

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/338594198>

A survey on Machine Learning-based Performance Improvement of Wireless Networks: PHY, MAC and network layer

Preprint · January 2020

CITATIONS

0

READS

10,811

4 authors:



Merima Kulic

Ghent University

14 PUBLICATIONS 353 CITATIONS

[SEE PROFILE](#)



Tarik Kazaz

Delft University of Technology

29 PUBLICATIONS 333 CITATIONS

[SEE PROFILE](#)



Ingrid Moerman

imec - Ghent University

676 PUBLICATIONS 12,915 CITATIONS

[SEE PROFILE](#)



Eli De Poorter

Ghent University

179 PUBLICATIONS 2,947 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



H2020 ORCA [View project](#)



SuperGPS [View project](#)

Article

A Survey on Machine Learning-Based Performance Improvement of Wireless Networks: PHY, MAC and Network Layer

Merima Kulin ^{1,*}, Tarik Kazaz ², Eli De Poorter ¹  and Ingrid Moerman ¹

¹ IDLab, Department of Information Technology, Ghent University-imec, Technologiepark-Zwijnaarde 126, B-9052 Gent, Belgium; Eli.DePoorter@ugent.be (E.D.P.); Ingrid.Moerman@ugent.be (I.M.)

² Faculty of EEMCS, Delft University of Technology, 2628 CD Delft, The Netherlands; t.kazaz@tudelft.nl

* Correspondence: merima.kulin@ugent.be

Abstract: This paper presents a systematic and comprehensive survey that reviews the latest research efforts focused on machine learning (ML) based performance improvement of wireless networks, while considering all layers of the protocol stack: PHY, MAC and network. First, the related work and paper contributions are discussed, followed by providing the necessary background on data-driven approaches and machine learning to help non-machine learning experts understand all discussed techniques. Then, a comprehensive review is presented on works employing ML-based approaches to optimize the wireless communication parameters settings to achieve improved network quality-of-service (QoS) and quality-of-experience (QoE). We first categorize these works into: radio analysis, MAC analysis and network prediction approaches, followed by subcategories within each. Finally, open challenges and broader perspectives are discussed.



Citation: Kulin, M.; Kazaz, T.; De Poorter, E.; Moerman, I. A Survey on Machine Learning-Based Performance Improvement of Wireless Networks: PHY, MAC and Network Layer. *Electronics* **2021**, *10*, 318. <https://doi.org/10.3390/electronics10030318>

Academic Editor: Woongsup Lee
Received: 30 December 2020
Accepted: 22 January 2021
Published: 29 January 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Science and the way we undertake research is rapidly changing. The increase of data generation is present in all scientific disciplines [1], such as computer vision, speech recognition, finance (risk analytics), marketing and sales (e.g., customer churn analysis), pharmacy (e.g., drug discovery), personalized health-care (e.g., biomarker identification in cancer research), precision agriculture (e.g., crop lines detection, weeds detection...), politics (e.g., election campaigning), etc. Until the recent years, this trend has been less pronounced in the wireless networking domain, mainly due to the lack of 'big data' and sufficient communication capacity [2]. However, with the era of the Fifth Generation (5G) cellular systems and the Internet-of-Things (IoT), the big data deluge in the wireless networking domain is under way. For instance, massive amounts of data are generated by the omnipresent sensors used in smart cities [3,4] (e.g., to monitor parking spaces availability in the cities, or monitor the conditions of road traffic to manage and control traffic flows), smart infrastructures (e.g., to monitor the condition of railways or bridges), precision farming [5,6] (e.g., monitor yield status, soil temperature and humidity), environmental monitoring (e.g., pollution, temperature, precipitation sensing), IoT smart grid networks [7] (e.g., to monitor distribution grids or track energy consumption for demand forecasting), etc. It is expected that 28.5 billion devices will be connected to the Internet by 2022 [8], which will create a huge global network of "things" and the demand for wireless resources will accordingly increase in an unprecedented way. On the other hand, the set of available communication technologies is expanding (e.g., the release of the new IEEE 802.11 standards such as IEEE 802.11ax and IEEE 802.11ay; and 5G technologies), which compete for the same finite and limited radio spectrum resources pressuring the need for enhancing

their coexistence and more effective use of the scarce spectrum resources. Similarly, on the mobile systems landscape, mobile data usage is tremendously increasing; according to the latest Ericsson's mobility report there are now 5.9 billion mobile broadband subscriptions globally, generating more than 25 exabytes per month of wireless data traffic [9], a growth close to 88% between Q4 2017 and Q4 2018!

So, big data today is a reality!

However, wireless networks and the generated traffic patterns are becoming more and more complex and challenging to understand. For instance, wireless networks yield many network *performance indicators* (e.g., signal-to-noise ratio (SNR), link access success/collision rate, packet loss rate, bit error rate (BER), latency, link quality indicator, throughput, energy consumption, etc.) and *operating parameters* at different layers of the network protocol stack (e.g., at the PHY layer: frequency channel, modulation scheme, transmitter power; at the MAC layer: MAC protocol selection, and parameters of specific MAC protocols such as CSMA: contention window size, maximum number of backoffs, backoff exponent; TSCH: channel hopping sequence, etc.) having significant impact on the communication performance.

Tuning of these operating parameters and achieving cross-layer optimization to maximize the end-to-end performance is a challenging task. This is especially complex due to the huge traffic demands and heterogeneity of deployed wireless technologies. To address these challenges, machine learning (ML) is increasingly used to develop advanced approaches that can autonomously extract *patterns* and *predict* trends (e.g., at the PHY layer: interference recognition, at the MAC layer: link quality prediction, at the network layer: traffic demand estimation) based on environmental measurements and performance indicators as input. Such patterns can be used to optimize the parameter settings at different protocol layers, e.g., PHY, MAC or network layer.

For instance, consider Figure 1, which illustrates an architecture with heterogeneous wireless access technologies, capable of *collecting* large amounts of observations from the wireless devices, *processing* them and feeding into ML algorithms which generate patterns that can help making better decisions to optimize the operating parameters and improve the network quality-of-service (QoS) and quality-of-experience (QoE).

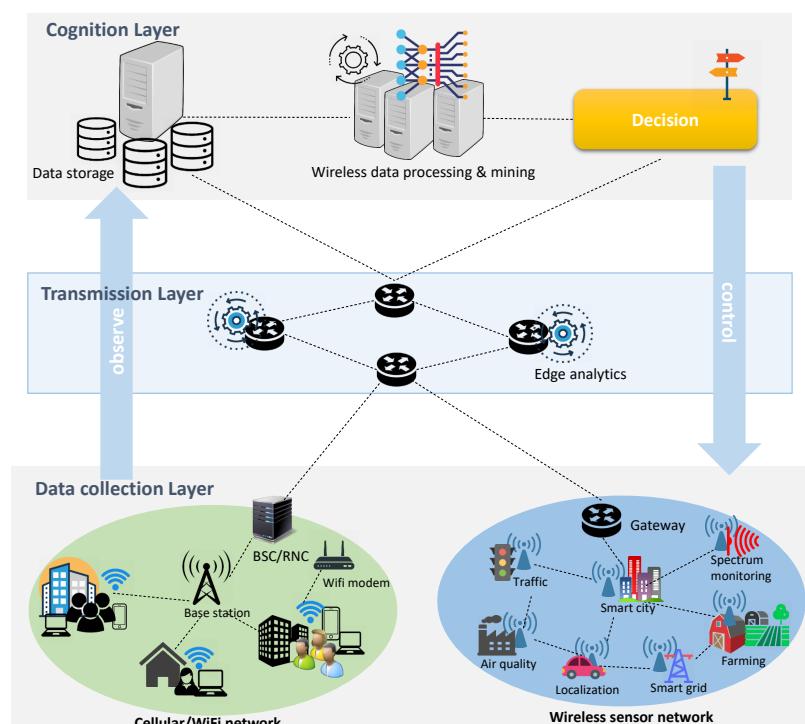


Figure 1. Architecture for wireless big data analysis.

Obviously, there is an urgent need for the development of novel intelligent solutions to improve the wireless networking performance. This has motivated this paper to structure the emerging interdisciplinary research area spanning: wireless networks and communications, machine learning, statistics, experimental-driven research and other research disciplines, to make it more approachable for the wireless networking community and empower wireless networking researchers to create their own predictive models. Furthermore, it aims to inspire researchers by showcasing the state-of-the-art employing ML to improve the performance of wireless networks, demonstrate novel ML-based solutions and discuss current research challenges and future research directives.

Although several survey papers exist, most of them focus on ML in a specific domain or network layer. To the best of our knowledge, this is the first survey that comprehensively reviews the latest research efforts focused on ML-based performance improvements of wireless networks while considering all layers of the protocol stack (PHY, MAC and network), whilst also providing the necessary tutorial for non-machine learning experts to understand all discussed techniques.

Paper organization: We structure this paper as shown on Figure 2.

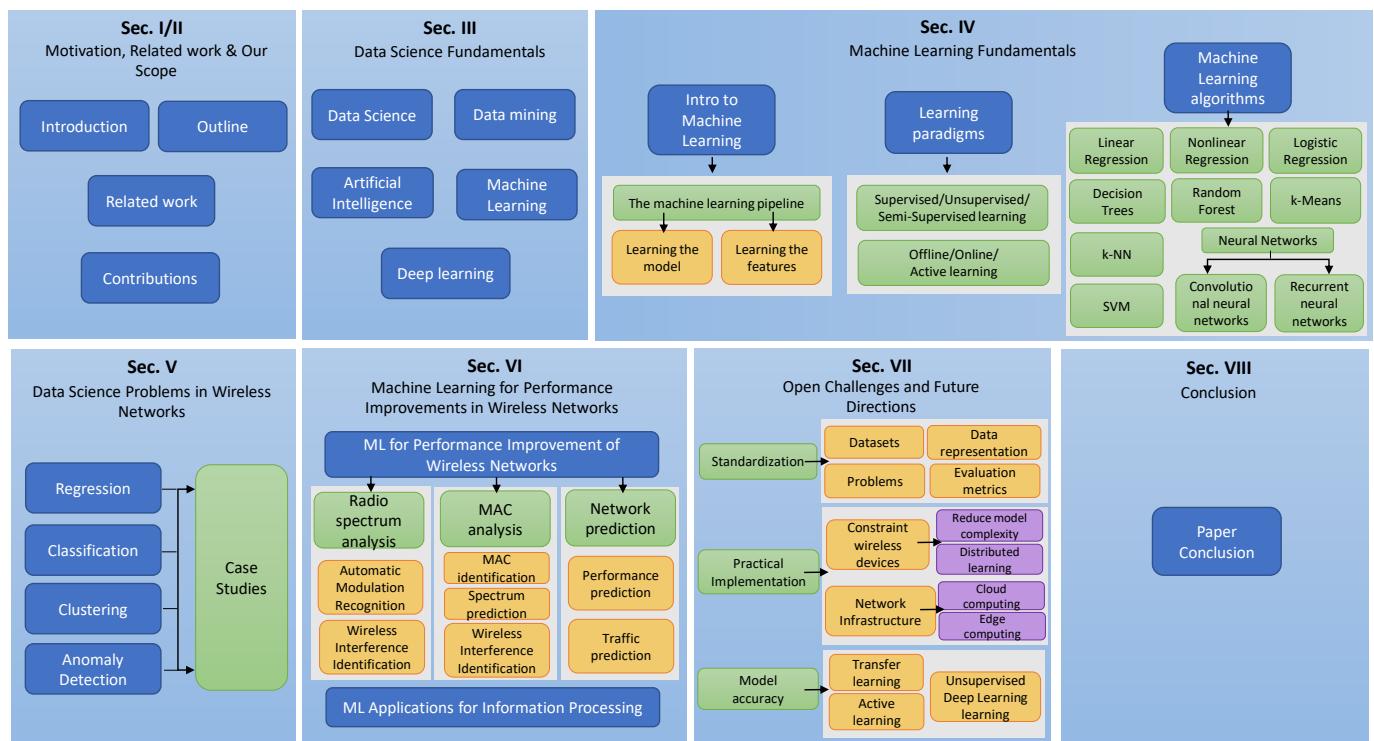


Figure 2. Paper outline.

We start with discussing the related work and distinguishing our work with the state-of-the-art, in Section 2. We conclude that section with a list of our contributions. In Section 3, we present a high-level introduction to data science, data mining, artificial intelligence, machine learning and deep learning. The main goal here is to define these interchangeably used terms and how they relate to each other. In Section 4 we provide a tutorial focused on machine learning, we overview various types of learning paradigms and introduce a couple of popular machine learning algorithms. Section 5 introduces four common types of data-driven problems in the context of wireless networks and provides examples of several case studies. The objective of this section is to help the reader formulate a wireless networking problem into a data-driven problem suitable for machine learning. Section 6 discusses the latest state-of-the-art about machine learning for performance improvements of wireless networks. First, we categorize these works into: radio analysis, MAC analysis and network prediction approaches; then we discuss

example works within each category and give an overview in tabular form, looking at various aspects including: input data, learning approach and algorithm, type of wireless network, achieved performance improvement, etc. In Section 7, we discuss open challenges and present future directions for each. Section 8 concludes the paper.

2. Related Work and Our Contributions

2.1. Related Work

With the advances in hardware and computing power and the ability to collect, store and process massive amounts of data, machine learning (ML) has found its way into many different scientific fields. The challenges faced by current 5G and future wireless networks pushed also the wireless networking domain to seek innovative solutions to ensure expected network performance. To address these challenges, ML is increasingly used in wireless networks. In parallel, a growing number of surveys and tutorials are emerging on ML for future wireless networks. Table 1 provides an overview and comparison with the existing survey papers (note that +– stands for partially available). For instance:

Table 1. Overview of the related work.

| Paper | Tutorial on ML | Wireless Network | Application Area | ML Paradigms | Year |
|-------|----------------|---|--|---|------|
| [10] | ✓ | CRN | Decision-making and feature classification in CRN | Supervised, unsupervised and reinforcement learning | 2012 |
| [11] | ✓ | Localization, security, event detection, routing, data aggregation, MAC | WSN | Supervised, unsupervised and reinforcement learning | 2014 |
| [12] | +– | HetNets | Self-configuration, self-healing, and self-optimization | AI-based techniques | 2015 |
| [13] | +– | CRN, WSN, Cellular and Mobile ad-hoc networks | Security, localization, routing, load balancing | NN | 2016 |
| [14] | | IoT | Big data analytics, event detection, data aggregation, etc. | Supervised, unsupervised and reinforcement learning | 2016 |
| [15] | ✓ | Cellular networks | Self-configuration, self-healing, and self-optimization | Supervised, unsupervised and reinforcement learning | 2017 |
| [16] | +– | CRN | Spectrum sensing and access | Supervised, unsupervised and reinforcement learning | 2018 |
| [17] | +– | IoT, Cellular networks, WSN, CRN | Routing, resource allocation, security, signal detection, application identification, etc. | Deep learning | 2018 |
| [18] | +– | IoT | Big data and stream analytics | Deep learning | 2018 |
| [19] | ✓ | IoT, Mobile networks, CRN, UAV | Communication, virtual reality and edge caching | ANN | 2019 |
| [20] | +– | CRN | Signal Recognition | Deep learning | 2019 |
| [21] | +– | IoT | Smart cities | Supervised, unsupervised and deep learning | 2019 |
| [22] | +– | Communications and networking | Wireless caching, data offloading, network security, traffic routing, resource sharing, etc. | Reinforcement learning | 2019 |
| This | ✓ | IoT, WSN, cellular networks, CRN | Performance improvement of wireless networks | Supervised, unsupervised and Deep learning | 2020 |

- In [10], the authors surveyed existing ML-based methods to address problems in Cognitive Radio Networks (CRNs).
- The authors of [11] survey ML approaches in WSNs (Wireless Sensor Networks) for various applications including location, security, routing, data aggregation and MAC.
- The authors of [12] surveyed the state-of-the-art Artificial Intelligence (AI)-based techniques applied to heterogeneous networks (HetNets) focusing on the research issues of self-configuration, self-healing, and self-optimization.
- ML algorithms and their applications in self organizing cellular networks also focusing on self-configuration, self-healing, and self-optimization, are surveyed in [15].
- In [16] ML applications in CRN are surveyed, that enable spectrum and energy efficient communications in dynamic wireless environments.
- The authors of [19] studied neural networks-based solutions to solve problems in wireless networks such as communication, virtual reality and edge caching.
- In [13], various applications of neural networks (NN) in wireless networks including security, localization, routing, load balancing are surveyed.
- The authors of [14] surveyed ML techniques used in IoT networks for big data analytics, event detection, data aggregation, power control and other applications.
- Paper [17] surveys deep learning applications in wireless networks looking at aspects such as routing, resource allocation, security, signal detection, application identification, etc.
- Paper [18] surveys deep learning applications in IoT networks for big data and stream analytics.
- Paper [20] studies and surveys deep learning applications in cognitive radios for signal recognition tasks.
- The authors of [21] survey ML approaches in the context of IoT smart cities.
- Paper [22] surveys reinforcement learning applications for various applications including network access and rate control, wireless caching, data offloading, network security, traffic routing, resource sharing, etc.

Nevertheless, some of the aforementioned works focus on reviewing specific wireless networking tasks (for example, wireless signal recognition [20]), some focus on the application of specific ML techniques (for instance, deep learning [13,19,20]) while some focus on the aspects of a specific wireless environment looking at broader applications (e.g., CRN [10,16,20], and IoT [14,21]). Furthermore, we noticed that some works miss out the necessary fundamentals for the readers who seek to learn the basics of an area outside their specialty. Finally, no existing work focuses on the literature on how to apply ML techniques to improve wireless network performance looking at possibilities at different layers of the network protocol stack.

To fill this gap, this paper provides a comprehensive introduction to ML for wireless networks and a survey of the latest advances in ML applications for performance improvement to address various challenges future wireless networks are facing. We hope that this paper can help readers develop perspectives on and identify trends of this field and foster more subsequent studies on this topic.

2.2. Contributions

The main contributions of this paper are as follows:

- Introduction for non-machine learning experts to the necessary fundamentals on ML, AI, big data and data science in the context of wireless networks, with numerous examples. It examines when, why and how to use ML.
- A systematic and comprehensive survey on the state-of-the-art that (i) demonstrates the diversity of challenges impacting the wireless networks performance that can be addressed with ML approaches and which (ii) illustrates how ML is applied to improve the performance of wireless networks from various perspectives: PHY, MAC and the network layer.

- References to the latest research works (up to and including 2020) in the field of predictive ML approaches for improving the performance of wireless networks.
- Discussion on open challenges and future directions in the field.

3. Data Science Fundamentals

The objective of this section is to introduce disciplines closely related to data-driven research and machine learning, and how they related to each other. Figure 3 shows a Venn diagram, which illustrates the relation between data science, data mining, artificial intelligence (AI), machine learning and deep learning (DL), explained in more detail in the following subsections. This survey, particularly, focuses on ML/DL approaches in the context of wireless networks.

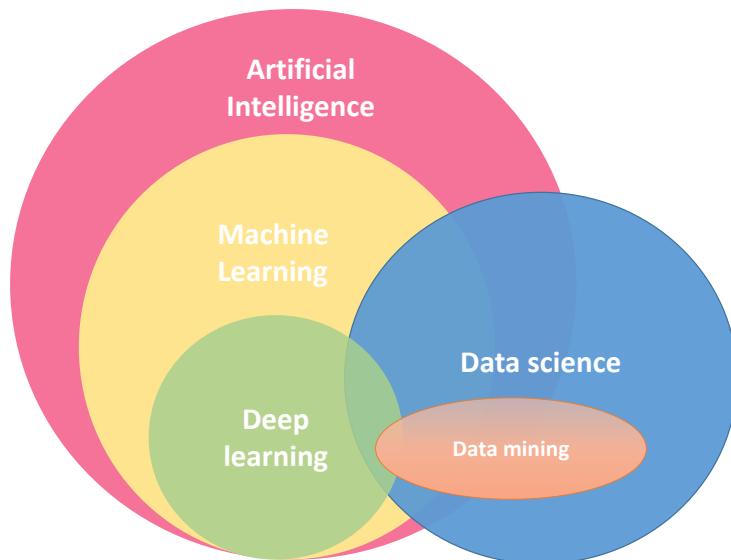


Figure 3. Data science vs. data mining vs. Artificial Intelligence (AI) vs. Machine learning (ML) vs. deep learning.

3.1. Data Science

Data science is the scientific discipline that studies everything related to data, from data acquisition, data storage, data analysis, data cleaning, data visualization, data interpretation, making decisions based on data, determining how to create value from data and how to communicate insights relevant to the business. One definition of the term data science, provided by Dhar [23], is: *Data science is the study of the generalizable extraction of knowledge from data*. Data science makes use of data mining, machine learning, AI techniques and also other approaches such as: heuristics algorithms, operational research, statistics, causal inference, etc. Practitioners of data science are typically skilled in mathematics, statistics, programming, machine learning, big data tools and communicating the results.

3.2. Data Mining

Data mining aims to understand and discover new, previously unseen knowledge in the data. The term mining refers to extracting content by digging. Applying this analogy to data, it may mean to extract insights by digging into data. A simple definition of data mining is: *Data mining refers to the application of algorithms for extracting patterns from data*. Data mining tends to focus on solving actual problems encountered in practice by exploiting algorithms developed by the ML community. For this purpose, a data-driven problem is first translated into a suitable data mining task [24], which will be in detail discussed in Section 5.

3.3. Artificial Intelligence

Artificial intelligence (AI) is concerned with making machines smart aiming to create a system which behaves like a human. This involves fields such as robotics, natural language processing, information retrieval, computer vision and *machine learning*. As coined by [25], AI is: *The science and engineering of making intelligent machines, especially computer systems by reproducing human intelligence through learning, reasoning and self-correction/adaption*. AI uses *intelligent agents* that perceive their environment and take actions that maximize their chance of successfully achieving their goals.

3.4. Machine Learning

Machine learning (ML) is a subset of AI. ML aims to develop algorithms that can learn from historical data and improve the system with experience. In fact, by feeding the algorithms with data it is capable of changing its own internal programming to become better at a certain task. As coined by [26]: *A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E*.

ML experts focus on proving mathematical properties of new algorithms, compared to data mining experts who focus on understanding empirical properties of existing algorithms that they apply. Within the broader picture of data science, ML is the step about taking the cleaned/transformed data and predicting future outcomes. Although ML is not a new field, with the significant increase of available data and the developments in computing and hardware technology ML has become one of the research hotspots in the recent years, in both academia and industry [27].

Compared to traditional signal processing approaches (e.g., estimation and detection), machine learning models are data-driven models; they do not necessarily assume a data model on the underlying physical processes that generated the data. Instead, we may say they “*let the data speak*”, as they are able to infer or *learn* from the data. For instance, when it is complex to model the underlying physics that generated the wireless data, and given that there is sufficient amount of data available that may allow to infer the model that generalizes well beyond what is has seen, ML may outperform traditional signal processing and expert-based systems. However, a representative amount and quality data is required. The advantage of ML is that the resulting models are less prone to the modeling errors of the data generation process.

3.5. Deep Learning

Deep learning is a subset of ML, in which data is passed via multiple number of non-linear transformations to calculate an output. The term *deep* refers to many steps in this case. A definition provided by [28], is: *Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction*. A key advantage of deep learning over traditional ML approaches is that it can automatically extract high-level features from complex data. The learning process does not need to be designed by a human, which tremendously simplifies prior feature handcrafting [28].

However, the performance of DNNs comes at the cost of the model’s interpretability. Namely, DNNs are typically seen as black boxes and there is lack of knowledge why they make certain decisions. Further, DNNs usually suffer from complex hyper-parameters tuning, and finding their optimal configuration can be challenging and time consuming. Furthermore, training deep learning networks can be computationally demanding and require advanced parallel computing such as graphics processing units (GPUs). Hence, when deploying deep learning models on embedded or mobile devices, considered should be the energy and computing constraints of the devices.

There is a growing interest in *deep learning* in the recent years. Figure 4 demonstrates the growing interest in the field, showing the Google search trend from the past few years.

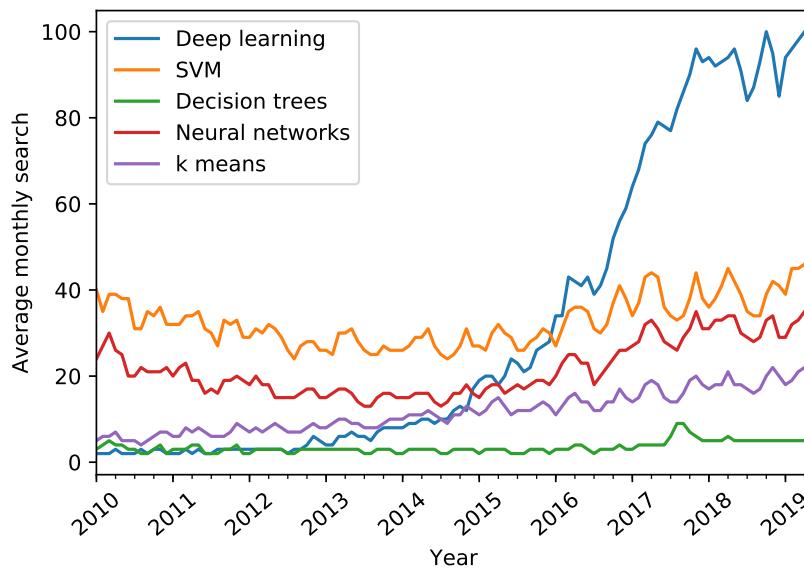


Figure 4. Google search trend showing increased attention in deep learning over the recent years.

4. Machine Learning Fundamentals

Due to their unpredictable nature, wireless networks are an interesting application area for data science because they are influenced by both, natural phenomena and man-made artifacts. This section sets up the necessary fundamentals for the reader to understand the concepts of machine learning.

4.1. The Machine Learning Pipeline

Prior to applying machine learning algorithms to a wireless networking problem, the wireless networking problem needs to be first *translated* into a data science problem. In fact, the whole process from problem to solution may be seen as a machine learning pipeline consisting of several steps.

Figure 5 illustrates those steps, which are briefly explained below:

- **Problem definition.** In this step the problem is identified and translated into a data science problem. This is achieved by formulating the problem as a data mining task. Section 5 further elaborates popular data mining methods such as classification and regression, and presents case studies of wireless networking problems of each type. In this way, we hope to help the reader understand how to formulate a wireless networking problem as a data science problem.
- **Data collection.** In this step, the needed amount of data to solve the formulated problem is identified and collected. The result of this step is *raw data*.
- **Data preparation.** After the problem is formulated and data is collected, the raw data is being preprocessed to be cleaned and transformed into a new space where each data pattern is represented by a vector, $\mathbf{x} \in \mathbb{R}^n$. This is known as the *feature vector*, and its n elements are known as *features*. Through, the process of *feature extraction* each pattern becomes a single point in a n -dimensional space, known as the *feature space* or the *input space*. Typically, one starts with some large value P of features and eventually selects the n most informative ones during the *feature selection* process.
- **Model training.** After defining the feature space in which the data lays, one has to train a machine learning algorithm to obtain a model. This process starts by forming the *training data* or *training set*. Assuming that M feature vectors and corresponding known output values (sometimes called *labels*) are available, the *training set* \mathcal{S} consists of M input-output pairs $((\mathbf{x}_i, y_i), i = 1, \dots, M)$ called *training examples*, that is,

$$\mathcal{S} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_M, y_M)\}, \quad (1)$$

where $\mathbf{x}_i \in \mathbb{R}^n$, is the feature vector of the i th observation,

$$\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{in}]^T, i = 1, \dots, M. \quad (2)$$

The corresponding output values (*labels*) to which $\mathbf{x}_i, i = 1, \dots, M$, belong, are

$$\mathbf{y} = [y_1, y_2, \dots, y_M]^T. \quad (3)$$

In fact, various ML algorithms are trained, tuned (by tuning their hyper-parameters) and the resulting models are evaluated based on standard performance metrics (e.g., mean squared error, precision, recall, accuracy, etc.) and the best performing model is chosen (i.e., model selection).

