that maximizes and minimizes $f(\mathbf{x})$ are respectively given by $\arg\max f(\mathbf{x})$ and $\arg\min f(\mathbf{x})$. The expected value of $f(x)$ is represented as $\mathbb{E}[\cdot]$. Finally, for the sake of convenience and brevity, **Table 1** summarizes the abbreviations that are used in this paper.

2 THE ROLE OF ML IN THZ WIRELESS SYSTEMS AND NETWORKS

Together with the promise of supporting high data-rate massive connectivity, THz wireless systems and networks come with several challenges. In particular, these challenges can be summarized as:

- Due to the high transmission frequency, i.e. the small wavelength, in the THz band, we can design high-directional antennas (with gains that may surpass 30 dBi) with unprecedented low beamwidths, which may be less than 4°[1]. These antennas are used to counterbalance the high channel attenuation by establishing high-directional links. On the one hand, high directionality creates additional DoFs, which, if they are appropriately exploited, they can enhance both dynamic spectrum access and network densification; thus, boost its connectivity capabilities. In this direction, new approaches to support intelligent interference monitoring and cognitive access are required. On the other hand, high directionality comes with the requirement of extremely accurate beam alignment between the fixed and moving communication nodes. To address this beam tracking and channel estimation approaches of high latency needs to be developed, which account for the latency requirement.
- Molecular absorption causes frequency- and distance-dependent path loss, which creates frequency windows that are unsuitable for establishing communication links (Boulogeorgos et al., 2018d). As a consequence, despite the high bandwidth availability in the THz band, windowed transmission with time varying loss and per-window adaptive bandwidth as well as power usage is expected to be employed. This characteristic is expected to influence both beamforming design as well as resource allocation and user association.
- The large penetration loss in the THz band, which may surpass 40 or even 50 dB (Kokkoniemi et al., 2016; Stratidakis et al., 2019; Petrov et al., 2020; Stratidakis et al., 2020a; Stratidakis et al., 2020b), renders questionable the establishment of the non-LoS links. As a result, blockage avoidance schemes are needed. Note that in lower frequency bands, such as mmW, the penetration loss is in the range of 20–30 dB (Mahapatra et al., 2015; Zhu et al., 2018; Zhang et al., 2020a).
- To exploit the spatial dimension of THz radio resources, support MU-connectivity as well as increase the link capacity in heterogeneous environments of moving nodes, suitable beamforming and mobility management designs that predict the number and motion of UEs need to be designed. Moreover, in mobile scenarios, accurate channel state information (CSI) is needed. In lower-frequency systems, this is achieved by performing

FIGURE 1 | ML-based applications to different layers of THz wireless systems and networks.
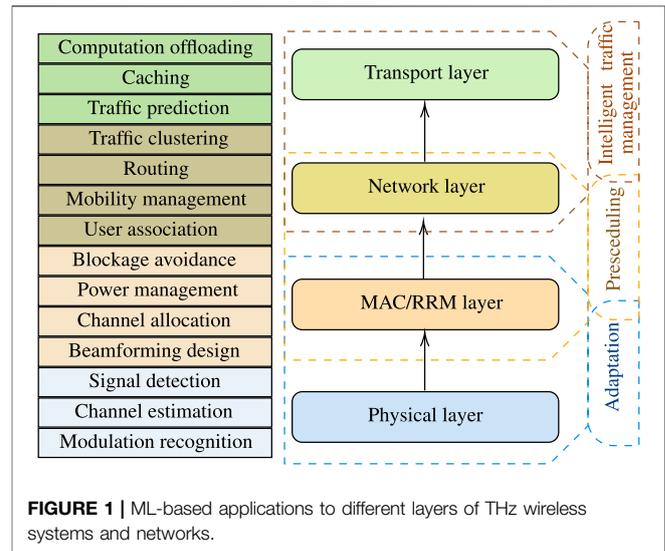
frequent channel estimation. However, in THz wireless systems, due to the small transmission wavelength, the channel can be affected by slight variations in the micrometer scale (Sarieddeen et al., 2021). As a result, the frequency of channel estimation is expected to be extremely high and the corresponding overhead unaffortable. This motivates the design of prediction-based channel estimation and beam tracking approaches.

- From the hardware point of view, high-frequency transceivers suffer from a number of hardware imperfections. In particular, the EVM of THz transceiver may even reach 0.4 (Koenig et al., 2013). It is questionable that conventional mitigation approaches would be able to limit the impact of hardware imperfections. As a result, smarter signal detection approaches need to be developed. Of note, as stated in (Schenk, 2008; Boulogeorgos, 2016; Boulogeorgos and Alexiou, 2020b), in lower-frequency communication systems including mmW, the EVM does not exceed 0.17.
- THz wireless systems are expected to support a large variety of applications with diverse set of requirements as well as UEs equipped with antennas of different gains. To guarantee high availability and association rate that tends to 100%, association schemes that take into account both the nature of THz resource block, the UEs' transceivers capabilities, as well as the applications requirements, need to be presented. Moreover, to ensure uninterrupted connectivity, these approaches should be highly adaptable to network topology changes. In other words, they should be able to predict network topologies changes and pre-actively perform hand-overs.
- Conventional routing strategies account neither the communication nodes distance nor their memory limitations. However, in THz mobile scenarios, where both the transmission range is limited to some decades of meter and the device memory is comparable to the packet length, routing may become a complex optimization problem.

---

[1]It is worth noting that in mmW wireless systems, antennas with beamwidths that are in the range of 10° are employed.

- Finally, for several realistic scenarios, the aggregated data-rate of the fronthaul is expected to reach 1 Tb/s, which is comparable with the backhaul's achievable data-rate. This may cause service latency due to data congestion in network nodes. To avoid this intelligent traffic prediction and caching strategies need to created to pre-actively bring the future-requested content near the end-user.

The rest of this section is focused on explaining the role of ML in THz wireless systems and networks by presenting the current state-of-the-art. As illustrated in **Figure 1** and in order to provide a comprehensive understanding of the need to formulate ML problems and devise such solutions, we classify the ML problems into four categories, namely 1) PHY, 2) MAC and RRM, 3) Network, and 4) transport. Of note, this classification is in line with the open system interconnection (OSI) model. For each category, we identify the communications and networks problems for which ML solutions have been proposed as well as the needs of utilizing them. Finally, this section provides a review on the ML problems and solutions that have been employed. Apparently, some of the aforementioned challenges have been also discussed in lower-frequency systems and networks. For the sake of completeness, in our literature review, we have also included contributions that although refer to lower frequency, can find application to THz wireless systems and networks.

2.1 PHY Layer

The additional DoFs, which have been brought by the ultra-wide band THz channels as well as their spatial nature, allow us to establish high data-rate links with limited transmission power. Moreover, advances in the fields of communication-components designs, fueled by new artificial materials, such as reconfigurable intelligent surfaces (RISs), created a controllable wireless propagation environment; thus, offered new opportunities in simplifying the PHY layer processes and further increased the systems DoFs (Boulogeorgos et al., 2021b). Finally, the use of direct conversion architectures (DCAs) in both transceivers created the need to utilize digital signal processing (DSP) algorithms in order to "decouple" the system's reliability from the hardware imperfection related degradation. The computational complexity of these algorithms increases as the modulation order of the transmission signal increases. The aforementioned factors create a rather dynamic high-dimensional complex environment with processes that are hard or even impossible to be analytically expressed. Motivated by this, the objectives of employing ML in the PHY layer is to provide adaptability to an ever changing wireless environment consisted of heterogeneous components (transceivers, RIS, active and passive relays, etc.) and to countermeasure the impact of transceivers hardware imperfections with increasing neither the communication and computation overheads.

In this direction, several researcher have focused on presenting ML-related solutions for automatic modulation recognition (AMR) (Khan et al., 2016; Li et al., 2018; Wu et al., 2018; Iqbal et al., 2019; Shah et al., 2019; Yang et al., 2019; Bu et al.,

2020), *channel estimation* (Satyanarayana et al., 2019; Zhu et al., 2019; Liu S. et al., 2020; Ma et al., 2020a; Ma et al., 2020b; Moon et al., 2020; Mai et al., 2021; Wang et al., 2021), and *signal detection* (Jeon et al., 2018; Aoudia and Hoydis, 2019; Samuel et al., 2019; Katla et al., 2020; Satyanarayana et al., 2020). In more detail, AMR has been identified as an important task for several wireless systems, since it enables dynamic spectrum access, interference monitoring, radio fault self-detection as well as other civil, government, and military applications. Moreover, it is considered as a key enabler of intelligent/cognitive nano- and macro-scale receivers (RXs). Fast AMR can significantly improve the spectrum utilization efficiency (Mao et al., 2018). However, it is a very challenges tasks, since it depends to the variation of the wireless channel. This aspires the introduction of intelligence in this task. In this sense, in (Yang et al., 2019), the authors employed convolutional NNs (CNNs) and recurrent NNs (RNNs) in order to perform AMR. As inputs, the algorithms used the in-phase and quadrature (I/Q) components of the unknown received signal and classified it to one of the following modulation schemes: binary frequency shift keying (BFSK), differential quadrature phase shift keying (DQPSK), 16 quadrature amplitude modulation (16-QAM), quaternary pulse amplitude modulation (4PAM), minimum shift keying (MSK), Gaussian minimum shift keying (GMSK). Moreover, in (Khan et al., 2016), a deep NN (DNN) approach was discussed for identifying the received signal modulation in coherent RXs. The DNN is comprised of two auto-encoders and an output perceptron layer. To train and verify the ML network, two datasets of numerous amplitude histograms are used. After training, the network is capable of accurately extracting the modulation format of the signal after receiving a number of symbols.

Similarly, in (Wu et al., 2018), the authors presented a DNN approach that is based on RNNs with short memory and is capable of to exploit the temporal and spatial correlation of the received samples in order to accurately extract the their modulation type. In this contribution, as an input, the ML algorithm requires a predetermined number of samples. Meanwhile, in (Shah et al., 2019), the authors reported a extreme supervised-learning ML algorithm capable of accurately and time-efficiently estimating the modulation type of the received samples. The disadvantage of the aforementioned approaches is that the training period is large and if the characteristics of channel changes, for example due to a RIS reconfiguration, a partial blockage phenomenon or the user equipment movement, the algorithm need to be re-trained.

Additionally, in (Li et al., 2018), a semi-supervised deep convolutional generative adversarial network (GAN) was presented that consists of a pair of GANs that collaboratively create a powerful modulator discriminator. The ML network receives as inputs the I/Q components of a number of received signal samples and matches them to a set of modulation formats. The main disadvantage of this approach is its high-computational resource demands as well as its sensitivity to received signal distribution variations. To deal with the data distribution variation, Bu et al. (Bu et al., 2020) introduced an adversarial transfer learning architecture that exploits transfer learning and is

capable of achieving accuracy comparable to the ones of supervised learning approaches. However, this approach demands careful handling of former knowledge since there may exist differences between wireless environments. In (Bu et al., 2020), the ML algorithm uses as an input the (I/Q) components of the received signal. Finally, in (Iqbal et al., 2019), an expectation maximization (EM) algorithm was employed in order to perform modulation mode detection and systematically differentiate between pulse- and carrier-based modulations. The presented results revealed the existence of a unique Pareto-optimal point for both the SNR and the classification threshold, where the error probability is minimized.

Another important task in PHY layer is *channel estimation and tracking*. In particular, in order to employ high directional beaforming in THz wireless systems, it is necessary to acquire channel information for all the transmitter (TX) and RX antenna pairs. The conventional approach that is supported by several standards, including IEEE 802.11ad and IEEE 802.11ay (Ghasempour et al., 2017; Silva et al., 2018; Boulogeorgos and Alexiou, 2019), depends on creating a set of transmission and reception beamforming vector pairs and scanning between them in order to identify the optimal one. During this process, the access point (AP) sends synchronization signals using all its possible beamforming vectors, while the user equipment (UE) performs energy detection in all possible reception directions. In the end of this process, the UE determines the optimal AP beamforming vector and feed it back to the AP. Next, the roles of the AP and UE interchange in order to allow the AP to identity the optimal UE beamforming vector and feed it back to it. Notice that in the second phase, the AP locks its RX to its optimal beamforming vector. Then, channel estimation is performed using the optimal beamforming pair and a classical DSP technique (e.g., minimum least square error, minimum mean square error, etc.). Let us assume that the AP and the UE respectively have L_A and L_U available beamforming vectors and that N_a and N_e received signal samples are needed for energy detection and channel estimation. As a consequence, the latency and power consumption due to channel estimation is proportionally to $L_A L_U N_a + L_U N_a + N_e + 2$. This indicates that as the number of the antennas at TX and RX increases, the number of beamforming pairs also increases; thus, the training overhead significantly increase and the conventional channel estimation approach becomes more complex as well as time and power inefficient.

To better understand the importance of this challenge in THz wireless systems, let us refer to an indicative example. Assume that we would like to support a virtual reality (VR) application for which the transmission distance between the AP and the UE is 20 m, a data-rate of 20 Gb/s is required with an uncoded bit error rate (BER) in the order of 10^{-6}. Furthermore, a false-alarm probability that is lower than 1% is required. This indicates that $N_a \geq 100$. Let us assume that the transmission bandwidth is 10 GHz, the transmission power is set to 10 dBm, while the receives mixer convention and miscellaneous losses are respectively 8 and 5 dB. Additionally, the RX's low noise amplifier (LNA) gain is 25 dB, whereas the RX's mixer and LNA noise figures are respectively 6 and 1 dB. If the

transmission frequency is 287.28 GHz, both the TX and RX need to be equipped with antennas of 35 dBi gain. Such antennas have a beamwidth of 3.6°; hence, $L_A = L_U = 100$. Also, by assuming that $N_e = 100$, the channel estimation latency becomes equal to approximately 1 ms. Of note, VR requires a latency that is lower than 1 μs

To address the aforementioned problem, researchers turned their attention to regression and clustering ML-based methods. Specifically, in (Moon et al., 2020), the authors employed a DNN-based algorithm to predict the user's channel in a sub-THz multiple-input multiple-output (MIMO) vehicular communications system. he ML algorithm of (Moon et al., 2020) takes as input the signals received by a predetermined number of APs and outputs a vector containing the estimated channel coefficients. Similarly, in (Ma et al., 2020b), the authors reported a deep learning compressed sensing (DLCS) algorithm for channel estimation scheme in multi-user (MU) massive MIMO sub-THz systems. The DLCS is a supervised learning algorithm that takes as input a simulation-based generated received signal vector as well as real measurements and performs two functionalities, i.e. beamspace channel amplitude estimation; and 2) channel reconstruction. The results indicate that this approach can achieve a minimum mean square error that is comparable with the one of the orthogonal matching pursuit scheme. Likewise, in (Mai et al., 2021), a manifold learning-extreme learning machine was presented for estimating high-directional channels. In more detail, manifold learning was employed for dimentionality reduction, whereas the extreme learning algorithm with one-shot training was employed for channel state information estimation. Moreover, in (Wang et al., 2021), the authors presented a Gaussian process based ML (GPML) algorithm to predict the channel in an unmanned aerial vehicle (UAV) aided coordinated multiple point (CoMP) communication system. The main idea was to provide real-time predictions of the location of the UAV and reconstruct the line-of-sight (LOS) channel between the AP and the UAV. Similarly, in (Zhu et al., 2019), a sparse Bayesian learning algorithm was introduced to estimate the propagation parameters of the wireless system. Meanwhile, in (Satyanarayana et al., 2019), the authors presented a neural network (NN)-based algorithm for channel prediction and showed that, after sufficient training, it can faithfully reproduce the channel state.

Likewise, in (Ma et al., 2020a), a NN-based algorithm, which consists of three hidden layers and one fully-connected layer, was reported to obtain the beam distribution vector and reproduce the channel state. The algorithm uses as inputs the (I/Q) components of the received signal. It is trained offline using simulations and is able to achieve similar performance in terms of total training time slots and the spectral efficiency with previously proposed approaches, like adaptive compressed sensing, hierarchical search, and multi-path decomposition. In order for this approach to work properly, the simulation and real world data should follow the same distribution. Also, the propagation parameters of both types of data should coincide. Furthermore, in (Liu S. et al., 2020), the authors presented a deep denoising NN assisted compressive sensing broadband

TABLE 2 | ML algorithm types applied in PHY.

	NN	CNN	RNN	DNN	GAN	DLCS	GPML	Bayesian learning	SOMP	kNN	EM
AMR											
Yang et al. (2019)	–	✓	✓	–	–	–	–	–	–	–	–
Khan et al. (2016)	–	–	–	✓	–	–	–	–	–	–	–
Wu et al. (2018)	–	–	–	✓	–	–	–	–	–	–	–
Shah et al. (2019)	✓	–	–	–	–	–	–	–	–	–	–
Li et al. (2018)	–	–	–	–	✓	–	–	–	–	–	–
Bu et al. (2020)	–	–	–	–	✓	–	–	–	–	–	–
Iqbal et al. (2019)	–	–	–	–	–	–	–	–	–	–	✓
Channel estimation and beam tracking											
Moon et al. (2020)	–	–	–	✓	–	–	–	–	–	–	–
Ma et al. (2020b)	–	–	–	–	–	✓	–	–	–	–	–
Mai et al. (2021)	✓	–	–	–	–	–	–	–	–	–	–
Wang et al. (2021)	–	–	–	–	–	–	✓	–	–	–	–
Zhu et al. (2019)	–	–	–	–	–	–	–	✓	–	–	–
Satyanarayana et al. (2019)	✓	–	–	–	–	–	–	–	–	–	–
Ma et al. (2020a)	✓	–	–	–	–	–	–	–	–	–	–
Liu et al. (2020b)	–	–	–	–	–	–	–	–	✓	–	–
Li et al. (2020d)	–	–	–	✓	–	–	–	–	–	–	–
Anton-Haro and Mestre (2019)	–	–	–	✓	–	–	–	–	–	✓	–
Signal detection											
Samuel et al. (2019)	–	–	–	✓	–	–	–	–	–	–	–
Aoudia and Hoydis (2019)	–	–	–	✓	–	–	–	–	–	–	–
Katla et al. (2020)	–	–	–	✓	–	–	–	–	–	–	–
Satyanarayana et al. (2020)	–	–	–	✓	–	–	–	–	–	–	–
Jeon et al. (2018)	–	–	–	✓	–	–	–	–	–	–	–

channel estimation algorithm that exploits the relation of angular-delay domain MIMO channels in sub-THz RIS-assisted wireless systems. In particular, the algorithm takes as inputs the received signals at a number of active elements of the RIS and forward them to a compressive sensing block, which feeds the NN. The proposed approach outperformed the well-known simultaneous orthogonal match pursuit (SOMP) algorithm in terms of normalized mean square error (NMSE). The main disadvantage of deep denoising NN approach is that it is not adaptable to changes in the propagation environment characteristics. Furthermore, in (Li et al., 2020d), Li et al. reported a deep leaning architecture for channel estimation in RIS-assisted THz MIMO systems. The idea behind this approach was to convert the channel estimation problem into the sparse recovery problem by exploiting the space nature of THz wireless channels. In this direction, the algorithm uses for training the (I/Q) components of received signals that carry pilot symbols. In the operation phase, the (I/Q) components of the received signal are used as inputs. Finally, in (Anton-Haro and Mestre, 2019), the authors employed k-nearest neighbors (kNN) and support vector classifiers (SVC) to estimate the angle-of-arrival (AoA) in hybrid beamforming wireless systems.

Signal detection is another PHY layer task. Conventional approaches require accurate estimation of both the channel model and the impact of hardware imperfections in order to design suitable equalizers and detectors. However, as the wireless environment becomes more complex and the influence of hardware imperfections become more severe, due to the high

number of transmission and reception antennas, data detection becomes more challenging. This fact motivates the study of ML-based solution for end-to-end signal detection, in which no channel and hardware imperfections equalization is required. In this sense, in (Samuel et al., 2019), the authors employed a DNN architecture for data detection, whereas, in (Aoudia and Hoydis, 2019), the effectiveness of deep learning for end-to-end signal detection was reported. Similarly, in (Katla et al., 2020), the authors presented a deep learning assisted approach for beam index modulation detection in high-frequency massive MIMO systems. Additionally, in (Satyanarayana et al., 2020), the authors demonstrated the use of deep learning assisted soft-demodulator in multi-set space-time shift keying millimeter wave (mmW) wireless systems. To cancel the impact of hardware imperfections without employing equalization units, a supervised ML signal detection approach was presented in (Jeon et al., 2018).

To sum up, PHY-specific ML algorithms usually employ as inputs I/Q components of the received signal samples. In case of modulation recognition, due to the lack of unlabeled data for training, unsupervised learning approaches are adopted. On the other hand, in channel estimation and signal detection, pilot signals can be exploited to train the ML algorithm. As a consequence, for channel estimation, supervised learning approaches are the usual choice. In particular, as observed in **Table 2**, both supervised and unsupervised learning approaches were applied that return classification, regression, dimensionality reduction and density estimation rules. Interestingly, it is observed that for signal detection only DNN-based algorithms

have been reported. Finally, it is worth noting that the main requirement for the algorithm selection in most cases was to provide high adaptation to the THz wireless system. It is worth mentioning that, for the sake of competence, **Table 2** includes some contributions that although do not refer to THz wireless systems, they can be straightforwardly applied to them. For example, (Shah et al., 2019; Yang et al., 2019) and (Li et al., 2018) can be applied to any system that employ I/Q modulation and demodulations approaches, whereas (Khan et al., 2016) is suitable for the ones that use coherent RXs. Of note, according to (Boulogeorgos et al., 2018b), coherence RXs are a usual choice in THz wireless fiber extenders. Similarly, since an inherent characteristic of THz channels is the high temporal correlation, the ML methodology presented in (Wu et al., 2018) is expected to find application in THz wireless systems. Additionally, despite the fact that the ML algorithms in (Zhu et al., 2019; Ma et al., 2020a; Liu S. et al., 2020; Ma et al., 2020b; Katla et al., 2020; Moon et al., 2020; Satyanarayana et al., 2020; Mai et al., 2021; Wang et al., 2021) were applied to mmW systems and exploit the spatio-temporal and directional characteristics of the channels in this band, the same approach can be used in THz systems that have the similar particularities. Finally, the ML approach presented in Jeon et al. (2018) can be applied in any MIMO regardless the operating frequency.

2.2 MAC and RRM Layer

MAC and RRM layers are responsible of providing uninterrupted high quality-of-experience (QoE) to mutliple end users. In contrast to lower-frequency communications, where omni-directional or quasi-omni-directional links are established, in THz wireless systems both the AP/BS and the UE employ beamforming. As a result, an additional to time and frequency resource, i.e., space, is created. The optimal exploitation of the tertiary nature of the channel require the design of beamforming approaches[2] and channel allocation strategies. Moreover, to satisfy the end user's data rate demands and to support non-orthogonal multiple access (NOMA), power management policies are required. Finally, to address the impact of blockage, environmental awareness need to be obtained by the wireless THz network and proactive blockage avoidance mechanisms are a necessity. Motivated by the above, the rest of this section discuss ML-based beamforming designs, channel allocation strategies, power management schemes and blockage avoidance mechanisms.

