Received: 16 May 2022 | Revised: 23 September 2022 | Accepted: 1 October 2022

DOI: 10.1049/rsn2.12337

IET Radar, Sonar & Navigation

The Institution of Engineering and Technology WILEY

REVIEW

Artificial intelligence meets radar resource management: A comprehensive background and literature review

Umair Sajid Hashmi[1,2] | Sunila Akbar[1] | Raviraj Adve[1] | Peter W. Moo[2] | Jack Ding[2]

[1]Department of Electrical and Computer Engineering, University of Toronto, OntarioToronto, Canada

[2]Radar Sensing and Exploitation Section, Defence Research and Development Canada, Ottawa Research Centre, OntarioOttawa, Canada

Correspondence

Umair Sajid Hashmi, Department of Electrical and Computer Engineering, University of Toronto, Toronto, Ontario, Canada.
Email: umair.hashmi@utoronto.ca; umair.hashmi@seecs.edu.pk

Funding information

Defence Research and Development Canada, Grant/Award Number: W7714-207097

Abstract

A multi-function radar is designed to perform disparate functions, such as surveillance, tracking, fire control, amongst others, within a limited resource (time, frequency, and energy) budget. A radar resource management (RRM) module within a radar system makes decisions on prioritisation, parameter selection, and scheduling of associated tasks. However, optimal RRM algorithms are generally computationally complex and operational radars resort to heuristics. On the other hand, algorithms based on artificial intelligence (AI) have been shown to yield near-optimal radar resource allocation results at manageable computational complexity. This survey study aims at enabling researchers and practitioners better understand the application of AI in RRM-related problems by providing a thorough literature review of AI-based RRM techniques. We first provide background concepts in RRM followed by a brief review of Symbolic-AI techniques for RRM. We mainly focus on the applications of state-of-the-art machine learning techniques to RRM. We emphasise on the recent findings and their potential within practical RRM scenarios for real-time resource allocation optimisation. The study concludes with a discussion of open research problems and future research directions in the light of the presented survey.

1 | INTRODUCTION

The operating principle of radio detection and ranging, commonly known as RADAR, starts with the transmission of an electromagnetic (EM) wave towards a potential target, scattering of the incident EM wave by the target, reception of the scattered signal by a receiver terminal and subsequent signal processing of the received energy to extract meaningful information regarding the target. Initially designed for military applications back during the World War II era, radar has found its way to myriad applications, whether military (e.g. multi-target tracking), security-related (e.g. through-the-wall detection and tracking), or civilian (e.g. bio-medical and automotive radar). There are three basic radar functions, namely: (i) searching, (ii) tracking, and (iii) imaging. During search operation, the radar systems attempt to detect and acquire targets of interest by scanning in a preset range of elevation and/or azimuth angles. To track a moving target, the radar detects the target multiple times, acquiring observations of the target state in range, azimuth/elevation angle. After detection and tracking of a target, the imaging mode may be activated that develops information regarding the target in terms of size, shape, azimuth, elevation, and speed of the target [1].

Multi-function radars (MFRs) [2], a relatively new development in radar systems, are capable of executing multiple radar functions simultaneously, such as surveillance, multiple target tracking, waveform generation, and electronic beam steering [3, 4]. However, all radars, indeed systems, have limited resources; in the context of a radar, the key resources are time, energy, frequency, and computation. When required to execute multiple tasks simultaneously, these resources must be allocated to the tasks in some structured manner. As such, radar resource management (RRM) [5] requires task prioritisation and scheduling, parameter selection, and resource allocation within an MFR [6]. Effective RRM is particularly important when the radar is overloaded with tasks, that is, a proper execution of all assigned tasks requires more resources than available. The basic premise behind RRM therefore revolves

This is an open access article under the terms of the Creative Commons Attribution-NoDerivs License, which permits use and distribution in any medium, provided the original work is properly cited and no modifications or adaptations are made.

© 2022 The Authors. *IET Radar, Sonar & Navigation* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology.

around optimisation and compromising between MFR tasks. With optimisation, the goal is to find a way to allocate resources in as efficient a manner as possible. In requiring some compromise, certain tasks are deemed more critical and are, hence, allocated resources before other tasks [7]. Indeed, some lower-priority tasks may be dropped in order to execute other, critical, tasks.