- **Model deployment.** The selected ML model is deployed into a practical wireless system where it is used to make predictions. For instance, given unknown raw data, first the feature vector \mathbf{x} is formed, and then it is fed into the ML model for making predictions. Furthermore, the deployed model is continuously monitored to observe how it behaves in real world. To make sure it is accurate, it may be retrained.

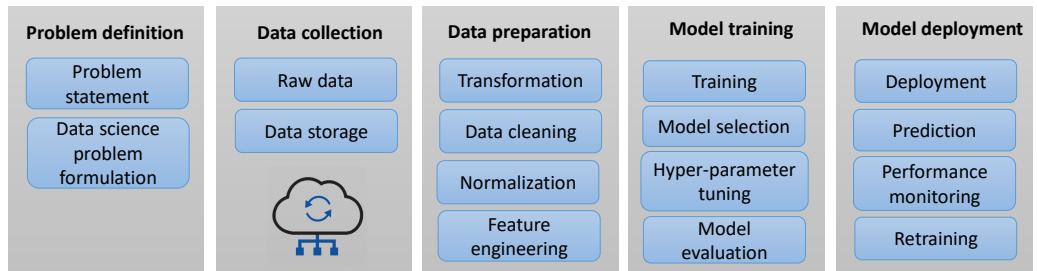


Figure 5. Steps in a machine learning pipeline.

Further below, the ML stage is elaborated in more detail.

Learning the Model

Given a set \mathcal{S} , the goal of a machine learning algorithm is to *learn* the mathematical model for f . Thus, f is some fixed but unknown function, that defines the relation between \mathbf{x} and y , that is

$$f : \mathbf{x} \rightarrow y. \quad (4)$$

The function f is obtained by applying the selected learning method to the training set, \mathcal{S} , so that f is a good estimator for new unseen data, i.e.

$$y \approx \hat{y} = \hat{f}(\mathbf{x}_{new}). \quad (5)$$

In machine learning, f is called the predictor, because its task is to *predict* the outcome y_i based on the input value of \mathbf{x}_i . Two popular predictors are the *regressor* and *classifier*, described by:

$$f(x) = \begin{cases} \text{regressor: if } y \in \mathbb{R} \\ \text{classifier: if } y \in \{0, 1\} \end{cases}. \quad (6)$$

In other words, when the output variable y is *continuous* or quantitative, the learning problem is a *regression* problem. But, if y predicts a discrete or *categorical* value, it is a *classification* problem.

In case, when the predictor f is parameterized by a vector $\Theta \in \mathbb{R}^n$, it describes a *parametric* model. In this setup, the problem of estimating f reduces down to one of estimating the parameters $\Theta = [\theta_1, \theta_2, \dots, \theta_n]^T$. In most practical applications, the observed data are noisy versions of the expected values that would be obtained under ideal circumstances.

These unavoidable errors, prevent the extraction of true parameters from the observations. With this in regard, the generic data model may be expressed as

$$y = f(\mathbf{x}) + \epsilon, \quad (7)$$

where $f(\mathbf{x})$ is the model and ϵ are additive measurement errors and other discrepancies. The goal of ML is to find the input-output relation that will “best” match the noisy observations. Hence, the vector θ may be estimated by solving a (*convex*) *optimization* problem. First, a *loss* or *cost function* $l(\mathbf{x}, \mathbf{y}, \theta)$ is set, which is a (point-wise) measure of the error between the observed data point y_i and the model prediction $\hat{f}(\mathbf{x}_i)$ for each value of θ . However, θ is estimated on the whole training set, S , not just one example. For this task, the average loss over all training examples called *training loss*, J , is calculated:

$$J(\theta) \equiv J(S, \theta) = \frac{1}{m} \sum_{(\mathbf{x}_i, y_i) \in S} l(\mathbf{x}_i, y_i, \theta), \quad (8)$$

where S indicates that the error is calculated on the instances from the training set and $i = 1, \dots, m$. The vector θ that minimizes the training loss $J(\theta)$, that is

$$\underset{\theta \in \mathbb{R}^n}{\operatorname{argmin}} J(\theta), \quad (9)$$

will give the desired model. Once the model is estimated, for any given input \mathbf{x} , the prediction for y can be made with $\hat{y} = \theta^T \mathbf{x}$.

The prediction accuracy of ML models heavily depends on the choice of the data representation or features used for training. For that reason, much effort in designing ML models goes into the composition of pre-processing and data transformation chains that result in a representation of the data that can support effective ML predictions. Informally, this is referred to as *feature engineering*. *Feature engineering* is the process of extracting, combining and manipulating features by taking advantage of human ingenuity and prior expert knowledge to arrive at more representative ones. The feature extractor ϕ transforms the data vector $\mathbf{d} \in \mathbb{R}^d$ into a new form, $\mathbf{x} \in \mathbb{R}^n$, $n \leq d$, more suitable for making predictions, that is

$$\phi(\mathbf{d}) : \mathbf{d} \rightarrow \mathbf{x}. \quad (10)$$

For instance, the authors of [29] engineered features from the RSSI (Received Signal Strength Indication) distribution to identify wireless signals. The importance of feature engineering highlights the bottleneck of ML algorithms: their inability to automatically extract the discriminative information from data. *Feature learning* is a branch of machine learning that moves the concept of learning from “learning the model” to “learning the features”. One popular feature learning method is *deep learning*, in detail discussed in Section 4.3.9.

4.2. Types of Learning Paradigms

This section discussed various types of learning paradigms in ML, summarized in Figure 6.

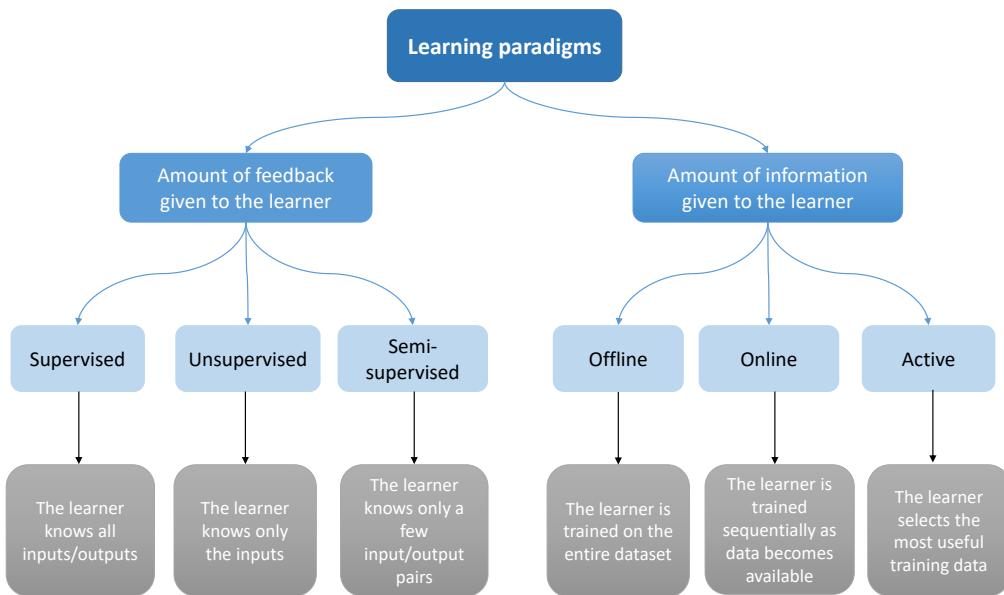


Figure 6. Summary of types of learning paradigms.

4.2.1. Supervised vs. Unsupervised vs. Semi-Supervised Learning

Learning can be categorized by the amount of knowledge or feedback that is given to the learner as either supervised or unsupervised.

Supervised Learning

Supervised learning utilizes predefined inputs and known outputs to build a system model. The set of inputs and outputs forms the labeled training dataset that is used to teach a learning algorithm how to predict future outputs for new inputs that were not part of the training set. Supervised learning algorithms are suitable for wireless network problems where prior knowledge about the environment exists and data can be labeled. For example, predict the location of a mobile node using an algorithm that is trained on signal propagation characteristics (inputs) at known locations (outputs). Various challenges in wireless networks have been addressed using supervised learning such as: medium access control [30–33], routing [34], link quality estimation [35,36], node clustering in WSN [37], localization [38–40], adding reasoning capabilities for cognitive radios [41–47], etc. Supervised learning has also been extensively applied to different types of wireless networks application such as: human activity recognition [48–53], event detection [54–58], electricity load monitoring [59,60], security [61–63], etc. Some of these works will be analyzed in more detail later.

Unsupervised Learning

Unsupervised learning algorithms try to find hidden structures in unlabeled data. The learner is provided only with inputs without known outputs, while learning is performed by finding similarities in the input data. As such, these algorithms are suitable for wireless network problems where no prior knowledge about the outcomes exists, or annotating data (labelling) is difficult to realize in practice. For instance, automatic grouping of wireless sensor nodes into clusters based on their current sensed data values and geographical proximity (without knowing a priori the group membership of each node) can be solved using unsupervised learning. In the context of wireless networks, unsupervised learning algorithms are widely used for: data aggregation [64], node clustering for WSNs [64–67], data clustering [68–70], event detection [71] and several cognitive radio applications [72,73], dimensionality reduction [74], etc.

Semi-Supervised Learning

Several mixes between the two learning methods exist and materialize into semi-supervised learning [75]. Semi-supervised learning is used in situations when a small amount of labeled data with a large amount of unlabeled data exists. It has great practical value because it may alleviate the cost of rendering a fully labeled training set, especially in situations where it is infeasible to label all instances. For instance, in human activity recognition systems where the activities change very fast so that some activities stay unlabeled or the user is not willing to cooperate in the data collection process, supervised learning might be the best candidate to train a recognition model [76–78]. Other potential use cases in wireless networks might be localization systems where it can alleviate the tedious and time-consuming process of collecting training data (calibration) in fingerprinting-based solutions [79] or semi-supervised traffic classification [80], etc.

4.2.2. Offline vs. Online vs. Active Learning

Learning can be categorized depending on the way the information is given to the learner as either offline or online learning. In offline learning the learner is trained on the entire training data at once, while in online learning the training data becomes available in a sequential order and is used to update the representation of the learner in each iteration.

Offline Learning

Offline learning is used when the system that is being modeled does not change its properties dynamically. Offline learned models are easy to implement because the models do not have to keep on learning constantly, and they can be easily retrained and redeployed in production. For example, in [81] a learning-based link quality estimator is implemented by deploying an offline trained model into the network stack of Tmote Sky wireless nodes. The model is trained based on measurements about the current status of the wireless channel that are obtained from extensive experiment setups from a wireless testbed.

Another use cases are human activity recognition systems, where an offline trained classifier is deployed to recognize actions from users. The classifier model can be trained based on information extracted from raw measurements collected by sensors integrated in a smartphone, which is at the same time the central processing unit that implements the offline learned model for online activity recognition [82].

Online Learning

Online learning is useful for problems where training examples arrive one at a time or when due to limited resources it is computationally infeasible to train over the entire dataset. For instance, in [83] a decentralized learning approach for anomaly detection in wireless sensor networks is proposed. The authors concentrate on detection methods that can be applied online (i.e., without the need of an offline learning phase) and that are characterized by a limited computational footprint, so as to accommodate the stringent hardware limitations of WSN nodes. Another example can be found in [84], where the authors propose an online outlier detection technique that can sequentially update the model and detect measurements that do not conform to the normal behavioral pattern of the sensed data, while maintaining the resource consumption of the network to a minimum.

Active Learning

A special form of online learning is active learning where the learner first reasons about which examples would be most useful for training (taking as few examples as possible) and then collects those examples. Active learning has proven to be useful in situations when it is expensive to obtain samples from all variables of interest. For instance, the authors in [85] proposed a novel active learning approach (for graphical model selection problems), where the goal is to optimize the total number of scalar samples obtained by allowing the collection of samples from only subsets of the variables. This technique could for instance

alleviate the need for synchronizing a large number of sensors to obtain samples from all the variables involved simultaneously.

Active learning has been a major topic in recent years in ML and an exhaustive literature survey is beyond the scope of this paper. We refer the reader for more details on active learning algorithms to [86–88].

4.3. Machine Learning Algorithms

This section reviews popular ML algorithms used in wireless networks research.

4.3.1. Linear Regression

Linear regression is a supervised learning technique used for modeling the relationship between a set of input (independent) variables (\mathbf{x}) and an output (dependent) variable (y), so that the output is a linear combination of the input variables:

$$y = f(\mathbf{x}) := \theta_0 + \theta_1 x_1 + \dots + \theta_n x_n + \epsilon = \theta_0 + \sum_{j=1}^n \theta_j x_j, \quad (11)$$

where $\mathbf{x} = [x_1, \dots, x_n]^T$, and $\boldsymbol{\theta} = [\theta_0, \theta_1, \dots, \theta_n]^T$ is the estimated parameter vector from a given training set (y_i, \mathbf{x}_i) , $i = 1, 2, \dots, m$.

4.3.2. Nonlinear Regression

Nonlinear regression is a supervised learning technique which models the observed data by a function that is a nonlinear combination of the model parameters and one or more independent input variables. An example of nonlinear regression is the polynomial regression model defined by:

$$y = f(\mathbf{x}) := \theta_0 + \theta_1 x + \theta_2 x^2 + \dots + \theta_n x^n, \quad (12)$$

4.3.3. Logistic Regression

Logistic regression [89] is a simple supervised learning algorithm widely used for implementing linear classification models, meaning that the models define smooth linear decision boundaries between different classes. At the core of the learning algorithm is the logistic function which is used to learn the model parameters and predict future instances. The logistic function, $f(z)$, is given by 1 over 1 plus e to the minus z , that is:

$$f(z) = \frac{1}{1 + e^{-z}}, \quad (13)$$

where, $z := \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n$, where x_1, x_2, \dots, x_n are the independent (input) variables, that we wish to use to describe or predict the dependent (output) variable $y = f(z)$.

The range of $f(z)$ is between 0 and 1, regardless of the value of z , which makes it popular for classification tasks. Namely, the model is designed to describe a probability, which is always some number between 0 and 1.

4.3.4. Decision Trees

Decision trees (DT) [90] is a supervised learning algorithm that creates a tree-like graph or model that represents the possible outcomes or consequences of using certain input values. The tree consists of one root node, internal nodes called decision nodes which test its input against a learned expression, and leaf nodes which correspond to a final class or decision. The learning tree can be used to derive simple decision rules that can be used for decision problems or for classifying future instances by starting at the root node and moving through the tree until a leaf node is reached where a class label is assigned. However, decision trees can achieve high accuracy only if the data is linearly separable,

i.e., if there exists a linear hyperplane between the classes. Hence, constructing an optimal decision tree is NP-complete [91].

There are many algorithms that can form a learning tree such as the simple Iterative Dichotomiser 3 (ID3), its improved version C4.5, etc.

4.3.5. Random Forest

Random forests (RF) are *bagged* decision trees. Bagging is a technique which involves training many classifiers and considering the average output of the ensemble. In this way, the variance of the overall ensemble classifier can be greatly reduced. Bagging is often used with DTs as they are not very robust to errors due to variance in the input data. Random forest are created by the following Algorithm 1:

Algorithm 1: Random Forest.

Input: Training set D

Output: Predicted value $h(x)$

Procedure:

- Sample k datasets D_1, \dots, D_k from D with replacement.
 - For each D_i train a decision tree classifier $h_i()$ to the maximum depth and when splitting the tree only consider a subset of features l . If d is the number of features in each training example, the parameter $l \leq d$ is typically set to $l = \sqrt{d}$.
 - The ensemble classifier is then the mean or majority vote output decision out of all decision trees.
-

Figure 7 illustrates this process.

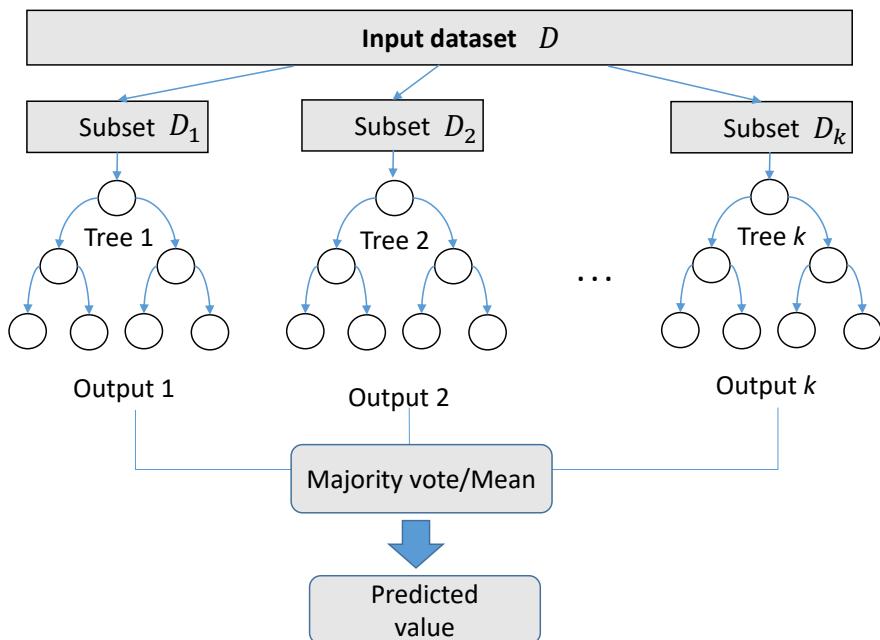


Figure 7. Graphical formulation for Random Forest.

4.3.6. SVM

Support Vector Machine (SVM) [92] is a learning algorithm that solves classification problems by first mapping the input data into a higher-dimensional feature space in which it becomes linearly separable by a hyperplane, which is used for classification. In Support vector regression, this hyperplane is used to predict the continuous value output. The mapping from the input space to the high-dimensional feature space is non-linear, which is achieved using *kernel* functions. Different kernel functions comply best for different

application domains. The most common kernel functions used in SVM are: linear kernel, polynomial kernel and basis kernel function (RBF), given as:

$$k(x_i, x_j) = x_i^T x_j k(x_i, x_j) = (x_i^T x_j + 1)^d k(x_i, x_j) = e^{-\frac{(x_i - x_j)^2}{\sigma^2}} \quad (14)$$

where σ is a user defined parameter.

4.3.7. k-NN

k nearest neighbors (k-NN) [93] is a learning algorithm that can solve classification and regression problems by looking into the distance (closeness) between input instances. It is called a non-parametric learning algorithm because, unlike other supervised learning algorithms, it does not learn an explicit model function from the training data. Instead, the algorithm simply memorizes all previous instances and then predicts the output by first searching the training set for the k closest instances and then: (1) for classification-predicts the majority class amongst those k nearest neighbors, while (2) for regression-predicts the output value as the average of the values of its k nearest neighbors. Because of this approach, k-NN is considered a form of instance-based or memory-based learning.

k-NN is widely used since it is one of the simplest forms of learning. It is also considered as *lazy* learning as the learner is passive until a prediction has to be performed, hence no computation is required until performing the prediction task. The pseudocode for k-NN [94] is summarized in Algorithm 2.

Algorithm 2: k-NN.

```

Input:  $(y_i, x_i)$ : Training set,  $i = 1, 2, \dots, m$ ;  $s$ : unknown sample
Output: Predicted value  $f(\mathbf{x})$ 
Procedure: for  $i \leftarrow 1$  to  $m$  do
    | Compute distance  $d(x_i, s)$ 
end
1. Compute set  $I$  containing indices for the  $k$  smallest distances  $d(x_i, s)$ 
2.  $f(\mathbf{x}) \leftarrow$  majority label/mean value for  $\{y_i \text{ where } i \in I\}$ 
return  $f(\mathbf{x})$ 
```

4.3.8. k-Means

k -Means is an unsupervised learning algorithm used for clustering problems. The goal is to assign a number of points, x_1, \dots, x_m into K groups or clusters, so that the resulting intra-cluster similarity is high, while the inter-cluster similarity low. The similarity is measured with respect to the mean value of the data points in a cluster. Figure 8 illustrates an example of k-means clustering, where $K = 3$ and the input dataset consisting of two features with data points plotted along the x and y axis.

On the left side of Figure 8 are data points before k-means is applied, while on the right side are the identified 3 clusters and their centroids represented with squares.

The pseudocode for k-means [94] is summarized in Algorithm 3.

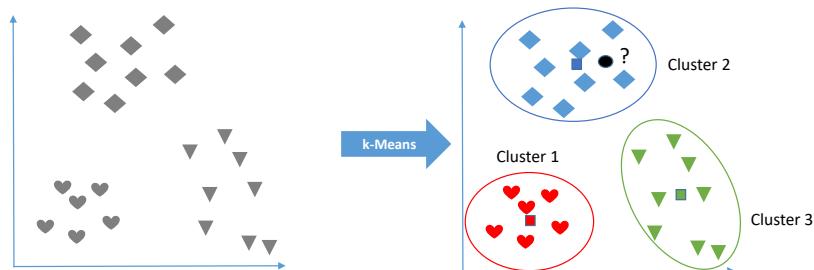


Figure 8. Graphical formulation for k-Means.

Algorithm 3: k-means.

Input: K : The number of desired clusters; $X = \{x_1, x_2, \dots, x_m\}$: Input dataset with m data points

Output: A set of K clusters

Procedure:

1. Set the cluster centroids $\mu_k, k = 1, \dots, K$ to arbitrary values;
2. **while** no change in μ_k **do**
 - (a) (Re)assign each item x_i to the cluster with the closest centroid.
 - (b) Update $\mu_k, k = 1, \dots, K$, as the mean value of the data points in each cluster.
- end**

return K clusters

4.3.9. Neural Networks

Neural Networks (NN) [95] or artificial neural networks (ANN) is a supervised learning algorithm inspired on the working of the brain, that is typically used to derive complex, non-linear decision boundaries for building a classification model, but are also suitable for training regression models when the goal is to predict real-valued outputs (regression problems are explained in Section 5.1). Neural networks are known for their ability to identify complex trends and detect complex non-linear relationships among the input variables at the cost of higher computational burden. A neural network model consists of one input, a number of hidden layers and one output layer, as shown on Figure 9.

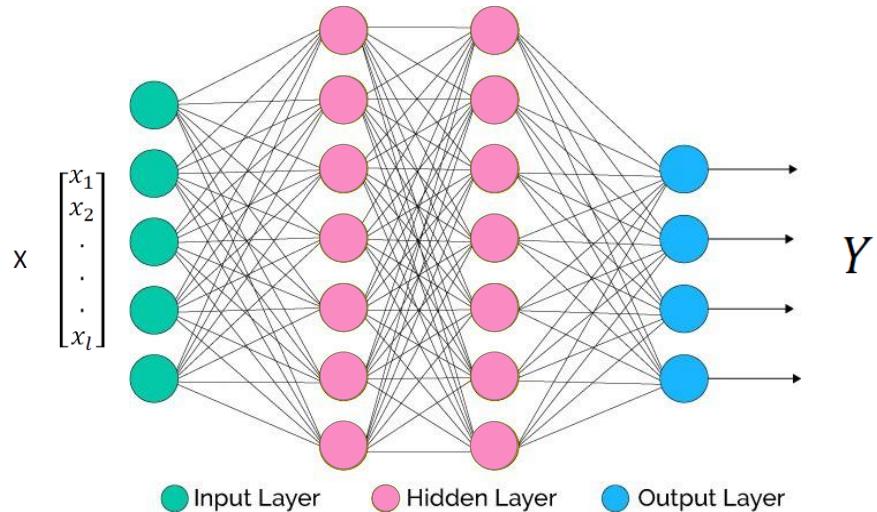


Figure 9. Graphical formulation for Neural networks.

The formulation for a single layer is as follow:

$$y = g(\mathbf{w}^T \mathbf{x} + b), \quad (15)$$

where \mathbf{x} is a training example input, and y is the layer output, \mathbf{w} are the layer weights, while b is the bias term.

The input layer corresponds to the input data variables. Each hidden layer consists of a number of processing elements called neurons that process its inputs (the data from the previous layer) using an activation or transfer function that translates the input signals to an output signal, $g()$. Commonly used activation functions are: unit step function, linear function, sigmoid function and the hyperbolic tangent function. The elements between

each layer are highly connected by connections that have numeric weights that are learned by the algorithm. The output layer outputs the prediction (i.e., the class) for the given inputs and according to the interconnection weights defined through the hidden layer. The algorithm is again gaining popularity in recent years because of new techniques and more powerful hardware that enable training complex models for solving complex tasks. In general, neural networks are said to be able to approximate any function of interest when tuned well, which is why they are considered as universal approximators [96].

Deep Neural Networks

Deep neural networks are a special type of NNs consisting of multiple layers able to perform feature transformation and extraction. Opposed to a traditional NN, they have the potential to alleviate manually extracting features, which is a process that depends much on prior knowledge and domain expertise [97].

Various deep learning techniques exist, including: deep neural networks (DNN), convolutional neural networks (CNN), recurrent neural networks (RNN) and deep belief networks (DBN), which have shown success in various fields of science including computer vision, automatic speech recognition, natural language processing, bioinformatics, etc., and increasingly also in wireless networks.

Convolutional Neural Networks

Convolutional neural networks (CNN) perform feature learning via non-linear transformations implemented as a series of nested *layers*. The input data is a multidimensional data array, called *tensor*, that is presented at the *visible layer*. This is typically a grid-like topological structure, e.g., time-series data, which can be seen as a 1D grid taking samples at regular time intervals, pixels in images with a 2D layout, a 3D structure of videos, etc. Then a series of *hidden layers* extract several abstract features. Hidden layers consist of a series of convolution, pooling and fully-connected layers, as shown on Figure 10.

Those layers are “hidden” because their values are not given. Instead, the deep learning model must determine which data representations are useful for explaining the relationships in the observed data. Each convolution layer consists of several *kernels* (i.e., filters) that perform a *convolution* over the input; therefore, they are also referred to as *convolutional layers*. Kernels are feature detectors, that convolve over the input and produce a transformed version of the data at the output. Those are banks of finite impulse response *filters* as seen in signal processing, just learned on a hierarchy of layers. The filters are usually multidimensional arrays of parameters that are learnt by the learning algorithm [98] through a training process called *backpropagation*.