Beamforming design is a crucial task for MIMO and MU-MIMO THz wireless systems. However, conventional beamforming approaches strongly rely on accurate channel estimation; as a result, their complexity is relatively high. To simplify the beamfoming design process, a great amount of research effort was put on investigating ML-based approaches.

In more detail, in (Kwon et al., 2019), the authors presented a DNN for determining the optimal beamforming vectors that maximizes the sum rate in a two-user multiple-input single-output (MISO) wireless system. The ML algorithm uses as inputs the real and imaginary part of the complex MISO channel coefficients as well as the transmitted power. For training the ML model, the cross-entropy is used as a cost function, which evaluates the errors by calculating the difference between probability distributions labeled and model outputted data. Three DNN architectures to approximate the hybrid beamformer's singular value decomposition, with varying levels of complexity were discussed in (Peken et al., 2020a). The architectures take as inputs the real and imaginary components of the MIMO channel coefficients. The first architecture predicts a predetermined number of the most important singular values and vectors of a given channel matrix by employing a single DNN. The second architecture employs k DNNs. Each one of them returns the largest singular value and corresponding right and left singular vectors of the MIMO channel matrix. The third architecture is suitable for channel matrices of rank-1 and outputs a predetermined number of singular values and vectors by recursively using the same DNN. The architectures are trained through comparison of the extracted channel matrices to the channel matrices extracted by the ML-models. The proposed ML-based architectures were shown, by means of Monte Carlo simulations, to improve the system's date rate by up to 50–70% compared to conventional approaches.

The main disadvantage of the ML-architectures presented in (Peken et al., 2020a) is that they require a large number of estimations of channel matrices for training, which may generate an unaffortable latency in a fast changing environment. To counterbalance this, in (Mo et al., 2019), an unsupervised K-means algorithm is employed, which exploits the electric-field response of each antenna element in order to design beam codebooks that optimize the average received power gain of UEs that are located within a cluster. Although this approach does not require large training periods, it is sensitive to drastical changes of the wireless environment, which may arise due to users or scatterers/blockers movement. Meanwhile, in (Sun et al., 2020), the authors formulated an interference managing problem by means of coordinated beamforming in ultra-dense networks that aims at "almost real-time" sum rate maximization. To solve the aforementioned problem, a Q-learning based ML algorithm was introduced, which only require large scale channel fading parameters and achieves similar results to the ones of the corresponding analytical approach that demands full channel state information. Note that the Q-learning algorithm in (Sun et al., 2020) takes as input the state of the system, i.e. a logarithmic transformation of the second-norm of the MIMO channel matrix. Likewise, in (Alkhateeb et al., 2018a), the authors reported a centralized deep learning based algorithm for coordinated beamforming vector design in high-mobility and high-directional wireless systems. This algorithm uses as inputs the I/Q components of a virtual omni-directional received signal and extracts a prediction for the optimal beamforming vectors. To train the deep-learning algorithm pilot symbols are employed,

---

[2]It is worth noting that in lower-frequency systems, beamforming design has been usually considered to be part of the PHY layer. However, according to IEEE 802.11.3ay (Boulogeorgos et al., 2017), since beamforming manages the spatial resources of THz systems, although it is conducted at the baseband, part of it belongs to the PHY, while the other to the MAC layer.

which are exchanged between the UE and basestations (BSs) within the coherence time. This approach provided high-adaptation with the cost of creating important overhead due to the message exchange need from/to BSs to/from a central/cloud processing unit. In (Aljumaily and Li, 2019), a supervised NN was employed, which is based on singular value decomposition, to design hybrid beamformers in massive MIMO systems that are capable of mitigating the impact of limited resolution phase shifters. The proposed approach is executed in a single BS and achieve a higher spectral efficiency compared with unsupervised learning ones. However, the performance enhancement demands a relatively high training period.

To address this inconvenience, in (Lizarraga et al., 2019), the authors presented a reinforcement learning algorithm to jointly-design the analog and digital layer vectors of a hybrid beamformer in large-antenna wireless systems. The algorithms take as input the achievable data rate and returns the phase shifts of each antenna element. The disadvantage of this algorithm is that it is unable to achieve the same performance as the corresponding supervised ML one. Moreover, in (Huang S. et al., 2020), the authors studied the use of extreme learning machine for jointly optimizing transmit and receive hybrid beamforming in MU-MIMO wireless systems. The algorithm requires as inputs the real and imaginary parts of the MIMO channel coefficients and returns the an optimal beamforming vectors estimation. As a training cost function the difference between the targeted and achievable SNR is used. In (Huang et al., 2020c), the extreme learning and NNs were employed in order to extract the transmit and receive beamforming vectors in full-duplex massive MIMO systems.

In (Elbir and Mishra, 2019), a CNN with quantized weights (Q-CNN) algorithm was utilized as a solution to the problem of jointly designing transmit and receive hybrid beamforming vectors. As input, the real and imaginary components of the MIMO channel matrix is used. Q-CNN has limited memory and low-overhead demands; hence, it is suitable for deployment in mobile devices. Furthermore, in (Chen J. et al., 2020), three NN-based approaches for designing hybrid beamforming schemes were reported. The first one is based on mapping various hybrid beamformers to NNs and thus transforming the beamforming codeword design non-convex optimization problem into a NN training one. The second approach is an extension of the first one that aims at optimizing the beam vectors for the case of MU access. In comparison to the aforementioned approach, the third one takes into account the hardware limitations, namely low-resolution phase shifters and analog-to-digital converters (ADCs). Simulation results revealed that the proposed approaches outperform analytical ones in terms of BER. All the aforementioned approaches in (Chen J. et al., 2020) require as input the MIMO channel matrix. In (Long et al., 2018), the authors presented a support vector machine (SVM) algorithm for analog beam selection in hybrid beamforming MIMO systems that uses as inputs the complex coefficients of the channel samples. This approach provides near-optimal uplink sum rates with reduced complexity compared to conventional strategies. On the other, it requires sufficient training data that

leads to high training periods, when the characteristics of the wireless channel changes.

To countermeasure the aforementioned problem, unsupervised learning approaches were employed in several works (Kao et al., 2018; Peken et al., 2020b; Lin et al., 2020). In more detail, in (Peken et al., 2020b), an autoencoding-based SVD methodology was used in order to estimating the optimal beamforming codes at the TX and RX, while, in (Lin et al., 2020), Lin et al. introduced a deep NN architecture for beamforming design that outperforms several previously presented deep learning approaches. Additionally, in (Kao et al., 2018), a ML-based clustering strategy with feature selection was employed to design three dimensional (3D) beamforming. In particular, the algorithm has three steps. In the first step, it uses pre-collected data in order to obtain a set of eigenbeams, while, in the second one, the aforementioned sets are used to estimate the channel state information, which in the third step is feed to a Rosenbrock search engine. The two first steps are executed offline, whereas, the third one is an online process. As the number of antenna elements increases, the size of the eigenbeam vector set also increases; thus, the practicality of this approach may be questionable in massive MIMO systems. All the aforementioned ML algorithms employ as input the MIMO channel matrix.

ML was also employed to optimize the beamforming vectors of relaying and reflected assisted high-directional THz links. For example, in (Li L. et al., 2020), Li et al. presented a cross-entropy hybrid beamforming vector estimation deep reinforcement learning-based scheme for unmanned aerial vehicle (UAV)-assisted massive MIMO network. The deep reinforcement learning method was employed in order to minimize the AP transmission power by jointly optimizing the AP and RIS active and passive beamforming vectors. Finally, in (Liu et al., 2020c), Liu et al. used Q-learning in order to jointly designing the movement of the UAV, phase shifts of the RIS, power allocation policy from the UAV to mobile UEs, as well as determining the dynamic decoding order of a NOMA scheme, in a RIS-UAV assisted THz wireless system. Both the ML algorithms utilized in (Li L. et al., 2020) and (Liu et al., 2020c) use as input the estimated channel coefficient matrix.

Another challenging and important task in THz wireless networks is *channel allocation*. Of note, in this band, the radio resource block (RRB) has three dimensions, i.e., time, frequency, and space. As a result, the additional DoF, namely space, creates a more complex resource allocation problem. Aspired by this fact, several contributions studied the use ML approaches in order to design suitable resource allocation policies. Indicative examples are (Ahmed and Khammari, 2018; Peng et al., 2019; Tauqir and Habib, 2019; Huang H. et al., 2020; Cao et al., 2020; Jang and Yang, 2020). In particular, in (Cao et al., 2020), a centralized NN was employed to return the channel allocation strategy that minimizes the co-channel interference in an ultra-dense wireless network. The NN takes as input a binary matrix that contains the user-channel association and estimates the up-link SINR. In (Peng et al., 2019), a centralized supervised cluster-based ML interference management channel allocation that takes into account the time-varying network load was introduced. As inputs

to the algorithm, the RRB allocation data, the acknowledgement (ACK) and the negative acknowledgement (NACK) data collected from the network are used. The algorithm outputs an estimation of the interference intensity.

To deal with the ever changing topology and time-varying channel conditions of ultra-dense mobile wireless networks, in (Jang and Yang, 2020), a deep Q-learning model was presented that uses quantized local AP and UE channel state information to cooperatively allocate the channels in a downlink scenario. The algorithm is self-adaptive and does not require any training. Additionally, in (Tauqir and Habib, 2019), a DNN was used that takes as inputs the channel state information and returns the spatial resource block (i.e. beam) allocation in massive MIMO wireless systems. Moreover, in (Huang H. et al., 2020), a deep learning approach was proposed for channel and power allocation in MIMO-NOMA wireless systems that aims at maximizing the sum data rate and energy efficiency of the overall network. The approach uses as inputs the channel vectors, precoding matrix and the power allocation factors. Finally, in (Ahmed and Khammari, 2018), a feedforward NN that takes as inputs the uplink channel state information and returns a channel allocation strategy in a rank and power constrained massive MIMO wireless system, was employed.

For multi-antenna THz transceivers, the hardware complexity and *power management* become a burden toward practical implementation (Han et al., 2015). To lighten the power management process, several researchers turned their eyes on ML. For instance, in (Zhang et al., 2020a), the K-means algorithm was employed to cluster the users of a NOMA-THz wireless network and to maximize the energy efficiency by optimizing the power allocation. The K-means algorithm in (Zhang et al., 2020a) requires as inputs the number of clusters and set of users as well as the channel vectors and outputs the UE-AP association matrix. Moreover, in (Kwon et al., 2020), Kwon et al. reported a self-adaptive DRL deterministic policy gradient-based power control of BS and proactive cache allocation toward BSs in distributed Internet-of-vehicle (iov) networks. In (Kwon et al., 2020), the system's state that is the input of the DRL, takes into account the available and total buffer capacity of each BS, the average e quality state of the provisioned video at each UE. Meanwhile, in (Meng et al., 2019), a transfer learning approach that is based to the Q-learning algorithm, was used to allocate power in MU cellular networks, in which each cell has different user densities. Similarly, Q-learning was employed in (Amiri and Mehrpouyan, 2018) to develop a self-organized power allocation strategy in mmW networks. Finally, in (Zhang et al., 2020b), Zhang et al. presented a semi-supervised learning and DNN for sub-channel and power allocation in directional NOMA wireless THz networks. The algorithm requires as inputs the set of users and their channel vectors as well as a predetermined number of clusters.

Another burden that THz wireless system face is *blockage*. Sudden blockage of the THz LOS path cause communication interruptions; thus, creates a detrimental impact on the system's reliability. Further, re-connections to the same or other BS/AP demands high beam training overhead, which in turn result to high latency. To avoid this, some contributions discuss the use of ML in order to predict dynamic (moving) obstacles position and their probability to block the LOS between the AP/BS and UE in order to proactively hand-over users to other AP/BS. Towards this direction, in (Alkhateeb et al., 2018b), a reinforcement learning algorithm was used to create a proactive hand-off blockage avoidance strategy. The state of the system that is defined by the beam index of each AP/BS at every time step is used as an input to the reinforcement learning agent. Moreover, in (Khan and Jacob, 2019), NN and CoMP clustering was employed to predict the channel state and avoid blockage. In the scenario under investigation, the authors consider dynamic blockers (i.e., cars) that perform deterministic motion. The presented algorithm use as inputs the UE location as well as the system's clock time. Similarly, in (Iimori et al., 2020), stochastic gradient descent (SGD) was employed to design outage-minimization robust directional CoMP systems by selecting communication paths that minimize the blockage probability. In this direction, the algorithm uses as inputs an initial estimation of the beamforming vector as well as the channel state information and returns the optimized beamforming vector. Finally, in (Jia et al., 2020), a DNN was employed to provide environmental awareness to an RIS-assisted wireless sub-THz network that performs beam switching between direct and RIS-assisted connectivity in order to avoid blockage. The inputs of the algorithm in (Jia et al., 2020) are the network topology and the links line-of-sight conditions.

To conclude and according to **Table 3**, supervised, unsupervised, and reinforcement learning were employed to solve different problems in MAC layer. In particular, for beamforming design, where the main requirement is adaptation to the ever changing propagation environment, unsupervised learning was used to cluster UEs, supervised learning was applied to design appropriate codebooks, and reinforcement learning was employed for beam refinement and fast adaptation. As a consequence, reinforcement learning approaches are attractive for mobile and non-deterministic varying wireless environments. On the other hand, supervised learning approaches are more suitable for static environments or environments that change in deterministic way. A key requirement that supervised learning approaches have is the need to be training through a set of channel or received signal vectors that are accompanied by an achievable performance indicator. The indicator can be the data-rate, outage probability or any other KPI of interest. Likewise, unsupervised clustering and supervised learning were used for channel allocation, which was performed based on the UE communication demands. Due to lack of extensive training data sets and need of adaptation, unsupervised and reinforcement learning approaches were employed for power management. Finally, supervised and reinforcement learning were used to provide proactive policies for blockage avoidance based on statistical or instantaneous information, respectively. Of note, some of the contributions in **Table 3** refer to wireless systems that employ the same technological enablers as THz communications without specifying the operating frequency band. Indicative examples are (Aljumaily and Li, 2019; Elbir and Mishra, 2019; Kwon et al., 2019; Peken et al., 2020a; Liu et al.,

TABLE 3 | ML algorithm types applied in MAC.

	NN	DNN	k-Means	Q-learning	DRL	Q-CNN	Autoencoder	SGD
Beamforming design								
Kwon et al. (2019)	–	✓	–	–	–	–	–	–
Peken et al. (2020a)	–	✓	–	–	–	–	–	–
Mo et al. (2019)	–	–	✓	–	–	–	–	–
Sun et al. (2020)	–	–	–	✓	–	–	–	–
Alkhateeb et al. (2018a)	–	✓	–	–	–	–	–	–
Aljumaily and Li (2019)	✓	–	–	–	–	–	–	–
Lizarraga et al. (2019)	–	–	–	–	✓	–	–	–
Huang et al. (2020b)	✓	–	–	–	–	–	–	–
Huang et al. (2020c)	✓	–	–	–	–	–	–	–
Elbir and Mishra (2019)	–	–	–	–	–	✓	–	–
Chen et al. (2020a)	✓	–	–	–	–	–	–	–
Long et al., (2018)	✓	–	–	–	–	–	–	–
Peken et al. (2020b)	–	–	–	–	–	–	✓	–
Lin et al. (2020)	–	✓	–	–	–	–	–	–
Kao et al. (2018)	–	–	✓	–	–	–	–	–
Li et al. (2020a)	–	–	–	–	✓	–	–	–
Liu et al. (2020c)	–	–	–	✓	–	–	–	–
Channel allocation								
Cao et al. (2020)	✓	–	–	–	–	–	–	–
Peng et al. (2019)	–	–	✓	–	–	–	–	–
Jang and Yang (2020)	–	–	–	✓	–	–	–	–
Tauqir and Habib (2019)	–	✓	–	–	–	–	–	–
Huang et al. (2020a)	–	✓	–	–	–	–	–	–
Ahmed and Khammari (2018)	✓	–	–	–	–	–	–	–
Power management								
Zhang et al. (2020a)	–	–	✓	–	–	–	–	–
Kwon et al. (2020)	–	–	–	–	✓	–	–	–
Meng et al. (2019)	–	–	–	✓	–	–	–	–
Amiri and Mehrpouyan (2018)	–	–	–	✓	–	–	–	–
Zhang et al. (2020a)	–	✓	–	–	–	–	–	–
Blockage avoidance								
Zhang et al. (2020a)	–	✓	–	–	–	–	–	–
Alkhateeb et al. (2018b)	–	–	–	✓	–	–	–	–
Khan and Jacob (2019)	✓	–	–	–	–	–	–	–
Iimori et al. (2020)	–	–	–	–	–	–	–	✓
Jia et al. (2020)	–	✓	–	–	–	–	–	–

2020c; Lin et al., 2020; Sun et al., 2020) that discuss ML approaches for beamforming design in analog and hybrid beamforming systems, as well as (Ahmed and Khammari, 2018; Peng et al., 2019; Tauqir and Habib, 2019; Cao et al., 2020; Jang and Yang, 2020), which present ML-based channel allocation approaches for ultra-dense networks in which the communication channels are high directional. Apparently, the aforementioned approaches are suitable for THz wireless deployments.

2.3 Network Layer
The ultra-wideband extremely directional nature of the sub-THz and THz links in combination with the non-uniform UE spatial distribution may lead to inefficient user association, when the classical minimum-distance criterion is employed. Networks operating in such frequencies can be considered noise- and blockage-limited, due to the fact that high path and penetration losses attenuate the interference (Boulogeorgos et al., 2018b; Papasotiriou et al., 2018). Hence, user association metrics designed for interference limited homogenous systems are not well suited to sub-THz and THz networks (Boulogeorgos A.-A. A. A. et al., 2018). As a result, *user association* should be designed to meet the dominant requirements of throughput and guarantee low blockage probability. Another challenge that user association schemes need to face is the user orientation, which is observed to have a detrimental effect on the performance of THz wireless systems (Boulogeorgos et al., 2019; Boulogeorgos and Alexiou, 2020a).

Scanning the technical literature, we can identify several contributions that employ ML for user association in sub-THz and THz wireless networks (Zhang H. et al., 2019; Khan et al., 2019; Liu R. et al., 2020; Khan L. U. et al., 2020; Chou et al., 2020; Li et al., 2020c; Elsayed et al., 2020; Ghadikolaei et al., 2020; Hassan et al., 2020). In more detail, in (Liu R. et al., 2020), the authors employed multi-label classification ML that takes as input

both topological as well as network characteristics and returns a user association policy that satisfy users' latency demands. Meanwhile, in (Li et al., 2020c), the authors presented an online deep reinforcement learning (DRL) based algorithm for heterogeneous networks, where multiple parallel DNNs generate user association solutions and shared memory is used to tore the best association scheme. Moreover, in (Khan L. U. et al., 2020), Khan et al. introduces a federate learning approach to jointly minimize the latency and the effect of model accuracy losses due to channel uncertainties.The inputs of the ML algorithm presented in (Khan L. U. et al., 2020) are the device association and the resource block matrices, while the output is the resource alocation matrix. Likewise, in (Chou et al., 2020), a deep gradient reinforcement learning based policy was presented as a solution to the joint user association and resource allocation problem in mobile edge computing. The reinforcement learning agent of (Chou et al., 2020) takes as input the system state that is described by the current backhaul and resource block usage.

In (Hassan et al., 2020), two clustering approaches, namely least standard deviation user clustering and redistribution of BSs load-based clustering were presented that take into account the characteristics of both radio frequency (RF) and THz as well as the traffic load across the network in order to provide appropriate associations in RF and THz heterogeneous networks. Furthermore, in (Ghadikolaei et al., 2020), a transfer learning methodology was employed for inter-operator spectrum sharing in mmW cellular networks. The aforementioned methodology takes as input the network topology, the association matrix, the coordination matrix, the effective channels and outputs approximate the achievable data-rate. In (Zhang H. et al., 2019), an asynchronous distributed DNN based scheme, which takes as inputs the channel coefficient matrix, was reported as a solution to the joint user association and power minimization problem. In (Elsayed et al., 2020), Elsayed et al. reported a transfer Q-learning based strategy for joint user-cell association and selection of number of beams for the purpose of maximizing the aggregate network capacity in NOMA-mmW networks. Finally, in (Khan et al., 2019), the authors exploited distributed deep reinforcement learning (DDRL) and the asynchronous actor critic algorithm (A3C) to design a low complexity algorithm that returns a suboptimal solution for the vehicle-cell association problem in mmW. The DDRL takes as input the current state of the network that is described by a set of a predetermined number of channel observation, the current achievable and required data rates.

After associating UEs to APs, uninterrupted connectivity needs to be guaranteed. However, the network is continuously undergoing change; thus, its management should be adaptive as well. This is where the conventional heuristic based exploration of state space needs to be extended to support UE mobility in an online manner. Aspired by this, several contributions presented ML-based *mobility management* solutions that aim at accurately tracking the UE and proactively steering the AP and UE beams (Burghal et al., 2019; Guo et al., 2019) as well as performing hand-overs between beams and APs/BSs (Yan et al., 2019; Ali et al., 2020). In particular, in (Burghal et al., 2019), a RNN with a

modified cost function that takes as input the observed received signal as well as the previous AoA estimation, was employed to track the AoA in a mmW network. The proposed approach was shown to outperform the corresponding Kalman-based one in terms of accuracy. Moreover, in (Guo et al., 2019), a long short term memory (LSTM) structure was designed to prevent the user position in order to proactively perform beam steering in mmW vehicular networks. The structure uses as input the estimated by the BS channel vector. Additionally, in (Yan et al., 2019), the authors employed kNN to predict handover decisions without involving time-consuming target selection and beam training processes in mmW vehicle-to-infrastructure (V2I) wireless topologies. Finally, in (Yajnanarayana et al., 2020), centralized Q-learning was employed that takes into account the current received signal strength in order to provide real-time controlling capabilities to the hand-over process between neighbor BS in directional wireless systems.

Another challenging task of THz wireless networks is *routing*. The limited transmission range in combination with the transmission power constraints and memory limitations of mobile devices render conventional routing strategies that are employed in lower-frequency networks unsuitable for THz ones. Motivated by this, in (Wang C.-C. et al., 2020), the authors presented a reinforcement learning routing algorithm and compared it with NN and decision tree-based solutions. The results showed that the reinforcing learning approach not only provides on-line routing optimization suggestions but also outperforms the NN and decision tree ones. To the best of the authors knowledge, the aforementioned contribution is the only published one that discuss ML-based routing policies in high-frequency wireless networks.

To create a fully automated THz wireless network or even integrate THz technologies into current cellular networks, one of the essential problems that a network manager need to solve is *traffic clustering*. An accurate traffic clustering allows the detection of suspicious data and can aid in the identification of security gaps. However, as the diversity of the data increases, due to their generation for different type of sources, e.g., sensors, artificial/virtual reality devices, robotics, etc, traffic clustering may become a difficult task. Additionally, labeled samples are usually scarce and difficult to obtain.

To address the aforementioned challenges, several researchers turned their eye to ML. In particular, in (Noorbehbahani and Mansoori, 2018), Noorbehbahani et al. presented a semi-supervised method for traffic classification, which is based on x-means clustering algorithm and a label propagation technique. This approach takes as input network traffic flow data. It was tested in real-data and achieved 95% accuracy using a limited labeled data. Similarly, in (Bin and Hao, 2010), the authors reported a semi-supervised classification method that exploits both labeled and unlabeled samples. This method combines offline particle swarm optimization (PSO) to cluster the labeled and unlabeled samples of the dataset with a mapping approach that enables matching clusters to applications.

In (Wang et al., 2011), Wang et al. applied K-means algorithm that takes into account the correlations of network domain

TABLE 4 | ML algorithm types applied in network layer.

	NN	DNN	kNN	Decision trees	Random forest	Naive bayes	Bayesian network	k-Means	x-Means	Feature selection	EM	Multi-layer perception	DRL	Q-learning	A3C
							User association								
Li et al. (2020c)	–	–	–	–	–	–	–	–	–	–	–	–	✓	–	–
Khan et al. (2020b)	✓	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Chou et al. (2020)	–	–	–	–	–	–	–	–	–	–	–	–	✓	–	–
Hassan et al. (2020)	–	–	–	–	–	–	✓	–	–	–	–	–	–	–	–
Ghadikolaei et al. (2020)	–	–	–	–	–	–	–	–	–	–	–	–	–	✓	–
Zhang et al. (2019b)	–	✓	–	–	–	–	–	–	–	–	–	–	–	–	–
Elsayed et al. (2020)	–	–	–	–	–	–	–	–	–	–	–	–	–	✓	–
Khan et al. (2019)	–	–	–	–	–	–	–	–	–	–	–	–	✓	–	✓
							Mobility management								
Burghal et al. (2019)	✓	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Guo et al. (2019)	✓	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Yan et al. (2019)	–	–	✓	–	–	–	–	–	–	–	–	–	–	–	–
Yajnanarayana et al. (2020)	–	–	–	–	–	–	–	–	–	–	–	–	–	✓	–
							Routing								
Wang et al. (2020a)	✓	–	–	✓	–	–	–	–	–	–	–	–	–	–	–
							Traffic clustering								
Noorbehbahani and Mansoori (2018)	–	–	–	–	–	–	–	–	✓	–	–	–	–	–	–
Bin and Hao (2010)	✓	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Wang et al. (2011)	–	–	–	–	–	–	–	✓	–	–	–	–	–	–	–
Liu et al. (2007)	–	–	–	–	–	–	–	✓	–	–	–	–	–	–	–
Kumari et al. (2016)	–	–	–	–	–	–	–	✓	–	–	–	–	–	–	–
Su et al. (2018)	–	–	–	–	–	–	–	–	–	✓	–	–	–	–	–
Wang et al. (2013)	–	–	–	–	✓	–	–	–	–	–	–	–	–	–	–
Auld et al. (2007)	–	–	–	–	–	–	✓	–	–	–	–	–	–	–	–
Kruber et al. (2018)	–	–	–	–	✓	–	–	–	–	–	–	–	–	–	–
Singh (2015)	–	–	–	–	–	–	–	✓	–	–	✓	–	–	–	–
Wang and Yu (2008)	–	–	–	✓	–	✓	✓	✓	–	–	–	✓	–	–	–
Zhang et al. (2019a)	✓	–	–	–	–	–	–	–	–	–	–	–	–	–	–

background information and transforms them into pair-wise must-link constraints that are incorporated in the process of clustering. Experimental results highlighted that incorporating constraints in the clustering process can significantly improve the overall accuracy. Furthermore, in (Liu et al., 2007), Liu et al. employed feature selection to identify optimal feature sets and log transformation to improve the accuracy of K-means based

network traffic classification. Another use of K-means was presented in (Kumari et al., 2016), where the authors used it to detect networks cyber-attacks. In particular, the K-means in (Kumari et al., 2016) takes as input network traffic-related data and identifies irregularities. Moreover, a network traffic feature selection scheme that provides accurate suspicious flow detection was reported in (Su et al., 2018), whereas, in (Wang et al., 2013),

random forest was employed to perform the same task. Also, Bayesian ML was used to (Auld et al., 2007), which take as input transport control protocol traffic flaws in order to identify internet traffic. Although, this approach does not require access to packet content, it demands a significant set of training data. To deal with the lack of training data, in (Kruber et al., 2018), the authors presented an unsupervised random forest clustering methodology for automatic network traffic categorization. Meanwhile, in (Singh, 2015), the performance of EM and K-means based algorithms for network traffic clustering were compared and it was shown that K-means outperforms EM in terms of accuracy. In addition, in (Wang and Yu, 2008), the authors evaluated and compared the effectiveness of a number of supervised, unsupervised and feature selection algorithms in real-time traffic classification problem, which takes as input network's statistical characteristics. Naive Bayes, Bayesian networks, multilayer perception, decision trees, K-means, and best-first search were among the algorithms that their performance were quantified. The results revealed that in terms of accuracy decision trees outperforms all the aforementioned algorithms. Finally, in (Zhang C. et al., 2019), the authors combined LSTM with CNN in order to develop a two-layer convolution LSTM mechanism capable of accurately clustering traffic generated by different application types.

Table 4 connects the published contributions to the ML algorithms that was applied in the network layer. Note that some of the contributions in these table does not explicitly refer to THz networks, however, the presented algorithms can find applications to THz wireless systems, due to system and network topologies commonalities. For example in (Li et al., 2020c), the ML approach can be used in any heterogeneous network that consists of macro-, pico- and femto-cells. Notice that femto cells can be established in the THz band. Similarly, the contributions in (Liu et al., 2007; Bin and Hao, 2010; Noorbehbahani and Mansoori, 2018; Zhang H. et al., 2019; Khan L. U. et al., 2020; Chou et al., 2020) and (Kumari et al., 2016) can be applied in any femto-cell network that support high data traffic demands, such as the THz wireless networks, independently from the operation frequency. Moreover, (Auld et al., 2007; Wang et al., 2013; Kruber et al., 2018; Su et al., 2018; Burghal et al., 2019; Guo et al., 2019; Khan et al., 2019; Yan et al., 2019; Elsayed et al., 2020; Ghadikolaei et al., 2020) and (Wang and Yu, 2008) refer to mmW networks that support data-rates in the order of 100 Gb/s in 60 GHz. In such networks, both the AP and UEs employ high-gain antennas. Notice that THz wireless network are also designed to support data-rates in the order of 100 Gb/s and establish high directional links in order to ensure an acceptable transmission range. Thus, the aforementioned contributions can be adopted to THz wireless networks.

From Table 4, we observe that based on the network nature, i.e. fixed or mobile topology, supervised and reinforcement learning approaches are respectively employed for user association. Additionally, when the network had prior knowledge of the UE possible direction supervised learning approaches were employed for mobility management. On the

other hand, in problems in which the UE motion is stochastic, reinforcement learning mechanisms were adopted. For routing, where accuracy plays an important role and no instantaneous adaptation is required, supervised learning was used. Finally, for traffic clustering both supervised and unsupervised learning were employed. In more detail, unsupervised learning seems an attractive approach when searching for data irregularities.

2.4 Transport Layer

In order to design self-management wireless network that embrace efficient automation, accurate *traffic prediction* is necessary. However, traffic prediction is a challenging task due to the nonlinear and complex nature of traffic patterns. In face of this challenge, ML-based approaches was recently discussed. In this sense, in (Zhang C. et al., 2019), Zhang et al. presented a deep transfer learning algorithm that combines spatial-temporal cross-domain NNs with LSTM in order to extract the relationship between cross-domain datasets and external factors that influence the traffic generation. Building upon the extracted relationship, traffic predictions were conducted. Similarly, in (Zeng et al., 2020), a cross-service and regional fusion transfer learning strategy, which was based on spatial-temporal cross-domain NNs was reported. The modeled reported in (Zeng et al., 2020) takes as input wireless cellular traffic data of Milan area. Likewise, in (Qiu et al., 2018), the authors investigated the use of RNNs that are fed by traffic data for extracting the data spatio-temporal correlation; thus, improve the data traffic prediction. Moreover, a CNN was used in (Zhang et al., 2018) to predict the traffic demands in cellular wireless networks.

In (Azari et al., 2019), a decision tree and random forest based method was presented for user traffic prediction that enables proactively resource management. The algorithm inputs are the number and size of both uplink and downlink packets, the ratio of number of uplink to downlink packets, as well as the used communication protocol. Additionally, in (Senevirathna et al., 2020), a LSTM architecture was designed in order to predict the traffic variations in a machine-type communication scenario, in which the devices access the network in a random fashion. The algorithm takes as input the network traffic flow. In (Li Y. et al., 2020), SVM was employed to predict video traffic and improve the video stream quality in beyond the fifth generation (B5G) networks. Finally, in (Chen M. et al., 2020), a single NN architecture, in which the weights connected to the output are adjusted by means of linear regression, whereas other weights are randomly initialized, was employed. Simulation results revealed that this approach outperforms LSTM in terms of execution time.

Wireless content *caching* in THz wireless networks is an attractive way to reduce the service latency and alleviate backhaul pressure. The main idea is to proactively transfer the content to be requested by a single of a cluster of UEs to the nearest possible BS/AP. Towards this direction, several researchers presented ML-based caching policies (Cheng et al., 2019; Jiang et al., 2019; Saputra et al., 2019; Wang X. et al., 2020; Kirilin et al., 2020; Ye et al., 2020). Specifically, in (Ye et al., 2020), the authors reported a device to primary and secondary BS clustering approach based on the requested content location in

TABLE 5 | ML algorithm types applied in transport layer.

	NN	DNN	Decision tree	k-Means	Bayesian network	DRL	Q-learning
Traffic prediction							
Zhang et al. (2019a)	✓	–	–	–	–	–	–
Zeng et al. (2020)	✓	–	–	–	–	–	–
Qiu et al. (2018)	✓	–	–	–	–	–	–
Zhang et al. (2018)	✓	–	–	–	–	–	–
Azari et al. (2019)	–	–	✓	–	–	–	–
Senevirathna et al. (2020)	✓	–	–	–	–	–	–
Li et al. (2020b)	–	✓	–	–	–	–	–
Chen et al. (2020b)	✓	–	–	–	–	–	–
Caching							
Ye et al. (2020)	–	–	–	✓	–	–	–
Kirilin et al. (2020)	–	–	–	–	–	–	–
Cheng et al. (2019)	–	–	–	–	✓	–	–
Jiang et al. (2019)	–	–	–	–	–	–	✓
Saputra et al. (2019)	–	✓	–	–	–	–	–
Wang et al. (2020c)	–	–	–	–	–	–	–
Computational offloading							
Xu et al. (2019)	–	✓	–	–	–	–	–
Le and Tham (2018)	–	–	–	–	–	✓	–
Khan et al. (2020a)	–	–	–	–	–	✓	–
Qi et al. (2018)	–	–	–	✓	–	–	–
Alfakih et al. (2020)	–	–	–	–	–	–	✓
Nduwayezu et al. (2020)	–	–	–	–	–	✓	–

mmW ultra-dense wireless networks. Moreover, in (Kirilin et al., 2020),the authors presented a reinforcement learning architecture, which increases the caching hit rate by deciding whether or not to admit a requested object into the content delivery network, and whether to evict contents, when the cache is full. Moreover, in (Cheng et al., 2019), Bayesian learning method to predict personal preferences and estimate the individual content request probability, which reflects preferences in order to precache the most popular contents at the wireless network edge, like small-cell BSs. The algorithm takes as input the channel matrix. Meanwhile, in (Jiang et al., 2019), Wei et al. proposed a Q-learning based approach to coordinate the caching decision in a mobile device-to-device network. Additionally, in (Saputra et al., 2019), a DNN architecture, which takes as inputs each user and requested content identifier, was introduced that enables network's mobile edge nodes to collaborate and exchange information in order to minimize the content demand prediction error with ensuring no mobile user privacy leakage. Finally, in (Wang X. et al., 2020), a federated reinforcement learning based algorithm was presented that allows BSs to cooperatively device a common predictive model by employing, as initial local training inputs, the first-round training parameters of BSs, and exchange near-optimal local parameters between the participating BSs.

Despite the paramount importance that latency plays in THz wireless systems, another challenge that need to be addressed is the limited computing resource and battery of mobile UEs. To address this challenge, *computation offloading* approaches were proposed. These approaches demand intelligence in order to decide in which of the networks nodes the tasks should be offloaded in order to guarantee applications' latency demands. Towards this direction, in (Xu et al., 2019), the authors described a DNN architecture that minimizes the total task transmission latency and overhead by optimizing the task placement in cloud and edge computing nodes based on the computation resources and load of the participating nodes. Likewise, in (Le and Tham, 2018), the authors solved the same problem applying deep reinforcement learning and assuming that the tasks can be offloaded either in nearby cloudlets by means of device-to-device communications. Similarly, in (Khan I. et al., 2020), Khan et al. used DRL to maximize the energy efficiency of cloud and mobile edge computing assisted wireless networks that support a large variety of machine type applications, under the constraints of computing power resources and delays. Moreover, in (Qi et al., 2018), Qi et al. presented an unsupervised K-means clustering algorithm in order to identify center user group sets that enhance task offloading and allow load balancing. Additionally, in (Alfakih et al., 2020), the authors reported a Q-learning based algorithm that decides whether a UE's tasks should be offloaded in the nearest edge server, adjacent edge server, or remote cloud in order to minimize the total system cost that is quantified in terms of energy consumption and computing time delay. In this algorithm the uploading and downloading bandwidths define the state of the agent. Finally, in (Nduwayezu et al., 2020), the authors applied DRL in order to decide whether to execute a task
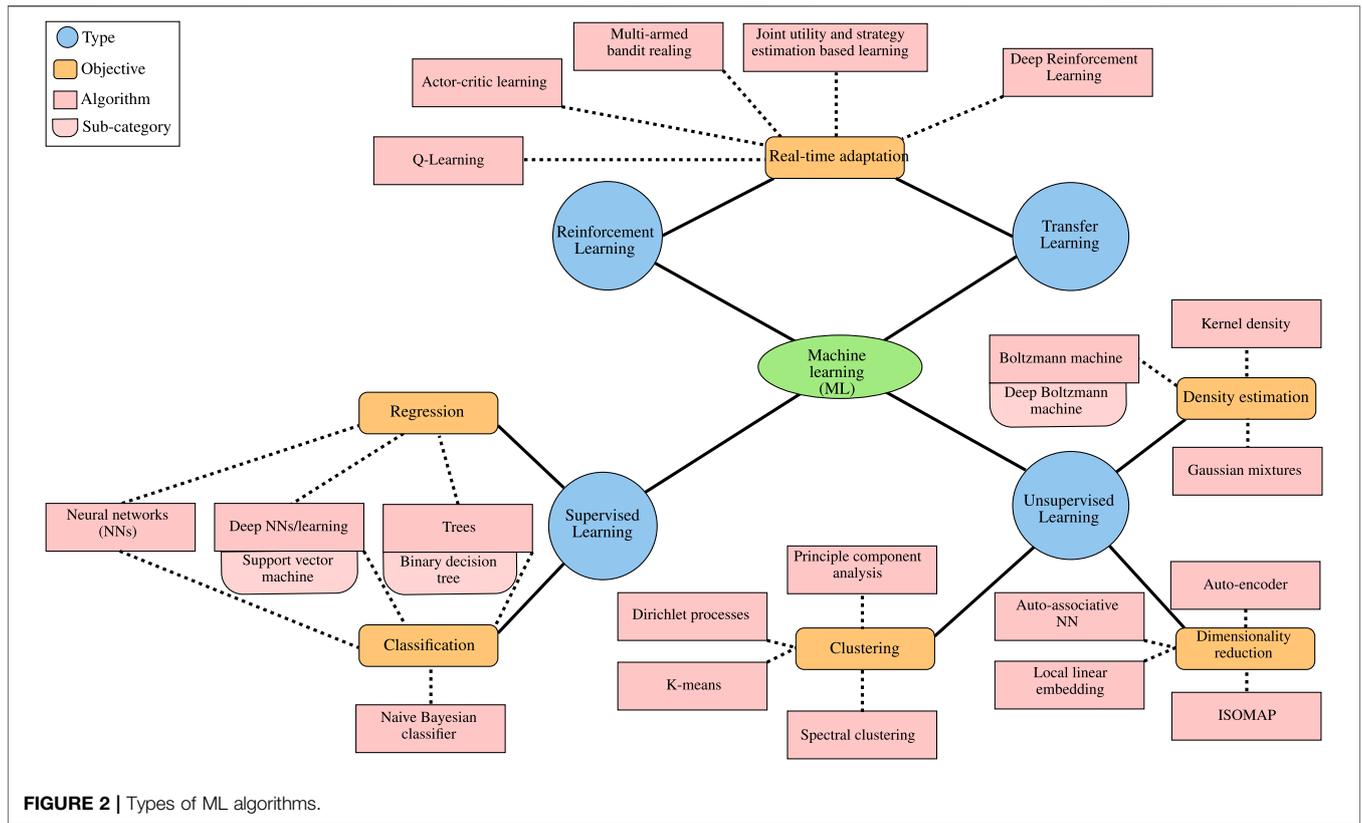
FIGURE 2 | Types of ML algorithms.
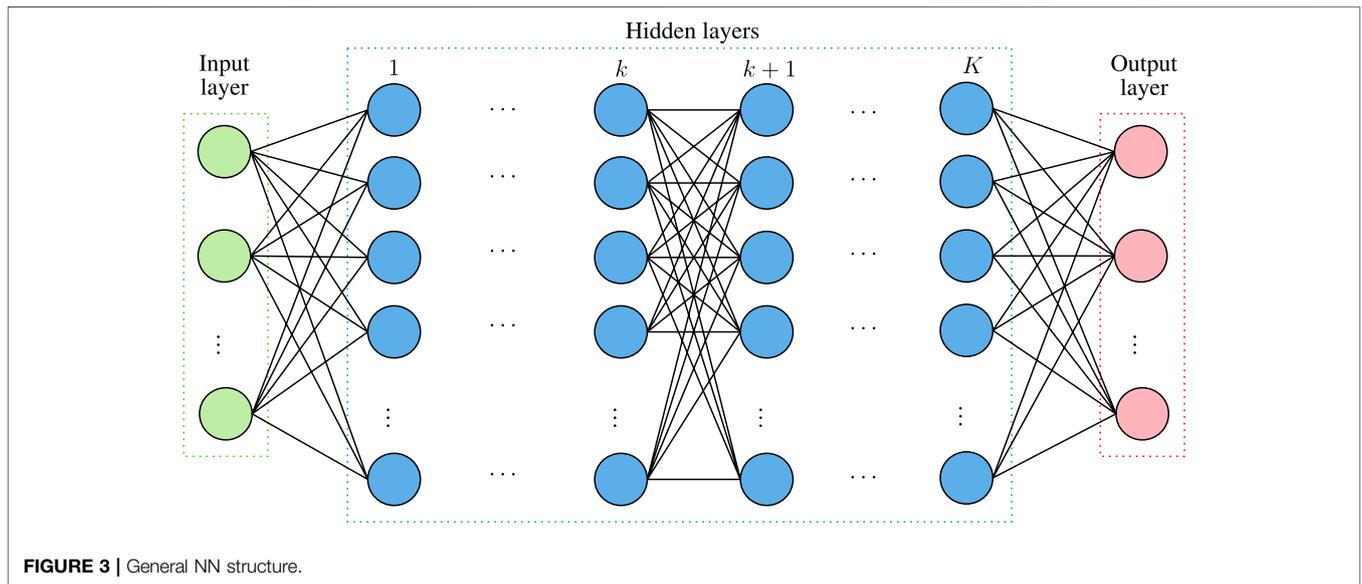
FIGURE 3 | General NN structure.

locally or remotely based on the computation rate maximization criterion in mobile edge computing assisted NOMA system. As input, the algorithm takes the input data size and workload.

Table 5 documents the different ML algorithms that have been used in transport layer. From this table, it becomes evident that supervised learning approaches have been the usual choice for traffic prediction problems, while unsupervised and reinforcement learning ones have been usually employed to solve caching and computational offloading problems. Of note, most of the contributions in **Table 5** refer to networks that have less backhaul than aggregated communication and computational fronthaul resources, without specifying their

FIGURE 4 | Three indicative examples of commonly used activation functions.
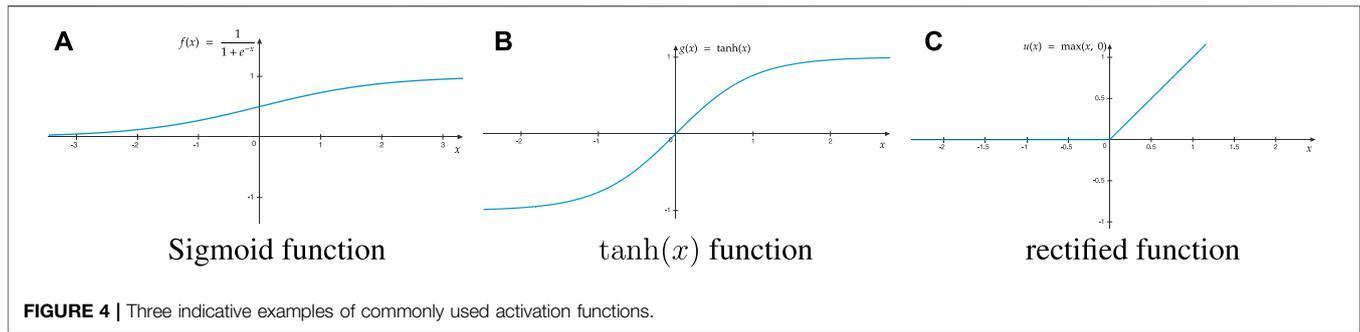
operation frequency. This is a common characteristic of stand-alone THz wireless deployments. As a consequence, they can be applied in THz wireless networks.

3 A METHODOLOGY TO SELECT A SUITABLE ML ALGORITHM

As illustrated in **Figure 2**, ML can be classified into four categories, namely 1) supervised, 2) unsupervised, 3) reinforcement, and 4) transfer learning. In what follows, we report the main features of each category and revisit indicative ML algorithms emphasizing their operation, training process, advantages and disadvantages.