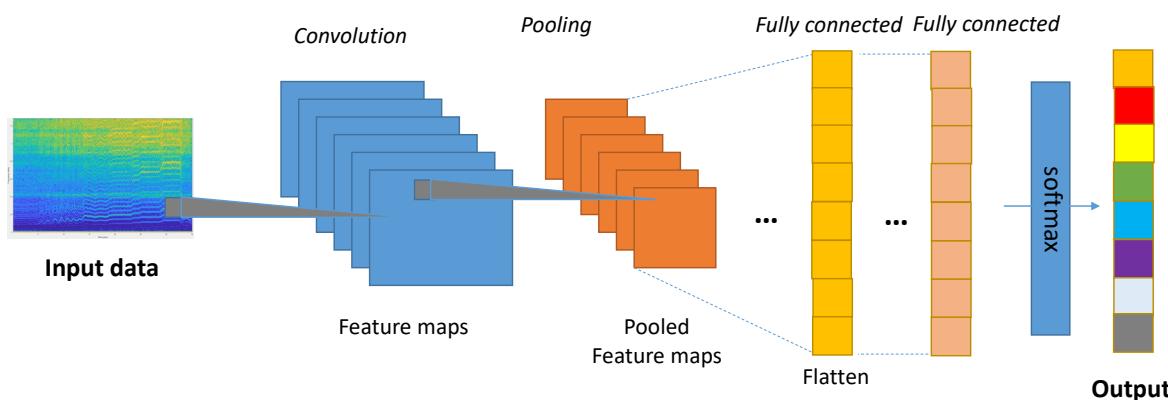


Figure 10. Graphical formulation of Convolutional Neural Networks.

For instance, given a two-dimensional input x , a two-dimensional kernel h computes the 2D convolution by

$$(x * h)_{i,j} = x[i,j] * h[i,j] = \sum_n \sum_m x[n,m] \cdot h[i-n][j-m] \quad (16)$$

i.e., the dot product between their weights and a small region they are connected to in the input.

After the convolution, a bias term is added and a point-wise nonlinearity g is applied, forming a *feature map* at the filter output. If we denote the l -th feature map at a given convolutional layer as h^l , whose filters are determined by the coefficients or *weights* W^l , the input x and the bias b_l , then the feature map h^l is obtained as follows

$$h^l_{i,j} = g((W^l * x)_{ij} + b_l), \quad (17)$$

where $*$ is the 2D convolution defined by Equation (16), while $g(\cdot)$ is the *activation function*.

Common activation functions encountered in deep neural networks are the *rectifier* that is defined as

$$g(x) = x^+ = \max(0, x), \quad (18)$$

the hyperbolic tangent function, *tanh*, $g(x) = \tanh(x)$, that is defined as

$$\tanh(x) = \frac{2}{1 + e^{-2x}} - 1, \quad (19)$$

and the *sigmoid* activation, $g(x) = \sigma(x)$, defined as

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \quad (20)$$

The *sigmoid* activation is rarely used because its activations saturate at either tail of 0 or 1 and they are not centered at 0 as is the *tanh*. The *tanh* normalizes the input to the range $[-1, 1]$, but compared to the rectifier its activations saturate which causes unstable gradients. Therefore, the *rectifier* activation function is typically used for CNNs. Kernels using the rectifier are called *ReLU* (Rectified Linear Unit) and have shown to greatly accelerate the convergence during the training process compared to other activation functions. They also do not cause vanishing or exploding of gradients in the optimization phase when minimizing the cost function. In addition, the ReLU simply thresholds the input, x , at zero, while other activation functions involve expensive operations.

In order to form a richer representation of the input signal, commonly, multiple filters are stacked so that each hidden layer consists of multiple *feature maps*, $\{h^{(l)}, l = 0, \dots, L\}$ (e.g., $L = 64, 128, \dots$, etc.). The number of filters per layer is a tunable parameter or *hyper-parameter*. Other tunable parameters are the filter size, the number of layers, etc. The selection of values for hyper-parameters may be quite difficult, and finding it commonly is much an art as it is science. An optimal choice may only be feasible by trial and error. The filter sizes are selected according to the input data size so as to have the right level of “granularity” that can create abstractions at the proper scale. For instance, for a 2D square matrix input, such as spectrograms, common choices are $3 \times 3, 5 \times 5, 9 \times 9$, etc. For a *wide* matrix, such as a real-valued representation of the complex I and Q samples of the wireless signal in $\mathbb{R}^{2 \times N}$, suitable filter sizes may be $1 \times 3, 2 \times 3, 2 \times 5$, etc.

After a convolutional layer, a *pooling* layer may be used to merge semantically similar features into one. In this way, the spatial size of the representation is reduced which reduces the amount of parameters and computation in the network. Examples of pooling units are *max pooling* (computes the maximum value of a local patch of units in one feature map), *neighbouring pooling* (takes the input from patches that are shifted by more than one row or column, thereby reducing the dimension of the representation and creating an invariance to small shifts and distortions, etc.

The penultimate layer in a CNN consists of *neurons* that are fully-connected with all feature maps in the preceding layer. Therefore, these layers are called *fully-connected* or *dense* layers. The very last layer is a *softmax* classifier, which computes the *posterior* probability of each class label over K classes as

$$\hat{y}_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}, i = 1, \dots, K \quad (21)$$

That is, the scores z_i computed at the output layer, also called *logits*, are translated into probabilities. A loss function, l , is calculated on the last fully-connected layer that measures the difference between the estimated probabilities, \hat{y}_i , and the one-hot encoding of the true class labels, y_i . The CNN parameters, θ , are obtained by minimizing the loss function on the training set $\{x_i, y_i\}_{i \in S}$ of size m ,

$$\min_{\theta} \sum_{i \in S} l(\hat{y}_i, y_i), \quad (22)$$

where $l(\cdot)$ is typically the mean squared error $l(y, \hat{y}) = \|y - \hat{y}\|_2^2$ or the *categorical cross-entropy* $l(y, \hat{y}) = \sum_{i=1}^m y_i \log(\hat{y}_i)$ for which a minus sign is often added in front to get the negative *log-likelihood*. Then the softmax classifier is trained by solving an optimization problem that minimizes the loss function. The optimal solution are the network parameters that fully describe the CNN model. That is $\hat{\theta} = \operatorname{argmin}_{\theta} J(S, \theta)$.

Currently, there is no consensus about the choice of the optimization algorithm. The most successful optimization algorithms seem to be: stochastic gradient descent (SGD), RMSProp, Adam, AdaDelta, etc. For a comparison on these, we refer the reader to [99].

To control over-fitting, typically regularization is used in combination with *dropout*, which is a new extremely effective technique that “drops out” a random set of activations in a layer. Each unit is retained with a fixed probability p , typically chosen using a validation set, or set to 0.5 which has shown to be close to optimal for a wide range of applications [100].

Recurrent Neural Networks

Recurrent neural networks (RNN) [101] are a type of neural networks where connections between nodes form a directed graph along a temporal sequence. They are called recurrent because of the recurrent connections between the hidden units. This is mathematically denoted as:

$$\mathbf{h}^{(t)} = f(\mathbf{h}^{(t-1)}, \mathbf{x}^{(t)}; \theta) \quad (23)$$

where function f is the activation output of a single unit, $\mathbf{h}^{(i)}$ are the state of the hidden units at time i , $\mathbf{x}^{(i)}$ is the input from the sequence at time index i , $\mathbf{y}^{(i)}$ is the output at time i , while θ are the network weight parameters used to compute the activation at all indices. Figure 11 shows a graphical representation of RNNs.

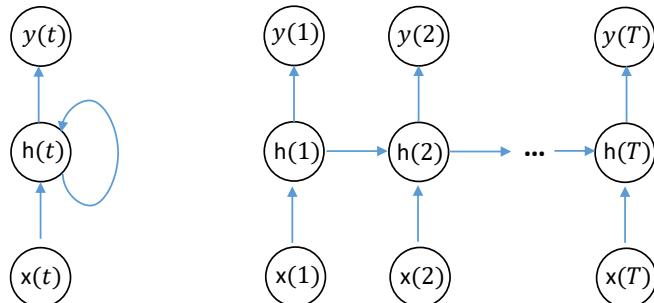


Figure 11. Graphical formulation of Recurrent Neural Networks.

The left part of Figure 11 presents the “folded” network, while the right part the “unfolded” network with its recurrent connections propagating information forward in time. An activation functional is applied in the hidden units and the *softmax* may be used to calculate the prediction.

There are various extensions of RNNs. A popular extension are LSTMs, which augment the traditional RNN model by adding a self-loop on the state of the network to better “remember” relevant information over longer periods in time.

5. Data Science Problems in Wireless Networks

The ultimate goal of data science is to extract knowledge from data, i.e., turn data into real value [102]. At the heart of this process are severe algorithms that can *learn* from and make predictions on data, i.e., machine learning algorithms. In the context of wireless networks, learning is a mechanism that enables context awareness and intelligence capabilities in different aspects of wireless communication. Over the last years, it has gained popularity due to its success in enhancing network-wide performance (i.e., QoS) [103], facilitating intelligent behavior by adapting to complex and dynamically changing (wireless) environments [104] and its ability to add automation for realizing concepts of self-healing and self-optimization [105]. During the past years, different data-driven approaches have been studied in the context of: mobile ad hoc networks [106], wireless sensor networks [107], wireless body area networks [50], cognitive radio networks [108,109] and cellular networks [110]. These approaches are focused on addressing various topics including: medium access control [30,111], routing [81,112], data aggregation and clustering [64,113], localization [114,115], energy harvesting communication [116], spectrum sensing [44,47], etc.

As explained in Section 4.1, prior to applying ML to a wireless networking problem, the problem needs to be first formulated as an adequate data mining task.

This section explains the following types of problems:

- Regression
- Classification
- Clustering
- Anomaly Detection

For each problem type, several wireless networking case studies are discussed together with the ML algorithms that are applied to solve the problem.

5.1. Regression

Regression is suitable for problems that aim to predict a real-valued output variable, y , as illustrated on Figure 12. Given a training set, \mathcal{S} , the goal is to estimate a function, f , whose graph fits the data. Once the function f is found, when an unknown point arrives, it is able to predict the output value. This function f is known as the *regressor*, and is defined as:

$$y = f(x) = \theta_0 + \theta_1 x_1 + \dots + \theta_n x_n \quad (24)$$

where $\theta_0, \dots, \theta_n$ are the function parameters.

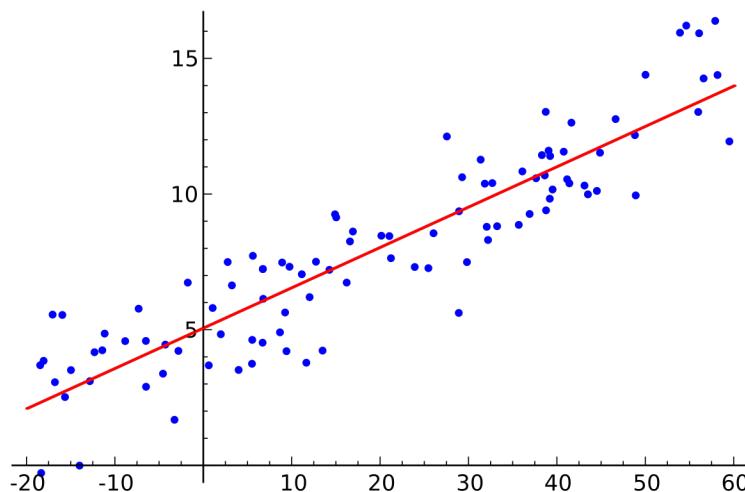


Figure 12. Illustration of regression.

Depending on the function representation, regression techniques are typically categorized into linear and non-linear regression algorithms, as explained in Section 4.3. For example, linear channel equalization in wireless communication can be seen as a regression problem.

5.1.1. Regression Example 1: Indoor Localization

In the context of wireless networks, linear regression is frequently used to derive an empirical log-distance model for the radio propagation characteristics as a linear mathematical relationship between the RSSI, usually in dBm, and the distance. This model can be used in RSSI-based indoor localization algorithms to estimate the distance towards each fixed node (i.e., anchor node) in the ranging phase of the algorithm [114].

5.1.2. Regression Example 2: Link Quality Estimation

Non-linear regression techniques are extensively used for modeling the relation between the PRR (Packet Reception Rate) and the RSSI, as well as between PRR and the Link Quality Indicator (LQI), to build a mechanism to estimate the link quality based on observations (RSSI, LQI) [117].

5.1.3. Regression Example 3: Mobile Traffic Demand Prediction

The authors in [118] use ML to optimize network resource allocation in mobile networks. Namely, each base station observes the traffic of a particular network slice in a mobile network. Then, a CNN model uses this information to predict the capacity required to accommodate the future traffic demands for services associated to each network slice. In this way, each slice gets optimal resources allocated.

5.2. Classification

A classification problem tries to understand and predict discrete values or categories. The term classification comes from the fact that it predicts the class membership of a particular input instance, as shown on Figure 13. Hence, the goal in classification is to assign an unknown *pattern* to one out of a number of classes that are considered to be known. For example, in digital communications, the process of demodulation can be viewed as a classification problem. Upon receiving the modulated transmitted signal, which has been impaired by propagation effects (i.e., the channel) and noise, the receiver has to decide which data symbol (out of a finite set) was originally transmitted. To evaluate the quality of the classification results, an intuitive way is to count the number of test

examples that are assigned to the right groups, which is also referred to as the accuracy rate (AR) defined by

$$AR = \frac{N_c}{N_t}, \quad (25)$$

where N_c denotes the number of test examples correctly assigned to the groups to which they belong; N_t the number of test patterns. To measure the details of the classification results, the so-called precision, $P = TP/(TP + FP)$, and recall, $R = TP/(TP + FN)$, are commonly used.

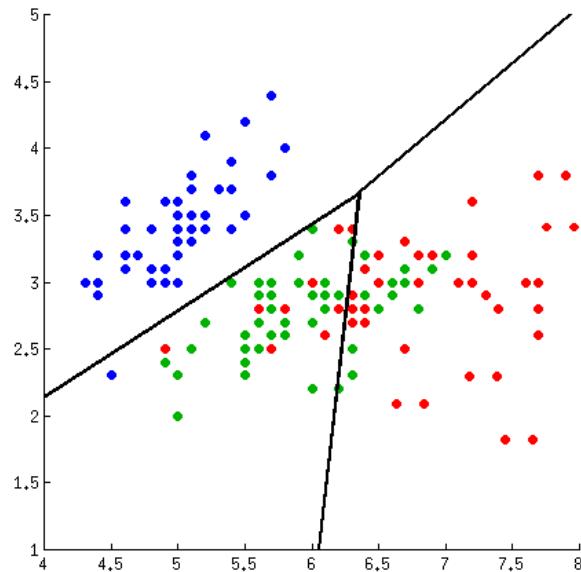


Figure 13. Illustration of classification.

Classification problems can be solved by supervised learning approaches, that aim to model boundaries between sets (i.e., classes) of similar behaving instances, based on known and labeled (i.e., with defined class membership) input values. There are many learning algorithms that can be used to classify data including decision trees, k-nearest neighbours, logistic regression, support vector machines, neural networks, convolutional neural networks, etc.

5.2.1. Classification Example 1: Cognitive MAC Layer

We consider the problem of designing an adaptive MAC layer as an application example of decision trees in wireless networks. In [30] a self-adapting MAC layer is proposed. It is composed of two parts: (i) a reconfigurable MAC architecture that can switch between different MAC protocols at run time, and (ii) a trained MAC engine that selects the most suitable MAC protocol for the current network condition and application requirements. The MAC engine is solved as a classification problem using a decision tree classifier which is learned based on: (i) two types of input variables which are (1) network conditions reflected through the RSSI statistics (i.e., mean and variance), and (2) the current traffic pattern monitored through the Inter-Packet Interval (IPI) statistics (i.e., mean and variance) and application requirements (i.e., reliability, energy consumption and latency), and (ii) the output which is the MAC protocol that is to be predicted and selected.

5.2.2. Classification Example 2: Intelligent Routing in WSN

Liu et al. [81] improved multi-hop wireless routing by creating a data-driven learning-based radio link quality estimator. They investigated whether machine learning algorithms (e.g., logistic regression, neural networks) can perform better than traditional, manually-constructed, pre-defined estimators such as STLE (Short-Term Link Estimator) [119] and 4 Bit (Four-Bit) [120]. Finally, they selected logistic regression as the most promising model

for solving the following classification problem: predict whether the next packet will be successfully received, i.e., output class is 1, or lost, i.e., output class is 0, based on the current wireless channel conditions reflected by statistics of the PRR, RSSI, SNR and LQI.

While in [81] the authors used offline learning to do prediction, in their follow-up work [112], they went a step further and both training and prediction were performed online by the nodes themselves using logistic regression with online learning (more specifically the stochastic gradient descent online learning algorithm). The advantage of this approach is that the learning and thus the model, adapt to changes in the wireless channel, that could otherwise be captured only by re-training the model offline and updating the implementation on the node.

5.2.3. Classification Example 3: Wireless Signal Classification

ML has been extensively used in cognitive radio applications to perform signal classification. For this purpose, typically flexible and reconfigurable SDR (software defined radio) platforms are used to sense the environment to obtain information about the wireless channel conditions and users' requirements, while intelligent algorithms build the cognitive learning engine that can make decisions on those reconfigurable parameters on SDR (e.g., carrier frequency, transmission power, modulation scheme).

In [44,47,121] SVMs are used as the machine learning algorithm to classify signals among a given set of possible modulation schemes. For instance, Huang et al. [47] identified four spectral correlation features that can be extracted from signals for distinction of different modulation types. Their trained SVM classifier was able to distinguish six modulation types with high accuracy: AM, ASK, FSK, PSK, MSK and QPSK.

5.3. Clustering

Clustering can be used for problems where the goal is to group sets of similar instances into clusters, as shown on Figure 14.

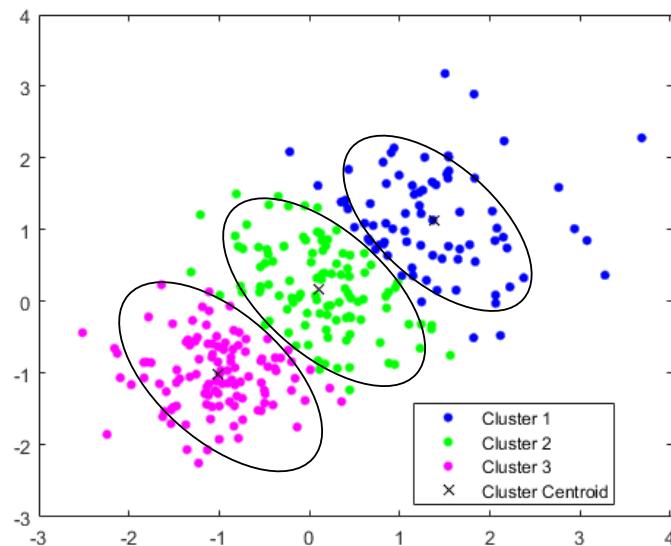


Figure 14. Illustration of clustering.

Opposed to classification, it uses *unsupervised* learning, which means that the input dataset instances used for training are not labeled, i.e., it is unknown to which group they belong. Given a set of unlabeled patterns $X = \{x_1, x_2, \dots, x_n\}$ in a d -dimensional space, the output of a clustering algorithm is a partitioning of X into k clusters $P = \{p_1, p_2, \dots, p_k\}$. The

output of a clustering problem also consists of a set of means or centroids $C = \{c_1, c_2, \dots, c_k\}$. A simple method for computing the means is as follows:

$$c_i = \frac{1}{|p_i|} \sum_{x \in p_i} x, \quad (26)$$

Clustering algorithms are widely adopted in wireless sensor networks, where they have found use for grouping sensor nodes into clusters to satisfy scalability and energy efficiency objectives, and finally elect the head of each cluster. A significant number of node clustering algorithms tends to be proposed for WSNs [122]. However, these *node* clustering algorithms typically do not use the data science clustering techniques directly. Instead, they exploit *data* clustering techniques to find data correlations or similarities between data of neighboring nodes, that can be used to partition sensor nodes into clusters.

Clustering can be used to solve other types of problems in wireless networks like anomaly detection, i.e., outliers detection, such as intrusion detection or event detection, for different data pre-processing tasks, cognitive radio application (e.g., identifying wireless systems [73]), etc. There are many learning algorithms that can be used for clustering, but the most commonly used is k -Means. Other popular clustering algorithms include hierarchical clustering methods such as single-linkage, complete-linkage, centroid-linkage; graph theory-based clustering such as highly connected subgraphs (HCS), cluster affinity search technique (CAST); kernel-based clustering as is support vector clustering (SVC), etc. A novel two-level clustering algorithm, namely TW- k -means, has been introduced by Chen et al. [113]. For a more exhaustive list of clustering algorithms and their explanation we refer the reader to [123]. Several clustering approaches have shown promise for designing efficient data aggregation for more efficient communication strategies in low power wireless sensor networks constrained. Given the fact that the most of the energy on the sensor nodes is consumed while the radio is turned on, i.e., while sending and receiving data [124], clustering may help to aggregate data in order to reduce transmissions and hence energy consumption.

5.3.1. Clustering Example 1: Summarizing Sensor Data

In [68] a distributed version of the k -Means clustering algorithm was proposed for clustering data sensed by sensor nodes. The clustered data is summarized and sent towards a sink node. Summarizing the data ensures to reduce the communication transmission, processing time and power consumption of the sensor nodes.

5.3.2. Clustering Example 2: Data Aggregation in WSN

In [64] a data aggregation scheme is proposed for in-network data summarization to save energy and reduce computation in wireless sensor nodes. The proposed algorithm uses clustering to form clusters of nodes sensing similar values within a given threshold. Then, only one sensor reading per cluster is transmitted which lowered extremely the number of transmissions in the wireless sensor network.

5.3.3. Clustering Example 3: Radio Signal Identification

The authors of [74] use clustering to separate and identify radio signal classes without to alleviate the need of using explicit class labels on examples of radio signals. First, dimensionality reduction is performed on signal examples to transform the signals into a space suitable for signal clustering. Namely, given an appropriate dimensionality reduction, signals are turned into a space where signals of the same or similar type have a low distance separating them while signals of differing types are separated by larger distances. Classification of radio signal types in such a space then becomes a problem of identifying clusters and associating a label with each cluster. The authors used the DBSCAN clustering algorithm [125].

5.4. Anomaly Detection

Anomaly detection (changes and deviation detection) is used when the goal is to identify unusual, unexpected or abnormal system behavior. This type of problem can be solved by supervised or unsupervised learning depending on the amount of knowledge present in the data (i.e., whether it is labeled or unlabeled, respectively).

Accordingly, classification and clustering algorithms can be used to solve anomaly detection problems. Figure 15 illustrates anomaly detection. A wireless example is the detection of suddenly occurring phenomena, such as the identification of suddenly disconnected networks due to interference or incorrect transmission power settings. It is also widely used for outliers detection in the pre-processing phase [126]. Other use-case examples include intrusion detection, fraud detection, event detection in sensor networks, etc.

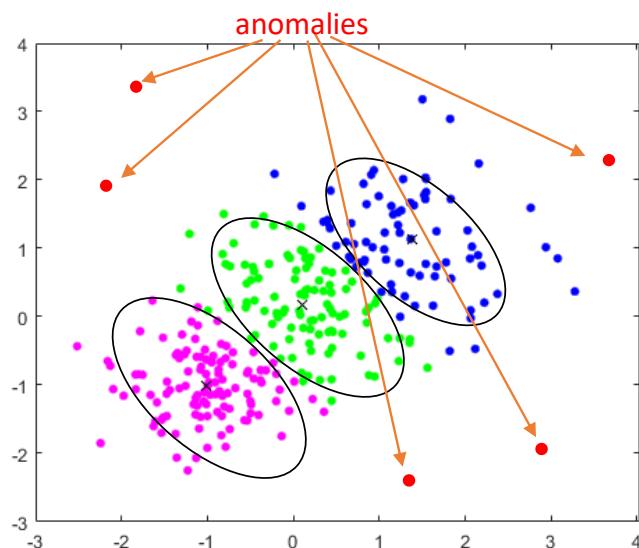


Figure 15. Illustration of anomaly detection.

5.4.1. Anomaly Detection Example 1: WSN Attack Detection

WSNs have been target of many types of DoS attacks. The goal of DoS attacks in WSNs is to transmit as many packets as possible whenever the medium is detected to be idle. This prevents a legitimate sensor node from transmitting their own packets. To combat a DoS attack, a secure MAC protocol based on neural networks has been proposed in [31]. The NN model is trained to detect an attack by monitoring variations of following parameters: collision rate R_c , average waiting time of a packet in MAC buffer T_w , arrival rate of RTS packets R_{RTS} . An anomaly, i.e., attack, is identified when the monitored traffic variations exceeds a preset threshold, after which the WSN node is switched off temporarily. The results is that flooding the network with untrustworthy data is prevented by blocking only affected sensor nodes.

5.4.2. Anomaly Detection Example 2: System Failure and Intrusion Detection

In [83] online learning techniques have been used to incrementally train a neural network for in-node anomaly detection in wireless sensor network. More specifically, the Extreme Learning Machine algorithm [127] has been used to implement classifiers that are trained online on resource-constrained sensor nodes for detecting anomalies such as: system failures, intrusion, or unanticipated behavior of the environment.

5.4.3. Anomaly Detection Example 3: Detecting Wireless Spectrum Anomalies

In [128] wireless spectrum anomaly detection has been studied. The authors use Power Spectral Density (PSD) data to detect and localize anomalies (e.g., unwanted signals

in the licensed band or the absence of an expected signal) in the wireless spectrum using a combination of Adversarial autoencoders (AAEs), CNN and LSTM.

6. Machine Learning for Performance Improvements in Wireless Networks

Obviously, machine learning is increasingly used in wireless networks [27]. After carefully looking at the literature, we identified two distinct categories or objectives where machine learning empowers wireless networks with the ability to learn and infer from data and extract patterns:

- **Performance improvements** of the wireless networks based on performance indicators and environmental insights (e.g., about the radio medium) as input, acquired from the devices. These approaches exploit ML to generate patterns or make predictions, which are used to modify operating parameters at the PHY, MAC and network layer.
- **Information processing** of data generated by wireless devices at the application layer. This category covers various applications such as: IoT environmental monitoring applications, activity recognition, localization, precision agriculture, etc.

This section presents tasks related to each of the aforementioned objectives achieved via ML and discusses existing work in the domain. First, the works are broadly summarized in tabular form in Table 2, followed by a detailed discussion of the most important works in each domain.

The focus of this paper is on the first category related to ML for performance improvement of wireless networks, therefore, a comprehensive overview of the existing work addressing problems pertaining to communication performance by making use of ML techniques is presented in the forthcoming subsection. These works provide a promising direction towards solving problems caused by the proliferation of wireless devices, networks and technologies in the near future, including: problems with interference (co-channel interference, inter-cell interference, cross technology interference, multi user interference, etc.), non-adaptive modulation scheme, static non-application cognizant MAC, etc.