3.1 Supervised Learning

Supervised learning focuses on extracting a mapping function between the input and output values based on a labeled dataset. As a consequence, it can be applied as a solution to regression and classification problems. In more detail, in this paper, the following supervised learning algorithms are discussed.

- *NNs*: are computing machines inspired by biological NN. In more detail and as illustrated in **Figure 3**, they consist of three types of layers, namely input, hidden, and output. Each layer has a certain number of nodes that are called neurons and process their input signal. A neuron in layer k implements a linear or non-linear manipulation, called *activation function*, of the input data and forwards its output to a number of edges, which connects neurons that belong to layer k with the ones of layer $k + 1$. In more detail, let

$$\mathbf{x}_k = \left[x_1^k, x_2^k, \ldots, x_l^k\right]^T \quad (1)$$

being the input vector of the k−th layer of the NN, and

$$\mathbf{w}_{k,n} = \left[w_1^{(k,n)}, w_2^{(k,n)}, \ldots, w_l^{(k,n)}\right]^T \quad (2)$$

being the weight vector of the n−th node of the k−th layer, then the output of this node would be

$$y_{k,n} = \sum_{i=1}^{l} w_i^{(k,n)} f\left(x_i^k\right) + b, \quad (3)$$

or equivalently

$$y_{k,n} = \left(\mathbf{w}_{k,n}\right)^T f\left(\mathbf{x}_k\right) + b, \quad (4)$$

where $f(\cdot)$ stands for a mapping function and b is a constant. Of note, if $f(\mathbf{x}_k) = \mathbf{x}_k$, then the mapping is linear; otherwise, the mapping is characterized as non-linear and usually returns a high-dimensional representation of $\mathbf{x}_k$.

The learning process aims at finding the optimal parameters $\mathbf{w}_{k,n}$ so that

$$\hat{y}_{k,n} = f\left(\mathbf{x}_k; \mathbf{w}_{k,n}\right) \quad (5)$$

to be as close as possible to the target $y_{k,n}$. To achieve this a cost/error function $\mathcal{J}(\mathbf{w}_{k,n})$ is defined and minimized, i.e.,

$$\frac{\partial \mathcal{J}\left(\mathbf{w}_{k,n}\right)}{\partial \mathbf{w}_{k,n}} = \mathbf{0}. \quad (6)$$

The analytical differentiation of **Eq. 6** is usually impossible; thus, numerical optimization methods are applied. The most commonly used methods are gradient descent as well as single and batch perceptron training (Graupe, 2013).

- *DNNs*: are NNs with multiple hidden layers that, as depicted in **Figure 4**, commonly employs tanh, sigmoid or rectified as an activation function. A DNN is segmented into two phases, i.e. training and execution. Training phase employs labeled data in order to extract the weights of all the activation functions of the DNN. Usually, the SGD with back-propagation algorithm is employed for this task. In general, as the number of hidden layers increases, the number of training data that is requires increases; however, the classification or regression accuracy also increases. In the execution phase, the DNN returns proper decisions based on its inputs, even when the input values have not been within the training data set. As a result, the main challenge of DNN is to optimally select its weights (Zhang L. et al., 2019).

The special types of DNNs have been extensively used in THz wireless systems and networks, namely CNN, and RNN.

RNN can be used for regression and classification. In contrast to conventional DNNs and as illustrated in **Figure 5**, it allows back-propagation by connecting neurons
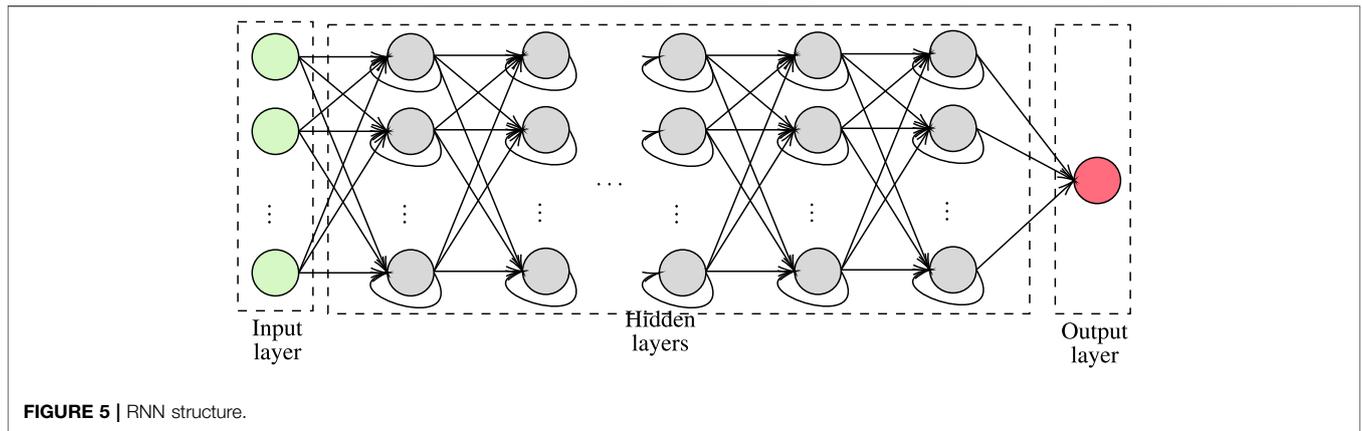
FIGURE 5 | RNN structure.

of layer k, with the ones of previous layers. In other words, it creates a memory that enables future inputs to be inherited by previous layers (Lipton et al., 2015). As a result, fewer tensor operations in comparison to the corresponding DNN need to be implemented, which is translated into lower computational complexity and training latency. Building upon this advantage, RNNs have been widely used for a large variety of applications ranging from automation modulation recognition, where channel correlation was discovered by exploiting the recurrent property, to traffic prediction, in which the data spatial-temporal correlation may play an important role.

Finally, *CNNs* have been employed as solutions to several THz wireless networks problems from AMR to traffic prediction. Their objective is to identify local correlations within the data and exploit them in order to reduce the number of parameters as we move from the input to the output through the hidden layers. In this type of networks, a hidden layer may play the role of a convolution, a rectifier linear unit (RELU), a pooling, or a flattening layer (Boulogeorgos et al., 2020). Convolution layers are used to extract the distinguished feature of each sample, while RELUs impose decision boundaries. Likewise, pooling layers are responsible for spatial dimensions down-sampling. Last, flattening is used to reorganize the values of high-dimensional matrices into vectors.

- *SVM*: can be employed as a solution for both high-dimensional regression and classification problems by mapping the original feature space into a higher-dimensional one, in which their discriminability is increased (Gholami and Fakhari, 2017). In other words, SVM aims at creating a space in which the minimum distance between nearest points are maximized. In this direction, let us describe the new space as a linear transformation of the original one, which can be described according to the following kernel function:

$$\mathbf{b}^t \hat{\mathbf{x}} + c = 0, \tag{7}$$

where $\mathbf{b}$ and c are SVM optimization parameters, while $\hat{\mathbf{x}}$ are the labeled sample that belongs to the set of
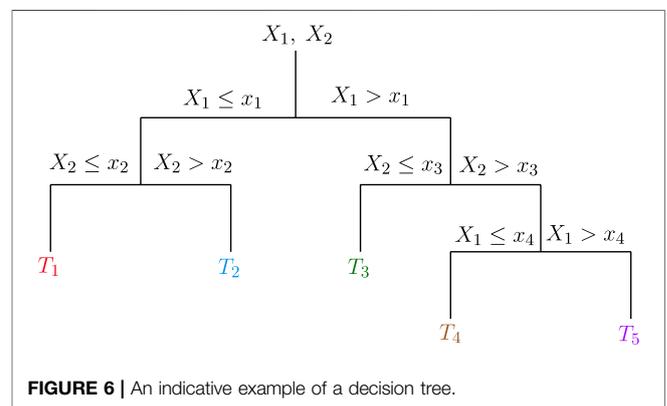
FIGURE 6 | An indicative example of a decision tree.

$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M]$, with the lowest separation distance. Note that $\mathbf{x}_m$, with $m = 1, 2, \ldots, M$, contains the N features of the n−th labeled sample. Then, their separation of the training samples can be expressed as

$$s_m = l_m \left(\mathbf{b}^t \mathbf{x}_m + c \right), \tag{8}$$

where l_m is the label of the m−th class. As a consequence, the optimization problem that describes SVM can be formulated as

$$\max_{\mathbf{b},c} \min_{m=1,2,\ldots,M} \frac{l_m s_m}{||\mathbf{b}||}$$

$$\text{s. t.} \quad C_1: s_m \geq \min_{m=1,2,\ldots,M} s_m, \quad m = 1, 2, \ldots, M \tag{9}$$

$$C_2: ||\mathbf{b}|| = 1$$

This problem may return a non-linear classification or regression of the original space. Another weakness is that training an SVM model is computationally expensive especially as the training data size increases. In practice, it can take long time to train an SVM model as the number of dimensions (features) increase in a dataset and the problem is exacerbated with increase in datapoints beyond a few hundreds of thousands. SVM has been extensively used for traffic clustering. The main challenge is the optimal selection of the kernel function. This function may be a linear, a polynomial, a radial, or an NN
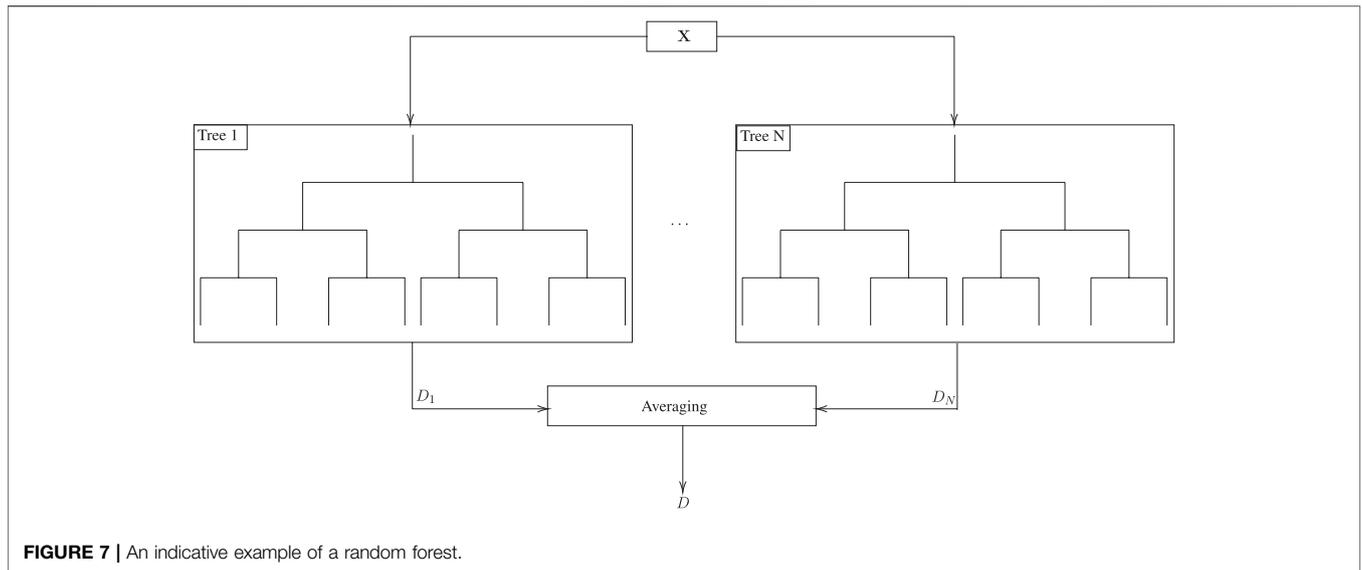
FIGURE 7 | An indicative example of a random forest.

one. To device a suitable kernel function, we usually apply inner product operations between input samples over the Hilbert space in order to extract feature mappings.

- *KNN*: A widely-used algorithm for classification is KNN. KNN consists of three steps, namely 1) distance calculation, 2) neighbor identification, and 3) label voting. To provide a comprehensive understanding of KNN, let us define a set of N training samples as $\mathcal{T} = \{(\mathbf{x}_1, l_1), (\mathbf{x}_2, l_2), \ldots, (\mathbf{x}_N, l_N),\}$, with $\mathbf{x}_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,M}]$, $i = 1, 2, \ldots, N$, being the samples of a class with label l_i, while $x_{i,m}$, $m = 1, 2, \ldots,$ M are M discrete features. Likewise, let us also define an unlabeled sample as $\tilde{\mathbf{x}} = [\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_M]$, where $\tilde{x}_m$ with $m = 1, 2, \ldots, M$ standing for the $m-$th feature of the unlabeled sample $\tilde{\mathbf{x}}$. During the first step, the Euclidean of the Manhattan distance between $\tilde{\mathbf{x}}$ and $\mathbf{x}_i$ is evaluated for all $i = 1, 2, \ldots, N$, according to

$$d_i = \begin{cases} \sqrt{\sum_{m=1}^{M} (\tilde{x}_m - x_{i,m})^2}, \text{Euclidean distance} \\ \sum_{m=1}^{M} |\tilde{x}_m - x_{i,m}|, \text{Manhattan distance} \end{cases} \quad (10)$$

In the second step, the K most similar labeled samples, i.e., with the lowest distance from $\tilde{\mathbf{x}}$, are identified. These samples are called "*K nearest neighbors*". In the final step, a majority rule is applied which classify $\tilde{\mathbf{x}}$ to the class in which the majority of the K nearest neighbors belongs to.

In THz wireless systems, KNN has been employed for channel estimation and beam tracking as well as mobility management purposes. Its main challenge is to appropriately select K. On the one hand, a large K can aim at counterbalancing the negative impact of noise. On the other hand, it may fuzzify the boundary of each class. This calls for heuristic approaches that returns approximations for K.

- *Decision trees*: are considered one the most attractive ML approach for both regression and clustering, due to their simplicity and intelligibility. They are defined by recursively segmenting the input space in order to create a local model for each one of the resulting regions. To provide a comprehensive understanding of decision trees operation, we consider an indicative tree that is depicted in **Figure 6**. We represent the target values by the tree's leaves, while branches stand for observations. In more detail, the first node checks whether the observation X_1 is lower or higher than the threshold x_1. If $X_1 \leq x_1$, then, we check whether the observation X_2 is lower or higher than another threshold x_2. If both $X_1 \leq x_1$ and $X_2 \leq x_2$, the decision tree returns the target value T_1. On the other hand, if $X_1 \leq x_1$ and $X_2 < x_2$, the target value T_2 is returned. Similarly, if $X_1 > x_1$, the decision tree checks whether X_2 is lower than the threshold x_3. If a positive answer is returned, the target value is set to T_3, otherwise, it checks whether X_1 is lower or higher of x_4. If it is lower, the target value T_4 is returned; otherwise, T_5 is returned.

In general, the decision tree model can be analytically expressed as

$$g(x) = \sum_{m=1}^{M} r_m \delta (\mathbf{X} \in \mathbf{R}_m), \quad (11)$$

where $\delta(\cdot)$ is an indicator function that is defined as

$$\delta (\mathbf{X} \in \mathbf{Y}) = \begin{cases} 1, \mathbf{X} \in \mathbf{Y} \\ 0, \text{otherwise} \end{cases} \quad (12)$$

Moreover, $\mathbf{R}_m$ stands for the $m-$th decision region, and r_m represents the mean response of this region. Finally, M represents the total number of regions. From **Eq. 11**, it becomes evident that training a decision tree network can be translated into finding the optimal partitioning, i.e., the optimal regions $\mathbf{R}_m$ with $m \in [1, M]$.
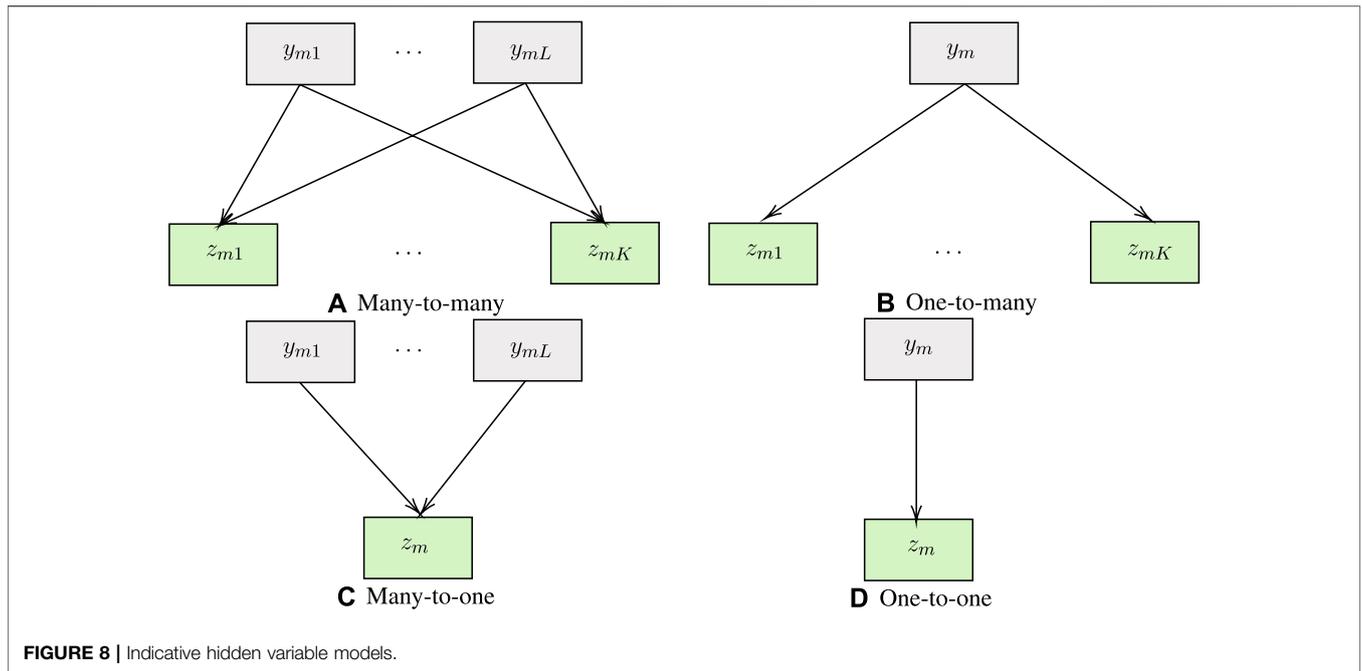
FIGURE 8 | Indicative hidden variable models.

This is usually an NP hard optimization problem and its solution require the implementation of greedy algorithms.

Although decision trees are easy to implement, they come with some fundamental limitations. In particular, they are have lower accuracy in comparison with NNs and DNNs. This is due to the greedy nature of the training process. Another disadvantage of decision trees is that their sensitive to changes to the input data. In other words, even small changes to the inputs may greatly affect the structure of the tree. In more detail, due to the hierarchical nature of the training process, errors that are caused at the top layers of the decision tree affect the rest of its structure.

- *Random forests*[3]: improve the accuracy of decision trees by averaging several estimations. In more detail and as illustrated in **Figure 7**, instead of training a single tree, random forest methodology is based on training N different trees using different sets of data, which are randomly chosen. The outputs of the N trees are averaged; hence, the random forest model can be described as

$$g(x) = \frac{1}{N} \sum_{n=1}^{N} g_n(x),$$ (13)

with $g_n(\cdot)$ standing for the n−th tree model.

The main challenge of random forests is to guarantee that the trees operate as uncorrelated predictors. To achieve this, the training data is randomly divided into subsets, where each subset is used to train a tree. It is very confusing to say "input variable subsets". Basically, one of the ideas behind the

---

[3]It is worth-noting that random forest is implementing a technique which is called bagging.

random splits of training data into subsets is that it makes the model resilient to outliers and overfitting. For example, if there is an outlier in one or more subsets, the model's accuracy is not skewed due to it, as this is an ensemble model and the outcome reflects joint decision of all trees, and since different trees have "seen" different distributions in data (due to random subsetting), it is expected to handle overfitting better. This indicates that there exists a trade-off between the accuracy and training latency/overhead. In more detail, as the number of trees increases and thus the accuracy of the random forest improves, the training set need to be lengthen. Therefore, both the training latency and the overhead in the network increases. Another disadvantage of random forests is the interpretability of the model is not as simple in comparison with singular or non-ensemble model such as a decision tree.

- *Naive Bayesian classifier*: aims at choosing the class that maximizes the posteriori probability of occurrence. In particular, let us define the vector $\mathbf{x} \in \{1,\ldots,R\}^S$, where R stands for the number of values for each feature, while D represents the number of features. Naive Bayesian classifier assigns a class conditional probability, $p(C_t | \mathbf{x})$ for each possible class C_t with $t \in [1, T]$. Of note, T stands for the number of different classes. By applying the Bayes' theorem, we can express $p(C_t | \mathbf{x})$ as

$$p(C_t | \mathbf{x}) = \frac{p(C_t) p(\mathbf{x} | C_t)}{p(\mathbf{x})}.$$ (14)

Moreover, by "naively" assuming that, for a given class label C_t, the features are conditionally independent, $p(\mathbf{x} | C_t)$ can be obtained as

FIGURE 9 | K-means algorithm.

$$p(\mathbf{x} \mid C_t) = \prod_{s=1}^{S} p(x_s \mid C_t). \qquad (15)$$

Here, conditional independence means that the algorithm treats all features as equally important and statistically independent of each other. This may apparently seem counter-intuitive as several features may indeed have some form of correlation. However, this "naive" assumption can often lead to good predictive accuracy due to the emphasis on evidence observed in the form of conditional probability of features, for a given outcome class of the predicted (target) variable. The independence assumption weakens the explainability of the predictions made by the Naive Bayes classifier but at the same time, the algorithm is very efficient to train because probabilities in 15) can be measured in a single data scan of the training dataset.

Based on **Eqs 14**, **15**, a class label can be assigned according to

$$\tilde{c} = \arg \max_{t \in [1,T]} p(C_t) \prod_{s=1}^{S} p(x_s \mid C_t). \qquad (16)$$

Note that based on the type of each feature, $\mathcal{U}_{s,t} = \{x_s | C_t\}$ may follow a Gaussian, Bernoulli, multinoulli well-defined distribution. Likewise, from **Eq. 16**, it becomes evident that the training problem is converted to a maximum likelihood one, which may generate overfitting issues and compromise the accuracy of the naive Bayes model.

3.2 Unsupervised Learning

Supervised learning highly depends on the existence of labeled datasets for training. However, in several practical scenario, no such datasets are available. In this case, *unsupervised learning* can be applied. Unsupervised learning aims at extracting data unknown features and identify the relationship between them and the system response. In more detail, unsupervised learning algorithms search for four types of relationships, namely: 1) many-to-many; 2) one-to-many; 3) many-to-one; and 4) one-to-one. This is graphically presented in **Figure 8**, where $y_{m1}, \cdots, y_{mL}$ denotes L hidden variables and $z_{m1}, \cdots, z_{mK}$ are K known ones,
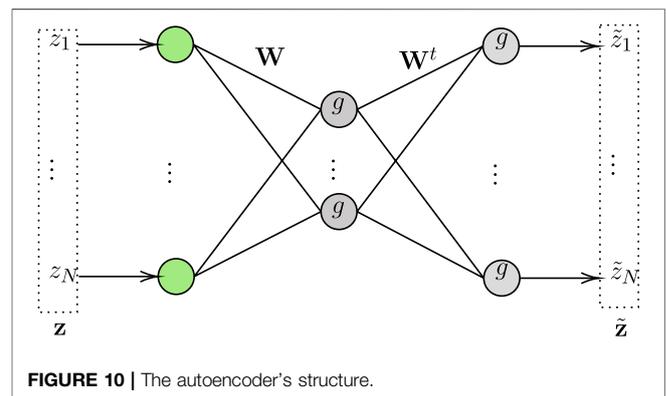
FIGURE 10 | The autoencoder's structure.

with $K \gg L$. These types of relationships can be used for clustering, density estimation, and dimensionality reduction. In more detail, in THz wireless systems and networks the following clustering approaches have been employed.

- *EM*: is a low-complexity iterative algorithm that aims at identifying maximum likelihood estimates of parameters by means of iteration between two phases, namely E and M. During the E phase, it infers the missing values, for a given set of parameters, while, in the M phase, it optimizes the parameters for a fixes "filled in" data set. In more detail, in the i–th iteration, at the E phase, EM computes an auxiliary function for the expectation of the log-likelihood using the parameters estimations of the $i - 1$ iteration, which can be expressed as

$$\mathcal{F}(\theta_{i-1} \mid \theta_i) = \mathbb{E}_{\mathbf{Z}|\mathbf{Y},\theta_i}[\log(p(\mathbf{Y}, \mathbf{Z} \mid \theta_{i-1}))], \qquad (17)$$

where $\mathbf{Y}$ and $\mathbf{Z}$ are respectively the set of known and hidden variables, whereas, θ_i is the set of the unknown parameter at the i–th iteration. In the M phase, the parameters that maximize the expected log-likelihood are determined based on the following formula:

$$\tilde{\theta}_i = \arg\max\boldsymbol{\theta}_i \mathcal{F}\left(\theta_{i-1} \mid \theta_i\right). \tag{18}$$

This process stops when parameters convergence is achieved.

The EM approach can be used to parameter estimation problems that are based on popular statistics models, like mixture Gaussian, hidden Markov, etc. However, it has an important disadvantage. It cannot guarantee convergence to a global optimum and not to a local one. As a result, it usually achieves poor performance in high-dimensional problems.

- *K-means*: The objective of K-means is to partition M unlabeled samples into K clusters, such as each sample to belong to exactly one cluster, based on their similarity in terms of distance. In order to achieve this a two step approach is followed, according to which, each training sample is assigned to one of the K clusters based on its distance from the cluster center[4], i.e., as a solution to the following optimization problem:

$$C^* = \arg\min_{\mathbf{c}} \sum_{l=1}^{K} \sum_{\mathbf{x}\in \mathbf{c}_k} \left\| \mathbf{x} - \mu_k \right\|^2, \tag{19}$$

where C^* is the optimal cluster segmentation, $\mathbf{x}$ is the set of samples and μ_k is the mean of the samples that belongs in the cluster $\mathbf{c}_k$. Then, the cluster center is updated, based on the new samples that are included or the ones that was removed from each cluster. This process is reaped until convergence is achieved. A graphical representation of the K-means algorithm is depicted in **Figure 9**.

From **Eq. 19**, it becomes evident that the clustering optimization problem is a NP-hard one. As a result, a heuristic algorithm needs to be employed in order to solve it. However, such algorithms cannot guarantee convergence in a global optimum. Its result is tightly connected to the initial cluster selections as well as their centers. Despite this disadvantage, it has been used as a solution to a wide range of problems spanning from beamfoming design to caching.

Feature selection or dimensionality reduction can be seen as a preprocessing phase of ML, since it enables the elimination of correlated features by means of feature transformation. In this direction, the following feature selection/dimensionality reduction approaches have been employed in THz wireless systems and networks:

- *Principle component analysis (PCA)*: implements an orthogonal transformation in order to convert potential correlated features of a dataset into uncorrelated ones that are called principle components. The operation pillar of PCA is based on the dogma that the first principle component has larger variance than the second, which in turn has larger than the third, and so on. As the variance decreases, the amount of the encaptulated information of the original features decreases, given that the original feature

FIGURE 11 | The Boltzmann machine's structure.

has a considerable correlation. Motivated by this, PCA aims at solving the following maximization problem:

$$\mathbf{v}^* = \max_{\mathbf{v}} \frac{1}{N} \sum_{n=1}^{N} \left(\mathbf{y}_n^t \mathbf{v}\right)^2, \tag{20}$$

or equivalently

$$\mathbf{v}^* = \max_{\mathbf{v}} \mathbf{v}_n^t \mathbf{G} \mathbf{v}, \tag{21}$$

where $\mathbf{G}$ stands for the covariance matrix of the training dataset that can be expressed as

$$\mathbf{G} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{y}_n \mathbf{y}_n^t, \tag{22}$$

while $\mathbf{y}_n$ represent the $n-$th training dataset, and $\mathbf{v}$ is a unit vector.

Notice that the solution of **Eq. 21**, is the eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_K$ of $\mathbf{G}$, with $K > M$, where M the number of the original features. As a result, the dimensionality reduction can be mathematically written as

$$\mathbf{z}_n = [\mathbf{v}_1, \ldots, \mathbf{v}_K]^t \mathbf{y}_n. \tag{23}$$

- *Auto-encoder*: is a feed-forward NN that it is trained to predict its inputs. As a consequence, the number of inputs is the same as the one of the outputs. As depicted in **Figure 10**, the auto-encoder is a three-layer network that can be described as (Vincent et al., 2010)

$$\tilde{\mathbf{z}} = \mathbf{g}\left(\mathbf{W}^t \left(\mathbf{W}\mathbf{z} + \mathbf{c}\right) + \mathbf{b}\right), \tag{24}$$

where $\mathbf{W}$, $\mathbf{b}$, and $\mathbf{c}$ are auto-encoder's parameters, while $\mathbf{g}$ stands for an activation function vector. Finally, $\mathbf{z}$ and $\tilde{\mathbf{z}}$ are $N-$dimensional vectors that contains the auto-encoder's inputs and outputs.

The first layer of the auto-encoder is a bottleneck one that is responsible for preventing the system from learning a trivial

---

[4]Note that as the cluster of the center, most K-means implementations use the mean of the samples that belong in the same cluster.
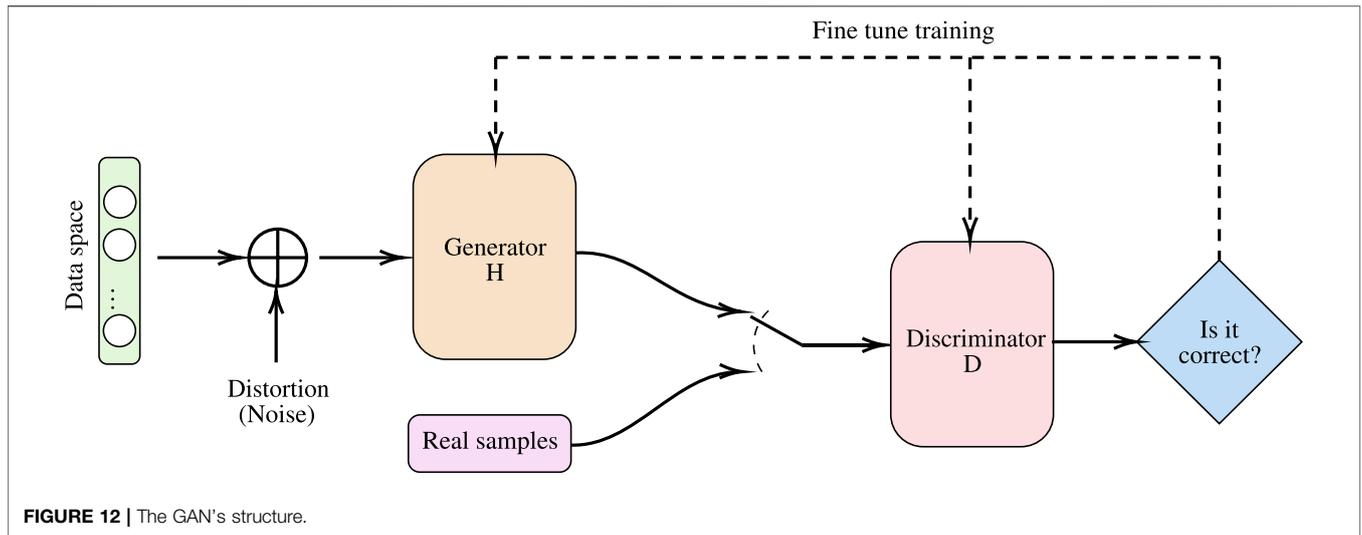
FIGURE 12 | The GAN's structure.

identity mapping. The connection weights between the first and the second as well as the second and the third layer are shared, i.e., $\mathbf{W}$ and $\mathbf{W}^t$. Note that the objective of the training phase of the auto-encoder is to select a suitable $\mathbf{W}$ in order to minimize the input-output error. This is usually performed by feeding inputs and outputs in the training algorithm. Finally, the hidden nodes are used to capture the most relevant dataset aspects.

In comparison with PCA, auto-encoder is capable of performing not only linear but also non-linear transformations. However, since a greedy algorithm is usually employed for its training, it is also sensitive to fitting errors. As a result, an important task for using auto-encoders is to appropriately select the activation function to be employed.

- *ISOMAP*: also called manifold learning, is a non-linear dimensionality reduction approach that is build upon the principle of preserving the geodesic distances[5] of the lower-dimension (Tenenbaum, 2000). In more detail, its implementation follows four stages. In the first stage, the neighbors of each points are determined. In particular, for each pair of points i, j, the input space distance $d_X(i, j)$, is calculated. Points that have an input distance lower than a predetermined fixed radius, ϵ, are considered neighboring points. Building upon the first stage, the second one generates neighborhood graph that connects each point with its neighbors. Then, in the third stage, the shortest path between two nodes of the neighborhood graph is evaluated, according to the Dijkstra's or Floyd-Warshall algorithm (Cormen et al., 2009). Towards this direction, the geodesic distance, $d_G(i, j$, between all pair of point on the manifold is calculated as

---

[5]Note that the geodestic distance is the distance between two points following the available/possible path that connects them.

$$d_G(i, j) = \min\left(d_a(i, j), d_a(i, k) + d_a(k, i)\right) \quad (25)$$

for each $k \in [1, N]$ were N stands for the total number of points. Moreover, $d_a(m, n)$ is an auxiliary variable that can be defined as

$$d_a(i, j) = \begin{cases} d_X(i, j), & \text{for } (i, j) \in \mathcal{A} \\ \infty & \text{for } (i, j) \notin \mathcal{A} \end{cases}, \quad (26)$$

with $\mathcal{A}$ being the set of neighboring points. Finally, in the forth stage, the lower-dimensional embedding, y_i, is extracted by minimizing the embedding cost function

$$J_c(j) = \sum_{i=1}^{N} \left(\|y_i - y_j\| - d_G(i, j)\right)^2. \quad (27)$$

This approach finds several applications in identifying non-linear correlated hidden variables, such as traffic clustering. However, it comes with an important disadvantage. In general, it is topologically unstable (Schwartz et al., 1988); thus, it should only be applied after extensive pre-processing of the data (Balasubramanian, 2002).

For density estimation the Boltzmann machine is usually used. *Boltzmann machines*: are employed to discover hidden features, which denote complex regulations in the training dataset. As a result, they can be used to extract the stochastic dynamics of datasets. Regarding its structure, as presented in **Figure 11**, it can be seen as a network, in which its units are bidirectionally symmetrically connected to each other with fixed weights and return stochastic binary decisions, i.e., 0 or 1. A unit can be a visible or a hidden node of the network. Notice that we can interact only with visible units. A unit that returns a state 0 indicates that the system rejects a hypothesis, while in state 1, it accepts it. The weight on a connection stands for a pairwise constraint between two hypotheses. In particular, positive weights refers to hypotheses that support each other, i.e., if one is accepted, the other should also be accepted, while negative ones indicate that only one of the two hypothesis can be

FIGURE 13 | Reinforcement learning structure.
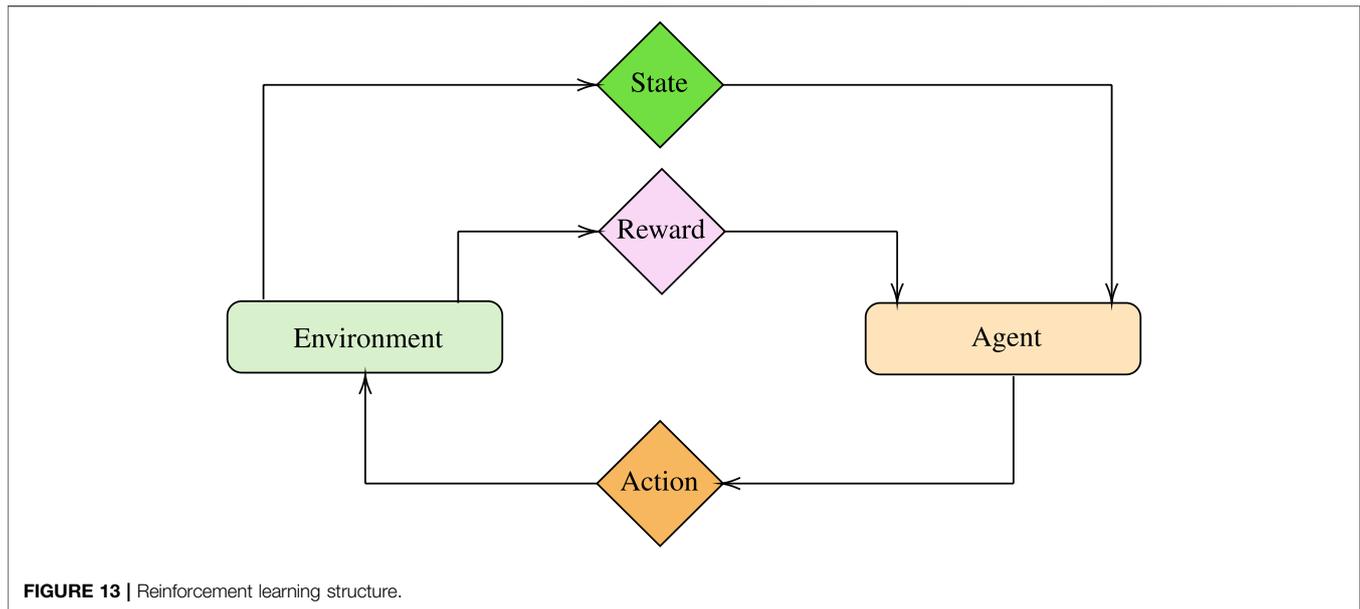
accepted. The objective of a Boltzmann machine is to minimize its global state, which is defined as (Alla and Adari, 2019)

$$E = -\sum_{i<j} w_{ij} s_i s_j + \sum_i \theta_i s_i, \qquad (28)$$

where w_{ij} is the i, j connection weight, s_i and s_j respectively stand for the i–th and j–th units states, and θ_i represents a threshold. To achieve this, we usually employ heuristic algorithms. As a consequence, Boltzmann machines suffer from performance degradation when the network is scaled up in size.

Finally, a special ML framework that can be used for both supervised and unsupervised learning is *GAN* (Goodfellow et al., 2014). GANs are usually used to generate new data that have the same statistics as the training ones (Ho and Ermon, 2016). As shown in **Figure 12**, they consist of two networks, a *generator* and a *discriminator*. The generator produces new samples after providing an estimation of the dataset distribution, while the discriminator compares the generated samples distribution to the one that arises by the unlabeled data. The generator's distribution, $p_Z(z)$, over data z is defined by introducing a prior on input samples distortion, which is distributed according to $p_N(n)$, and a mapping to the data space, which is represented by $H(n\,|\,\theta)$, where H describes a multi-layer preceptor (MLP) with parameters θ. The discriminator that follows utilizes another MLP, which we denote $D(z\,|\,\theta_D)$ with parameter θ_D, that outputs a scalar that indicates the probability that the training samples and the data generated by H are labeled correctly. In this direction, the generator is trained in order to minimize the term

$$\mathcal{F}_H = \log(1 - D(H(n\,|\,\theta))), \qquad (29)$$

whereas the discriminator aims at maximizing the term

$$\mathcal{F}_G = \log(H(z)). \qquad (30)$$

In other words, a two-players min-max game is formulated, which can be solved by employing iterative numerical methods.

In THz wireless systems, GANs have been applied as the solution to AMR problems, due to their capability to predict the different versions of the received signal, when a specific symbol is transmitted. However, it comes with some disadvantages. First of all, the training phase may be unstable, if not considerable amount of time is not spend in this phase. Moreover, to deal with training instability, visual examination may be needed in each step; this creates an important workload to the ML designer. Finally, it has no density estimation capabilities. This indicates that it cannot be used for anomaly detection.

3.3 Reinforcement and Transfer Learning

This section is devoted to reporting the reinforcement and transfer learning approaches that have been employed in THz wireless systems and networks with emphasis to their operation principles, applications and challenges. In this direction, in **Section 3.3.1**, reinforcement learning approaches are documented, while, in **Section 3.3.2**, the transfer learning ones are reported.

3.3.1 Reinforcement Learning

As illustrated in **Figure 13**, the fundamental idea of reinforcement learning is to resemble the trail and error process by employing an agent that continuously interacts with the environment (Kiumarsi et al., 2018). In more detail, an agent sense the environment state and applies an action that affect the environment. As a response, the environment returns a quantified reward. Of note, the environment stage is influenced by two factors, namely 1) the environment itself; and 2) the agent's action. Similarly, the award is evaluated based on its impact to the environment and the action of the reinforcement learning method. As a result, reinforcement learning approaches allows real or almost-real time interactions to environmental changes. This characteristic is a key requirement in several system and network operation processes in all the OSI layers. Thus, they have extensively adopted as solutions to a wide range of problems,
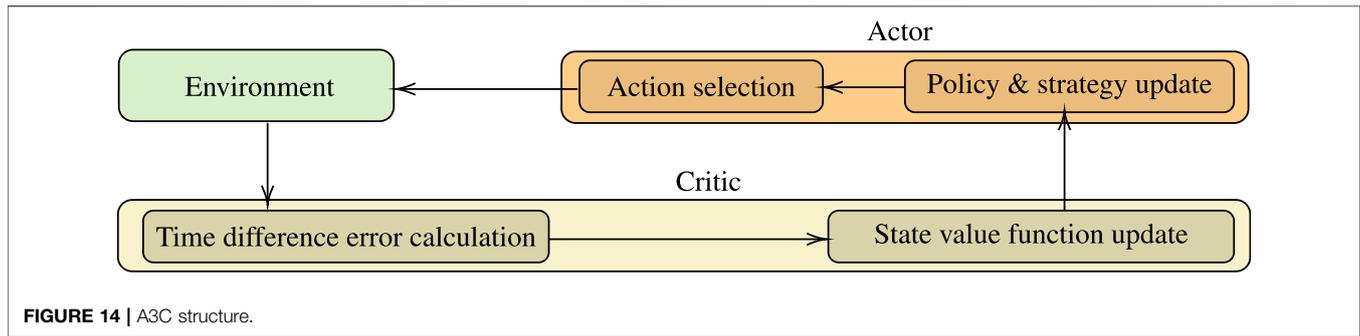
FIGURE 14 | A3C structure.

such as beamforming design, power management, blockage avoidance, user association, mobility management, caching, and computational offloading. Likewise, in contrast to conventional optimization approaches that focuses on immediate reward maximization, reinforcement learning aims at long-term reward. This is achieved by taking into account in the optimization process both the immediate and the future reward; thus, allowing intelligent prediction of the future system's state. The following reinforcement learning algorithms have been applied in THz wireless systems and networks:

- *Q-learning*: or as alternatively called *temporal-difference learning*, is a commonly adopted model-free reinforcement learning approach capable of directly acquiring knowledge from raw experience without requiring either an environmental model or a delayed reward system. Its interaction with the environment is based on a state-action value function that is called Q-function, which is continuously updated in order to achieve maximization by means of selecting an appropriate action, i.e.,

$$A^* = \arg\max_{A \in \mathcal{A}} Q(S, A), \qquad (31)$$

where S stands for the system state, A represents the selected action among the available ones that are included in the set $\mathcal{A}$. Moreover, $Q(\cdot, \cdot)$ is the Q-function, which is updated, according to (Watkins and Dayan, 1992), as

$$\hat{Q}(S, A) = (1 - c)Q(S, A) + c\left(r + d\max_{A \in \mathcal{A}} Q(S', A)\right). \qquad (32)$$

In **Eq. 32**, c and d respectively denote the updated weight and the discount factor, while r is a constant.

- *Deep reinforcement learning*: or *deep Q-learning* is usually applied in problems in which the dimensions of the state and action spaces are quite large. In these scenarios, the use of a Q-function as a table that contains values for each state and action is deemed impractical. To address this issue, we train a NN with parameters θ, which is responsible for the estimation of the Q values through a Q-function approximation, i.e., $Q_N(S, A; \theta) \approx Q(S, A)$. The training phase aims at minimizing in each step i a loss function that can be expressed as

$$L_i(\theta_i) = \mathbb{E}_{S, A, r, S'|\rho_s}\left[\left(T\left(S, S', A, A', r; \theta_i, \theta_{i-1}\right)\right)^2\right], \qquad (33)$$

where

$$T\left(S, S', A, A', r; \theta_i, \theta_{i-1}\right) = r + d\max_{A'} Q_N\left(S', A'; \theta_{i-1}\right)$$
$$- Q_N(S, A; \theta_i) \qquad (34)$$

stands for the temporal difference, while ρ_s represents the system's behavior distribution.

- *A3C*: is another reinforcement learning approach that, as presented in **Figure 14**, is composed by three units, namely *actor*, *critic*, and *environment*. The actor makes an initial action selection A from the set of available actions $\mathcal{A}$, based on a current policy or strategy. Next, the critic computes the new state value, which was extracted due to the environment variation, and updates the a time difference error (TDE). The new TDE is fed to the actor, which create a revised policy and/or strategy. The policy update is usually based on a Boltzmann distribution. The A3C is going to converge to an optimal state, after revisiting each action for each state by infinite times (Singh et al., 2000).