Table 2. An overview of the applications of machine learning in wireless networks.

| Goal | Scope/Area | Example of Problem | References |
|-------------------------|-------------------------------|--|---|
| Performance Improvement | Radio spectrum analysis | <ul style="list-style-type: none"> • AMR • Wireless interference identification | [74,129–176] [128,169,177–193] |
| | MAC analysis | <ul style="list-style-type: none"> • MAC identification • Wireless interference detection at packet level • Spectrum prediction | [194–199] [200–203] [204–209] |
| | Network prediction | <ul style="list-style-type: none"> • Network performance prediction • Network traffic prediction | [30,81,112,210–216] [30,81,112,210–217] |
| Information processing | IoT Infrastructure monitoring | <ul style="list-style-type: none"> • Smart farming • Smart mobility • Smart city • Smart grid | [4–7] [218–221] |
| | Wireless security | Device fingerprinting | [222–228] |
| | Wireless localization | <ul style="list-style-type: none"> • Indoor • Outdoor | [229–234] |
| | Activity recognition | Via wireless signals | [235–239] |

6.1. Machine Learning Research for Performance Improvement

Data generated during monitoring of wireless networking infrastructure (e.g., throughput, end-to-end delay, jitter, packet loss, etc.) and by the wireless sensor devices (e.g., spectrum monitoring) and analyzed by ML techniques has the potential to optimize wireless networks configurations, thereby improving end-users' QoE. Various works have applied ML techniques for gaining insights that can help improve the network perfor-

mance. Depending on the type of data used as input for ML algorithms, we first categorize the researched literature into three types, summarized in Table 2:

- **Radio spectrum analysis**
- **Medium access control (MAC) analysis**
- **Network prediction**

Furthermore, within each of the above categories, we identified several classes of research approaches illustrated in Figure 16. In what follows, the work in these directions is reviewed and each class is presented in tabular form throughout Tables 3–7. Below is a definition of the columns used in these tables:

[Research Problem] *The problem addressed in the work.*

[Performance improvement] *Performance improvement achieved via ML in the work.*

[Type of wireless network] *The type of wireless networks considered in the work for which the problem is solved (e.g., Cognitive radio, Wi-Fi, etc.).*

[Data Type] *Type of data used in the work, e.g., synthetic or real.*

[Input Data] *The data used as input for the developed machine learning algorithms.*

[Learning Approach] *Type of learning approach, e.g., traditional machine learning (ML) or deep learning (DL).*

[Learning Algorithm] *List of learning algorithms used (e.g., CNN, SVM etc.).*

[Year] *The year when the work was published.*

[Reference] *The reference to the analyzed work.*

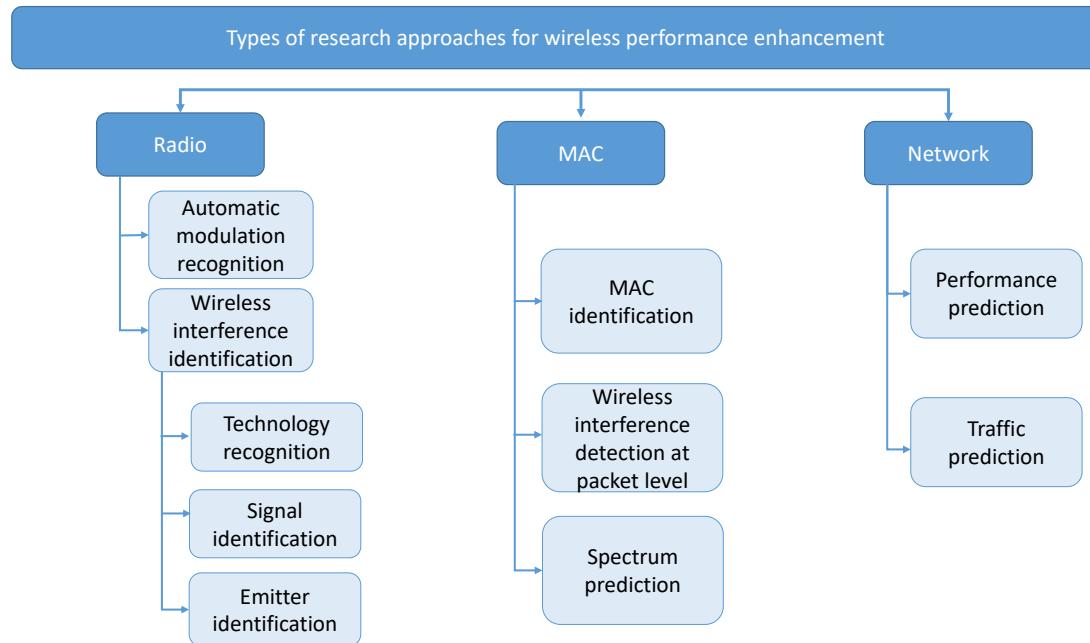


Figure 16. Types of research approaches for performance improvement of wireless networks.

6.1.1. Radio Spectrum Analysis

Radio spectrum analysis refers to investigating wireless data sensed by the wireless devices to infer the radio spectrum usage. Typically, the goal is to detect unused spectrum portions in order to share it with other coexisting users within the network without exorbitant interference with each other. Namely, as wireless devices become more pervasive throughout society the available radio spectrum, which is a scarce resource, will contain more non-cooperative signals than seen before. Therefore, collecting information about the signals within the spectrum of interest is becoming ever more important and complex. This has motivated the use of ML for analyzing the signals occupying the radio spectrum.

Automatic modulation recognition. AMR plays a key role in various civilian and military applications, where friendly signals shall be securely transmitted and received,

whereas hostile signals must be located, identified and jammed. In short, the goal of this task is to recognize the type of modulation scheme an emitter is using to modulate its transmitting signal based on raw samples of the detected signal at the receiver side. This information can provide insight about the type of communication systems and emitters present in the radio environment.

Traditional AMR algorithms were classified into likelihood-based (LB) approaches [240–242] and feature-based (FB) approaches [243,244]. LB approaches are based on detection theory (i.e., hypothesis testing) [245]. They can offer good performance and are considered optimal classifiers, however they suffer high computation complexity. Therefore, FB approaches were developed as suboptimal classifiers suitable for practical use. Conventional FB approaches heavily rely on expert knowledge, which may perform well for specialized solutions, however they are poor in generality and are time-consuming. Namely, in the preprocessing phase of designing the AMR algorithm, traditional FB approaches extracted complex hand engineered features (e.g., some signal parameters) computed from the raw signal and then employed an algorithm to determine the modulation schemes [246].

To remedy these problems, ML-based classifiers that aim to learn on preprocessed received data have been adopted and shown great advantages. ML algorithms usually provide better generalization to new unseen datasets, making their application preferable over solely FB approaches. For instance, the authors of [130,131,140] used the support vector machine (SVM) machine learning algorithm to classify modulation schemes. While strictly FB approaches may become obsolete with the advent of the employment of ML classifiers for AMR, hand engineered features can provide useful input to ML techniques. For instance in the following works [137,153], the authors engineered features using expert experience applied on the raw received signal and feeding the designed features as input for a neural network ML classifier.

Although ML methods have the advantage of better generality, classification efficiency and performance, the feature engineering step to some extent still depends on expert knowledge. As a consequence, the overall classification accuracy may suffer and depend on the expert input. On the other hand, current communication systems tend to become more complex and diverse, posing new challenges to the coexistence of homogeneous and heterogeneous signals and a heavy burden on the detection and recognition of signals in the complex radio environment. Therefore, the ability of self-learning is becoming a necessity when confronted with such complex environment.

Recently, the wireless communication community experienced a breakthrough by adopting deep learning techniques to the wireless domain. In [139], deep convolution neural networks (CNNs) are applied directly on complex time domain signal data to classify modulation formats. The authors demonstrated that CNNs outperform expert-engineered features in combination with traditional ML classifiers, such as SVMs, k-Nearest Neighbors (k-NN), Decision Trees (DT), Neural Networks (NN) and Naive Bayes (NB). An alternative method, is to learn the modulation format of the received signal from different representations of the raw signal. In our work in [247], CNNs are employed to learn the modulation of various signals using the in-phase and quadrature (IQ) data representation of the raw received signal and two additional data representations without affecting the simplicity of the input. We showed that the amplitude/phase representation outperformed the other two, demonstrating the importance of the choice of the wireless data representation used as input to the deep learning technique so as to determine the most optimal mapping from the raw signal to the modulation scheme. Other, follow-up works include [154–158,162,163,165–169], etc.

Recently, the authors of [172] proposed a novel accumulated polar feature based deep learning algorithm with a channel compensation mechanism for AMR that is capable to learn from A/Ph domain with historical information to reduce the offline training overhead. In addition, two mechanisms for online retraining are proposed to deal with the time-varying fading channel.

In [171] the authors were able to outperform state-of-the-art work on AMR based on the well-known RadioML2016.10a dataset, by proposing a novel data preprocessing on the signal samples to improve CNN-based AMR.

In [173] were able to increase the AMR classification accuracy by 11.5% at 10 dB SNR by adding two expert features in the data processing stage. Namely, the algorithm first detects whether the modulation is WBFM or not by means of the maximum of zero-center normalization amplitude spectrum density. Additionally, Haar-wavelet transform rest searching is used to classify QAM16 and QAM64.

The authors of [174] propose an effective LightAMC method using CNN and compressive sensing under varying noise regimes. The proposed LightAMC method uses L1 regularization-based neuron pruning technology is applied to cut down redundant neurons in the CNN for the M-AMC method. Hence, the proposed LightAMC method requires fewer device memories and has faster computational speed under the limited performance loss.

In [175] an AMR using a feature clustering based two-lane capsule network (AMC2N) is proposed. The authors were able to improve the classification accuracy by designing a new two-layer capsule network (TL-CapsNet), and the classification time is reduced by introducing a new feature clustering approach in the TL-CapsNet.

The authors of [176] introduce an improved CNN-based automatic modulation classification network (IC-AMCNet) by applying dropout and Gaussian noise layer, and instead of using many filters, they apply small number of filters to reduce the computing time, so that the algorithm can be applied in real-world network systems that require low-latency communications (such as Beyond 5G communications).

For a more comprehensive overview of the state-of-the art work on AMR we refer the reader to Tables 4 and 5.

Table 3. Description of the structure for Tables 4–7.

| Column Name | Description |
|--------------------------|---|
| Research Problem | The problem addressed in the work |
| Performance improvement | Performance improvement achieved in the work |
| Type of wireless network | The type of wireless networks considered in the work and/or for which the problem is solved |
| Data Type | Type of data used in the work, e.g., synthetic or real |
| Input Data | The data used as input for the developed machine learning algorithms |
| Learning Approach | Type of learning approach, e.g., traditional machine learning (ML) or deep learning (DL) |
| Learning Algorithm | List of learning algorithms used |
| Year | The year when the work was published |
| Reference | The reference to the analyzed work |

Wireless interference identification. WII essentially refers to identifying the type of wireless emitters (signal or technology) existing in the local radio environment, which can be immensely helpful information to investigate an effective interference avoidance and coexistence mechanisms. For instance, for technologies operating in the ISM bands in order to efficiently coexist it is crucial to know what type of other emitters are present in the environment (e.g., Wi-Fi, Zigbee, Bluetooth, etc.). Similar to AMR, FB and ML approaches (e.g., using time or frequency features) may be employed for technology recognition and signal identification approaches. Due to the development of deep learning applications for wireless signals classification, there has been significant success in applying it also for WII approaches.

For instance, the authors of [248] exploit the amplitude/phase difference representation to train a CNN model network to discriminate several radar signals from Wi-Fi and LTE transmissions. Their method was able to successfully recognize radar signals even under the presence of several interfering signals (i.e., LTE and Wi-Fi) at the same time, which is a key step for reliable spectrum monitoring.

In [157], the authors make use of the average magnitude spectrum representation of the raw observed signal on a distributed architecture with low-cost spectrum sensors together with an LSTM deep learning classifier to discriminate between different wireless emitters, such as TETRA, DVB, RADAR, LTE, GSM and WFM. Results showed that their method is able to outperform conventional ML approaches and a CNN based architecture for the given task.

In [179] the authors use the time domain quadrature (i.e., IQ) representation of the received signal and amplitude/phase vectors as input for CNN classifiers to learn the type of interfering technology present in the ISM spectrum. The results demonstrate that the proposed scheme is well suited for discriminating between Wi-Fi, ZigBee and Bluetooth signals. In [247], we introduce a methodology for end-to-end learning from various signal representations and investigate also the frequency domain (FFT) representation of the ISM signals and demonstrate that the CNN classifier that used FFT data as input outperforms the CNN models used by the authors in [179]. Similarly, the authors of [178] developed a CNN model to facilitate the detection and identification of frequency domain signatures for 802.x standard compliant technologies. Compared to [179] the authors in [178] make use of spectrum scans across the entire ISM region (80-MHz) and feed as input to a CNN model.

In [184] the authors used a CNN model to perform recognition of LTE and Wi-Fi transmissions based on two wireless signal representations, namely, the IQ and the frequency domain representation. The motivation behind this approach was to obtain accurate information about the technologies present in the local wireless environment so as to select an appropriate mLTE-U configuration that will allow fair coexistence with Wi-Fi in the unlicensed spectrum band.

Other examples include [128,177,182,183], etc.

In some applications like cognitive radio (CR) and spectrum sensing, the goal is however to identify the presence or absence of a signal. Namely, spectrum sensing is a process by which unlicensed users, also known as secondary users (SUs), acquire information about the status of the radio spectrum allocated to a licensed user, also known as primary user (PU), for the purpose of accessing unused licensed bands in an opportunistic manner without causing intolerable interference to the transmissions of the licensed user [249].

For instance, in [185] four ML techniques are examined k-NN, SVM, DT and logistic regression (LR) in order to predict the presence or absence of a PU in CR applications. The authors in [188] go a step further and design a spectrum sensing framework based on CNNs to facilitate a SU to achieve higher sensing accuracy compared with conventional approaches. For more examples, we refer the reader to [180,186,187].

Recently, the authors of [190] study the Wi-Fi/LTE coexistence problem by assuming a legacy operation of the interfering technology (LTE), which degrades the throughput of the incumbent technology (Wi-Fi) due to the interference. They propose a distributed spectrum management framework based on ML techniques that provides a global view of the spectral resources in wireless environments and helps Wi-Fi identify which wireless technology is transmitting. Based on that, different strategies for spectrum access can be enforced. The authors use a CNN-based architecture trained in a semi-supervised way using Autoencoders (AEs) to identify the wireless technology emitting and report the spectrum occupancy per channel. Results showed that using our approach, Wi-Fi maintains its throughput even when LTE uses all the spectral resources of a given channel.

Table 4. An overview of work on machine learning for radio level analysis for performance optimization—2010–2018.

| Research Problem | Performance Improvement | Type of Wireless Network | Data Type | Input Data | Learning Approach | Learning Algorithm | Year | Reference |
|------------------------|--|--------------------------|-----------|--|-------------------|-----------------------------|------|-----------|
| AMR | More efficient spectrum utilization | Cognitive radio | Synthetic | FR, ZCR, RE | ML | SVM | 2010 | [131] |
| AMR | More accurate signal modulation recognition for cognitive radio applications | Cognitive radio | Synthetic | CWT, HOM | ML | ANN | 2010 | [132] |
| Emitter identification | More accurate Radar Specific Emitter Identification | Radar | Real | Cumulants | ML | k-NN | 2011 | [133] |
| AMR | More accurate signal modulation recognition for cognitive radio and DSA applications | Cognitive radio | Synthetic | Max(PSD), NORM(A), AVG(x) | ML | ANN | 2011 | [134] |
| AMR | More accurate signal modulation recognition for cognitive radio applications | Cognitive radio | Synthetic | Cumulants | ML | k-NN | 2012 | [135] |
| AMR | More efficient spectrum utilization | Cognitive radio | Synthetic | Max(PSD), STD(ap), STD(dp), STD(aa), STD(df), F_c , cumulants, CWT | ML | SVM | 2012 | [136] |
| AMR | More efficient spectrum utilization | Cognitive radio | Synthetic | v_{20} , AVG(A), β , Max(PSD), STD(ap), STD(dp), STD(aa) | ML | FC-FFNN, FC-RNN | 2013 | [137] |
| AMR | More accurate signal modulation recognition for cognitive radio and DSA applications | Cognitive radio | Synthetic | ST, WT | ML | NN, SVM, LDA, NB, k-NN | 2015 | [138] |
| AMR | More efficient spectrum utilization | Cognitive radio | Synthetic | STD(dp), CWT, AVG(NORM()) | ML | SVM | 2016 | [140] |
| AMR | More efficient spectrum utilization | Cognitive radio | Real | IQ samples | DL & ML | CNN, DNN, k-NN, DT, SVM, NB | 2016 | [141] |
| AMR | More accurate signal modulation recognition for cognitive radio applications | Cognitive radio | Synthetic | IQ samples | ML | DNN | 2016 | [142] |
| AMR | More accurate signal modulation recognition for cognitive radio applications | Cognitive radio | Synthetic | IA signal samples, cumulants | DL & ML | CNN, SVM | 2017 | [143] |

Table 4. *Cont.*

| Research Problem | Performance Improvement | Type of Wireless Network | Data Type | Input Data | Learning Approach | Learning Algorithm | Year | Reference |
|------------------------|--|----------------------------|------------------|---|-------------------|-------------------------|------|-----------|
| AMR | More accurate signal modulation recognition for cognitive radio applications | Cognitive radio | Synthetic | Cumulants | DL | DNN, ANC, SAE | 2017 | [144] |
| AMR | More accurate signal modulation recognition for cognitive radio applications | Cognitive radio | Synthetic | IQ samples | DL | CLDNN, ResNet, DenseNet | 2017 | [145] |
| AMR | More accurate signal modulation recognition for cognitive radio | Cognitive radio | Synthetic | IQ samples | DL | CNN, AE | 2017 | [146] |
| Emitter identification | More efficient spectrum utilization | Cognitive radio | Synthetic | IQ samples | DL | AE, CNN | 2017 | [74] |
| AMR | More efficient spectrum utilization | Cognitive radio | Synthetic | IQ samples | DL | CLDNN, CNN, ResNet | 2017 | [147] |
| TR | More efficient management of the wireless spectrum | Cellular, WLAN, WPAN, WMAN | Real | Spectrograms | DL | CNN | 2017 | [177] |
| SI | More efficient spectrum utilization | Cognitive radio | Synthetic | CFD, (non)standardized IQ samples | DL | CNN | 2017 | [180] |
| AMR | More accurate and simple spectral events detection | ISM | Real | Spectrograms | DL | YOLO | 2017 | [139] |
| AMR | More efficient spectrum utilization | Cognitive radio | Synthetic | IQ samples | DL | CNN | 2017 | [148] |
| AMR | More efficient spectrum utilization | Cognitive radio | Synthetic | IQ samples | DL | CNN | 2017 | [149] |
| SI | More efficient spectrum monitoring | Radar | Real | Spectrograms, A/Ph | DL | CNN | 2017 | [248] |
| WII | Improved spectrum utilization | Bluetooth, Zigbee, Wi-Fi | Real | Power-frequency | DL | CNN | 2017 | [178] |
| SI | More efficient spectrum monitoring | Cognitive radio | Real | FFT | DL | CNN | 2017 | [150] |
| WII | More efficient spectrum management via wireless interference identification | Bluetooth, Zigbee, Wi-Fi | Synthetic | FFT | DL | CNN, NFSC | 2017 | [179] |
| Emitter identification | More accurate Radar Specific Emitter Identification | Radar | Real & Synthetic | FD-curves | ML | SVM | 2018 | [151] |
| AMR | More efficient spectrum utilization | Cognitive radio | Real | In-band spectral variation, deviation from unit circle, cumulants | ML | ANN, HH-AMC | 2018 | [153] |

Table 4. *Cont.*

| Research Problem | Performance Improvement | Type of Wireless Network | Data Type | Input Data | Learning Approach | Learning Algorithm | Year | Reference |
|------------------|--|--------------------------|-----------|-------------------------|-------------------|--------------------------|------|-----------|
| AMR | More efficient spectrum utilization | Cognitive radio | Synthetic | IQ samples, HOMs | DL | CNN, RN | 2018 | [154] |
| AMR | More accurate signal modulation recognition for cognitive radio and DSA applications | Cognitive radio | Synthetic | WT, CFD, HOMs, HOCs, ST | | SVM | 2018 | [250] |
| AMR | Modulation and Coding Scheme recognition for cognitive radio and DSA applications | Wi-Fi | Synthetic | IQ samples | DL | CNN | 2018 | [155] |
| AMR | More accurate signal modulation recognition for cognitive radio and DSA applications | Cognitive radio | Synthetic | IQ samples, FFT | DL | CNN, CLDNN, MTL-CNN | 2018 | [156] |
| AMR | More efficient spectrum utilization | Cognitive radio | Synthetic | A/Ph, AVG_{magFFT} | DL & ML | CNN, LSTM, RF, SVM, k-NN | 2018 | [157] |

Table 5. An overview of work on machine learning for radio level analysis for performance optimization—2018–2020.

| Research Problem | Performance Improvement | Type of Wireless Network | Data Type | Input Data | Learning Approach | Learning Algorithm | Year | Reference |
|------------------|---|--------------------------|-----------|---|-------------------|-------------------------|------|-----------|
| TR | More efficient spectrum management via wireless interference identification | Bluetooth, Zigbee, Wi-Fi | Synthetic | IQ samples | DL | CNN | 2018 | [181] |
| WII | More efficient spectrum utilization by detecting interference | Bluetooth, Zigbee, Wi-Fi | Real | RSSI samples | DL | CNN | 2018 | [183] |
| AMR | More efficient spectrum utilization | Cognitive radio | Synthetic | Contour Stellar Image | DL | CNN, AlexNet, ACGAN | 2018 | [158] |
| AMR | More efficient spectrum utilization | Cognitive radio | Synthetic | Constellation diagram | DL | CNN, AlexNet, GoogLeNet | 2018 | [160] |
| AMR | More efficient spectrum utilization | UAV | Synthetic | IQ samples | DL | CNN, LSTM | 2018 | [159] |
| SI | More efficient spectrum utilization | Cognitive radio | Synthetic | Activity, non-activity, “amplitude node” and permanence probability | ML | k-NN, SVM, LR, DT | 2018 | [185] |

Table 5. *Cont.*

| Research Problem | Performance Improvement | Type of Wireless Network | Data Type | Input Data | Learning Approach | Learning Algorithm | Year | Reference |
|------------------------|---|----------------------------------|------------------|---|-------------------|------------------------------|------|-----------|
| AMR | More efficient spectrum utilization | Cognitive radio | Synthetic | IQ samples | DL | CNN | 2018 | [162] |
| AMR | More efficient spectrum utilization by detecting interference | RF signals | Synthetic | IQ samples | DL & ML | SVM, DNN, CNN, MST | 2018 | [152] |
| AMR | More efficient spectrum utilization | Cognitive radio | Synthetic | IQ samples | DL & ML | CNN, LSTM, SVM | 2018 | [163] |
| AMR | More efficient spectrum utilization | VHF | Synthetic | IQ samples | DL | CNN | 2018 | [164] |
| Emitter identification | More efficient spectrum utilization | Cognitive radio | Synthetic | IQ signal samples, FOC | DL | CNN, LSTM | 2018 | [165] |
| SI | More efficient spectrum utilization | Cellular | Synthetic | IQ and Amplitude data | DL | CNN | 2018 | [161] |
| AMR | More efficient spectrum utilization | Cognitive radio | Synthetic | IQ samples | DL & ML | ACGAN, SCGAN, SVM, CNN, SSTM | 2018 | [166] |
| SI | Improved spectrum sensing | Cognitive radio | Synthetic | Beamformed IQ samples | ML | SVM | 2018 | [187] |
| AMR | More efficient spectrum utilization | Cognitive radio | Synthetic | IQ samples | DL & ML | DCGAN, NB, SVM, CNN | 2018 | [167] |
| AMR | More efficient spectrum utilization | Cognitive radio | Synthetic | IQ | DL | RNN | 2018 | [251] |
| AMR | More efficient spectrum utilization | Cognitive radio | Synthetic | IQ samples with corrected frequency and phase | DL | CNN, CLDNN | 2019 | [168] |
| TR | More efficient management of the wireless spectrum | LTE and Wi-Fi | Real | IQ samples, FFT | DL | CNN | 2019 | [184] |
| AMR | More efficient spectrum utilization | Cognitive radio | Synthetic | IQ samples | DL | CNN | 2019 | [169] |
| SI | Improved spectrum sensing | Cognitive radio | Synthetic | RSSI | DL | CNN | 2019 | [188] |
| WII | More efficient management of the wireless spectrum | Bluetooth, Zigbee, Wi-Fi | Real | FFT, A/Ph | DL | CNN, LSTM, ResNet, CLDNN | 2019 | [252] |
| WII | More efficient management of the wireless spectrum | Sigfox, LoRa and IEEE 802.15. 4g | Real | IQ, FFT | DL | CNN | 2019 | [253] |
| WII | More accurate and simple spectral events detection | Cognitive radio | Synthetic & Real | PSD data | DL | AAE | 2019 | [128] |

Table 5. *Cont.*

| Research Problem | Performance Improvement | Type of Wireless Network | Data Type | Input Data | Learning Approach | Learning Algorithm | Year | Reference |
|------------------|---|--|-----------|--|-------------------|---------------------|------|-----------|
| TR | More efficient management of the wireless spectrum | GSM, WCDMA and LTE | Synthetic | SCF, FFT, ACF, PSD | DL | SVM | 2019 | [254] |
| TR | More efficient management of the wireless spectrum | LTE, Wi-Fi and DVB-T | Real | RSSI, IQ and Spectrogram | DL | CNN | 2019 | [189] |
| AMR | More efficient spectrum utilization | Cognitive radio | Synthetic | IQ, A/Ph | DL | CLDNN, ResNet, LSTM | 2019 | [169] |
| AMR | More efficient spectrum utilization with online retraining | Cognitive radio | Synthetic | IQ, A/Ph | DL | ResNet, LSTM, CNN | 2020 | [172] |
| TR | More efficient spectrum access strategies to improve spectrum utilization | LTE and Wi-Fi | Real | IQ samples | DL | CNN, AEs | 2020 | [190] |
| TR | More efficient spectrum access strategies to improve spectrum utilization | Sigfox, LoRA, IEEE 802.15.4g and IEEE 802.11ah | Real | IQ samples, FFT | DL | CNN | 2020 | [191] |
| TR | More efficient spectrum access strategies to improve spectrum utilization | Incumbent and interference | Real | FFT | DL | CNN | 2020 | [192] |
| AMR | More efficient spectrum utilization | Cognitive radio | Synthetic | Novel data preprocessing method | DL | CNN | 2020 | [171] |
| AMR | More efficient spectrum utilization | Cognitive radio | Synthetic | IQ, Zero-center normalization amplitude spectrum density, Haar-wavelet transform | DL | CNN, LSTM | 2020 | [173] |
| AMR | More efficient spectrum utilization | Cognitive radio | Synthetic | IQ, Compressive sensing | DL | CNN | 2020 | [174] |
| AMR | More efficient spectrum utilization | Cognitive radio | Synthetic | IQ with feature extraction and clustering | DL | AMC2N | 2020 | [175] |
| AMR | More efficient spectrum utilization | Cognitive radio | Synthetic | IQ | DL | ICAMCNet | 2020 | [176] |
| WII | More efficient spectrum management via STAs identification | Wi-Fi | Synthetic | IQ | DL | CNN | 2020 | [193] |

The authors of [191] propose a CNN based low complexity near-real time multi-band sub-GHz technology recognition approach which supports a wide variety of technologies (Sigfox, LoRA, IEEE 802.15.4g and IEEE 802.11ah) using multiple settings. Results showed accuracies comparable with state-of-the-art solutions (around 99%), while the classification time remained small offering real-time execution, without the need of expensive and high power consuming hardware.