Reinforcement learning faces an important challenge. In more detail, the methodology to design the optimal state, action, reward/cost in different that enables convergence into optimal system performance depends on the scenario under investigation. As a consequence, in low-dimension state-action spaces in which all the space-action pairs can be documented into Q-value tables and explored by the reinforcement learning algorithm, a near-optimal solution can be rapidly identified. However, as the state-action space dimension increase, the reinforcement learning algorithm performance degrades, since several state-action pairs may remain unexplored. To counterbalance this, deep reinforcement learning is adopted. However, this approach comes with the training latency of NNs.

3.3.2 Transfer Learning

To avoid training latency, the concept of *transfer learning* was born, according to which knowledge from a specific domain can be used to speed up the learning process. In more detail, the aforementioned knowledge can be represented as Q-values of Q-learning and A3C algorithms, or NN weights in deep

reinforcement learning approaches. The Q values and weights may have been learned by an agent in a former and similar environment (Taylor and Stone, 2009).

Transfer layer have enabled several operation in wireless THz systems that vary from beamforming design to computational offloading. However, it comes with an important drawback. If the difference between former and current tasks and environments are important, the knowledge that is transferred will cause a negative impact to the system performance (Sun et al., 2019).

3.4 ML Algorithm Selection Guidelines

This section provide a systematic methodology that can be used to select the appropriate family of algorithms in order to solve a ML problem. In this direction, the first step is to categorize the problem under investigation. This can be achieved by examining the type of the input data and the expected outcomes. In more details, if labeled input data are provided, a supervised learning ML strategy can be selected. On the other hand, if the input data are unlabeled, unsupervised ML algorithms should be employed. Finally, if no data exists and the model need to interact with the environment, reinforcement or transfer learning algorithms should be applied, based on the absence or existence of simulation data.

Next, the outputs should be examined in order to identify the category of the problem that we want to solve. In particular, if the output data of the ML algorithm is a number, the problem is a regression one. Regression is usually used for AMR, beam training, signal detection, beam tracking, beamforming design, blockage avoidance, mobility management, and traffic prediction. Additionally, if the output of the model is a class and the number of the expected classes is predefined, then the problem is a classification one. An indicative example of a classification problem is user association. In both cases of regression and classification, NNs, DNN, naive Bayes, decision trees or random forests will be applied. To choose between the aforementioned algorithms, we should first examine the variation of the input data. If they have a small variation and low-latency or interpretability are key requirements decision trees will most likely used. This is the reason why decision trees are attractive approaching fro routing problems. On the other hand, if they have small variation but the latency and interpretability existence are not the main requirements, a random forest will be applied. On the contrary, if the input data have a relatively large variation, naive Bayes or NNs/DNNs will be employed based on whether they follow or not a well-known distribution.

On the other hand, if the objective of the problem is to categorize data into an initially unknown-number of classes, its a clustering problem. Indicative examples of such problems are traffic clustering, caching, and computational offloading. These problems can be solved by employing k-Means, k-Median, EM and hierarchical clustering. k-Means and k-Medians are usually selected in high-dimensional problems, while EM and hierarchical clustering in low-dimensional problems.

Another objective of ML problem is to improve the system and/or network performance. Such problems belong to the optimization category and are usually multi-variate ones. In

THz wireless systems and networks, several researchers have employed gradient descent and reinforcement/transfer learning algorithms, either to predict the optimal operation point or to correct it. Similarly, recommendation problems are the ones that return options based on the history of actions and are usually solved by employing reinforcement/transfer learning algorithms. Reinforcement/transfer learning provides high-adaptability to environmental changes and requires no training. Therefore, their suitable for problems that require fast adaptation, like channel allocation, power management, blockage avoidance and user association in mobile THz wireless networks, as well as computational offloading.

Finally, if the goal is to obtain insight from data for pattern recognition or anomaly detection, then dimensional reduction or feature selection algorithms can be applied, such as PCA, auto-encoder or ISOMAP. PCA is usually applied if the data are linearly correlated. On the other hand, auto-encoder and ISOMAP achieve acceptable performance in non-linear correlated datasets. These algorithms can find application in beamforming design and traffic clustering.

4 DEPLOYMENT STRATEGIES

B5G THz systems are expected to satisfy an ever increasing data connectivity, data rate and throughput demands. AI methods are positioned to play a central role to enable the functional and non functional demands expected in these systems. AI would be integrated in various layers of the network management stack as shown in **Figure 1** and presented in **Section 3**. In this section, we present an overview of deployment time strategies that must be considered before AI and/or ML based models are operationalized. Here, operationalization refers to the deployment of a well-trained and tested AI/ML model in production to help automate decision making in real time. This stage is usually preceded by staging where the models are observed in production settings and monitored but do not drive the decision making process.

In wireless architectures or any complex service architecture expected to perform complex decision making in real time, deployment aspects of AI/ML models are considered early on, during the design and implementation of management functions that would integrate AI/ML models. This integration must consider that AI/ML models arguably follow a lifecycle that needs to be regularly managed. This lifecycle begins firstly with the conceptual problem formulation phase, which targets a certain question to be answered with a prediction. Secondly, a physical solution usually in the form of a training process is devised which consumes training data, processes it and outputs a model. Within this training process, essential steps such feature engineering, normalization, regularization and/or up/down sampling may have been employed to train a model that best serves a performance criteria such as accuracy, recall, precision, F1 or quality metrics related to error margins. The training process requires some craft and good understanding of the AI/ML fundamentals and how the target attribute (often termed as label) is shaped. As a standard cross-validation design pattern is used for training and multiple testing methods are employed to establish the model's overfitting,
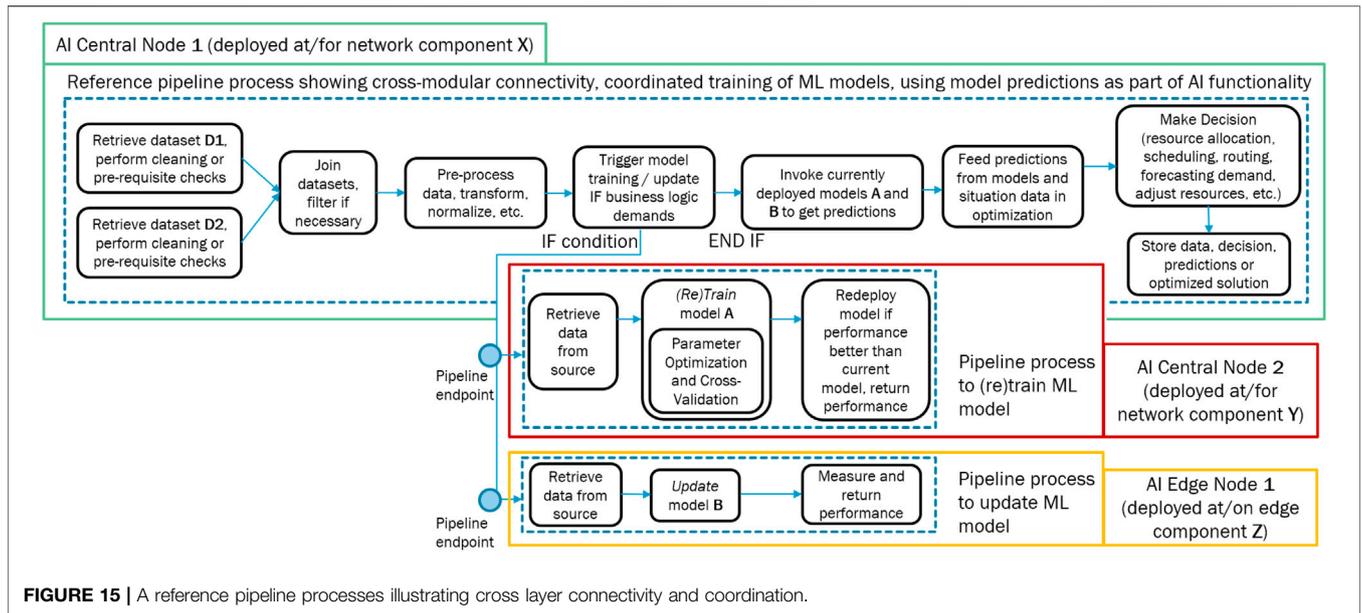
FIGURE 15 | A reference pipeline processes illustrating cross layer connectivity and coordination.

underfitting and generalization capabilities and further requirements. This activity is at first manual and done in collaboration with domain experts and data analysts. Once the training and testing process is completed, the third stage of the lifecylce is approached namely which is deployment.

Deployment of AI/ML models needs to take into consideration more than just the application of the model to get predictions. We especially sketch out two parallel formations have to be put in place for deployment. First, the performance in terms of accuracy and response time is to be continually monitored, and secondly, for some kind of models, it may be necessary to update or retrain the model at regular intervals, on new training data that may have become available over time. It should be noted that this data may become available on the periphery of the network or on edges, and may need to be transmitted or processed at the edge.

On the one hand, the AI/ML model deployment shares some commonalities with how the software components in an information technology (IT) system are updated e.g., following blue green deployment, but also some major differences, such as the deployment usually follows a primary-secondary; or sometimes referred to as "primary-challenger" dual models. The primary model is the one being used for automated decision making whereas the challenger is used as a stand by if at any point in time, the observed functional performance (e.g., accuracy) of the primary drops significantly lower than the primary model's performance. This also helps as a counter measure to drift or shift of concept phenomenon that is observed in ML systems when the data gradually exhibits different distributions unlike what the models are originally trained on. Finally, the models may need to be scaled up when the overall management function faces an increased demand in connectivity, hand-over or adaptation scenarios. For instance, a large amount of UEs approach an area which is governed by a few APs, or the data-rate demand increases significantly due to a major event or news. In such scenarios, horizontal scaling of the model plays an effective role to

continue to deliver QoS levels expected by the users and hence, enough infrastructure resources should be available for manual or automatic scaling of the models in production.

4.1 Centralized and Distributed ML Deployments

Having presented some of the high level deployment mechanics of AI/ML models, one can better appreciate the deployment level requirements when the components of the wireless architecture start consuming AI/ML models in a distributed setting. Towards this direction, it becomes important to consider topological and resource-constrained aspects of various components of the wireless networks. Some of these are more central and resource-rich such as the APs - also called BSs, while other elements such as the RIS, passive elements such as metasurface reflectors, or consumer devices such as UEs or IoT sensor devices are distributed and may find themselves at edges of the network while also being resource-constrained. Hence, future wireless networks present a distributed system which can benefit from various development and deployment capabilities seen in the broader field of Distributed ML, where the models may be developed, updated or deployed at any of the cited entities. In the following, we present a short summary of popular methods, frameworks and paradigms that target or enable specific scenarios how AI/ML models can be trained and deployed for scalable consumption. Afterwards, in the last sub-section of this section, we attempt to relate how some of the opportunities presented in **section 2** may benefit from these methods.

4.2 Deployment Units and Deployment Enabling Paradigms

In B5G/6G networks, AI is expected to be utilized all over the network components from the core down to the terminals (UEs)

and at all communication layers from the PHY (L1) to the application layer (L7). Keeping this comprehensive view in mind, the long term management and sustainability of the AI/ML functionality requires that a unit of deployment must be identified. Harnessing certain well-established practices from the Data Science community, the concept of pipelines emerges as a modular, configurable and reusable implementation unit for ML and its affiliated data processing requirements. The ITU-T standardization body Focus Group FG-ML5G have recognized this potential of ML and extract, transform, load (ETL) pipelines and developed a technical specification codenamed Y.3172, entitled "Architectural framework for ML in future networks including IMT-2020" (On Machine Learning for 5G (FG-ML5G), I.-T. F. G., 2019). This specification provides guidelines on training, deployment and orchestration aspects centered around the notion of ML/ETL pipelines, which are bundled as Cloud Computing containers. Harnessing Cloud Computing techniques, the pipeline containers may be exposed as REST web-services, and deployed at the core or edges of the network. In this way, complex interactions can be realized among distributed components of the network from whom data can be collected widely and frequently, while complex data processing tasks such as model training or updates, can be triggered on infrastructure nodes that possess better resource capacity. The objective of this synergy with Cloud Computing is to deliver the expected QoS for connectivity, adaptation and other low-latency requirements of THz systems. We try to present this concept diagrammatically in **Figure 15**.

As exemplified in **Figure 15**, a pipeline process is a set of operations, arranged as nodes, which ingest data, transform or pre-process it, may trigger training, retraining or update of ML models locally or remotely (by invoking externally deployed pipeline's endpoint). It may also conditionally use predictions from ML models on new data or in the evolving network environment, by running or interfacing with ongoing optimizations. Predictions may be a classification score, regression value, or a decision artefact e.g., a schedule, resource allocation plan or an advice, which is returned to the invoking component to take automated action. Outcomes can also be stored in a data sink for later reuse. Pipeline processes may be invoked on demand or scheduled fashion either offline or online scenarios. Such pipeline-based distributed AI/ML pipelines also integrate disparate B5G network functions that need to cooperate or coordinate with each other to achieve cross-layer decision making.

As recognized and advocated in Y.3172 and related specifications by ITU-T FG-5GML as well the predecessor focus group on autonomous networks (FG-AN), enabling AI/ML in future wireless networks would require synergies with Cloud Computing paradigm to realize effective deployment, adoption and continuous upgrade of AI/Ml models by utilizing compute and storage capabilities of the Cloud. The specifications also highlight the challenges and opportunities by exploiting pipelines for loosely interfacing data from different upstream and downstream layers of the network. This fusion of data can be exploited for instance in UE-to-AP assignment predictions, which usually consume link level properties of the network including location, topology, resource requirements of UEs and resource capacity of APs, to assign a UE to the best AP with whom minimal LoS blockages or interferes are expected. However, with the overlay of interacting pipelines, such connectivity may be established by also considering channel level properties by consuming upstreaming data from lower layers of the network and vice versa. Such interactive constellations to fuse and merge data from different network components and layers holds promising opportunities for AI/ML to tackle many hard problems as shared in **section 3** under more realistic settings.

We now take a brief look at the emerging paradigm of Edge Computing, which is expected to be adopted by the fast growing Internet-of-Things (IoT) industry. IoT would arguably be one of the largest beneficiaries of B5G and THz communications as many industries today face the challenge how best to utilize the enormous amounts of data being produced by a large number of sensors. This data can be used to derive insights from industrial processes (as seen in assembly lines, supply chain logistics and manufacturing plants) to optimize these operations for higher efficiency and cost-effectiveness. Edge computing and its relevance with ML raises various questions. These include challenges related to training models on partial data, establishing a holistic, correct and balanced view of the prediction objectives, deployment of processes at the core and edge of the network, while also realizing complex interactions of these processes - all of which remain a topic of further research and investigations within the field of distributed ML.

4.2.1 Distributed ML

Distributed ML focuses on establishing mechanisms to train ML models in a collaborative fashion, with the objective to harness distributed compute infrastructure. Certain parts of this infrastructure may be composed of small devices that are restrained in their system level and communication resources, while other parts may be resource-rich e.g., Cloud based virtual machines or containers that can be horizontally and vertically scaled.

In (S. Teerapittayanon et al., 2017) some ideas were presented to train a deep learning model over a hierarchy of distributed compute nodes. These include end devices, edge nodes and a cloud node. A DNN is trained and maintained at each layer, whereby the end devices and edge nodes have fewer NN layers, while the network in the cloud has more NN layers. Using this architecture, a multi-sensor multi-camera surveillance application deployed at various end nodes is able to efficiently perform inference with required accuracy. The communication latency between the end device (or edge node) and the cloud is limited by sending aggregate data to the cloud node for inference when the end devices cannot reach a high degree of classification confidence. The presented approach is 20 times faster in comparison to the alternative if all raw data from end (sensor) devices were to be sent to the central cloud node for the purpose of inference. While this approach seems to hold good promise for sensor fusion based applications, its applicability in other IoT-based systems remains questionable because the models at end devices are rather limited NN models and may need to rely on the central NN model more frequently.

Although not limited to the proposed framework in (S. Teerapittayanon et al., 2017), distributed ML requires the adopters to setup and configure additional communication mechanisms. These mechanisms are being addressed in the emerging field of federated learning.

4.2.2 Federated Learning

Federated learning makes use of decentralized infrastructure to train a shared ML model with the collaboration of potentially resource constrained end devices distributed at edges of a network and a centralized node that acts as a core. In federated learning, end devices or edge nodes train and maintain local ML models, using the frequently available batches of data that is available (or produced) by the device. This model's complexity and capability is subject to the resource limitations of the end device. Infrequently, a summary of local model's parameters e.g., weights or coefficients are securely transmitted to the core node which updates a consolidated ML model. Hence, the consolidated model is a result of joint but disparate training conducted at potentially millions of end nodes. This approach retains data and the inference (application of prediction model) only at the end nodes so the communication overhead to transfer all raw data to the core is avoided, while also preserving the privacy and confidentiality of local data.

In (McMahan and Ramage, 2017), Google researchers introduced federated learning for a query suggesting application that is installed on mobile applications. The work presents proposals to train a miniature version of Tensorflow based model on mobile devices that consumes minimum energy and minimally interferes with the user experience. Additionally, several technical challenges to achieve federated learning are highlighted. A fundamental challenge is to update the shared model. In (McMahan et al., 2017), the same team presented their solution in the form of a federated averaging algorithm that combines local SGD based updates of local model with a server that performs model averaging on the shared model. However, to ensure the integrity of the consolidated (shared) model, these updates are governed through a Secure Aggregation protocol that only performs cryptographically-secured model updates after a sufficient number (hundreds or thousands) of end devices share their updates i.e. an aggregate data structure must be assembled first. This helps 1) to prevent local phenomenon such as concept drift which happens when distributions of data start exhibiting gradual or abrupt differences, 2) to tackle corner cases such as anomalous data collection or existence of outliers, which can negatively skew the parameters of the consolidated model and 3) to preserve privacy of the local model parameters which are shared in cryptographically secured format (Bonawitz et al., 2017). The updated shared model is made available to selected end devices, which test the model on locally available data, after which the shared model is updated on all end devices.

The cited solutions are applicable in many other domains as well, e.g., IoT where heterogeneous sensor devices can federate, autonomous driving where a large fleet of vehicles can federate or network industries where multiple sensors or devices communicate e.g., railways, airlines, gas networks and power grids. Arguably however, the future THz networks with dense topologies comprising of potentially very large number of UEs and APs hosted within a short geographical area, may find many use cases to adopt federated learning. To conclude, federated learning can be helpful in many industries and use cases where general purpose privacy-aware pattern recognition has precedence over personalized pattern recognition. However, federated learning has yet to mature in terms of widely available deployment and orchestration tooling that can be applied in a variety of domains and applications.

5 RESEARCH DIRECTIONS

Although ML-empowered THz wireless systems and networks will gradually arrive to our lives and are expected to commercialize with the dawn of 6G era, researchers should start looking at the challenges and solutions that the combination of these technologies would bring. Aspired by this, the following research directions are identified:

- *New KPIs definition*: THz wireless systems and networks are expected to support three new type of services, namely: computation-oriented communications (COC), contextual agile enhanced mobile broadband (CAeC), and event defined ultra-reliable low-latency communication (EDuRLLC) (Boulogeorgos et al., 2018a; Letaief et al., 2019). COC refers to collaboration between a number of smart devices in order to create distributed and edge intelligence. High data rates, which can be provided by THz links, as well as lightweight and green AI approaches are considered the key enablers of COC. COC is expected to flexibly select the operation point in the data rate-latency-reliability space. CAeC comes with the promise of agility and adaptability to the network context, physical environment, and social network context. In other words, it is expected to become the solution to the backhaul congestion due to high-fronthaul resource demands by pre-caching the requested content as near the end-user as possible. Moreover, it is expected to provide mobility management functionalities that is essential for THz wireless systems and networks. Finally, EDuRLLC envisions to be able to support spatially and temporally changes of device densities that execute data rate hungry applications, such as virtual reality. A possible solution to this problem is AI-empowered THz networks that are capable of supporting high data rates and prefetching the requesting content.

In contrast to previous generation services, COC, CAeC, and EDuRLLC come with stricter requirements that should not only quantify the efficiency of either the core and the access network, but also from the end-user to the requested information holder. Additionally, to evaluate and enhance the ML algorithms efficiency, its learning ability should be measured. In other words, the performance and feasibility should not only be quantified by one of the conventional metrics, i.e. latency, data-rate and error rate, but from new ones that also captures ML-related metrics, such as computational complexity and

computational resources needs. This calls for formulating and solving novel optimization problems that target in optimizing multiple metrics, for which the relationship is beyond the existing mathematical models.

- *Novel usage scenarios*: AI-empowered THz systems will not only provide ubiquitous connectivity but also enable sensing solutions with very fine range, Doppler and angular resolutions, as well as localization with cm-level accuracy (Bourdoux et al., 2020). This is due to the fact that THz signal experience high absorption, making them unable to penetrate objects. As a results, there are usually LoS. This leads to a relatively direct relation that connects the propagation paths to the wireless environment. Moreover, in THz wireless systems, pencil-beamforming is employed in order to countermeasure the high-pathloss. Pencil-beamforming requires and enables angle estimation of high-accuracy. Finally, the high data rates in these systems allow fast and reliable information sharing between different sensing devices. Except from THz technologies, in order to realize high-level sensing and localisation from low-level raw data, such as the received signal or the channel coefficient estimation, ML-based predictive models and pattern recognition techniques are required. To further boost the sensing and localization accuracy, while maintaining acceptable levels of computational complexity and latency, the aforementioned models should be combined with physics-based signal propagation models. This observation opens the door to the design and testing of new hybrid ML-models that combine physical-models with data-driven learning approaches for localization and sensing.
- *Cross-layer optimization*: In **Section 3**, a number of different ML-problems for enhancing the performance of individual tasks in THz wireless systems were presented. However, for the wireless network, co-designing and co-optimizing a variation of tasks that take into account parameters ranging from PHY to transport layer is expected to bring unrepresented excellence in terms of overall performance. For example, optimizing user association by taking into account not only their position but also the location of the requested content could provide significant overall latency and power consumption minimization. This observation highlights the need of formulating new cross-layer optimization problems.
- *Novel deployment strategies development*: Two deployment strategies were discussed in **Section 4**, namely distributed

and centralized one. Distributed deployment with incomplete local information may result to inaccurate results, which in turn affect the THz system and network performance. Moreover, distributed ML may cause competition between neighboring-area agents that negatively influence the overall network performance. On the other hand, centralized ML demands information that are periodically collected. This may result in unaffordable signaling and computing overhead; hence, an increased end-to-end delay. These remarks reveal the necessity of quantifying the trade-off between centralized global accuracy and high overhead. Moreover, hybrid deployment strategies, like federated learning, should be investigates, which enables part of the operations to run locally in the end or edge units, and feed their results to centralized units. In such deployments, an open research question concerns the optimal amount of signaling.