In [192], the authors proposed a two-step CNN-based algorithm that uses spectrum data and information provided by the incumbent to recognize, learn, and proactively predict the incumbent transmission pattern with an accuracy above 95% in near-real-time. The CNN outputs if a given technology is present in some spectrum voxel, where a spectrum voxel is a geometrical realization of the spectrum in terms of time, frequency, location, and power.

An interesting approach is available in [193], where the authors propose a MAC protocol based on intelligent spectrum learning for future WLAN networks. An access point (AP) is installed with a pre-trained CNN model able to identify the number of stations (STAs) involved in the collisions based on RF traces.

6.1.2. Medium Access Control (MAC) Analysis

Sharing the limited spectrum resources is the main concern in wireless networks [255]. One of the key functionalities of the MAC layer in wireless networks is to negotiate the access to the wireless medium to share the limited resources in an ad hoc manner. Opposed to centralized designs where entities like base stations control and distribute resources, nodes in Wireless Ad hoc Networks (WANETs) have to coordinate resources.

For this purpose, several MAC protocols have been proposed in the literature. Traditional MAC protocols designed for WANETs include Time Division Multiple Access (TDMA) [256,257], Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) [258,259], Code Division Multiple Access (CDMA) [260,261] and hybrid approaches [262,263]. However, given the changing network and environment conditions, designing a MAC protocol that fits all possible conditions and various application requirements is a challenge especially when these conditions are not available or known *a priori*. This subsection investigates the advances made related to the MAC layer to tackle the problem of efficient spectrum sharing with the help of machine learning. We identify two categories of MAC analysis (1) MAC identification and (2) Interference recognition. The reviewed MAC analysis tasks are listed in Table 6.

MAC identification. These approaches are typically employed in cognitive radio (CR) applications to foster communication and coexistence between protocol-distinct technologies. CRs rely on information gathered during spectral sensing to infer the environment conditions, presence of other technologies and spectrum holes. Spectrum holes are frequency bands that have been allocated to licensed network users but are not used at a particular time, which can be utilized by a CR user. Usually, spectrum sensing can determine the frequency range of a spectrum hole, while the timing information, which is also a channel access parameter, is unknown.

MAC protocol identification approaches may help CR users determine the timing information of a spectrum hole and accordingly tailor its packet transmission duration, which provides the potential benefits for network performance improvement. For this purpose, several MAC layer characteristics can be exploited.

For example, in [194] the TDMA and slotted ALOHA MAC protocols are identified based on two features, the power mean and the power variance of the received signal combined with a SVM classifier. The authors in [196] utilized power and time features to distinguish between four MAC protocols, namely TDMA, CSMA/CA, pure ALOHA, and slotted ALOHA using a SVM classifier. Similar, in [197] the authors captured MAC layer temporal features of 802.11 b/g/n homogeneous and heterogeneous networks and employed a k-NN and a NB ML classifier to distinguish between all three. In [198] the authors prove that a CNN model outperforms classical ML techniques such a SVM for

detecting MAC protocols especially when more types of MAC protocols are added in the dataset.

The authors of [199] introduce further improvements, by converting the sampled data into the form of a spectrogram and propose a CNN-based identification approach which combines the spectrogram and the CNN to identify TDMA, CSMA/CA, ALOHA and Pure ALOHA MAC protocols.

Wireless interference detection at packet level. Similar to the approaches of recognizing interference based on radio spectrum analysis, the goal here is to identify the type of radio interference which degrades the network performance. However, compared to the previously introduced work, the works in the MAC analysis level category focus on identifying distinct features of interfered channel and packets to detect and quantify interference in order to pinpoint the viability of opportunistic transmissions in interfered channels and select an appropriate strategy to co-exist with the present interference. This is realized based on information available on low-cost off-the-shelf devices, such as 802.15.4 and Wi-Fi radios, which is used as input for ML classifiers.

For instance, in [203] the authors investigated two possibilities for detecting interference: (1) the energy variations during packet reception captured by sampling the radio's RSSI register and (2) monitoring the Link Quality Indicator (LQI) of received corrupted packets. This information is combined with a DT classifier, considered as a computationally and memory efficient candidate for the implementation on 802.15.4 devices. Another work on interference identification in WSNs is [201]. The authors were able to accurately distinguish Wi-Fi, Bluetooth and microwave oven interference based on features of corrupted packets (i.e., mean normalized RSSI, LQI, RSSI range, error burst spanning and mean error burst spacing) used as input to a SVM and DT classifier.

In [200] the authors were able to detect non-Wi-Fi interference on Wi-Fi commodity hardware. They collected energy samples across the spectrum from the Wi-Fi card to extract a diverse set of features that capture the spectral and temporal properties of wireless signals (e.g., central frequency, bandwidth, spectral signature, duty cycle, pulse signature, inter-pulse timing signature, etc.). They used these features and investigated performance of two classifiers, SVM and DT. The idea is to embed these functionalities in Wi-Fi APs and clients, which can then implement an appropriate mitigation mechanism that can quickly react to the presence of significant non Wi-Fi interference.

Table 6. An overview of work on machine learning for medium access control (MAC) level analysis for performance optimization.

| Research Problem | Performance Improvement | Type of Wireless Network | Data Type | Input Data | Learning Approach | Learning Algorithm | Year | Reference |
|---|--|-------------------------------------|-----------|--|-------------------|--------------------|------|-----------|
| MAC identification | More efficient spectrum utilization | Cognitive radio | Synthetic | Mean and variance of power samples | ML | SVM | 2010 | [194] |
| Wireless interference detection at packet level | Enhanced spectral efficiency by interference mitigation | Wi-Fi | Real | Duty cycle, Spectral signatures, Frequency and bandwidth, Pulse signatures, Pulse spread, Inter-pulse timing signatures, Frequency sweep | ML | DT | 2011 | [200] |
| MAC identification | More efficient spectrum utilization | Cognitive radio | Synthetic | Power mean, Power variance, Maximum power, Channel busy duration, Channel idle duration | ML | SVM, NN, DT | 2012 | [195] |
| Wireless interference detection at packet level | Enhanced spectral efficiency by interference mitigation | Wi-Fi, Bluetooth, Microwave for WSN | Real | LQI, range(RSSI), Mean error burst spacing, Error burst spanning, AVG(NORM(RSSI)), 1 - mode(RSSI _{normed}) | ML | SVM, DT | 2013 | [201] |
| MAC identification | More efficient spectrum utilization | Cognitive radio | Synthetic | Received power mean, Power variance, Channel busy state duration, Channel idle state duration | ML | SVM | 2014 | [196] |
| Wireless interference detection at packet level | Reduced power consumption by interference identification | Wireless sensor network | Real | On-air time, Minimum Packet Interval, Peak to Average Power Ratio, Under Noise Floor | ML | DT | 2014 | [202] |
| MAC identification | More efficient spectrum utilization | WLAN | Real | Number of activity fragments at 111 µs, between 150 µs and 200 µs, between 200 µs and 300 µs, between 300 µs and 500 µs, and number of fragments between 1100 µs and 1300 µs | ML | k-NN, NB | 2015 | [197] |

Table 6. Cont.

| Research Problem | Performance Improvement | Type of Wireless Network | Data Type | Input Data | Learning Approach | Learning Algorithm | Year | Reference |
|---|--|--------------------------|--------------------|--|-------------------|--------------------|------|-----------|
| Wireless interference detection at packet level | Enhanced spectral efficiency by interference mitigation | Wireless sensor network | Real | Packet corruption rate, Packet loss rate, Packet length, Error rate, Error burstiness, Energy perception per packet, Energy perception level per packet, Backoffs, Occupancy level, Duty cycle, Energy span during packet reception, Energy level during packet reception, RSSI regularity during packet reception | ML | DT | 2015 | [203] |
| Spectrum prediction | Enhanced performance via more efficient spectrum usage achieved with medium availability prediction | ISM technologies | Synthetic | State matrix of the network, $X^{f,n}$ for each node n in frame f | ML | NN | 2018 | [204] |
| Spectrum prediction | Enhanced performance via more efficient spectrum usage achieved with select a predicted free channel | WSN | Synthetic and Real | DNN with network states as input and Q values as output | DL | DQN | 2018 | [205] |
| Spectrum prediction | More efficient radio resource utilization and enhanced link scheduling and power control | Cellular | Synthetic | The channel matrix H containing $ h_{i,j} ^2$ between all pairs of transmitter and receivers and the weight matrix W | DL | DQN and DNN | 2018 | [206] |
| Spectrum prediction | More efficient spectrum utilization and increased network performance | CRN | Synthetic | Environmental states s . | DL | ResNet, DNN, DQN | 2019 | [207] |
| Spectrum prediction | More efficient spectrum utilization and increased network performance via predicting PUs future activity | CRN | Synthetic | Vector x with n channel sensing results, where each result has value “−1” (idle) or “1” (busy). | ML | NN | 2019 | [208] |

Table 6. *Cont.*

| Research Problem | Performance Improvement | Type of Wireless Network | Data Type | Input Data | Learning Approach | Learning Algorithm | Year | Reference |
|---------------------|---|--------------------------|--------------------|---|-------------------|--------------------|------|-----------|
| Spectrum prediction | Enhanced performance via more efficient spectrum usage achieved with medium availability prediction | ISM technologies | Synthetic and Real | Channel observations $O_{s,c}^{f,n}$ on channel c made by node n at time (f, s) , where s is a timeslot in superframe f | DL | CNN | 2019 | [209] |
| MAC identification | More efficient spectrum utilization | Cognitive radio | Synthetic | Spectrograms of TDMA, Slotted ALOHA and FH signals | DL | CNN | 2020 | [198] |
| MAC identification | More efficient spectrum utilization | Cognitive radio | Synthetic | Spectrograms of TDMA, CSMA/CA, Slotted ALOHA and Pure ALOHA | DL | CNN | 2020 | [199] |

The authors of [202] propose an energy efficient rendezvous mechanism resilient to interference for WSNs based on ML. Namely, due to the energy constraints on sensor nodes, it is of great importance to save energy and extend the network lifetime in WSNs. Traditional rendezvous mechanism such as Low Power Listening (LPL) and Low Power Probe (LPP) rely on low duty cycling (scheduling the radio of a sensor node between ON and OFF compared to always-ON methods) depending on the presence of a signal (e.g., signal strength). However, both suffer performance degradation in noisy environments with signal interference incorrectly regarding a non-ZigBee interfering signal as an interested signal and improperly keeping the radio ON, which increases the probability of false wake-ups. To remedy this, the proposed approach in [202] is capable of detecting potential ZigBee transmissions and accordingly decide whether to turn the radio ON. For this purpose, they extracted signal features from time domain RSSI samples (i.e., On-air time, Minimum Packet Interval, Peak to Average Power Ratio and Under Noise Floor) and used it as input to a DT classifier to effectively distinguish ZigBee signals from other interfering ones.

Spectrum prediction. In order to share the available spectrum in a more efficient way, there are various attempts in predicting the wireless medium availability to minimize transmission collisions and, therefore, increase the overall performance of the network.

For instance, an intelligent wireless device may monitor the medium and based on MAC-level measurements predict if the medium is likely to be busy or idle. In another variation of this approach, a device may predict the quality of the channels in terms of properties such as idle probabilities or idle durations and then select the channel with the highest quality for transmission.

For instance, the authors in [204] use NNs to predict if a slot will be free based on some history to minimize collisions and optimize the usage of the scarce spectrum. In their follow up work [209], they exploit CNNs to predict the spectrum usage of the other neighboring networks. Their approach is aimed for devices with limited capabilities for retraining.

In [205], a Deep Q-Network (DQN) is proposed to predict and select a free channel for WSNs. In [208], the authors design a NN predictor to predict PUs future activity based on past channel occupancy sensing results, with the goal of improving secondary users (SUs) throughput while alleviating collision to primary user (PU) in full-duplex (FD) cognitive networks.

The authors of [207] consider the problem of sharing time slots among a multiple of time-slotted networks so as to maximize the sum throughput of all the networks. The authors utilize the ResNet and compare performance to a plain DNN.

MAC analysis approaches are listed in Table 6.

6.1.3. Network Prediction

Network prediction refers to tasks related to inferring the wireless network performance or network traffic, given historical measurements or related data. Table 7 gives an overview of the works on machine learning for network level prediction tasks, i.e., (1) Network performance prediction and (2) Network traffic prediction.

Network performance prediction. ML approaches are used, extensively, to create prediction models for many wireless networking applications. Typically, the goal is to forecast the performance or optimal device parameters/settings and use this knowledge to adapt the communication parameters to the changing environment conditions and application QoS requirements so as to optimize the overall network performance.

For instance, in [216] the authors aim to select the optimal MAC parameter settings in 6LoWPAN networks to reduce excessive collisions, packet losses and latency. First, the MAC layer parameters are used as input to a NN to predict the throughput and latency, followed by an optimization algorithm to achieve high throughput with minimum delay. The authors of [212] employ NNs to predict the users' QoE in cellular networks, based on

average user throughput, number of active users in a cell, average data volume per user and channel quality indicators, demonstrating high prediction accuracy.

Given the dynamic nature of wireless communications, a traditional one-MAC-fit-all approach cannot meet the challenges under significant dynamics in operating conditions, network traffic and application requirements. The MAC protocol may deteriorate significantly in performance as the network load becomes heavier, while the protocol may waste network resources when the network load turns lighter. To remedy this, [30,213] study an adaptive MAC layer with multiple MACs available that is able to select the MAC protocol most suitable for the current conditions and application requirements. In [30] a MAC selection engine for WSNs based on a DT model decides which is the best MAC protocol for the given application QoS requirements, current traffic pattern and ambient interference levels as input. The candidate protocols are TDMA, BoX-MAC and RI-MAC. The authors of [213] compare the accuracy of NB, Random Forest (RF), decision trees and SMO [264] to decide between the DCF and TDMA protocol to best respond to the dynamic network circumstances.

As an integral part of reliable communication in WSNs, accurate link estimation is essential for routing protocols, which is a challenging task due to the dynamic nature of wireless channels. To address this problem, the authors in [81] use ML (i.e., LR, NB and NN) to predict the link quality based on physical layer parameters of last received packets and the PRR, demonstrating high accuracy and improved routing. The same authors go a step further in [112] and employ online machine learning to adapt their link quality prediction mechanism real-time to the notoriously dynamic wireless environment.

The authors in [214] develop a ML engine that predicts the packet loss rate in a WSN using machine learning techniques network performance as an integral part for an adaptive MAC layer.

Table 7. An overview of work on machine learning for network level analysis for performance optimization.

| Research Problem | Performance Improvement | Type of Wireless Network | Data Type | Input Data | Learning Approach | Learning Algorithm | Year | Reference |
|--------------------------------|--|--------------------------|-----------|--|-------------------|-------------------------|------|-----------|
| Network performance prediction | Enhanced performance by accurate link quality prediction | Wireless sensor network | Real | PRR, RSSI, SNR and LQI | ML | LR, NB and NN | 2011 | [81] |
| Network performance prediction | Enhanced performance by accurate link quality prediction | Wireless sensor network | Real | PRR, RSSI, SNR and LQI | ML | LR | 2012 | [210] |
| Network performance prediction | Enhanced performance by selecting the optimal MAC scheme | Wireless sensor network | Real | RSSI statistics (mean and variance), IPI statistics (mean and variance), reliability, energy consumption and latency | ML | DT | 2013 | [30] |
| Network performance prediction | Enhanced performance by accurate link quality prediction | Wireless sensor network | Real | PRR, RSSI, SNR and LQI | ML | LR, SGD | 2014 | [112] |
| Network performance prediction | Enhanced network performance via performance characterization and optimal radio parameters prediction | Cellular | Synthetic | SINR (Signal to Interference plus Noise Ratio), ICI, MCS, Transmit power | ML | Random NN | 2015 | [211] |
| Network traffic prediction | Enhanced resource allocation by predicting mobile traffic demand | Cellular | Real | Traffic (Bytes)per 10 min | ML | Hierarchical clustering | 2015 | [265] |
| Network traffic prediction | Enhanced base station sleeping mechanism which reduced power consumption by predicting mobile traffic demand | Cellular | Synthetic | Machine learning | | | 2015 | [266] |
| Network traffic prediction | Enhanced resource allocation by predicting mobile traffic demand | Cellular | Real | User traffic | ML | MWNN | 2015 | [267] |

Table 7. *Cont.*

| Research Problem | Performance Improvement | Type of Wireless Network | Data Type | Input Data | Learning Approach | Learning Algorithm | Year | Reference |
|--------------------------------|--|--------------------------|-----------|---|-------------------|---------------------|------|-----------|
| Network performance prediction | Enhanced QoE by predicting KPI parameters | Cellular | Real | Mobile network KPIs | ML | NN | 2016 | [212] |
| Network performance prediction | Enhancing performance by selecting the optimal MAC | Cognitive radio | Synthetic | Protocol Type, Packet Length, Data Rate , Inter-arrival Time , Transmit Power, Node Number, Average load, Average throughput, Transmitting delay, Minimum throughput, Maximum throughput, Throughput standard deviation and classification result | ML | NB, DT, RF, SMO | 2016 | [213] |
| Network traffic prediction | Enhanced resource allocation by predicting mobile traffic demand | Cellular | Real | Mobile traffic volume | ML | Regression analysis | 2016 | [268] |
| Network traffic prediction | Enhanced resource allocation by predicting mobile traffic demand | Cellular | Real | Mobile traffic | ML | SVM, MLPWD, MLP | 2016 | [269] |
| Network performance prediction | Enhanced performance by predicting packet loss rate | WSN | Real | Number of detected nodes, IPI, Number of received packets, Number of erroneous packets LR, RT, NN | ML | | 2017 | [214] |
| Network traffic prediction | Enhanced resource allocation by predicting mobile traffic demand | Cellular | Real | Average traffic load per hour | DL | LSTM, GSSE, LSAE | 2017 | [270] |
| Network traffic prediction | Enhanced resource allocation by predicting mobile traffic demand | Cellular | Real | Number of CDRs generated during each time interval in a square of the Milan Grid | DL | RNN, 3D CNN | 2017 | [271] |
| Network performance prediction | Maximized reliability and minimized end-to-end delay by selecting optimal MAC parameters | 6LoWPAN | Synthetic | Maximum CSMA backoff, Backoff exponent, Maximum frame retries limit | ML | ANN | 2017 | [216] |

Table 7. *Cont.*

| Research Problem | Performance Improvement | Type of Wireless Network | Data Type | Input Data | Learning Approach | Learning Algorithm | Year | Reference |
|--------------------------------|---|--------------------------|-----------|---|-------------------|--------------------|------|-----------|
| Network traffic prediction | Enhanced resource allocation by predicting mobile traffic demand | Cellular | Real | Traffic volume snapshots every 10 min | DL | STN, LSTM, 3D CNN | 2018 | [272] |
| Network traffic prediction | Enhanced resource allocation by predicting mobile traffic demand | Cellular | Real | Cellular traffic load per half-hour | DL | LSTM,GNN | 2018 | [273] |
| Network traffic prediction | Enhanced resource allocation by predicting mobile traffic demand | Cellular | Real | CDRs with an interval of 10 min | DL | DNN, LSTM | 2018 | [274] |
| Network performance prediction | Enhanced network resource allocation by predicting network parameters | Wireless sensor network | Synthetic | Network lifetime, Power level, Internode distance | ML | NN | 2018 | [215] |
| Network traffic prediction | Enhanced resource allocation by predicting mobile traffic demand | Cellular | Real | SMS and Call volume per 10 min interval | DL | CNN | 2018 | [275] |
| Network traffic prediction | Enhanced resource allocation by predicting mobile traffic demand | Cellular | Real | Traffic load per 10 min | DL | LSTM | 2018 | [276] |
| Network traffic prediction | Enhanced resource allocation by more efficiently predicting mobile traffic load | Cellular | Real | Traffic logs recorded at 10 min intervals | ML | RF | 2018 | [277] |
| Network traffic prediction | Enhanced resource allocation by predicting mobile traffic demand | Cellular | Real | Traffic logs recorded at 15 min intervals | DL | LSTM | 2019 | [278] |

Table 7. *Cont.*

| Research Problem | Performance Improvement | Type of Wireless Network | Data Type | Input Data | Learning Approach | Learning Algorithm | Year | Reference |
|--------------------------------|--|--------------------------|-----------|--|-------------------|-----------------------------|------|-----------|
| Network traffic prediction | Enhanced resource allocation by predicting mobile traffic demand | Cellular | Real | Traffic load per 5 min intervals | DL | 3D CNN | 2019 | [118] |
| Network traffic prediction | Enhanced resource allocation by predicting idle time windows | Cellular | Real | Number of unique subscribers observed and Number of communication events occurring in a counting time window for a specific cell | DL | TGCN, TCN, LSTM, and GCLSTM | 2019 | [279] |
| Network performance prediction | Enhanced end user QoE for video streaming applications via accurate performance prediction | Cellular | Real | PHY and application data | DL & ML | RF, SVM, LSTM | 2020 | [217] |

Network traffic prediction. Accurate prediction of user traffic in cellular networks is crucial to evaluate and improve the system performance. For instance, the functional base station sleeping mechanism may be adapted by utilizing knowledge about future traffic demands, which are in [266] predicted based on a NN model. This knowledge helped reduce the overall power consumption, which is becoming an important topic with the growth of the cellular industry.

In another example, consider the need for efficient management of expensive mobile network resources, such as spectrum, where finding a way to predict future network use can help for network resource management and planning. A new paradigm for future 5G networks is *network slicing* enabling the network infrastructure to be divided into slices devoted to different services and tailored to their needs [280]. With this paradigm, it is essential to allocate the needed resources to each slice, which requires the ability to forecast their respective demands. The authors in [118] employed a CNN model that, based on traffic observed at base stations of a particular network slice, predicts the capacity required to accommodate the future traffic demands for services associated to it.

In [270] LSTMs are used to model the temporal correlations of the mobile traffic distribution and perform forecasting together with stacked Auto Encoders for spatial feature extraction. Experiments with a real-world dataset demonstrate superior performance over SVM and the Autoregressive Integrated Moving Average (ARIMA) model.

Deep learning was also employed in [271,274,276] where the authors utilize CNNs and LSTMs to perform mobile traffic forecasting. By effectively extracting spatio-temporal features, their proposals gain significantly higher accuracy than traditional approaches, such as ARIMA.

Forecasting with high accuracy the volume of data traffic that mobile users will consume is becoming increasingly important for demand-aware network resource allocation. More example approaches can be found in [118,269,270,272,273,275,277–279].

6.2. Machine Learning Applications for Information Processing

Wireless sensor nodes and mobile applications installed on various mobile devices record application level data frequently, making them act as sensor hubs responsible for data acquisition and preprocessing and subsequently storing the data in the “cloud” for further “offline” data storage and real-time computing using big data technologies (e.g., Storm [281], Spark [282], Kafka [283], Hadoop [284], etc). Example applications are (1) *IoT infrastructure monitoring* such as smart farming [5,6], smart mobility [4,218], smart city [7,219,220] and smart grid [221], (2) *device fingerprinting*, (3) *localization* and (4) *activity recognition*.

For instance, the works [222–228] exploit various time and radio patterns of the data with machine learning classifiers to distinguish legitimate wireless devices from adversarial ones, so as to increase the wireless network security.

In the works of [229–234] ML or deep learning is employed to localize users in indoor or outdoor environments, based on different signals received from wireless devices or about the wireless channels such as amplitude and phase channel state information (CSI), RSSI, etc.

The goal in the works [235–239] is to identify the activity of a person based on various wireless signal properties in combination with a machine learning technique. For instance, in [236] the authors demonstrate accurate human pose estimation through walls and occlusions based on properties of Wi-Fi wireless signals and how they reflect off the human body, used as input to a CNN classifier. In [237] the authors detect intruders based on how their movement patterns affect Wi-Fi signals in combination with a Gaussian Mixture Model (GMM).

For a more throughout overview of the applications and works on wireless information processing the reader is referred to [285].

7. Open Challenges and Future Directions

Previous sections presented the significant amount of research work focused on exploiting ML to address the spectrum scarcity problem in future wireless networks. However, despite the growing state-of-the-art with more and more different ML algorithms being explored and applied at various layers of the network protocol stack, there are still open challenges that need to be addressed in order to employ these paradigms in real radio environments to enable a fully intelligent wireless network in the near future.

This section discusses a set of open challenges and explores future research directions which are expected to accelerate the adoption of ML in future wireless network deployments.

7.1. Standard Datasets, Problems, Data Representation and Evaluation Metrics

7.1.1. Standard Datasets

To allow the comparison between different ML approaches, it is essential to have common benchmarks and standard datasets available, similar to the open dataset MNIST that is often used in computer vision. In order to effectively learn, ML algorithms will require a considerable amount of data. Furthermore, preferably standardized data generation/collection procedures should be created to allow reproducing the data. Research attempts in this direction include [286,287], showing that synthetic generation of RF signals is possible, however some wireless problems may require to inhibit specifics of a real system in the data (e.g., RF device fingerprinting).

Therefore, standardizing these datasets and benchmarks remains an open challenge. Significant research efforts need to be put in building large-scale datasets and sharing them with the wireless research community.

7.1.2. Standard Problems

Future research initiatives should identify a set of common problems in wireless networks to facilitate researchers in benchmarking and comparing their supervised/unsupervised learning algorithms. These problems should be supported with standard datasets. For instance, in computer vision for benchmarking computer vision algorithms for image recognition tasks, the MNIST and ImageNet datasets are typically used. Examples of standard problems in wireless networks may be: wireless signal identification, beamforming, spectrum management, wireless network traffic demand prediction, etc. Special research attention must be focused on designing these problems.

7.1.3. Standard Data Representation

DL is increasingly used in wireless networks, however it is still unclear what the optimal data representation is. For instance, an I/Q sample may be represented as a single complex number, a tuple of real numbers or via the amplitude and phase values of their polar coordinates. It is a debate that there is no one-size-fits-all data representation solution for every learning problem [247]. The optimal data representation might depend among other factors on the DL architecture, the learning objective and choice of the loss function [146].

7.1.4. Standard Evaluation Metrics

After identifying standard datasets and problems, future research initiatives should identify a set of standard metrics for evaluating and comparing different ML models. For instance, a set of standard metrics may be determined per standardized problem. Examples of standardized metrics might be: confusion matrix, F-score, precision, recall, accuracy, mean squared error, etc. In addition, the evaluation part may take into account other evaluation metrics such as: model complexity, memory overhead, training time, prediction time, required data size, etc.

7.2. Implementation of Machine Learning Models in Practical Wireless Platforms/Systems

There is no doubt that ML will play a prominent role in the evolution of future wireless networks. However, although ML is powerful, it may be a burden when running on a single device. Furthermore, DL which has shown great success, requires significant amount of data to perform well, which poses extra challenges on the wireless network. It is therefore of paramount importance to advance our understanding of how to simply and efficiently integrate ML/DL breakthroughs within constrained computing platforms. A second question that requires particular attention is which requirements does the network need to meet to support collection and transfer of large volumes of data?