6 CONCLUSION

This article reported the applications of THz wireless systems and netwotks in the B5G era as well as their enabling technologies and fundamental challenges that could be formulated as ML problems. These problems were categorized into PHY, MAC and RRM, network as well as transport layer. For each of them, we documented the ML approaches, which had been so far used, emphasizing their principles and limitations. Additionally, useful guidelines that are expected to help the reader to select an appropriate ML algorithm to solve the problem that they investigate were reported. Moreover, ML deployment strategies as well as their enablers were discussed. Finally, we presented research gaps and possible future directions.

AUTHOR CONTRIBUTIONS

A-AAB envisioned the concept of the paper and prepared the initial draft. EY prepared **Section 4** and reviewed the paper. MR, AA, RD, and RK performed internal reviews.

FUNDING

REFERENCES

Ahmed, I., and Khammari, H. (2018). "Joint Machine Learning Based Resource Allocation and Hybrid Beamforming Design for Massive MIMO Systems.," in IEEE Globecom Workshops (*GC Wkshps*) (Abu Dhabi, United Arab Emirates: IEEE). doi:10.1109/glocomw.2018.8644454

Alfakih, T., Hassan, M. M., Gumaei, A., Savaglio, C., and Fortino, G. (2020). Task Offloading and Resource Allocation for mobile Edge Computing by Deep Reinforcement Learning Based on SARSA. *IEEE Access* 8, 54074–54084. doi:10.1109/access.2020.2981434

Ali, Z., Miozzo, M., Giupponi, L., Dini, P., Denic, S., and Vassaki, S. (2020). "Recurrent Neural Networks for Handover Management in Next-Generation Self-Organized Networks," in IEEE 31st Annual International Symposium on

Personal, Indoor and Mobile Radio Communications. London, United Kingdom. IEEE. doi:10.1109/pimrc48278.2020.9217178

Aljumaily, M. S., and Li, H. (2019). "Machine Learning Aided Hybrid Beamforming in Massive-MIMO Millimeter Wave Systems," in IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN). IEEE. doi:10.1109/dyspan.2019.8935814

Alkhateeb, A., Alex, S., Varkey, P., Li, Y., Qu, Q., and Tujkovic, D. (2018a). Deep Learning Coordinated Beamforming for Highly-mobile Millimeter Wave Systems. IEEE Access 6, 37328–37348. doi:10.1109/access.2018.2850226

Alkhateeb, A., Beltagy, I., and Alex, S. (2018b). "Machine Learning for Reliable mmWave Systems: Blockage Prediction and Proactive Handoff," in IEEE Global Conference on Signal and Information Processing (GlobalSIP). Anaheim, CA, USA. IEEE. doi:10.1109/globalsip.2018.8646438

Alla, S., and Adari, S. K. (2019). "Boltzmann Machines," in Beginning Anomaly Detection Using Python-Based Deep Learning (Berkeley, CA: Apress), 179–212. doi:10.1007/978-1-4842-5177-5_5

Amiri, R., and Mehrpouyan, H. (2018). "Self-organizing mm Wave Networks: A Power Allocation Scheme Based on Machine Learning," in 11th Global Symposium on Millimeter Waves (GSMM). Boulder, CO, USA. IEEE. doi:10.1109/gsmm.2018.8439323

Anton-Haro, C., and Mestre, X. (2019). Learning and Data-Driven Beam Selection for mmWave Communications: An Angle of Arrival-Based Approach. IEEE Access 7, 20404–20415. doi:10.1109/access.2019.2895594

Aoudia, F. A., and Hoydis, J. (2019). Model-free Training of End-To-End Communication Systems. IEEE J. Select. Areas Commun. 37, 2503–2516. doi:10.1109/jsac.2019.2933891

Auld, T., Moore, A. W., and Gull, S. F. (2007). Bayesian Neural Networks for Internet Traffic Classification. IEEE Trans. Neural Netw. 18, 223–239. doi:10.1109/TNN.2006.883010

Azari, A., Papapetrou, P., Denic, S., and Peters, G. (2019). "User Traffic Prediction for Proactive Resource Management: Learning-Powered Approaches," in IEEE Global Communications Conference (GLOBECOM). Waikoloa, HI, USA. IEEE. doi:10.1109/globecom38437.2019.9014115

Balasubramanian, M. (2002). The Isomap Algorithm and Topological Stability. Science 295, 7a–7. doi:10.1126/science.295.5552.7a

Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., et al. (2017). "Practical Secure Aggregation for Privacy-Preserving Machine Learning," in Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. New York, NY, USA. Dallas, TX: Association for Computing Machinery, 1175–1191. CCS '17. doi:10.1145/3133956.3133982

Boulogeorgos, A.-A. A. (2016). Interference Mitigation Techniques in Modern Wireless Communication systems Ph.D. Thesis. Thessaloniki, Greece: Aristotle University of Thessaloniki.

Boulogeorgos, A.-A. A., Goudos, S. K., and Alexiou, A. (2018c). "Users Association in Ultra Dense THz Networks," in IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC). Kalamata, Greece. IEEE. doi:10.1109/spawc.2018.8445950

Boulogeorgos, A.-A. A., and Alexiou, A. (2020c). "Antenna Misalignment and Blockage in THz Communications," in Next Generation Wireless Terahertz Communication Networks (CRC Press). chap.

Boulogeorgos, A.-A. A., and Alexiou, A. (2020a). Error Analysis of Mixed THz-RF Wireless Systems. IEEE Commun. Lett. 24, 277–281. doi:10.1109/lcomm.2019.2959337

Boulogeorgos, A.-A. A., and Alexiou, A. (2020b). How Much Do Hardware Imperfections Affect the Performance of Reconfigurable Intelligent Surface-Assisted Systems? IEEE Open J. Commun. Soc. 1, 1185–1195. doi:10.1109/ojcoms.2020.3014331

Boulogeorgos, A.-A. A., Alexiou, A., Kritharidis, D., Katsiotis, A., Ntouni, G., Kokkoniemi, J., et al. (2018a). Wireless Terahertz System Architectures for Networks beyond 5G. TERRANOVA CONSORTIUM. White paper 1.0.

Boulogeorgos, A.-A. A., Alexiou, A., Merkle, T., Schubert, C., Elschner, R., Katsiotis, A., et al. (2018b). Terahertz Technologies to Deliver Optical Network Quality of Experience in Wireless Systems beyond 5G. IEEE Commun. Mag. 56, 144–151. doi:10.1109/mcom.2018.1700890

Boulogeorgos, A.-A. A., and Alexiou, A. (2020d). "Outage Probability Analysis of THz Relaying Systems," in IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications. London, United Kingdom: IEEE. doi:10.1109/pimrc48278.2020.9217121

Boulogeorgos, A.-A., and Alexiou, A. (2019). "Performance Evaluation of the Initial Access Procedure in Wireless THz Systems," in 16th International Symposium on Wireless Communication Systems (ISWCS). Oulu, Finland. IEEE. doi:10.1109/iswcs.2019.8877185

Boulogeorgos, A.-A. A., Papasotiriou, E. N., and Alexiou, A. (2019). Analytical Performance Assessment of THz Wireless Systems. IEEE Access 7, 11436–11453. Dataset. doi:10.1109/access.2019.2892198

Boulogeorgos, A.-A. A., Chatzidiamantis, N., Sandalidis, H. G., Alexiou, A., and Renzo, M. D. (2021b). Cascaded Composite Turbulence and Misalignment: Statistical Characterization and Applications to Reconfigurable Intelligent Surface-Empowered Wireless Systems. Dataset Available at: https://arxiv.org/abs/2106.15082/.

Boulogeorgos, A.-A. A., Papasotiriou, E. N., and Alexiou, A. (2018d). "A Distance and Bandwidth Dependent Adaptive Modulation Scheme for THz Communications," in 19th IEEE International Workshop on Signal Processing Advances in Wireless Communications, Kalamata, Greece. SPAWC. doi:10.1109/spawc.2018.8445864

Boulogeorgos, A.-A. A., Stratidakis, G., Papasotirou, E., Lehtomaki, J., Kokkoniemi, J., Mushtaq, M. S., and et al (2017). D4.2-THz Driven MAC Layer Design and Caching Overlay Method. Report.

Boulogeorgos, A.-A. A., Trevlakis, S. E., Tegos, S. A., Papanikolaou, V. K., and Karagiannidis, G. K. (2021a). Machine Learning in Nano-Scale Biomedical Engineering. IEEE Trans. Mol. Biol. Multi-scale Commun. 7, 10–39. doi:10.1109/TMBMC.2020.3035383

Boulogeorgos, A.-A. A., Trevlakis, S. E., Tegos, S. A., Papanikolaou, V. K., and Karagiannidis, G. K. (2021). Machine Learning in Nano-Scale Biomedical Engineering. IEEE Trans. Mol. Biol. Multi-scale Commun. 7, 10–39. doi:10.1109/tmbmc.2020.3035383

Bourdoux, A., Barreto, A. N., van Liempd, B., de Lima, C., Dardari, D., Belot, D., and et al. (2020). 6G white Paper on Localization and Sensing.

Bu, K., He, Y., Jing, X., and Han, J. (2020). Adversarial Transfer Learning for Deep Learning Based Automatic Modulation Classification. IEEE Signal. Process. Lett. 27, 880–884. doi:10.1109/lsp.2020.2991875

Burghal, D., Abbasi, N. A., and Molisch, A. F. (2019). "A Machine Learning Solution for Beam Tracking in mmWave Systems," in 53rd Asilomar Conference on Signals, Systems, and Computers. Pacific Grove, CA, USA. IEEE. doi:10.1109/ieeeconf44664.2019.9048770

Cao, J., Peng, T., Liu, X., Dong, W., Duan, R., Yuan, Y., et al. (2020). Resource Allocation for Ultradense Networks with Machine-Learning-Based Interference Graph Construction. IEEE Internet Things J. 7, 2137–2151. doi:10.1109/jiot.2019.2959232

Chen, J., Feng, W., Xing, J., Yang, P., Sobelman, G. E., Lin, D., et al. (2020a). Hybrid Beamforming/combining for Millimeter Wave MIMO: A Machine Learning Approach. IEEE Trans. Veh. Technol. 69, 11353–11368. doi:10.1109/tvt.2020.3009746

Chen, M., Wei, X., Gao, Y., Huang, L., Chen, M., and Kang, B. (2020b). "Deep-broad Learning System for Traffic Flow Prediction toward 5g Cellular Wireless Network," in International Wireless Communications and Mobile Computing (IWCMC). Limassol, Cyprus. IEEE. doi:10.1109/iwcmc48107.2020.9148092

Cheng, P., Ma, C., Ding, M., Hu, Y., Lin, Z., Li, Y., et al. (2019). Localized Small Cell Caching: A Machine Learning Approach Based on Rating Data. IEEE Trans. Commun. 67, 1663–1676. doi:10.1109/tcomm.2018.2878231

Chou, P.-Y., Chen, W.-Y., Wang, C.-Y., Hwang, R.-H., and Chen, W.-T. (2020). "Deep Reinforcement Learning for MEC Streaming with Joint User Association and Resource Management," in IEEE International Conference on Communications (ICC). Dublin, Ireland. IEEE. doi:10.1109/icc40277.2020.9149086

Cormen, T. H. D. C., Leiserson, C. E. M., Rivest, R. L. M., and Stein, C. C. U. (2009). Introduction to Algorithms. MIT Press Ltd.

Da Silva, C. R. C. M., Kosloff, J., Chen, C., Lomayev, A., and Cordeiro, C. (2018). "Beamforming Training for IEEE 802.11 Ay Millimeter Wave Systems," in Information Theory and Applications Workshop (ITA) (San Diego, CA, USA: IEEE). doi:10.1109/ita.2018.8503112

Dang, S., Amin, O., Shihada, B., and Alouini, M.-S. (2020). What Should 6G Be? Nat. Electron. 3, 20–29. doi:10.1038/s41928-019-0355-6

Elbir, A. M., and Mishra, K. V. (2019). "Robust Hybrid Beamforming with Quantized Deep Neural Networks," in IEEE 29th International Workshop

on Machine Learning for Signal Processing (MLSP). Pittsburgh, PA, USA. IEEE. doi:10.1109/mlsp.2019.8918866

Elsayed, M., Erol-Kantarci, M., and Yanikomeroglu, H. (2021). Transfer Reinforcement Learning for 5G New Radio mmWave Networks. *IEEE Trans. Wireless Commun.* 20, 2838–2849. doi:10.1109/twc.2020.3044597

Ghadikolaei, H. S., Ghauch, H., Fodor, G., Skoglund, M., and Fischione, C. (2020). A Hybrid Model-Based and Data-Driven Approach to Spectrum Sharing in mmWave Cellular Networks. *IEEE Trans. Cogn. Commun. Netw.* 6, 1269–1282. doi:10.1109/tccn.2020.2981031

Ghasempour, Y., da Silva, C. R. C. M., Cordeiro, C., and Knightly, E. W. (2017). IEEE 802.11ay: Next-Generation 60 GHz Communication for 100 Gb/s Wi-Fi. *IEEE Commun. Mag.* 55, 186–192. doi:10.1109/mcom.2017.1700393

Gholami, R., and Fakhari, N. (2017). "Support Vector Machine: Principles, Parameters, and Applications," in *Handbook of Neural Computation* (Elsevier, 515–535. doi:10.1016/b978-0-12-811318-9.00027-2

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). *Generative Adversarial Networks. arXiv preprint arXiv:1406.2661*.

Graupe, D. (2013). *Principles of Artificial Neural Networks*. World Scientific. doi:10.1142/8868Principles of Artificial Neural Networks

Guo, Y., Wang, Z., Li, M., and Liu, Q. (2019). "Machine Learning Based mmWave Channel Tracking in Vehicular Scenario," in IEEE International Conference on Communications Workshops (ICC Workshops). Shanghai, China. IEEE. doi:10.1109/iccw.2019.8757185

Han, S. I. C.-l. C., Xu, Z., and Rowell, C. (2015). Large-scale Antenna Systems with Hybrid Analog and Digital Beamforming for Millimeter Wave 5G. *IEEE Commun. Mag.* 53, 186–194. doi:10.1109/mcom.2015.7010533

Hassan, N., Hossan, M. T., and Tabassum, H. (2020). "User Association in Coexisting RF and TeraHertz Networks in 6g," in IEEE Canadian Conference on Electrical and Computer Engineering (CCECE). London, ON, Canada. IEEE. doi:10.1109/ccece47787.2020.9255737

Ho, J., and Ermon, S. (2016). "Generative Adversarial Imitation Learning," in *Advances in Neural Information Processing Systems*. Editors D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (Curran Associates, Inc.), Vol. 29.

Huang, H., Yang, Y., Ding, Z., Wang, H., Sari, H., and Adachi, F. (2020a). Deep Learning-Based Sum Data Rate and Energy Efficiency Optimization for MIMO-NOMA Systems. *IEEE Trans. Wireless Commun.* 19, 5373–5388. doi:10.1109/twc.2020.2992786

Huang, S., Ye, Y., and Xiao, M. (2020b). Hybrid Beamforming for Millimeter Wave Multi-User MIMO Systems Using Learning Machine. *IEEE Wireless Commun. Lett.* 9, 1914–1918. doi:10.1109/lwc.2020.3007990

Huang, S., Ye, Y., and Xiao, M. (2021c). *Learning Based Hybrid Beamforming Design for Full-Duplex Millimeter Wave Systems*, 7. IEEE Trans. on Cogn. Commun. Netw., 120–132. doi:10.1109/tccn.2020.3019604Learning-Based Hybrid Beamforming Design for Full-Duplex Millimeter Wave Systems*IEEE Trans. Cogn. Commun. Netw.*

IEEE Standard for High Data Rate Wireless Multi-Media Networks (2017). *IEEE Standard for High Data Rate Wireless Multi-media Networks–Amendment 2: 100 Gb/s Wireless Switched point-to-point Physical Layer*. Dataset. doi:10.1109/ieeestd.2017.8066476

IEEE Standard for Information technology (2009). *IEEE Standard for Information Technology– Local and Metropolitan Area Networks– Specific Requirements– Part 15.3: Amendment 2: Millimeter-Wave-Based Alternative Physical Layer Extension*. Dataset. doi:10.1109/ieeestd.2009.5284444)

Iimori, H., de Abreu, G. T. F., Taghizadeh, O., Stoica, R.-A., Hara, T., and Ishibashi, K. (2020). Stochastic Learning Robust Beamforming for Millimeter-Wave Systems with Path Blockage. *IEEE Wireless Commun. Lett.* 9, 1557–1561. doi:10.1109/lwc.2020.2997366

Iqbal, M. O., Ur Rahman, M. M., Imran, M. A., Alomainy, A., Abbasi, Q. H., and Abbasi, Q. H. (2019). Modulation Mode Detection and Classification for *In Vivo* Nano-Scale Communication Systems Operating in Terahertz Band. *IEEE Trans.on Nanobioscience* 18, 10–17. doi:10.1109/TNB.2018.2882063

Jang, J., and Yang, H. J. (2020). Deep Reinforcement Learning-Based Resource Allocation and Power Control in Small Cells with Limited Information Exchange. *IEEE Trans. Veh. Technol.* 69, 13768–13783. doi:10.1109/tvt.2020.3027013

Jeon, Y.-S., Hong, S.-N., and Lee, N. (2018). Supervised-learning-aided Communication Framework for MIMO Systems with Low-Resolution ADCs. *IEEE Trans. Veh. Technol.* 67, 7299–7313. doi:10.1109/tvt.2018.2832845

Jia, C., Gao, H., Chen, N., and He, Y. (2020). Machine Learning Empowered Beam Management for Intelligent Reflecting Surface Assisted MmWave Networks. *China Commun.* 17, 100–114. doi:10.23919/jcc.2020.10.007

Jiang, J.-R. (2020). "Short Survey on Physical Layer Authentication by Machine-Learning for 5G-Based Internet of Things," in 3rd IEEE International Conference on Knowledge Innovation and Invention (ICKII). Kaohsiung, Taiwan. IEEE, 41–44. doi:10.1109/ICKII50300.2020.9318879

Jiang, W., Feng, G., Qin, S., Yum, T. S. P., and Cao, G. (2019). Multi-agent Reinforcement Learning for Efficient Content Caching in mobile D2d Networks. *IEEE Trans. Wireless Commun.* 18, 1610–1622. doi:10.1109/twc.2019.2894403

Kao, W.-C., Zhan, S.-Q., and Lee, T.-S. (2018). "AI-aided 3-d Beamforming for Millimeter Wave Communications," in International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS). Ishigaki, Okinawa, Japan. IEEE. doi:10.1109/ispacs.2018.8923234

Katla, S., Xiang, L., Zhang, Y., El-Hajjar, M., Mourad, A. A. M., and Hanzo, L. (2020). Deep Learning Assisted Detection for index Modulation Aided mmWave Systems. *IEEE Access* 8, 202738–202754. doi:10.1109/access.2020.3035961

Khan, F. N., Zhong, K., Al-Arashi, W. H., Yu, C., Lu, C., and Lau, A. P. T. (2016). Modulation Format Identification in Coherent Receivers Using Deep Machine Learning. *IEEE Photon. Technol. Lett.* 28, 1886–1889. doi:10.1109/lpt.2016.2574800

Khan, H., Elgabli, A., Samarakoon, S., Bennis, M., and Hong, C. S. (2019). Reinforcement Learning-Based Vehicle-Cell Association Algorithm for Highly mobile Millimeter Wave Communication. *IEEE Trans. Cogn. Commun. Netw.* 5, 1073–1085. doi:10.1109/tccn.2019.2941191

Khan, I., Tao, X., Rahman, G. M. S., Rehman, W. U., and Salam, T. (2020a). Advanced Energy-Efficient Computation Offloading Using Deep Reinforcement Learning in MTC Edge Computing. *IEEE Access* 8, 82867–82875. doi:10.1109/access.2020.2991057

Khan, J., and Jacob, L. (2019). "Learning Based CoMP Clustering for URLLC in Millimeter Wave 5g Networks with Blockages," in IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS). Goa, India. IEEE. doi:10.1109/ants47819.2019.9117984

Khan, L. U., Majeed, U., and Hong, C. S. (2020b). "Federated Learning for Cellular Networks: Joint User Association and Resource Allocation," in 21st Asia-Pacific Network Operations and Management Symposium (APNOMS). Daegu, Korea (South). IEEE. doi:10.23919/apnoms50412.2020.9237045

Kirilin, V., Sundarrajan, A., Gorinsky, S., and Sitaraman, R. K. (2020). RL-cache: Learning-Based Cache Admission for Content Delivery. *IEEE J. Select. Areas Commun.* 38, 2372–2385. doi:10.1109/jsac.2020.3000415

Kiumarsi, B., Vamvoudakis, K. G., Modares, H., and Lewis, F. L. (2018). Optimal and Autonomous Control Using Reinforcement Learning: A Survey. *IEEE Trans. Neural Netw. Learn. Syst.* 29, 2042–2062. doi:10.1109/tnnls.2017.2773458

Koenig, S., Lopez-Diaz, D., Antes, J., Boes, F., Henneberger, R., Leuther, A., et al. (2013). Wireless Sub-thz Communication System with High Data Rate. *Nat. Photon* 7, 977–981. doi:10.1038/nphoton.2013.275

Kokkoniemi, J., Lehtomäki, J., and Juntti, M. (2016). Measurements on Penetration Loss in Terahertz Band. doi:10.1109/EuCAP.2016.7481176

Kruber, F., Wurst, J., and Botsch, M. (2018). "An Unsupervised Random forest Clustering Technique for Automatic Traffic Scenario Categorization," in 21st International Conference on Intelligent Transportation Systems (ITSC). Maui, HI, USA. IEEE. doi:10.1109/itsc.2018.8569682

Kumari, R., SheetanshuSingh, M. K., Jha, R., and Singh, N. K. (2016). "Anomaly Detection in Network Traffic Using K-Mean Clustering," in 3rd International Conference on Recent Advances in Information Technology (RAIT). Dhanbad, India. IEEE. doi:10.1109/rait.2016.7507933

Kwon, D., Kim, J., Mohaisen, D. A., and Lee, W. (2020). Self-adaptive Power Control with Deep Reinforcement Learning for Millimeter-Wave Internet-Of-Vehicles Video Caching. *J. Commun. Netw.* 22, 326–337. doi:10.1109/jcn.2020.000022

Kwon, H. J., Lee, J. H., and Choi, W. (2019). "Machine Learning-Based Beamforming in Two-User MISO Interference Channels," in International

Conference on Artificial Intelligence in Information and Communication (ICAIIC). IEEE. doi:10.1109/icaiic.2019.8669027

Letaief, K. B., Chen, W., Shi, Y., Zhang, J., and Zhang, Y.-J. A. (2019). The Roadmap to 6g: AI Empowered Wireless Networks. *IEEE Commun. Mag.* 57, 84–90. doi:10.1109/mcom.2019.1900271

Li, L., Ren, H., Cheng, Q., Xue, K., Chen, W., Debbah, M., et al. (2020a). Millimeter-wave Networking in the Sky: A Machine Learning and Mean Field Game Approach for Joint Beamforming and Beam-Steering. *IEEE Trans. Wireless Commun.* 19, 6393–6408. doi:10.1109/twc.2020.3003284

Li, M., Liu, G., Li, S., and Wu, Y. (2018). "Radio Classify Generative Adversarial Networks: A Semi-supervised Method for Modulation Recognition," In IEEE 18th International Conference on Communication Technology (ICCT) (Chongqing, China: IEEE). doi:10.1109/icct.2018.8600032

Li, Y., Wang, J., Sun, X., Li, Z., Liu, M., and Gui, G. (2020b). Smoothing-aided Support Vector Machine Based Nonstationary Video Traffic Prediction towards B5g Networks. *IEEE Trans. Veh. Technol.* 69, 7493–7502. doi:10.1109/tvt.2020.2993262

Li, Z., Chen, M., Wang, K., Pan, C., Huang, N., and Hu, Y. (2020c). "Parallel Deep Reinforcement Learning Based Online User Association Optimization in Heterogeneous Networks," in IEEE International Conference on Communications Workshops (ICC Workshops). Dublin, Ireland. IEEE. doi:10.1109/iccworkshops49005.2020.9145209

Li, Z., Chen, Z., Ma, X., and Chen, W. (2020d). IEEE. doi:10.1109/icccworkshops49972.2020.9209937

Lin, C.-H., Lee, Y.-T., Chung, W.-H., Lin, S.-C., and Lee, T.-S. (2020). "Unsupervised ResNet-Inspired Beamforming Design Using Deep Unfolding Technique," in IEEE Global Communications Conference (GLOBECOM). Taipei, Taiwan. IEEE. doi:10.1109/globecom42002.2020.9322638

Lipton, Z. C., Berkowitz, J., and Elkan, C. (2015). *A Critical Review of Recurrent Neural Networks for Sequence Learning*. ArXiV.