7.2.1. Constraint Wireless Devices

Wireless nodes, such as for instance seen in the IoT (e.g., phones, watches and embedded sensors), are typically inexpensive devices with scarce resources: limited storage resource, energy, computational capability and communication bandwidth. These device constraints bring several challenges when it comes to implement and run complex ML models. Certainly, ML models with a large number of neurons, layers and parameters will necessarily require additional hardware and energy consumption not just for performing training but also for inference.

Reducing Complexity of Machine Learning Models

ML/DL is well on its way to becoming mainstream on constraint devices [288]. Promising early results are appearing across many domains, including hardware [289], systems and learning algorithms. For example, in [290] binary deep architectures are proposed, that are composed solely of 1-bit weights instead of 32-bit or 16-bit parameters, allowing for smaller models and less expensive computations. However, their ability to generalize and perform well in real-world problems is still an open question.

Distributed Machine Learning Implementation

Another approach to address this challenge, may be to distribute the ML computation load across multiple nodes. Some questions that need to be addressed here are: “Which part of the learning algorithms can be decomposed and distributed?”, “How are the input data and output calculation results communicated among the devices?”, “Which device is responsible for the assembly for the final prediction results?”, etc.

7.2.2. Infrastructure for Data Collection and Transfer

The tremendously increasing number of wireless devices and their traffic demands, require a scalable networking architecture to support large scale wireless transmissions. The transmission of large volumes of data is a challenging task due to the following reasons: (1) there are no standards/protocols that can efficiently deliver $> 100T$ bits of data per second, (2) it is extremely difficult to monitor the network in real-time, due to the huge traffic density in short time.

A promising direction in addressing this challenge is the concept of *fog computing/analytics* [18]. The idea of fog computing is to bring computing and analytics closer to the end-devices, which may improve the overall network performance by reducing or completely avoiding the transmission of large amounts of raw data to the cloud. Still, special efforts need to be devoted to employ these concepts in practical systems. Finally, *cloud computing* technologies (using virtualized resources, parallel processing and scalable data storage) may help reduce the computational cost when it comes to processing and analysis of data.

7.2.3. Software Defined Networking for Network Control

Recently, software-defined networking (SDN) has attracted great interest as a new paradigm in networking [291]. Due to the inherently distributed nature of traditional networks, machine learning techniques are challenging to be applied and deployed to

control and operate networks. To facilitate this, SDN appeared as a promising enabler to provide intelligence inside the networks. The main idea of SDN is to detach the control plane from the forwarding plane, to break vertical integration, and to introduce the ability to program the network. Namely, SDN allows logical centralization of feedback control, and decisions are made by the logically centralized network controller which has a global network view. The global view of the network enables the controller to obtain data from different layers of the network protocol stack with arbitrary granularity, such as channel state information at the physical layer, packet information at the data link/network layers, and also application information at the application layer. On the other hand, such global network view, makes the network easy to control and manage.

The capabilities of SDN (e.g., logically centralized control, global view of the network, software-based traffic analysis, and dynamic updating of forwarding rules) make it easier to apply machine learning techniques to make optimal decisions to adapt to the network environments which can significantly improve network control and management processes. For instance, the SDN controller may feed a ML model with all the cross-layer information from all the different layers, which uses this data to make predictions that provide insights about the network that can be used by the SDN controller to implement network control strategies (e.g., physical layer parameters adaptation, resource allocation, topology construction, routing mechanisms, congestion control, etc.) [292].

7.3. Machine Learning Model Accuracy in Practical Wireless Systems

Machine learning has been commonly used in static contexts, when the model speed is usually not a concern. For example, consider recognizing images in computer vision. Whilst, images are considered as stationary data, wireless data (e.g., signals) are inherently time-varying and stochastic. Training a robust ML model on wireless data that generalizes well is a challenging task due to the fact that wireless networks are inherently dynamic environments with changing channel conditions, user traffic demands and changing operating parameters (e.g., due to changes in standardization bodies). Considering that stability is one of the main requirements of wireless communication systems, rigorous theoretical studies are essential to ensure ML based approaches always work well in practical systems. The open question here is “How to efficiently train a ML model that generalizes well to unseen data in such dynamically changing system?”. The following paragraphs discuss promising directions in addressing this challenge.

7.3.1. Transfer Learning

With typical supervised learning a learned model is applicable for a specific scenario and likely biased to the training dataset. For instance, a learned model for recognizing a set of wireless technologies is trained to recognize only those technologies and also tight to the specific wireless environment characteristic where the data is collected. What if new technologies need to be identified? What if the conditions in the wireless environment change? Obviously, the ability of generalization of the trained learning models are still open questions. How can we efficiently adapt our model to these new circumstances?

Traditional approaches may require to retrain the model based on new data (i.e., incorporating new technologies or specifics of a new environment together with new labels). Fortunately, with the advances in ML it turns out that it is not necessary to fully retrain a ML model. A new popular method called *transfer learning* may solve this. Transfer learning is a method that allows to *transfer* the knowledge gained from one task to another similar task, and hereby alleviate the need to train ML models from scratch [293]. The advantage of this approach is that the learning process in new environments can be speeded up, with a smaller amount of data needed to train a good performing model. In this way, wireless networking researchers may solve new but similar problems in a more efficient manner. For instance, if the new task requires to recognize new modulation formats, the model parameters for an already trained CNN model may be reused as the initialization for training the new CNN.

7.3.2. Active Learning

Active learning is a subfield of ML that allows to update a learning model on-the-fly in a short period of time. For instance, in wireless networks, the benefit is that updating the model depending on the wireless networking conditions allows the model to be more accurate with respect to the current state [294].

The learning model adjusts its parameters whenever it receives new labeled data. The learning process stops when the system achieves the desired prediction accuracy.

7.3.3. Unsupervised/Semi-Supervised Deep Learning

Typical supervised learning approaches, especially the recently popular deep learning techniques, require a large amount of training data with a set of corresponding labels. The disadvantage here is that so much data might either not always be available or comes at a great expense to prepare. This is especially a time consuming task in wireless networks, where one has to wait for the occurrence of certain types of events (e.g., appearance of emission from a specific wireless technology or on a specific frequency band) for creating training instances to build robust models. At the same time, this process requires significant expert knowledge to construct labels, which is not a sufficiently automated process and generic for practical implementations.

To reduce the need for much domain knowledge and labeling data, *deep unsupervised learning* [128] and semi-supervised learning [74] is recently used. For instance, the AE (autoencoders) have become a powerful deep unsupervised learning tool [295], which have also shown the ability to compress the input information by possibly learning a lower dimensional encoding of the input. However, these new tools, require further research to fulfill their full potentials in (practical) wireless networks.

8. Conclusions

With the advances in hardware and computing power and the ability to collect, store and process massive amounts of data, machine learning (ML) has found its way into many different scientific fields, including wireless networks. The challenges wireless networks are faced with, pushed the wireless networking domain to seek more innovative solutions to ensure expected network performance. To address these challenges, ML is increasingly used in wireless networks.

In parallel, a growing number of surveys and tutorials emerged on ML applied in wireless networks. We noticed that some of the existing works focus on addressing specific wireless networking tasks (e.g., wireless signal recognition), some on the usage of specific ML techniques (e.g., deep learning techniques), while others on the aspects of a specific wireless environment (e.g., IoT, WSN, CRN, etc.) looking at broad application scenarios (e.g., localization, security, environmental monitoring, etc.). Therefore, we realized that none of the works elaborate ML for optimizing the performance of wireless networks, which is critically affected by the proliferation of wireless devices, networks, technologies and increased user traffic demands. We further noticed that some works are missing out the fundamentals, necessary for the reader to understand ML and data-driven research in general. To fill this gap, this paper presented (i) a well-structured starting point for non-machine learning experts, providing fundamentals on ML in an accessible manner, and (ii) a systematic and comprehensive survey on ML for performance improvements of wireless networks looking at various perspectives of the network protocol stack. To the best of our knowledge, this is the first survey that comprehensively reviews the latest research efforts (up until and including 2019) in applying prediction-based ML techniques focused on improving the performance of wireless networks, while looking at all protocol layers: PHY, MAC and network layer. The surveyed research works are categorized into: radio analysis, MAC analysis and network prediction approaches. We reviewed works in various wireless networks including IoT, WSN, cellular networks and CRNs. Within *radio analysis* approaches we identified the following: Automatic modulation recognition, and wireless interference identification (i.e., technology recognition, signal identification and emitter

identification). *MAC analysis* approaches are divided into: MAC identification, wireless interference identification and spectrum prediction tasks. *Network prediction* approaches are classified into: performance prediction, and traffic prediction approaches.

Finally, open challenges and exciting research directions in this field are elaborated. We discussed where standardization efforts are required, including standard: datasets, problems, data representations and evaluations metrics. Further, we discussed the open challenges when implementing machine learning models in practical wireless systems. Herewith, we discussed future directions at two levels: (i) implementing ML on constraint wireless devices (via reducing complexity of ML models or distributed implementation of ML models) and (ii) adapting the infrastructure for massive data collection and transfer (via edge analytics and cloud computing). Finally, we discussed open challenges and future directions on the generalization of ML models in practical wireless environments.

We hope that this article will become a source of inspiration and guide for researchers and practitioners interested in applying machine learning for complex problems related to improving the performance of wireless networks.

Author Contributions: Conceptualization, M.K. and T.K.; methodology, M.K.; software, M.K.; validation, M.K. T.K. and E.D.P.; formal analysis, M.K.; investigation, T.K.; resources, M.K.; data curation, M.K.; writing—original draft preparation, M.K.; writing—review and editing, M.K.; supervision, I.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Abbasi, A.; Sarker, S.; Chiang, R.H. Big data research in information systems: Toward an inclusive research agenda. *J. Assoc. Inf. Syst.* **2016**, *17*, 1. [CrossRef]
2. Qian, L.; Zhu, J.; Zhang, S. Survey of wireless big data. *J. Commun. Inf. Netw.* **2017**, *2*, 1–18. [CrossRef]
3. Rathore, M.M.; Ahmad, A.; Paul, A. IoT-based smart city development using big data analytical approach. In Proceedings of the 2016 IEEE International Conference on Automatica (ICA-ACCA), Curico, Chile, 19–21 October 2016; pp. 1–8.
4. Nguyen, H.N.; Krishnakumari, P.; Vu, H.L.; van Lint, H. Traffic congestion pattern classification using multi-class svm. In Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016; pp. 1059–1064.
5. Lottes, P.; Khanna, R.; Pfeifer, J.; Siegwart, R.; Stachniss, C. Uav-based crop and weed classification for smart farming. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 3024–3031.
6. Sa, I.; Chen, Z.; Popović, M.; Khanna, R.; Liebisch, F.; Nieto, J.; Siegwart, R. weednet: Dense semantic weed classification using multispectral images and mav for smart farming. *IEEE Robot. Autom. Lett.* **2018**, *3*, 588–595. [CrossRef]
7. Strohbach, M.; Ziekow, H.; Gazis, V.; Akiva, N. Towards a big data analytics framework for iot and smart city applications. In *Modeling and Processing for Next-Generation Big-Data Technologies*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 257–282.
8. Cisco. Cisco Visual Networking Index: Forecast and Trends. 2019. Available online: <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html> (accessed on 1 January 2021).
9. Cisco Systems White Paper. Available online: <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-738429.html> (accessed on 9 May 2019).
10. Bkassiny, M.; Li, Y.; Jayaweera, S.K. A survey on machine-learning techniques in cognitive radios. *IEEE Commun. Surv. Tutor.* **2012**, *15*, 1136–1159. [CrossRef]
11. Alsheikh, M.A.; Lin, S.; Niyato, D.; Tan, H.-P. Machine learning in wireless sensor networks: Algorithms, strategies, and applications. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 1996–2018. [CrossRef]
12. Wang, X.; Li, X.; Leung, V.C. Artificial intelligence-based techniques for emerging heterogeneous network: State of the arts, opportunities, and challenges. *IEEE Access* **2015**, *3*, 1379–1391. [CrossRef]
13. Ahad, N.; Qadir, J.; Ahsan, N. Neural networks in wireless networks: Techniques, applications and guidelines. *J. Netw. Comput. Appl.* **2016**, *68*, 1–27. [CrossRef]
14. Park, T.; Abuzaïnab, N.; Saad, W. Learning how to communicate in the internet of things: Finite resources and heterogeneity. *IEEE Access* **2016**, *4*, 7063–7073. [CrossRef]
15. Klaine, P.V.; Imran, M.A.; Onireti, O.; Souza, R.D. A survey of machine learning techniques applied to self-organizing cellular networks. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 2392–2431. [CrossRef]
16. Zhou, X.; Sun, M.; Li, G.Y.; Juang, B.-H.F. Intelligent wireless communications enabled by cognitive radio and machine learning. *China Commun.* **2018**, *15*, 16–48.

17. Mao, Q.; Hu, F.; Hao, Q. Deep learning for intelligent wireless networks: A comprehensive survey. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 2595–2621. [[CrossRef](#)]
18. Mohammadi, M.; Al-Fuqaha, A.; Sorour, S.; Guizani, M. Deep learning for iot big data and streaming analytics: A survey. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 2923–2960. [[CrossRef](#)]
19. Chen, M.; Challita, U.; Saad, W.; Yin, C.; Debbah, M. Artificial neural networks-based machine learning for wireless networks: A tutorial. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 3039–3071. [[CrossRef](#)]
20. Li, X.; Dong, F.; Zhang, S.; Guo, W. A survey on deep learning techniques in wireless signal recognition. *Wirel. Commun. Mob. Comput.* **2019**. [[CrossRef](#)]
21. Din, I.U.; Guizani, M.; Rodrigues, J.J.; Hassan, S.; Korotaev, V.V. Machine learning in the internet of things: Designed techniques for smart cities. *Future Gener. Comput. Syst.* **2019**, *100*, 826–843. [[CrossRef](#)]
22. Luong, N.C.; Hoang, D.T.; Gong, S.; Niyato, D.; Wang, P.; Liang, Y.-C.; Kim, D.I. Applications of deep reinforcement learning in communications and networking: A survey. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 3133–3174. [[CrossRef](#)]
23. Dhar, V. Data science and prediction. *Commun. ACM* **2013**, *56*, 64–73. [[CrossRef](#)]
24. Kulin, M.; Fortuna, C.; Poorter, E.D.; Deschrijver, D.; Moerman, I. Data-driven design of intelligent wireless networks: An overview and tutorial. *Sensors* **2016**, *16*, 790. [[CrossRef](#)]
25. McCarthy, J. Artificial intelligence, logic and formalizing common sense. In *Philosophical Logic and Artificial Intelligence*; Springer: Berlin/Heidelberg, Germany, 1989; pp. 161–190.
26. Mitchell, T.; Buchanan, B.; DeJong, G.; Dietterich, T.; Rosenbloom, P.; Waibel, A. Machine learning. *Annu. Rev. Comput. Sci.* **1990**, *4*, 417–433. [[CrossRef](#)]
27. Jiang, C.; Zhang, H.; Ren, Y.; Han, Z.; Chen, K.-C.; Hanzo, L. Machine learning paradigms for next-generation wireless networks. *IEEE Wirel. Commun.* **2017**, *24*, 98–105. [[CrossRef](#)]
28. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436. [[CrossRef](#)] [[PubMed](#)]
29. Liu, W.; Kulin, M.; Kazaz, T.; Shahid, A.; Moerman, I.; Poorter, E.D. Wireless technology recognition based on rssi distribution at sub-nyquist sampling rate for constrained devices. *Sensors* **2017**, *17*, 2081. [[CrossRef](#)] [[PubMed](#)]
30. Sha, M.; Dor, R.; Hackmann, G.; Lu, C.; Kim, T.-S.; Park, T. Self-adapting mac layer for wireless sensor networks. In Proceedings of the 2013 IEEE 34th Real-Time Systems Symposium, Vancouver, BC, Canada, 3–6 December 2013; pp. 192–201.
31. Kulkarni, R.V.; Venayagamoorthy, G.K. Neural network based secure media access control protocol for wireless sensor networks. In Proceedings of the Neural Networks, Atlanta, GA, USA, 14–19 June 2009; pp. 1680–1687.
32. Kim, M.H.; Park, M.-G. Bayesian statistical modeling of system energy saving effectiveness for mac protocols of wireless sensor networks. In *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 233–245.
33. Shen, Y.-J.; Wang, M.-S. Broadcast scheduling in wireless sensor networks using fuzzy hopfield neural network. *Expert Syst. Appl.* **2008**, *34*, 900–907. [[CrossRef](#)]
34. Barbancho, J.; León, C.; Molina, J.; Barbancho, A. Giving neurons to sensors. qos management in wireless sensors networks. In Proceedings of the Emerging Technologies and Factory Automation, Diplomat Hotel Prague, Czech Republic, 20–22 September 2006; pp. 594–597.
35. Liu, T.; Cerpa, A.E. Data-driven link quality prediction using link features. *ACM Trans. Sens. Netw.* **2014**, *10*, 37. [[CrossRef](#)]
36. Wang, Y.; Martonosi, M.; Peh, L.-S. Predicting link quality using supervised learning in wireless sensor networks. *ACM Sigmobile Mob. Comput. Commun. Rev.* **2007**, *11*, 71–83. [[CrossRef](#)]
37. Ahmed, G.; Khan, N.M.; Khalid, Z.; Ramer, R. Cluster head selection using decision trees for wireless sensor networks. In Proceedings of the Intelligent Sensors, Sensor Networks and Information Processing, Sydney, Australia, 15–18 December 2008; pp. 173–178.
38. Shareef, A.; Zhu, Y.; Musavi, M. Localization using neural networks in wireless sensor networks. In Proceedings of the 1st international Conference on MOBILE Wireless MiddleWARE, Operating Systems, and Applications, London, UK, 22–24 June 2008; p. 4.
39. Chagas, S.H.; Martins, J.B.; Oliveira, L.L.D. An approach to localization scheme of wireless sensor networks based on artificial neural networks and genetic algorithms. In Proceedings of the New Circuits and systems Conference (NEWCAS), Montreal, QC, Canada, 17–20 June 2012; pp. 137–140.
40. Tran, D.A.; Nguyen, T. Localization in wireless sensor networks based on support vector machines. *IEEE Trans. Parallel Distrib. Syst.* **2008**, *19*, 981–994. [[CrossRef](#)]
41. Tumuluru, V.K.; Wang, P.; Niyato, D. A neural network based spectrum prediction scheme for cognitive radio. In Proceedings of the 2010 IEEE International Conference on Communications (ICC), Cape Town, South Africa, 23–27 May 2010; pp. 1–5.
42. Baldo, N.; Zorzi, M. Learning and adaptation in cognitive radios using neural networks. In Proceedings of the 5th 2008 Consumer Communications and Networking Conference, Las Vegas, NV, USA, 10–12 January 2008; pp. 998–1003.
43. Tang, Y.-J.; Zhang, Q.-Y.; Lin, W. Artificial neural network based spectrum sensing method for cognitive radio. In Proceedings of the 2010 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM), Chengdu, China, 23–25 September 2010; pp. 1–4.

44. Hu, H.; Song, J.; Wang, Y. Signal classification based on spectral correlation analysis and svm in cognitive radio. In Proceedings of the AINA 2008, 22nd International Conference on Advanced Information Networking and Applications, Okinawa, Japan, 25–28 March 2008; pp. 883–887.
45. Xu, G.; Lu, Y. Channel and modulation selection based on support vector machines for cognitive radio. In Proceedings of the WiCOM 2006, International Conference on Wireless Communications, Networking and Mobile Computing, Wuhan, China, 22–24 September 2006; pp. 1–4.
46. Petrova, M.; Mähönen, P.; Osuna, A. Multi-class classification of analog and digital signals in cognitive radios using support vector machines. In Proceedings of the 2010 7th International Symposium on Wireless Communication Systems (ISWCS), York, UK, 19–22 September 2010; pp. 986–990.
47. Huang, Y.; Jiang, H.; Hu, H.; Yao, Y. Design of learning engine based on support vector machine in cognitive radio. In Proceedings of the CiSE 2009, International Conference on Computational Intelligence and Software Engineering, Wuhan, China, 11–13 December 2009; pp. 1–4.
48. Mannini, A.; Sabatini, A.M. Machine learning methods for classifying human physical activity from on-body accelerometers. *Sensors* **2010**, *10*, 1154–1175. [[CrossRef](#)]
49. Hong, J.H.; Kim, N.J.; Cha, E.J.; Lee, T.S. Classification technique of human motion context based on wireless sensor network. In Proceedings of the 27th Annual International Conference of the Engineering in Medicine and Biology Society, Shanghai, China, 31 August 2005; pp. 5201–5202.
50. Lara, O.D.; Labrador, M.A. A survey on human activity recognition using wearable sensors. *Commun. Surv. Tutor.* **2013**, *15*, 1192–1209. [[CrossRef](#)]
51. Bulling, A.; Blanke, U.; Schiele, B. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Comput. Surv.* **2014**, *46*, 33.
52. Bao, L.; Intille, S.S. Activity recognition from user-annotated acceleration data. In *Pervasive Computing*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 1–17.
53. Bulling, A.; Ward, J.A.; Gellersen, H. Multimodal recognition of reading activity in transit using body-worn sensors. *ACM Trans. Appl. Percept.* **2012**, *9*, 2. [[CrossRef](#)]
54. Yu, L.; Wang, N.; Meng, X. Real-time forest fire detection with wireless sensor networks. In Proceedings of the 2005 International Conference on Wireless Communications, Networking and Mobile Computing, Zhangjiajie, China, 2–4 August 2005; Volume 2, pp. 1214–1217.
55. Bahrepour, M.; Meratnia, N.; Havinga, P.J. Use of ai techniques for residential fire detection in wireless sensor networks. In Proceedings of the AIAI Workshops, Thessaloniki, Greece, 23–25 April 2009.
56. Bahrepour, M.; Meratnia, N.; Poel, M.; Taghikhaki, Z.; Havinga, P.J. Distributed event detection in wireless sensor networks for disaster management. In Proceedings of the 2nd International Conference on Intelligent Networking and Collaborative Systems (INCOS), Thessaloniki, Greece, 24–26 November 2010; pp. 507–512.
57. Zoha, A.; Imran, A.; Abu-Dayya, A.; Saeed, A. A machine learning framework for detection of sleeping cells in lte network. In Proceedings of the Machine Learning and Data Analysis Symposium, Doha, Qatar, 3–6 April 2014.
58. Khanafer, R.M.; Solana, B.; Triola, J.; Barco, R.; Moltsen, L.; Altman, Z.; Lazaro, P. Automated diagnosis for umts networks using bayesian network approach. *IEEE Trans. Veh. Technol.* **2008**, *57*, 2451–2461.
59. Ridi, A.; Gisler, C.; Hennebert, J. A survey on intrusive load monitoring for appliance recognition. In Proceedings of the 2014 22nd International Conference on Pattern Recognition (ICPR), Stockholm, Sweden, 24–28 August 2014; pp. 3702–3707.
60. Chang, H.-H.; Yang, H.-T.; Lin, C.-L. Load identification in neural networks for a non-intrusive monitoring of industrial electrical loads. In *Computer Supported Cooperative Work in Design IV*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 664–674.
61. Branch, J.W.; Giannella, C.; Szymanski, B.; Wolff, R.; Kargupta, H. In-network outlier detection in daa wireless sensor networks. *Knowl. Inf. Syst.* **2013**, *34*, 23–54. [[CrossRef](#)]
62. Kaplantzis, S.; Shilton, A.; Mani, N.; Sekercioğlu, Y.A. Detecting selective forwarding attacks in wireless sensor networks using support vector machines. In Proceedings of the 3rd International Conference on Intelligent Sensors, Sensor Networks and Information, Melbourne, Australia, 3–6 December 2007; pp. 335–340.
63. Kulkarni, R.V.; Venayagamoorthy, G.K.; Thakur, A.V.; Madria, S.K. Generalized neuron based secure media access control protocol for wireless sensor networks. In Proceedings of the IEEE Symposium on Computational Intelligence in Miulti-Criteria Decision-Making, Nashville, TN, USA, 30 March–2 April 2009; pp. 16–22.
64. Yoon, S.; Shahabi, C. The clustered aggregation (cag) technique leveraging spatial and temporal correlations in wireless sensor networks. *ACM Trans. Sens. Netw.* **2007**, *3*, 3. [[CrossRef](#)]
65. He, H.; Zhu, Z.; Mäkinen, E. A neural network model to minimize the connected dominating set for self-configuration of wireless sensor networks. *IEEE Trans. Neural Netw.* **2009**, *20*, 973–982. [[PubMed](#)]
66. Kanungo, T.; Mount, D.M.; Netanyahu, N.S.; Piatko, C.D.; Silverman, R.; Wu, A.Y. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 881–892. [[CrossRef](#)]
67. Liu, C.; Wu, K.; Pei, J. A dynamic clustering and scheduling approach to energy saving in data collection from wireless sensor networks. In Proceedings of the 2005 Second Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, SECON, Santa Clara, CA, USA, 4–7 October 2005; Volume 5, pp. 374–385.

68. Taherkordi, A.; Mohammadi, R.; Eliassen, F. A communication-efficient distributed clustering algorithm for sensor networks. In Proceedings of the 22nd International Conference on Advanced Information Networking and Applications-Workshops, AINAW, Okinawa, Japan, 25–28 March 2008; pp. 634–638.
69. Guo, L.; Ai, C.; Wang, X.; Cai, Z.; Li, Y. Real time clustering of sensory data in wireless sensor networks. In Proceedings of the IPCCC, Scottsdale, AZ, USA, 14–16 December 2009; pp. 33–40.
70. Wang, K.; Ayyash, S.A.; Little, T.D.; Basu, P. Attribute-based clustering for information dissemination in wireless sensor networks. In Proceedings of the 2nd Annual IEEE Communications Society Conference on Sensor and AD Hoc Communications and Networks (SECON'05), Santa Clara, CA, USA, 26–29 September 2005.
71. Ma, Y.; Peng, M.; Xue, W.; Ji, X. A dynamic affinity propagation clustering algorithm for cell outage detection in self-healing networks. In Proceedings of the Wireless Communications and Networking Conference (WCNC), Sydney, Australia, 18–21 July 2013; pp. 2266–2270.
72. TClancy, C.; Khawar, A.; Newman, T.R. Robust signal classification using unsupervised learning. *IEEE Trans. Wirel. Commun.* **2011**, *10*, 1289–1299. [[CrossRef](#)]
73. Shetty, N.; Pollin, S.; Pawełczak, P. Identifying spectrum usage by unknown systems using experiments in machine learning. In Proceedings of the Wireless Communications and Networking Conference, Budapest, Hungary, 5–8 April 2009; pp. 1–6.
74. O’Shea, T.J.; West, N.; Vondal, M.; Clancy, T.C. Semi-supervised radio signal identification. In Proceedings of the 2017 19th International Conference on Advanced Communication Technology (ICACT), Pyeongchang, Korea, 19–22 February 2017; pp. 33–38.
75. Witten, I.H.; Frank, E. *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed.; Morgan Kaufmann Series in Data Management Systems; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2005.
76. Guan, D.; Yuan, W.; Lee, Y.-K.; Gavrilov, A.; Lee, S. Activity recognition based on semi-supervised learning. In Proceedings of the 13th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, Daegu, Korea, 21–24 August 2007; pp. 469–475.
77. Stikic, M.; Larlus, D.; Ebert, S.; Schiele, B. Weakly supervised recognition of daily life activities with wearable sensors. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 2521–2537. [[CrossRef](#)]
78. Huỳnh, T.; Schiele, B. Towards less supervision in activity recognition from wearable sensors. In Proceedings of the 10th IEEE International Symposium on Wearable Computers, Cambridge, MA, USA, 11–14 October 2006; pp. 3–10.
79. Pulkkinen, T.; Roos, T.; Myllymäki, P. Semi-supervised learning for wlan positioning. In *Artificial Neural Networks and Machine Learning—ICANN 2011*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 355–362.
80. Erman, J.; Mahanti, A.; Arlitt, M.; Cohen, I.; Williamson, C. Offline/realtime traffic classification using semi-supervised learning. *Perform. Eval.* **2007**, *64*, 1194–1213.
81. Liu, T.; Cerpa, A.E. Foresee (4c): Wireless link prediction using link features. In Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks, Chicago, IL, USA, 12–14 April 2011; pp. 294–305.
82. Abdullah, M.F.A.B.; Negara, A.F.P.; Sayeed, M.S.; Choi, D.-J.; Muthu, K.S. Classification algorithms in human activity recognition using smartphones. *Int. J. Comput. Inf. Eng.* **2012**, *6*, 77–84.
83. Bosman, H.H.; Iacca, G.; Tejada, A.; Wörtche, H.J.; Liotta, A. Ensembles of incremental learners to detect anomalies in ad hoc sensor networks. *Ad Hoc Netw.* **2015**, *35*, 14–36. [[CrossRef](#)]
84. Zhang, Y.; Meratnia, N.; Havinga, P. Adaptive and online one-class support vector machine-based outlier detection techniques for wireless sensor networks. In Proceedings of the International Conference on Advanced Information Networking and Applications Workshops, WAINA'09, Bradford, UK, 26–29 May 2009; pp. 990–995.
85. Dasarathy, G.; Singh, A.; Balcan, M.-F.; Park, J.H. Active learning algorithms for graphical model selection. *arXiv* **2016**, arXiv:1602.00354.
86. Castro, R.M.; Nowak, R.D. Minimax bounds for active learning. *IEEE Trans. Inf. Theory* **2008**, *54*, 2339–2353. [[CrossRef](#)]
87. Beygelzimer, A.; Langford, J.; Tong, Z.; Hsu, D.J. Agnostic active learning without constraints. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, USA, 6–9 December 2010; pp. 199–207.
88. Hanneke, S. Theory of disagreement-based active learning. *Found. Trends Mach. Learn.* **2014**, *7*, 131–309. [[CrossRef](#)]
89. Freedman, D.A. *Statistical Models: Theory and Practice*; Cambridge University Press: Cambridge, UK, 2009.
90. Maimon, O.; Rokach, L. *Data Mining with Decision Trees: Theory and Applications*; World Scientific: Singapore, 2008.
91. Safavian, S.R.; Landgrebe, D. A survey of decision tree classifier methodology. *IEEE Trans. Syst. Man Cybern.* **1991**, *21*, 660–674. [[CrossRef](#)]
92. Vapnik, V.N.; Vapnik, V. *Statistical Learning Theory*; Wiley: New York, NY, USA, 1998; Volume 1.
93. Larose, D.T. *k-Nearest Neighbor Algorithm, Discovering Knowledge in Data: An Introduction to Data Mining*; John Wiley & Sons: Hoboken, NJ, USA, 2005; pp. 90–106.
94. Dunham, M.H. *Data Mining: Introductory and Advanced Topics*; Pearson Education India: Delhi, India, 2006.
95. Haykin, S.S.; Haykin, S.S.; Haykin, S.S.; Haykin, S.S. *Neural Networks and Learning Machines*; Pearson Education Upper Saddle River: Hoboken, NJ, USA, 2009; Volume 3.
96. Hornik, K.; Stinchcombe, M.; White, H. Multilayer Feedforward Networks are Universal Approximators. *Neural Netw.* **1989**, *2*, 359–366. [[CrossRef](#)]

97. Jia, F.; Lei, Y.; Lin, J.; Zhou, X.; Lu, N. Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data. *Mech. Syst. Signal Process.* **2016**, *72*, 303–315. [CrossRef]
98. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016. Available online: <http://www.deeplearningbook.org> (accessed on 1 January 2021).
99. Schaul, T.; Antonoglou, I.; Silver, D. Unit tests for stochastic optimization. *arXiv* **2013**, arXiv:1312.6055.
100. Srivastava, N.; Hinton, G.E.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
101. DRumelhart, E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors, *Cogn. Model.* **1988**, *5*, 1.
102. Fayyad, U.; Piatetsky-Shapiro, G.; Smyth, P. The kdd process for extracting useful knowledge from volumes of data. *Commun. ACM* **1996**, *39*, 27–34. [CrossRef]
103. Yau, K.-L.A.; Komisarczuk, P.; Teal, P.D. Reinforcement learning for context awareness and intelligence in wireless networks: Review, new features and open issues. *J. Netw. Comput. Appl.* **2012**, *35*, 253–267. [CrossRef]
104. Venayagamoorthy, G.K.K. A successful interdisciplinary course on computational intelligence. *IEEE Comput. Intell. Mag.* **2009**, *4*, 14–23. [CrossRef]
105. Khatib, E.J.; Barco, R.; Munoz, P.; Bandera, L.; De, I.; Serrano, I. Self-healing in mobile networks with big data. *IEEE Comput. Intell. Mag.* **2016**, *54*, 114–120. [CrossRef]
106. Förster, A. Machine learning techniques applied to wireless ad-hoc networks: Guide and survey. In Proceedings of the 3rd International Conference on Intelligent Sensors, Sensor Networks and Information, Melbourne, Australia, 3–6 December 2007; pp. 365–370.
107. Kulkarni, R.V.; Förster, A.; Venayagamoorthy, G.K. Computational intelligence in wireless sensor networks: A survey. *IEEE Commun. Surv. Tutor.* **2011**, *13*, 68–96. [CrossRef]
108. Thilina, K.M.; Choi, K.W.; Saquib, N.; Hossain, E. Machine learning techniques for cooperative spectrum sensing in cognitive radio networks. *IEEE J. Sel. Areas Commun.* **2013**, *31*, 2209–2221. [CrossRef]
109. Clancy, C.; Hecker, J.; Stuntebeck, E.; Shea, T.O. Applications of machine learning to cognitive radio networks. *IEEE Wirel. Commun.* **2007**, *14*, 47–52. [CrossRef]
110. Anagnostopoulos, T.; Anagnostopoulos, C.; Hadjiefthymiades, S.; Kyriakakos, M.; Kalousis, A. Predicting the location of mobile users: A machine learning approach. In Proceedings of the 2009 International Conference on Pervasive Services, London, UK, 13–16 July 2009; pp. 65–72.
111. Esteves, V.; Antonopoulos, A.; Kartsakli, E.; Puig-Vidal, M.; Miribel-Català, P.; Verikoukis, C. Cooperative energy harvesting-adaptive mac protocol for wbans. *Sensors* **2015**, *15*, 12635–12650. [CrossRef]
112. Liu, T.; Cerpa, A.E. Temporal adaptive link quality prediction with online learning. *ACM Trans. Sen. Netw.* **2014**, *10*, 1–41. [CrossRef]
113. Chen, X.; Xu, X.; Huang, J.Z.; Ye, Y. Tw-k-means: Automated two-level variable weighting clustering algorithm for multiview data. *IEEE Trans. Knowl. Data Eng.* **2013**, *25*, 932–944. [CrossRef]
114. Vanheel, F.; Verhaeverte, J.; Laermans, E.; Moerman, I.; Demeester, P. Automated linear regression tools improve rssi wsn localization in multipath indoor environment. *EURASIP J. Wirel. Commun. Netw.* **2011**, *2011*, 1–27. [CrossRef]
115. Tennina, S.; Renzo, M.D.; Kartsakli, E.; Graziosi, F.; Lalos, A.S.; Antonopoulos, A.; Mekikis, P.V.; Alonso, L. Wsn4qol: A wsn-oriented healthcare system architecture. *Int. J. Distrib. Sens. Netw.* **2014**, *10*, 503417. [CrossRef]
116. Blasco, P.; Gunduz, D.; Dohler, M. A learning theoretic approach to energy harvesting communication system optimization. *IEEE Trans. Wirel. Commun.* **2013**, *12*, 1872–1882. [CrossRef]
117. Levis, K. Rssi is under appreciated. In Proceedings of the Third Workshop on Embedded Networked Sensors, Cambridge, MA, USA, 30–31 May 2006; Volume 3031, p. 239242.
118. Bega, D.; Gramaglia, M.; Fiore, M.; Banchs, A.; Costa-Perez, X. Deepcog: Cognitive network management in sliced 5g networks with deep learning. In Proceedings of the IEEE INFOCOM, Paris, France, 29 April–2 May 2019.
119. Alizai, M.H.; Landsiedel, O.; Link, J.B.; Götz, S.; Wehrle, K. Bursty traffic over bursty links. In Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, Berkeley, CA, USA, 4–6 November 2009; pp. 71–84.
120. Fonseca, R.; Gnawali, O.; Jamieson, K.; Levis, P. Four-bit wireless link estimation. In Proceedings of the HotNets, Atlanta, GA, USA, 14–15 November 2007.
121. Ramon, M.M.; Atwood, T.; Barbin, S.; Christodoulou, C.G. Signal classification with an svm-fft approach for feature extraction in cognitive radio. In Proceedings of the 2009 SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference (IMOC), Belem, Brazil, 3–6 November 2009; pp. 286–289.
122. Abbasi, A.A.; Younis, M. A survey on clustering algorithms for wireless sensor networks. *Comput. Commun.* **2007**, *30*, 2826–2841. [CrossRef]
123. Xu, R.; Wunsch, D. Survey of clustering algorithms. *IEEE Trans. Neural Netw.* **2005**, *16*, 645–678. [CrossRef]
124. Kimura, N.; Latifi, S. A survey on data compression in wireless sensor networks. In Proceedings of the International Conference on Information Technology: Coding and Computing, Las Vegas, NV, USA, 4–6 April 2005; Volume 2, pp. 8–13.
125. Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Kdd, Cambridge, UK, 20 December 1996; Volume 96, pp. 226–231.

126. Zhang, Y.; Meratnia, N.; Havinga, P. Outlier detection techniques for wireless sensor networks: A survey. *IEEE Commun. Surv. Tutor.* **2010**, *12*, 159–170. [[CrossRef](#)]
127. Huang, G.-B.; Zhu, Q.-Y.; Siew, C.-K. Extreme learning machine: A new learning scheme of feedforward neural networks. In Proceedings of the 2004 IEEE International Joint Conference on Neural Networks, Budapest, Hungary, 25–29 July 2004; Volume 2, pp. 985–990.
128. Rajendran, S.; Meert, W.; Lenders, V.; Pollin, S. Unsupervised wireless spectrum anomaly detection with interpretable features. *IEEE Trans. Cogn. Commun. Netw.* **2019**, *5*, 637–647. [[CrossRef](#)]
129. Wong, M.D.; Nandi, A.K. Automatic digital modulation recognition using artificial neural network and genetic algorithm. *Signal Process.* **2004**, *84*, 351–365. [[CrossRef](#)]
130. Wang, L.-X.; Ren, Y.-J. Recognition of digital modulation signals based on high order cumulants and support vector machines. In Proceedings of the 2009 ISECS International Colloquium on Computing, Communication, Control, and Management, Sanya, China, 8–9 August 2009; Volume 4, pp. 271–274.
131. Tabatabaei, T.S.; Krishnan, S.; Anpalagan, A. Svm-based classification of digital modulation signals. In Proceedings of the 2010 IEEE International Conference on Systems, Man and Cybernetics, Istanbul, Turkey, 10–13 October 2010; pp. 277–280.
132. Hassan, K.; Dayoub, I.; Hamouda, W.; Berbineau, M. Automatic modulation recognition using wavelet transform and neural networks in wireless systems. *Eurasip J. Adv. Signal Process.* **2010**, *2010*, 532898. [[CrossRef](#)]
133. Aubry, A.; Bazzoni, A.; Carotenuto, V.; Maio, A.D.; Failla, P. Cumulants-based radar specific emitter identification. In Proceedings of the 2011 IEEE International Workshop on Information Forensics and Security, Washington, DC, USA, 29 November–2 December 2011; pp. 1–6.
134. Popoola, J.J.; Olst, R.V. A novel modulation-sensing method. *IEEE Veh. Technol. Mag.* **2011**, *6*, 60–69. [[CrossRef](#)]
135. Aslam, M.W.; Zhu, Z.; Nandi, A.K. Automatic modulation classification using combination of genetic programming and knn. *IEEE Trans. Wirel. Commun.* **2012**, *11*, 2742–2750.
136. Valipour, M.H.; Homayounpour, M.M.; Mehralian, M.A. Automatic digital modulation recognition in presence of noise using svm and pso. In Proceedings of the 6th International Symposium on Telecommunications (IST), Tehran, Iran, 6–8 November 2012; pp. 378–382.
137. Popoola, J.J.; van Olst, R. Effect of training algorithms on performance of a developed automatic modulation classification using artificial neural network. In Proceedings of the 2013 Africon, Pointe-Aux-Piments, Mauritius, 9–12 September 2013; pp. 1–6.
138. Satija, U.; Mohanty, M.; Ramkumar, B. Automatic modulation classification using s-transform based features. In Proceedings of the 2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN), Noida, New Delhi, 19–20 February 2015; pp. 708–712.
139. O’Shea, T.; Roy, T.; Clancy, T.C. Learning robust general radio signal detection using computer vision methods. In Proceedings of the 2017 51st Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, 29 October–1 November 2017; pp. 829–832.
140. Hassanpour, S.; Pezeshk, A.M.; Behnia, F. Automatic digital modulation recognition based on novel features and support vector machine. In Proceedings of the 2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), Naples, Italy, 28 November–1 December 2016; pp. 172–177.
141. O’Shea, T.J.; Corgan, J.; Clancy, T.C. Convolutional radio modulation recognition networks. In Proceedings of the International Conference on Engineering Applications of Neural Networks, Halkidiki, Greece, 5–7 June 2016; pp. 213–226.
142. Kim, B.; Kim, J.; Chae, H.; Yoon, D.; Choi, J.W. Deep neural network-based automatic modulation classification technique. In Proceedings of the 2016 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Korea, 19–21 October 2016; pp. 579–582.
143. Peng, S.; Jiang, H.; Wang, H.; Alwageed, H.; Yao, Y.-D. Modulation classification using convolutional neural network based deep learning model. In Proceedings of the 2017 26th Wireless and Optical Communication Conference (WOCC), Newark, NJ, USA, 7–8 April 2017; pp. 1–5.
144. Ali, A.; Yangyu, F. Automatic modulation classification using deep learning based on sparse autoencoders with nonnegativity constraints. *IEEE Signal Process. Lett.* **2017**, *24*, 1626–1630. [[CrossRef](#)]
145. Liu, X.; Yang, D.; Gamal, A.E. Deep neural network architectures for modulation classification. In Proceedings of the 2017 51st Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, 29 October–1 November 2017; pp. 915–919.
146. O’Shea, T.; Hoydis, J. An introduction to deep learning for the physical layer. *IEEE Trans. Cogn. Commun. Netw.* **2017**, *3*, 563–575. [[CrossRef](#)]
147. West, N.E.; O’Shea, T. Deep architectures for modulation recognition. In Proceedings of the 2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN), Baltimore, MD, USA, 6–9 March 2017; pp. 1–6.
148. Karra, K.; Kuzdeba, S.; Petersen, J. Modulation recognition using hierarchical deep neural networks. In Proceedings of the 2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN), Baltimore, MD, USA, 6–9 March 2017; pp. 1–3.
149. Hauser, S.C.; Headley, W.C.; Michaels, A.J. Signal detection effects on deep neural networks utilizing raw iq for modulation classification. In Proceedings of the MILCOM 2017—2017 IEEE Military Communications Conference (MILCOM), Baltimore, MD, USA, 23–25 October 2017; pp. 121–127.

150. Paisana, F.; Selim, A.; Kist, M.; Alvarez, P.; Tallon, J.; Bluemm, C.; Puschmann, A.; DaSilva, L. Context-aware cognitive radio using deep learning. In Proceedings of the 2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN), Baltimore, MD, USA, 6–9 March 2017; pp. 1–2.
151. Zhao, Y.; Wui, L.; Zhang, J.; Li, Y. Specific emitter identification using geometric features of frequency drift curve. *Bull. Pol. Acad. Sci. Tech. Sci.* **2018**, *66*, 99–108.
152. Youssef, K.; Bouchard, L.; Haigh, K.; Silovsky, J.; Thapa, B.; Valk, C.V. Machine learning approach to rf transmitter identification. *IEEE J. Radio Freq. Identif.* **2018**, *2*, 197–205. [[CrossRef](#)]
153. Jagannath, J.; Polosky, N.; O'Connor, D.; Theagarajan, L.N.; Sheaffer, B.; Foulke, S.; Varshney, P.K. Artificial neural network based automatic modulation classification over a software defined radio testbed. In Proceedings of the 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018; pp. 1–6.
154. O’Shea, T.J.; Roy, T.; Clancy, T.C. Over-the-air deep learning based radio signal classification. *IEEE J. Sel. Top. Signal Process.* **2018**, *12*, 168–179. [[CrossRef](#)]
155. Cheong, P.S.; Camelo, M.; Latré, S. Evaluating deep neural networks to classify modulated and coded radio signals. In Proceedings of the International Conference on Cognitive Radio Oriented Wireless Networks, Poznan, Poland, 11–12 June 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 177–188.
156. Mossad, O.S.; ElNainay, M.; Torki, M. Deep convolutional neural network with multi-task learning scheme for modulations recognition. In Proceedings of the 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC), Tangier, Morocco, 24–28 June 2019.
157. Rajendran, S.; Meert, W.; Giustiniano, D.; Lenders, V.; Pollin, S. Deep learning models for wireless signal classification with distributed low-cost spectrum sensors. *IEEE Trans. Cogn. Commun. Netw.* **2018**, *4*, 433–445.
158. Tang, B.; Tu, Y.; Zhang, Z.; Lin, Y. Digital signal modulation classification with data augmentation using generative adversarial nets in cognitive radio networks. *IEEE Access* **2018**, *6*, 15713–15722. [[CrossRef](#)]
159. Zhang, D.; Ding, W.; Zhang, B.; Xie, C.; Li, H.; Liu, C.; Han, J. Automatic modulation classification based on deep learning for unmanned aerial vehicles. *Sensors* **2018**, *18*, 924. [[CrossRef](#)]
160. Peng, S.; Jiang, H.; Wang, H.; Alwageed, H.; Zhou, Y.; Sebdani, M.M.; Yao, Y.-D. Modulation classification based on signal constellation diagrams and deep learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *99*, 718–727.
161. Duan, S.; Chen, K.; Yu, X.; Qian, M. Automatic multicarrier waveform classification via pca and convolutional neural networks. *IEEE Access* **2018**, *6*, 51365–51373.
162. Meng, F.; Chen, P.; Wu, L.; Wang, X. Automatic modulation classification: A deep learning enabled approach. *IEEE Trans. Veh. Technol.* **2018**, *67*, 10760–10772. [[CrossRef](#)]
163. Wu, Y.; Li, X.; Fang, J. A deep learning approach for modulation recognition via exploiting temporal correlations. In Proceedings of the 2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Kalamata, Greece, 25–28 June 2018; pp. 1–5.
164. Wu, H.; Wang, Q.; Zhou, L.; Meng, J. Vhf radio signal modulation classification based on convolution neural networks. In Proceedings of the Matec Web of Conferences, EDP Sciences, Lille, France, 8–10 October 2018; Volume 246, p. 03032.
165. Zhang, M.; Zeng, Y.; Han, Z.; Gong, Y. Automatic modulation recognition using deep learning architectures. In Proceedings of the 2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Kalamata, Greece, 25–28 June 2018; pp. 1–5.
166. Li, M.; Li, O.; Liu, G.; Zhang, C. Generative adversarial networks-based semi-supervised automatic modulation recognition for cognitive radio networks. *Sensors* **2018**, *18*, 3913. [[CrossRef](#)]
167. Li, M.; Liu, G.; Li, S.; Wu, Y. Radio classify generative adversarial networks: A semi-supervised method for modulation recognition. In Proceedings of the 2018 IEEE 18th International Conference on Communication Technology (ICCT), Chongqing, China, 8–11 October 2018; pp. 669–672.
168. Yashashwi, K.; Sethi, A.; Chaporkar, P. A learnable distortion correction module for modulation recognition. *IEEE Wirel. Commun. Lett.* **2019**, *8*, 77–80. [[CrossRef](#)]
169. Sadeghi, M.; Larsson, E.G. Adversarial attacks on deep-learning based radio signal classification. *IEee Wirel. Commun. Lett.* **2019**, *8*, 213–216. [[CrossRef](#)]
170. Ramjee, S.; Ju, S.; Yang, D.; Liu, X.; Gamal, A.E.; Eldar, Y.C. Fast deep learning for automatic modulation classification. *arXiv* **2019**, arXiv:1901.05850.
171. Zhang, H.; Huang, M.; Yang, J.; Sun, W. A data preprocessing method for automatic modulation classification based on cnn. *IEEE Commun. Lett.* **2020**. [[CrossRef](#)]
172. Teng, C.-F.; Chou, C.-Y.; Chen, C.-H.; Wu, A.-Y. Accumulated polar feature based deep learning with channel compensation mechanism for efficient automatic modulation classification under time varying channels. *arXiv* **2020**, arXiv:2001.01395.
173. Yao, T.; Chai, Y.; Wang, S.; Miao, X.; Bu, X. Radio signal automatic modulation classification based on deep learning and expert features. In Proceedings of the 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Moscow, Russia, 15–17 October 2020; Volume 1, pp. 1225–1230.
174. Wang, Y.; Yang, J.; Liu, M.; Gui, G. Lightamc: Lightweight automatic modulation classification via deep learning and compressive sensing. *IEEE Trans. Veh. Technol.* **2020**, *69*, 3491–3495. [[CrossRef](#)]

175. Al-Nuaimi, D.H.; Akbar, M.F.; Salman, L.B.; Abidin, I.S.Z.; Isa, N.A.M. Amc2n: Automatic modulation classification using feature clustering-based two-lane capsule networks. *Electronics* **2021**, *10*, 76.
176. Hermawan, A.P.; Ginanjar, R.R.; Kim, D.-S.; Lee, J.-M. Cnn-based automatic modulation classification for beyond 5g communications. *IEEE Commun. Lett.* **2020**, *24*, 1038–1041. [[CrossRef](#)]
177. O’Shea, T.J.; Roy, T.; Erpek, T. Spectral detection and localization of radio events with learned convolutional neural features. In Proceedings of the 2017 25th European Signal Processing Conference (EUSIPCO), Kos Island, Greece, 28 August–2 September 2017; pp. 331–335.
178. Bitar, N.; Muhammad, S.; Refai, H.H. Wireless technology identification using deep convolutional neural networks. In Proceedings of the 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Montreal, QC, Canada, 8–13 October 2017; pp. 1–6.
179. Schmidt, M.; Block, D.; Meier, U. Wireless interference identification with convolutional neural networks. *arXiv* **2017**, arXiv:1703.00737.
180. Han, D.; Sobabe, G.C.; Zhang, C.; Bai, X.; Wang, Z.; Liu, S.; Guo, B. Spectrum sensing for cognitive radio based on convolution neural network. In Proceedings of the 2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Shanghai, China, 14–16 October 2017; pp. 1–6.
181. Grunau, S.; Block, D.; Meier, U. Multi-label wireless interference classification with convolutional neural networks. In Proceedings of the 2018 IEEE 16th International Conference on Industrial Informatics (INDIN), Porto, Portugal, 18–20 July 2018; pp. 187–192.
182. Sun, H.; Chen, X.; Shi, Q.; Hong, M.; Fu, X.; Sidiropoulos, N.D. Learning to optimize: Training deep neural networks for interference management. *IEEE Trans. Signal Process.* **2018**, *66*, 5438–5453.
183. Yi, S.; Wang, H.; Xue, W.; Fan, X.; Wang, L.; Tian, J.; Matsukura, R. Interference source identification for ieee 802.15. 4 wireless sensor networks using deep learning. In Proceedings of the 2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Bologna, Italy, 9–12 September 2018; pp. 1–7.
184. Maglogiannis, V.; Shahid, A.; Naudts, D.; Poorter, E.D.; Moerman, I. Enhancing the coexistence of lte and wi-fi in unlicensed spectrum through convolutional neural networks. *IEEE Access* **2019**, *7*, 28464–28477. [[CrossRef](#)]
185. Soto, D.D.Z.; Parra, O.J.S.; Sarmiento, D.A.L. Detection of the primary user’s behavior for the intervention of the secondary user using machine learning. In Proceedings of the International Conference on Future Data and Security Engineering, Ho Chi Minh City, Vietnam, 28–30 November 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 200–213.
186. Lee, W. Resource allocation for multi-channel underlay cognitive radio network based on deep neural network. *IEEE Commun. Lett.* **2018**, *22*, 1942–1945. [[CrossRef](#)]
187. Awe, O.P.; Deligiannis, A.; Lambotharan, S. Spatio-temporal spectrum sensing in cognitive radio networks using beamformer-aided svm algorithms. *IEEE Access* **2018**, *6*, 25377–25388. [[CrossRef](#)]
188. Lee, W.; Kim, M.; Cho, D.-H. Deep cooperative sensing: Cooperative spectrum sensing based on convolutional neural networks. *IEEE Trans. Veh. Technol.* **2019**, *68*, 3005–3009. [[CrossRef](#)]
189. Fontaine, J.; Fonseca, E.; Shahid, A.; Kist, M.; DaSilva, L.A.; Moerman, I.; Poorter, E.D. Towards low-complexity wireless technology classification across multiple environments. *Hoc Netw.* **2019**, *91*, 101881. [[CrossRef](#)]
190. Soto, P.; Camelo, M.; Fontaine, J.; Girmay, M.; Shahid, A.; Maglogiannis, V.; Poorter, E.D.; Moerman, I.; Botero, J.F.; Latré, S. Augmented wi-fi: An ai-based wi-fi management framework for wi-fi/lte coexistence. In Proceedings of the 2020 16th International Conference on Network and Service Management (CNSM), Izmir, Turkey, 2–6 November 2020; pp. 1–9.
191. Fontaine, J.; Shahid, A.; Elsas, R.; Seferagic, A.; Moerman, I.; Poorter, E.D. Multi-band sub-ghz technology recognition on nvidia’s jetson nano. In Proceedings of the IEEE Vehicular Technology Conference VTC-Fall2020, Honolulu, HI, USA, 18 November–16 December 2020; pp. 1–7.
192. Camelo, M.; Mennes, R.; Shahid, A.; Struye, J.; Donato, C.; Jabandzic, I.; Giannoulis, S.; Mahfoudhi, F.; Maddala, P.; Seskar, I.; et al. An ai-based incumbent protection system for collaborative intelligent radio networks. *IEEE Wirel. Commun.* **2020**, *27*, 16–23. [[CrossRef](#)]
193. Yang, B.; Cao, X.; Omotere, O.; Li, X.; Han, Z.; Qian, L. Improving medium access efficiency with intelligent spectrum learning. *IEEE Access* **2020**, *8*, 94484–94498. [[CrossRef](#)]
194. Yang, Z.; Yao, Y.-D.; Chen, S.; He, H.; Zheng, D. Mac protocol classification in a cognitive radio network. In Proceedings of the The 19th Annual Wireless and Optical Communications Conference (WOCC 2010), Shanghai, China, 14–15 May 2010; pp. 1–5.
195. Hu, S.; Yao, Y.-D.; Yang, Z. Mac protocol identification approach for implement smart cognitive radio. In Proceedings of the 2012 IEEE International Conference on Communications (ICC), Ottawa, ON, Canada, 10–15 June 2012; pp. 5608–5612.
196. Hu, S.; Yao, Y.-D.; Yang, Z. Mac protocol identification using support vector machines for cognitive radio networks. *IEEE Wirel. Commun.* **2014**, *21*, 52–60. [[CrossRef](#)]
197. Rajab, S.A.; Balid, W.; Kalaa, M.O.A.; Refai, H.H. Energy detection and machine learning for the identification of wireless mac technologies. In Proceedings of the 2015 International Wireless Communications and Mobile Computing Conference (IWCMC), Dubrovnik, Croatia, 24–28 August 2015; pp. 1440–1446.
198. Zhou, Y.; Peng, S.; Yao, Y. Mac protocol identification using convolutional neural networks. In Proceedings of the 2020 29th Wireless and Optical Communications Conference (WOCC), Newark, NJ, USA, 1–2 May 2020; pp. 1–4.