Liu Bin, L., and Tu Hao, T. (2010). "An Application Traffic Classification Method Based on Semi-supervised Clustering," in 2nd International Symposium on Information Engineering and Electronic Commerce. Ternopil, Ukraine. IEEE. doi:10.1109/ieec.2010.5533239

Liu, R., Lee, M., Yu, G., and Li, G. Y. (2020a). User Association for Millimeter-Wave Networks: A Machine Learning Approach. *IEEE Trans. Commun.* 68, 4162–4174. doi:10.1109/tcomm.2020.2983036

Liu, S., Gao, Z., Zhang, J., Renzo, M. D., and Alouini, M.-S. (2020b). Deep Denoising Neural Network Assisted Compressive Channel Estimation for mmWave Intelligent Reflecting Surfaces. *IEEE Trans. Veh. Technol.* 69, 9223–9228. doi:10.1109/tvt.2020.3005402

Liu, X., Liu, Y., and Chen, Y. (2021c). Machine Learning Empowered Trajectory and Passive Beamforming Design in UAV-RIS Wireless Networks. *IEEE J. Select. Areas Commun.* 39, 2042–2055. doi:10.1109/jsac.2020.3041401

Liu, Y., Bi, S., Shi, Z., and Hanzo, L. (2020d). When Machine Learning Meets Big Data: A Wireless Communication Perspective. *IEEE Veh. Technol. Mag.* 15, 63–72. doi:10.1109/mvt.2019.2953857

Liu, Y., Li, W., and Li, Y. (2007). "Network Traffic Classification Using K-Means Clustering," in Second International Multi-Symposiums on Computer and Computational Sciences (IMSCCS 2007). Iowa City, IA, USA. IEEE. doi:10.1109/imsccs.2007.52

Lizarraga, E. M., Maggio, G. N., and Dowhuszko, A. A. (2019). "Hybrid Beamforming Algorithm Using Reinforcement Learning for Millimeter Wave Wireless Systems," in XVIII Workshop on Information Processing and Control (RPIC). Salvador da Bahia, Argentina. IEEE. doi:10.1109/rpic.2019.8882140

Long, Y., Chen, Z., Fang, J., and Tellambura, C. (2018). Data-driven-based Analog Beam Selection for Hybrid Beamforming under Mm-Wave Channels. *IEEE J. Sel. Top. Signal. Process.* 12, 340–352. doi:10.1109/jstsp.2018.2818649

Ma, W., Qi, C., and Li, G. Y. (2020a). Machine Learning for Beam Alignment in Millimeter Wave Massive MIMO. *IEEE Wireless Commun. Lett.* 9, 875–878. doi:10.1109/lwc.2020.2973972

Ma, W., Qi, C., Zhang, Z., and Cheng, J. (2020b). Sparse Channel Estimation and Hybrid Precoding Using Deep Learning for Millimeter Wave Massive MIMO. *IEEE Trans. Commun.* 68, 2838–2849. doi:10.1109/tcomm.2020.2974457

Mahapatra, S. K., Mohapatra, S. K., Behera, S., and Kanoje, L. (2015). An Experimental Analysis of Penetration Loss Around Buildings of an Institution. doi:10.1109/ICGCIoT.2015.7380431

Mai, Z., Chen, Y., and Du, L. (2021). A Novel Blind mmWave Channel Estimation Algorithm Based on ML-ELM. *IEEE Commun. Lett.*, 1–1 10.1109/lcomm 2021, 3049885.

Mao, Q., Hu, F., and Hao, Q. (2018). Deep Learning for Intelligent Wireless Networks: A Comprehensive Survey. *IEEE Commun. Surv. Tutorials* 20, 2595–2621. doi:10.1109/comst.2018.2846401

McMahan, B., Moore, E., Ramage, D., and Hampson, S. (2017). "y Arcas, B. ACommunication-Efficient Learning of Deep Networks from Decentralized Data.,"Proceedings of the 20th International Conference on Artificial Intelligence and Statistics. Fort Lauderdale, FL, USA. Editors A. Singh and J. Zhu (FL: PMLR), 54, 1273–1282. of *Proceedings of Machine Learning Research*.

McMahan, B., and Ramage, D. (2017). *Collaborative Machine Learning without Centralized Training Data*. Mountain View, CA: Google AI.

Meng, F., Chen, P., and Wu, L. (2019). "Power Allocation in Multi-User Cellular Networks with Deep Q Learning Approach," in IEEE International Conference on Communications (ICC). Shanghai, China. IEEE. doi:10.1109/icc.2019.8761431

Mo, J., Ng, B. L., Chang, S., Huang, P., Kulkarni, M. N., Alammouri, A., et al. (2019). Beam Codebook Design for 5g mmWave Terminals. *IEEE Access* 7, 98387–98404. doi:10.1109/access.2019.2930224

Moon, S., Kim, H., and Hwang, I. (2020). Deep Learning-Based Channel Estimation and Tracking for Millimeter-Wave Vehicular Communications. *J. Commun. Netw.* 22, 177–184. doi:10.1109/jcn.2020.000012

Nduwayezu, M., Pham, Q.-V., and Hwang, W.-J. (2020). Online Computation Offloading in NOMA-Based Multi-Access Edge Computing: A Deep Reinforcement Learning Approach. *IEEE Access* 8, 99098–99109. doi:10.1109/access.2020.2997925

Noorbehbahani, F., and Mansoori, S. (2018). "A New Semi-supervised Method for Network Traffic Classification Based on X-Means Clustering and Label Propagation," in 8th International Conference on Computer and Knowledge Engineering (ICCKE). Mashhad, Iran. IEEE. doi:10.1109/iccke.2018.8566608

On Machine Learning for 5G (FG-ML5G), I.-T. F. G.(2019). *Architectural Framework for Machine Learning in Future Networks Including Imt-2020 (y.3172)*. Dataset.

Papasotiriou, E. N., Kokkoniemi, J., Boulogeorgos, A.-A. A., Lehtomaki, J., Alexiou, A., and Juntti, M. (2018). "A New Look to 275 to 400 GHz Band: Channel Model and Performance Evaluation," in IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC). Bologna, Italy. IEEE. doi:10.1109/pimrc.2018.8580934

Peken, T., Adiga, S., Tandon, R., and Bose, T. (2020a). Deep Learning for SVD and Hybrid Beamforming. *IEEE Trans. Wireless Commun.* 19, 6621–6642. doi:10.1109/twc.2020.3004386

Peken, T., Tandon, R., and Bose, T. (2020b). "Unsupervised mmWave Beamforming *via* Autoencoders," in IEEE International Conference on Communications (ICC). IEEE. doi:10.1109/icc40277.2020.9149222

Peng, T., Cao, J., Liu, X., Dong, W., Duan, R., Yuan, Y., et al. (2019). A Data-Driven and Load-Aware Interference Management Approach for Ultra-dense Networks. *IEEE Access* 7, 129514–129528. doi:10.1109/access.2019.2939709

Petrov, V., Eckhardt, J. M., Moltchanov, D., Koucheryavy, Y., and Kurner, T. (2020). "Measurements of Reflection and Penetration Losses in Low Terahertz Band Vehicular Communications," in 14th European Conference on Antennas and Propagation. Copenhagen, Denmark: EuCAP, 1–5. doi:10.23919/EuCAP48036.2020.9135389

Qi, W., Zhang, B., Chen, B., and Zhang, J. (2018). "A User-Based K-Means Clustering Offloading Algorithm for Heterogeneous Network," in IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC). Las Vegas, NV, USA. IEEE. doi:10.1109/ccwc.2018.8301769

Qiu, C., Zhang, Y., Feng, Z., Zhang, P., and Cui, S. (2018). Spatio-temporal Wireless Traffic Prediction with Recurrent Neural Network. *IEEE Wireless Commun. Lett.* 7, 554–557. doi:10.1109/lwc.2018.2795605

Saad, W., Bennis, M., and Chen, M. (2020). A Vision of 6g Wireless Systems: Applications, Trends, Technologies, and Open Research Problems. *IEEE Netw.* 34, 134–142. doi:10.1109/MNET.001.1900287

Samuel, N., Diskin, T., and Wiesel, A. (2019). Learning to Detect. *IEEE Trans. Signal. Process.* 67, 2554–2564. doi:10.1109/tsp.2019.2899805

Saputra, Y. M., Hoang, D. T., Nguyen, D. N., Dutkiewicz, E., Niyato, D., and Kim, D. I. (2019). Distributed Deep Learning at the Edge: A Novel Proactive and

Cooperative Caching Framework for mobile Edge Networks. *IEEE Wireless Commun. Lett.* 8, 1220–1223. doi:10.1109/lwc.2019.2912365

Sarieddeen, H., Alouini, M.-S., and Al-Naffouri, T. Y. (2021). *An Overview of Signal Processing Techniques for Terahertz Communications*. Dataset.

Satyanarayana, K., El-Hajjar, M., Mourad, A. A. M., and Hanzo, L. (2019). Multi-user Full Duplex Transceiver Design for mmWave Systems Using Learning-Aided Channel Prediction. *IEEE Access* 7, 66068–66083. doi:10.1109/access.2019.2916799

Satyanarayana, K., El-Hajjar, M., Mourad, A. A. M., Pietraski, P., and Hanzo, L. (2020). Soft-decoding for Multi-Set Space-Time Shift-Keying mmWave Systems: A Deep Learning Approach. *IEEE Access* 8, 49584–49595. doi:10.1109/access.2020.2973318

Schenk, T. (2008). *RF Imperfections in High-Rate Wireless Systems*. Netherlands: Springer.:

Schwartz, E. L., Merker, B., Wolfson, E., and Shaw, A. (1988). Applications of Computer Graphics and Image Processing to 2d and 3d Modeling of the Functional Architecture of Visual Cortex. *IEEE Comput. Grap. Appl.* 8, 13–23. doi:10.1109/38.7745

Senevirathna, T., Thennakoon, B., Sankalpa, T., Seneviratne, C., Ali, S., and Rajatheva, N. (2020). "Event-driven Source Traffic Prediction in Machine-type Communications Using LSTM Networks," in IEEE Global Communications Conference (GLOBECOM). Taipei, Taiwan. IEEE. doi:10.1109/globecom42002.2020.9322417

Shah, S. I. H., Alam, S., Ghauri, S. A., Hussain, A., and Ahmed Ansari, F. (2019). A Novel Hybrid Cuckoo Search- Extreme Learning Machine Approach for Modulation Classification. *IEEE Access* 7, 90525–90537. doi:10.1109/access.2019.2926615

Singh, H. (2015). "Performance Analysis of Unsupervised Machine Learning Techniques for Network Traffic Classification," in Fifth International Conference on Advanced Computing & Communication Technologies. IEEE. doi:10.1109/acct.2015.54

Singh, S., Jaakkola, T., Littman, M. L., and Szepesvári, C. (2000). *Machine Learn.* 38, 287–308. doi:10.1023/a:1007678930559

Stratidakis, G., Boulogeorgos, A.-A. A. A., and Alexiou, A. (2019). "A Cooperative Localization-Aided Tracking Algorithm for Thz Wireless Systems," in IEEE Wireless Communications and Networking Conference. Marrakesh, Morocco: WCNC, 1–7. doi:10.1109/WCNC.2019.8885710

Stratidakis, G., Ntouni, G. D., Boulogeorgos, A.-A. A. A., Kritharidis, D., and Alexiou, A. (2020a). "A Low-Overhead Hierarchical Beam-Tracking Algorithm for Thz Wireless Systems," in European Conference on Networks and Communications. Dubrovnik, Croatia: EuCNC, 74–78. doi:10.1109/EuCNC48522.2020.9200946

Stratidakis, G., Papastiriou, E. N., Konstantinis, H., Boulogeorgos, A.-A. A. A., and Alexiou, A. (2020b). "Relay-based Blockage and Antenna Misalignment Mitigation in Thz Wireless Communications," in *2nd 6G Wireless Summit* (6G SUMMIT), 1–4. doi:10.1109/6GSUMMIT49458.2020.9083750

Su, L., Yao, Y., Li, N., Liu, J., Lu, Z., and Liu, B. (2018). "Hierarchical Clustering Based Network Traffic Data Reduction for Improving Suspicious Flow Detection," In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE) (New York, NY, USA: IEEE). doi:10.1109/trustcom/bigdatase.2018.00108

Sun, C., Shi, Z., and Jiang, F. (2020). A Machine Learning Approach for Beamforming in Ultra Dense Network Considering Selfish and Altruistic Strategy. *IEEE Access* 8, 6304–6315. doi:10.1109/access.2019.2963468

Sun, Y., Peng, M., Zhou, Y., Huang, Y., and Mao, S. (2019). Application of Machine Learning in Wireless Networks: Key Techniques and Open Issues. *IEEE Commun. Surv. Tutorials* 21, 3072–3108. doi:10.1109/comst.2019.2924243

Tauqir, H. P., and Habib, A. (2019). "Deep Learning Based Beam Allocation in Switched-Beam Multiuser Massive MIMO Systems," in Second International Conference on Latest trends in Electrical Engineering and Computing Technologies (INTELLECT). Karachi, Pakistan. IEEE. doi:10.1109/intellect47034.2019.8955466

Taylor, M. E., and Stone, P. (2009). Transfer Learning for Reinforcement Learning Domains: A Survey. *J. Mach. Learn. Res.* 10, 1633–1685.

Teerapittayanon, S., McDanel, B., and McDanel, H. T. (2017). "Distributed Deep Neural Networks over the Cloud, the Edge and End Devices," in 37th International Conference on Distributed Computing Systems (ICDCS). IEEE, 328–339. doi:10.1109/ICDCS.2017.226

Tenenbaum, J. B. (2000). A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science* 290, 2319–2323. doi:10.1126/science.290.5500.2319

Van Le, D., and Tham, C.-K. (2018). "A Deep Reinforcement Learning Based Offloading Scheme in Ad-Hoc mobile Clouds," in IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). Honolulu, HI, USA. IEEE. doi:10.1109/infcomw.2018.8406881

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P.-A. (2010). Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *J. Mach. Learn. Res.* 11, 3371–3408.

Wang, C.-C., Yao, X., Wang, W.-L., and Jornet, J. M. (2020a). Multi-hop Deflection Routing Algorithm Based on Reinforcement Learning for Energy-Harvesting Nanonetworks. *IEEE Trans. Mobile Comput.*, 1. doi:10.1109/tmc.2020.3006535

Wang, J., Han, R., Bai, L., Zhang, T., Liu, J., and Choi, J. (2021). Coordinated Beamforming for UAV-Aided Millimeter-Wave Communications Using GPML-Based Channel Estimation. *IEEE Trans. Cogn. Commun. Netw.* 7, 100–109. doi:10.1109/tccn.2020.3048399

Wang, J., Jiang, C., Zhang, H., Ren, Y., Chen, K.-C., and Hanzo, L. (2020b). Thirty Years of Machine Learning: The Road to Pareto-Optimal Wireless Networks. *IEEE Commun. Surv. Tutorials* 22, 1472–1514. doi:10.1109/comst.2020.2965856

Wang, X., Wang, C., Li, X., Leung, V. C. M., and Taleb, T. (2020c). Federated Deep Reinforcement Learning for Internet of Things with Decentralized Cooperative Edge Caching. *IEEE Internet Things J.* 7, 9441–9455. doi:10.1109/jiot.2020.2986803

Wang, Y., Xiang, Y., and Zhang, J. (2013). "Network Traffic Clustering Using Random forest Proximities," in IEEE International Conference on Communications (ICC). Budapest, Hungary. IEEE. doi:10.1109/icc.2013.6654829

Wang, Y., Xiang, Y., Zhang, J., and Yu, S. (2011). "A Novel Semi-supervised Approach for Network Traffic Clustering," in 5th International Conference on Network and System Security. Milan, Italy. IEEE. doi:10.1109/icnss.2011.6059997

Wang, Y., and Yu, S.-Z. (2008). "Machine Learned Real-Time Traffic Classifiers," in Second International Symposium on Intelligent Information Technology Application. Shanghai, China. IEEE. doi:10.1109/iita.2008.536

Watkins, C. J. C. H., and Dayan, P. (1992). Q-learning. *Machine Learn.* 8, 279–292. doi:10.1007/bf0099269810.1023/a:1022676722315

Wu, Y., Li, X., and Fang, J. (2018). "A Deep Learning Approach for Modulation Recognition *via* Exploiting Temporal Correlations," in IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC). Kalamata, GR. IEEE. doi:10.1109/spawc.2018.8445938

Xu, X., Li, D., Dai, Z., Li, S., and Chen, X. (2019). A Heuristic Offloading Method for Deep Learning Edge Services in 5g Networks. *IEEE Access* 7, 67734–67744. doi:10.1109/access.2019.2918585

Yajnanarayana, V., Ryden, H., and Hevizi, L. (2020). "5g Handover Using Reinforcement Learning," in *IEEE 3rd 5G World Forum (5GWF)* (Bangalore, India: IEEE). doi:10.1109/5gwf49715.2020.9221072

Yan, L., Ding, H., Zhang, L., Liu, J., Fang, X., Fang, Y., et al. (2019). Machine Learning-Based Handovers for Sub-6 GHz and mmWave Integrated Vehicular Networks. *IEEE Trans. Wireless Commun.* 18, 4873–4885. doi:10.1109/twc.2019.2930193

Yang, C., He, Z., Peng, Y., Wang, Y., and Yang, J. (2019). Deep Learning Aided Method for Automatic Modulation Recognition. *IEEE Access* 7, 109063–109068. doi:10.1109/access.2019.2933448

Ye, Y., Huang, S., Xiao, M., Ma, Z., and Skoglund, M. (2020). Cache-enabled Millimeter Wave Cellular Networks with Clusters. *IEEE Trans. Commun.* 68, 7732–7745. doi:10.1109/tcomm.2020.3022896

Zeng, Q., Sun, Q., Chen, G., Duan, H., Li, C., and Song, G. (2020). Traffic Prediction of Wireless Cellular Networks Based on Deep Transfer Learning and Cross-Domain Data. *IEEE Access* 8, 172387–172397. doi:10.1109/access.2020.3025210

Zhang, C., Zhang, H., Qiao, J., Yuan, D., and Zhang, M. (2019a). Deep Transfer Learning for Intelligent Cellular Traffic Prediction Based on Cross-Domain Big Data. *IEEE J. Select. Areas Commun.* 37, 1389–1401. doi:10.1109/jsac.2019.2904363

Zhang, C., Zhang, H., Yuan, D., and Zhang, M. (2018). Citywide Cellular Traffic Prediction Based on Densely Connected Convolutional Neural Networks. *IEEE Commun. Lett.* 22, 1656–1659. doi:10.1109/lcomm.2018.2841832

Zhang, H., Zhang, H., Huangfu, W., Liu, W., Dong, J., Long, K., et al. (2019b). "Distributed DNN Based User Association and Resource Optimization in mmWave Networks," in IEEE Global Communications Conference (GLOBECOM). Waikoloa, HI, USA. IEEE. doi:10.1109/globecom38437.2019.9014077

Zhang, H., Zhang, H., Liu, W., Long, K., Dong, J., and Leung, V. C. M. (2020a). Energy Efficient User Clustering, Hybrid Precoding and Power Optimization in Terahertz MIMO-NOMA Systems. *IEEE J. Select. Areas Commun.* 38, 2074–2085. doi:10.1109/jsac.2020.3000888

Zhang, H., Zhang, H., Long, K., and Karagiannidis, G. K. (2020b). Deep Learning Based Radio Resource Management in NOMA Networks: User Association, Subchannel and Power Allocation. *IEEE Trans. Netw. Sci. Eng.* 7, 2406–2415. doi:10.1109/tnse.2020.3004333

Zhang, L., Liang, Y.-C., and Niyato, D. (2019c). 6G Visions: Mobile Ultra-broadband, Super Internet-Of-Things, and Artificial Intelligence. *China Commun.* 16, 1–14. doi:10.23919/jcc.2019.08.001

Zhu, F., Liu, A., and Lau, V. K. N. (2019). "Channel Estimation and Localization for mmWave Systems: A Sparse Bayesian Learning Approach," in IEEE International Conference on Communications (ICC). Shanghai, China. IEEE. doi:10.1109/icc.2019.8761825

Zhu, R., Wang, Y. E., Xu, Q., Liu, Y., and Li, Y. D. (2018). "Millimeter-wave to Microwave Mimo Relays (M4r) for 5g Building Penetration Communications,"

in IEEE Radio and Wireless Symposium. Anaheim, CA: RWS, 206–208. doi:10.1109/RWS.2018.8304988