199. Zhang, X.; Shen, W.; Xu, J.; Liu, Z.; Ding, G. A mac protocol identification approach based on convolutional neural network. In Proceedings of the 2020 International Conference on Wireless Communications and Signal Processing (WCSP), Nanjing, China, 21–23 October 2020; pp. 534–539.
200. Rayanchu, S.; Patro, A.; Banerjee, S. Airshark: Detecting non-wifi rf devices using commodity wifi hardware. In Proceedings of the 2011 ACM Sigcomm Conference on Internet Measurement Conference, Miami Beach, FL, USA, 27–29 October 2011; pp. 137–154.
201. Hermans, F.; Rensfelt, O.; Voigt, T.; Ngai, E.; Norden, L.-A.; Gunningberg, P. Sonic: Classifying interference in 802.15. 4 sensor networks. In Proceedings of the 12th International Conference on Information Processing in Sensor Networks, Philadelphia, PA, USA, 12–16 April 2013; pp. 55–66.
202. Zheng, X.; Cao, Z.; Wang, J.; He, Y.; Liu, Y. Zisense: Towards interference resilient duty cycling in wireless sensor networks. In Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems, Memphis, TN, USA, 3–6 November 2014; pp. 119–133.
203. Hithnawi, A.; Shafagh, H.; Duquennoy, S. Tiim: Technology-independent interference mitigation for low-power wireless networks. In Proceedings of the Proceedings of the 14th International Conference on Information Processing in Sensor Networks, Seattle, WA, USA, 14–16 April 2015; pp. 1–12.
204. Mennes, R.; Camelo, M.; Claeys, M.; Latre, S. A neural-network-based mf-tdma mac scheduler for collaborative wireless networks. In Proceedings of the 2018 IEEE Wireless Communications and Networking Conference (WCNC), Sydney, Australia, 18–21 May 2018; pp. 1–6.
205. Wang, S.; Liu, H.; Gomes, P.H.; Krishnamachari, B. Deep reinforcement learning for dynamic multichannel access in wireless networks. *IEEE Trans. Cogn. Commun. Netw.* **2018**, *4*, 257–265. [[CrossRef](#)]
206. Xu, S.; Liu, P.; Wang, R.; Panwar, S.S. Realtime scheduling and power allocation using deep neural networks. *arXiv* **2018**, arXiv:1811.07416.
207. Yu, Y.; Wang, T.; Liew, S.C. Deep-reinforcement learning multiple access for heterogeneous wireless networks. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 1277–1290. [[CrossRef](#)]
208. Zhang, Y.; Hou, J.; Towhidou, V.; Shikh-Bahaei, M. A neural network prediction based adaptive mode selection scheme in full-duplex cognitive networks. *IEEE Trans. Cogn. Commun. Netw.* **2019**, *5*, 540–553. [[CrossRef](#)]
209. Mennes, R.; Claeys, M.; Figueiredo, F.A.D.; Jabandžić, I.; Moerman, I.; Latré, S. Deep learning-based spectrum prediction collision avoidance for hybrid wireless environments. *IEEE Access* **2019**, *7*, 45818–45830. [[CrossRef](#)]
210. Liu, T.; Cerpa, A.E. Talent: Temporal adaptive link estimator with no training, In Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems, Toronto, ON, Canada, 6–9 November 2012; pp. 253–266.
211. Adeel, A.; Larjani, H.; Javed, A.; Ahmadinia, A. Critical analysis of learning algorithms in random neural network based cognitive engine for lte systems, In Proceedings of the 2015 IEEE 81st Vehicular Technology Conference (VTC Spring), Glasgow, UK, 11–14 May 2015; pp. 1–5.
212. Pierucci, L.; Micheli, D. A neural network for quality of experience estimation in mobile communications. *IEEE Multimed.* **2016**, *23*, 42–49. [[CrossRef](#)]
213. Qiao, M.; Zhao, H.; Wang, S.; Wei, J. Mac protocol selection based on machine learning in cognitive radio networks. In Proceedings of the 2016 19th International Symposium on Wireless Personal Multimedia Communications (WPMC), Shenzhen, China, 14–16 November 2016; pp. 453–458.
214. Kulin, M.; Poorter, E.D.; Kazaz, T.; Moerman, I. Poster: Towards a cognitive mac layer: Predicting the mac-level performance in dynamic wsn using machine learning. In Proceedings of the 2017 International Conference on Embedded Wireless Systems and Networks, Uppsala, Sweden, 20–22 February 2017; pp. 214–215.
215. Akbas, A.; Yildiz, H.U.; Ozbayoglu, A.M.; Tavli, B. Neural network based instant parameter prediction for wireless sensor network optimization models. *Wirel. Netw.* **2018**, *25*, 3405–3418. [[CrossRef](#)]
216. Al-Kaseem, B.R.; Al-Raweshidy, H.S.; Al-Dunainawi, Y.; Banitsas, K. A new intelligent approach for optimizing 6lowpan mac layer parameters. *IEEE Access* **2017**, *5*, 16229–16240.
217. Raca, D.; Zahran, A.H.; Sreenan, C.J.; Sinha, R.K.; Halepovic, E.; Jana, R.; Gopalakrishnan, V. On leveraging machine and deep learning for throughput prediction in cellular networks: Design, performance, and challenges. *IEEE Commun. Mag.* **2020**, *58*, 11–17. [[CrossRef](#)]
218. Rathore, M.M.; Ahmad, A.; Paul, A.; Jeon, G. Efficient graph-oriented smart transportation using internet of things generated big data. In Proceedings of the 2015 11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), Bangkok, Thailand, 23–27 November 2015; pp. 512–519.
219. Amato, G.; Carrara, F.; Falchi, F.; Gennaro, C.; Meghini, C.; Vairo, C. Deep learning for decentralized parking lot occupancy detection. *Expert Syst. Appl.* **2017**, *72*, 327–334. [[CrossRef](#)]
220. Mittal, G.; Yagnik, K.B.; Garg, M.; Krishnan, N.C. Spotgarbage: smartphone app to detect garbage using deep learning. In Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing, Heidelberg, Germany, 12–16 September 2016; pp. 940–945.
221. Gillis, J.M.; Alshareef, S.M.; Morsi, W.G. Nonintrusive load monitoring using wavelet design and machine learning. *IEEE Trans. Smart Grid* **2016**, *7*, 320–328. [[CrossRef](#)]
222. Lv, J.; Yang, W.; Man, D. Device-free passive identity identification via wifi signals. *Sensors* **2017**, *17*, 2520. [[CrossRef](#)] [[PubMed](#)]

223. Jafari, H.; Omotere, O.; Adesina, D.; Wu, H.-H.; Qian, L. IoT devices fingerprinting using deep learning. In Proceedings of the MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM), Los Angeles, CA, USA, 29–31 October 2018; pp. 1–9.
224. Merchant, K.; Revay, S.; Stantchev, G.; Nousain, B. Deep learning for rf device fingerprinting in cognitive communication networks. *IEEE J. Sel. Top. Signal Process.* **2018**, *12*, 160–167. [[CrossRef](#)]
225. Thing, V.L. IEEE 802.11 network anomaly detection and attack classification: A deep learning approach. In Proceedings of the 2017 IEEE Wireless Communications and Networking Conference (WCNC), San Francisco, CA, USA, 19–22 March 2017; pp. 1–6.
226. Uluagac, A.S.; Radhakrishnan, S.V.; Corbett, C.; Baca, A.; Beyah, R. A passive technique for fingerprinting wireless devices with wired-side observations. In Proceedings of the in: 2013 IEEE Conference on Communications and Network Security (CNS), Washington, DC, USA, 14–16 October 2013; pp. 305–313.
227. Bezawada, B.; Bachani, M.; Peterson, J.; Shirazi, H.; Ray, I. Behavioral fingerprinting of iot devices. In Proceedings of the 2018 Workshop on Attacks and Solutions in Hardware Security, Cambridge, UK, 22–23 March 2018; pp. 41–50.
228. Riyaz, S.; Sankhe, K.; Ioannidis, S.; Chowdhury, K. Deep learning convolutional neural networks for radio identification. *IEEE Commun. Mag.* **2018**, *56*, 46–152.
229. Wang, X.; Gao, L.; Mao, S. Phasefi: Phase fingerprinting for indoor localization with a deep learning approach. In Proceedings of the 2015 IEEE Global Communications Conference (GLOBECOM), San Diego, CA, USA, 6–10 December 2015; pp. 1–6.
230. Wang, X.; Gao, L.; Mao, S. Csi phase fingerprinting for indoor localization with a deep learning approach. *IEEE Internet Things J.* **2016**, *3*, 1113–1123. [[CrossRef](#)]
231. Wang, X.; Gao, L.; Mao, S.; Pandey, S. Csi-based fingerprinting for indoor localization: A deep learning approach. *IEEE Trans. Veh. Technol.* **2017**, *66*, 763–776.
232. Wang, X.; Gao, L.; Mao, S.; Pandey, S. Deepfi: Deep learning for indoor fingerprinting using channel state information. In Proceedings of the 2015 IEEE Wireless Communications and Networking Conference (WCNC), New Orleans, LA, USA, 9–12 March 2015; pp. 1666–1671.
233. Wang, J.; Zhang, X.; Gao, Q.; Yue, H.; Wang, H. Device-free wireless localization and activity recognition: A deep learning approach. *IEEE Trans. Veh. Technol.* **2017**, *66*, 6258–6267. [[CrossRef](#)]
234. Zhang, W.; Liu, K.; Zhang, W.; Zhang, Y.; Gu, J. Deep neural networks for wireless localization in indoor and outdoor environments. *Neurocomputing* **2016**, *194*, 279–287. [[CrossRef](#)]
235. Zeng, Y.; Pathak, P.H.; Mohapatra, P. Wiwho: Wifi-based person identification in smart spaces. In Proceedings of the 15th International Conference on Information Processing in Sensor Networks, Vienna, Austria, 11–14 April 2016; p. 4.
236. Zhao, M.; Li, T.; Alsheikh, M.A.; Tian, Y.; Zhao, H.; Torralba, A.; Katabi, D. Through-wall human pose estimation using radio signals. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7356–7365.
237. Lv, J.; Man, D.; Yang, W.; Gong, L.; Du, X.; Yu, M. Robust device-free intrusion detection using physical layer information of wifi signals. *Appl. Sci.* **2019**, *9*, 175. [[CrossRef](#)]
238. Shahzad, M.; Zhang, S. Augmenting user identification with wifi based gesture recognition. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **2018**, *2*, 134.
239. Wang, W.; Liu, A.X.; Shahzad, M.; Ling, K.; Lu, S. Understanding and modeling of wifi signal based human activity recognition. In Proceedings of the 21st Annual International Conference on Mobile Computing and Networking, Paris, France, 7–11 September 2015; pp. 65–76.
240. Ozdemir, O.; Li, R.; Varshney, P.K. Hybrid maximum likelihood modulation classification using multiple radios. *IEEE Commun. Lett.* **2013**, *17*, 1889–1892. [[CrossRef](#)]
241. Ozdemir, O.; Wimalajeewa, T.; Dulek, B.; Varshney, P.K.; Su, W. Asynchronous linear modulation classification with multiple sensors via generalized em algorithm. *IEEE Trans. Wirel. Commun.* **2015**, *14*, 6389–6400. [[CrossRef](#)]
242. Wimalajeewa, T.; Jagannath, J.; Varshney, P.K.; Drozd, A.; Su, W. Distributed asynchronous modulation classification based on hybrid maximum likelihood approach. In Proceedings of the MILCOM 2015-2015 IEEE Military Communications Conference, Tampa, FL, USA, 26–28 October 2015; pp. 1519–1523.
243. Azzouz, E.; Nandi, A.K. *Automatic Modulation Recognition of Communication Signals*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013.
244. Alharbi, H.; Mobien, S.; Alshebeili, S.; Alturki, F. Automatic modulation classification of digital modulations in presence of hf noise. *EURASIP J. Adv. Signal Process.* **2012**, *2012*, 238. [[CrossRef](#)]
245. Chepuri, S.P.; Francisco, R.D.; Leus, G. Performance evaluation of an IEEE 802.15. 4 cognitive radio link in the 2360–2400 mhz band. In Proceedings of the 2011 IEEE Wireless Communications and Networking Conference, Cancun, Mexico, 28–31 March 2011; pp. 2155–2160.
246. Dobre, O.A.; Abdi, A.; Bar-Ness, Y.; Su, W. Survey of automatic modulation classification techniques: Classical approaches and new trends. *IET Commun.* **2007**, *1*, 137–156. [[CrossRef](#)]
247. Kulin, M.; Kazaz, T.; Moerman, I.; Poorter, E.D. End-to-end learning from spectrum data: A deep learning approach for wireless signal identification in spectrum monitoring applications. *IEEE Access* **2018**, *6*, 18484–18501. [[CrossRef](#)]
248. Selim, A.; Paisana, F.; Arokiam, J.A.; Zhang, Y.; Doyle, L.; DaSilva, L.A. Spectrum monitoring for radar bands using deep convolutional neural networks. *arXiv* **2017**, arXiv:1705.00462.

249. Akyildiz, I.F.; Lee, W.-Y.; Vuran, M.C.; Mohanty, S. A survey on spectrum management in cognitive radio networks. *IEEE Commun. Mag.* **2008**, *46*, 40–48. [[CrossRef](#)]
250. Ghasemzadeh, P.; Banerjee, S.; Hempel, M.; Sharif, H. Performance evaluation of feature-based automatic modulation classification. In Proceedings of the 2018 12th International Conference on Signal Processing and Communication Systems (ICSPCS), Cairns, Australia, 17–19 December 2018; pp. 1–5.
251. Hu, S.; Pei, Y.; Liang, P.P.; Liang, Y.-C. Robust modulation classification under uncertain noise condition using recurrent neural network. In Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, UAE, 9–13 December 2018; pp. 1–7.
252. Zhang, X.; Seyfi, T.; Ju, S.; Ramjee, S.; Gamal, A.E.; Eldar, Y.C. Deep learning for interference identification: Band, training snr, and sample selection. *arXiv* **2019**, arXiv:1905.08054.
253. Shahid, A.; Fontaine, J.; Camelo, M.; Haxhibeqiri, J.; Saelens, M.; Khan, Z.; Moerman, I.; Poorter, E.D. A convolutional neural network approach for classification of lpwan technologies: Sigfox, lora and ieee 802.15. 4g. In Proceedings of the 2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), Boston, MA, USA, 10–13 June 2019; pp. 1–8.
254. Tekbiyik, K.; Akbunar, I.A.R.E.; In, A.G.; Kurt, G.K. Multi-dimensional wireless signal identification based on support vector machines. *IEEE Access* **2019**, *7*, 138890–138903. [[CrossRef](#)]
255. Isolani, P.H.; Claeys, M.; Donato, C.; Granville, L.Z.; Latré, S. A survey on the programmability of wireless mac protocols. *IEEE Commun. Surv. Tutor.* **2018**, *21*, 1064–1092. [[CrossRef](#)]
256. Cordeiro, C.; Challapali, K. C-mac: A cognitive mac protocol for multi-channel wireless networks. In Proceedings of the 2007 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, Dublin, Ireland, 17–20 April 2007; pp. 147–157.
257. Haddad, M.; Muhlethaler, P.; Laouiti, A.; Zagrouba, R.; Saidane, L.A. Tdma-based mac protocols for vehicular ad hoc networks: A survey, qualitative analysis, and open research issues. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 2461–2492. [[CrossRef](#)]
258. Lien, S.-Y.; Tseng, C.-C.; Chen, K.-C. Carrier sensing based multiple access protocols for cognitive radio networks. In Proceedings of the 2008 IEEE International Conference on Communications, Honolulu, HI, USA, 25–27 May 2008; pp. 3208–3214.
259. Jain, N.; Das, S.R.; Nasipuri, A. A multichannel csma mac protocol with receiver-based channel selection for multihop wireless networks. In Proceedings of the Tenth International Conference on Computer Communications and Networks (Cat. No. 01EX495), Scottsdale, Arizona, 15–17 October 2001; pp. 432–439.
260. Muqattash, A.; Krunz, M. Cdma-based mac protocol for wireless ad hoc networks. In Proceedings of the 4th ACM International Symposium on Mobile AD Hoc Networking & Computing, Annapolis, MD, USA, 1–3 June 2003; pp. 153–164.
261. Kumar, S.; Raghavan, V.S.; Deng, J. Medium access control protocols for ad hoc wireless networks: A survey. *Ad Hoc Netw.* **2006**, *4*, 326–358. [[CrossRef](#)]
262. Sitanayah, L.; Sreenan, C.J.; Brown, K.N. Er-mac: A hybrid mac protocol for emergency response wireless sensor networks. In Proceedings of the 2010 Fourth International Conference on Sensor Technologies and Applications, Venice, Italy, 18–25 July 2010; pp. 244–249.
263. Su, H.; Zhang, X. Opportunistic mac protocols for cognitive radio based wireless networks. In Proceedings of the 2007 41st Annual Conference on Information Sciences and Systems, Baltimore, MD, USA, 14–16 March 2007; pp. 363–368.
264. Keerthi, S.S.; Shevade, S.K.; Bhattacharyya, C.; Murthy, K.R.K. Improvements to platt's smo algorithm for svm classifier design. *Neural Comput.* **2001**, *13*, 637–649. [[CrossRef](#)]
265. Wang, H.; Xu, F.; Li, Y.; Zhang, P.; Jin, D. Understanding mobile traffic patterns of large scale cellular towers in urban environment. In Proceedings of the 2015 Internet Measurement Conference, Tokyo, Japan, 28–30 October 2015; pp. 225–238.
266. Hu, J.; Heng, W.; Zhang, G.; Meng, C. Base station sleeping mechanism based on traffic prediction in heterogeneous networks. In Proceedings of the 2015 International Telecommunication Networks and Applications Conference (ITNAC), Sydney, Australia, 18–20 November 2015; pp. 83–87.
267. Zang, Y.; Ni, F.; Feng, Z.; Cui, S.; Ding, Z. Wavelet transform processing for cellular traffic prediction in machine learning networks. In Proceedings of the 2015 IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP), Chengdu, China, 12–15 July 2015; pp. 458–462.
268. Xu, F.; Lin, Y.; Huang, J.; Wu, D.; Shi, H.; Song, J.; Li, Y. Big data driven mobile traffic understanding and forecasting: A time series approach. *IEEE Trans. Serv. Comput.* **2016**, *9*, 796–805. [[CrossRef](#)]
269. Nikravesh, A.Y.; Ajila, S.A.; Lung, C.-H.; Ding, W. Mobile network traffic prediction using mlp, mlpwd, and svm. In Proceedings of the 2016 IEEE International Congress on Big Data (BigData Congress), San Francisco, CA, USA, 27 June–2 July 2016; pp. 402–409.
270. Wang, J.; Tang, J.; Xu, Z.; Wang, Y.; Xue, G.; Zhang, X.; Yang, D. Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach. In Proceedings of the IEEE INFOCOM 2017–IEEE Conference on Computer Communications, Atlanta, GA, USA, 1–4 May 2017; pp. 1–9.
271. Huang, C.-W.; Chiang, C.-T.; Li, Q. A study of deep learning networks on mobile traffic forecasting. In Proceedings of the 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Montreal, QC, Canada, 8–13 October 2017; pp. 1–6.

272. Zhang, C.; Patras, P. Long-term mobile traffic forecasting using deep spatio-temporal neural networks. In Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing, Los Angeles, CA, USA, 26 June 2018; pp. 231–240.
273. Wang, X.; Zhou, Z.; Xiao, F.; Xing, K.; Yang, Z.; Liu, Y.; Peng, C. Spatio-temporal analysis and prediction of cellular traffic in metropolis. *IEEE Trans. Mob. Comput.* **2018**, *18*, 2190–2202. [[CrossRef](#)]
274. Alawe, I.; Ksentini, A.; Hadjadj-Aoul, Y.; Bertin, P. Improving traffic forecasting for 5g core network scalability: A machine learning approach. *IEEE Netw.* **2018**, *32*, 42–49. [[CrossRef](#)]
275. Zhang, C.; Zhang, H.; Yuan, D.; Zhang, M. Citywide cellular traffic prediction based on densely connected convolutional neural networks. *IEEE Commun. Lett.* **2018**, *22*, 1656–1659. [[CrossRef](#)]
276. Feng, J.; Chen, X.; Gao, R.; Zeng, M.; Li, Y. Deepptp: An end-to-end neural network for mobile cellular traffic prediction. *IEEE Netw.* **2018**, *32*, 108–115. [[CrossRef](#)]
277. Yamada, Y.; Shinkuma, R.; Sato, T.; Oki, E. Feature-selection based data prioritization in mobile traffic prediction using machine learning. In Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, UAE, 9–13 December 2018; pp. 1–6.
278. Hua, Y.; Zhao, Z.; Liu, Z.; Chen, X.; Li, R.; Zhang, H. Traffic prediction based on random connectivity in deep learning with long short-term memory. In Proceedings of the 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall), Porto, Portugal, 3–6 June 2019; pp. 1–6.
279. Fang, L.; Cheng, X.; Wang, H.; Yang, L. Idle time window prediction in cellular networks with deep spatiotemporal modeling. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 1441–1454
280. Zhang, H.; Liu, N.; Chu, X.; Long, K.; Aghvami, A.-H.; Leung, V.C. Network slicing based 5g and future mobile networks: Mobility, resource management, and challenges. *IEEE Commun. Mag.* **2017**, *55*, 138–145.
281. Iqbal, M.H.; Soomro, T.R. Big data analysis: Apache storm perspective. *Int. J. Comput. Trends Technol.* **2015**, *19*, 9–14. [[CrossRef](#)]
282. Zaharia, M.; Xin, R.S.; Wendell, P.; Das, T.; Armbrust, M.; Dave, A.; Meng, X.; Rosen, J.; Venkataraman, S.; Franklin, M.J.; et al. Apache spark: A unified engine for big data processing. *Commun. ACM* **2016**, *59*, 56–65. [[CrossRef](#)]
283. Ranjan, R. Streaming big data processing in datacenter clouds. *IEEE Cloud Comput.* **2014**, *1*, 78–83. [[CrossRef](#)]
284. Dittrich, J.; Quiané-Ruiz, J.-A. Efficient big data processing in hadoop mapreduce. *Proc. VLDB Endow.* **2012**, *5*, 2014–2015. [[CrossRef](#)]
285. Mahdavinejad, M.S.; Rezvan, M.; Barekatain, M.; Adibi, P.; Barnaghi, P.; Sheth, A.P. Machine learning for internet of things data analysis: A survey. *Digit. Commun. Netw.* **2018**, *4*, 161–175. [[CrossRef](#)]
286. O’shea, T.J.; West, N. Radio machine learning dataset generation with gnu radio. In Proceedings of the GNU Radio Conference, Boulder, CO, USA, 12–16 September 2016; Volume 1.
287. Rajendran, S.; Calvo-Palomino, R.; Fuchs, M.; den Bergh, B.V.; Cordobés, H.; Giustiniano, D.; Pollin, S.; Lenders, V. Electrosense: Open and big spectrum data. *IEEE Commun. Mag.* **2018**, *56*, 210–217. [[CrossRef](#)]
288. Lane, N.D.; Bhattacharya, S.; Mathur, A.; Georgiev, P.; Forlivesi, C.; Kawzar, F. Squeezing deep learning into mobile and embedded devices. *IEEE Pervasive Comput.* **2017**, *16*, 82–88. [[CrossRef](#)]
289. Han, S.; Kang, J.; Mao, H.; Hu, Y.; Li, X.; Li, Y.; Xie, D.; Luo, H.; Yao, S.; Wang, Y.; et al. Ese: Efficient speech recognition engine with sparse lstm on fpga. In Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, Monterey, CA, USA, 22–24 February 2017; pp. 75–84.
290. McDanel, B.; Teerapittayanon, S.; Kung, H. Embedded binarized neural networks, *arXiv* **2017**, arXiv:1709.02260.
291. Kreutz, D.; Ramos, F.M.; Verissimo, P.E.; Rothenberg, C.E.; Azodolmolky, S.; Uhlig, S. Software-defined networking: A comprehensive survey. *Proc. IEEE* **2014**, *103*, 14–76. [[CrossRef](#)]
292. Xie, J.; Yu, F.R.; Huang, T.; Xie, R.; Liu, J.; Wang, C.; Liu, Y. A survey of machine learning techniques applied to software defined networking (sdn): Research issues and challenges. *IEEE Commun. Surv. Tutor.* **2018**, *21*, 393–430. [[CrossRef](#)]
293. Baştug, E.; Bennis, M.; Debbah, M. A transfer learning approach for cache-enabled wireless networks. In Proceedings of the 2015 13th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), Mumbai, India, 25–29 May 2015; pp. 161–166.
294. Yang, K.; Ren, J.; Zhu, Y.; Zhang, W. Active learning for wireless iot intrusion detection. *IEEE Wirel. Commun.* **2018**, *25*, 19–25. [[CrossRef](#)]
295. O’Shea, T.J.; Corgan, J.; Clancy, T.C. Unsupervised representation learning of structured radio communication signals. In Proceedings of the 2016 First International Workshop on Sensing, Processing and Learning for Intelligent Machines (SPLINE), Aalborg, Denmark, 6–8 July 2016; pp. 1–5.