In the first phase of RRM, task parameters, such as the priority, dwell time, and revisit interval are determined by heuristics [8] or joint optimisation techniques, under strict resource constraints [9]. The priority assignment and takes place at the situation level, while the parameter optimisation is carried out at an object level [4] following the Joint Director of Laboratories data fusion model [10] for resource management. In the second phase, task scheduling to determine the exact time and order of task execution is performed at the measurement level, such that as many tasks as possible are accommodated on the radar timeline without imposing significant delays.

Task scheduling may be performed using queue- or frame-based schedulers [11]. Queue-based schedulers execute tasks from an ordered list based on some criteria, such as earliest start time (EST) and earliest deadline first schedulers [6]. On the other hand, frame-based schedulers estimate the optimal tasks to be executed on a frame-by-frame basis using a variety of heuristics [12] or, of importance here, machine learning (ML)-based algorithms [13]. The RRM scheduling algorithms can be classified as adaptive or non-adaptive algorithms. In contrast to adaptive algorithms, where task prioritisation and scheduling is performed to optimise radar performance radar in a dynamically changing environment, the non-adaptive algorithms have predefined task priorities and the task scheduling is performed using some preset heuristic rules without any optimisation [6].

As modern radars must execute increasingly complex tasks, a recent research thrust has been the development of *cognitive* radar, that is, a computational system that learns from the environment and past actions to improve performance [14]. Importantly, a cognitive radar, having learnt from its past performance and being environmentally aware, would make near-optimal decisions computationally efficiently. While the initial proposal in ref. [14] was conceptual, the recent exponential growth in the use of ML techniques makes it now possible to implement a *cognitive* radar.

Artificial intelligence (AI) has found applications in many diverse fields, such as wireless communications [15–18], speech signal processing, computer vision, and natural language processing [19] to name just a few. Artificial intelligence algorithms have multiple domains, like logic programming, recommender systems, and ML [20], to name a few. It can broadly be classified as Symbolic-AI and ML, the former is based on symbolic reasoning through human intervention such as rules engines, expert systems, and knowledge graphs, whereas the later is based on learning from data, identifying patterns, and making decisions with minimal human intervention [21]. In recent years, Defence Advanced Research Projects Agency has initiated many projects related to ML applications in radar, such as the radio frequency ML system project [22], the behaviour learning for adaptive electronic warfare project [23], and the adaptive radar countermeasures project [24]. Applications of ML for radar-based applications include emitter recognition and classification [25, 26], image processing [27, 28], image denoising [29, 30], automatic target reconstruction [31, 32], target detection [33, 34], anti-jamming [35], optimal waveform design [36], and array antenna selection [37]. Some of the ML-based algorithms used in such applications include the traditional ML techniques, such as decision trees (DT), support vector machines (SVM), K-means algorithm, and random forests (RF). Some noteworthy Deep learning (DL) techniques include convolutional neural networks (CNN), auto encoders (AE), deep belief networks, recurrent neural networks (RNN), and generative adversarial networks (GAN). A more detailed description on these algorithms will be provided in the following sections of this paper. For a detailed review of the application of ML concepts in the wireless communications domain, readers can see refs. [38–41].

1.1 | Contributions and organisation

Recognising the broad applicability of ML techniques in different domains, the radar research community has also started applying ML-based algorithms to RRM tasks apart from the traditional Symbolic-AI techniques. Since these works are relatively recent, a thorough and systematic survey of the literature is yet to be carried out in this domain. Our work in this paper fills this gap by providing an extensive overview of the existing literature in ML applications in RRM, while also highlighting some of the key areas which demand attention by the radar research community. Some relevant survey papers (summarised in Table 1) have discussed RRM in joint radar and communications (JRC) [42], ML applications in radar signal processing [43], and an overview of RRM algorithms [44]. A survey of the research in AI with a particular focus on ML for RRM is still a Terra incognita and addressed in this work. In short, the contributions of this paper are as follows:

- We provide fundamental knowledge of RRM with some basic concepts using RRM model. In addition, we briefly discuss RRM for networked and cognitive radar along with important metrics to quantify the performance in RRM domain (Section 2).

TABLE 1 Comparisons with other recent survey papers

Ref	RRM	ML	Topic
[42]	✓		RRM in JRC
[43]		✓	ML in radar signal processing
[44]	✓		Overview of RRM algorithms
This paper	✓	✓	ML for RRM algorithms

Abbreviations: RRM, Radar resource management; ML, machine learning.

- We provide a brief review of some notable RRM works in the Symbolic-AI domain which serve as the baseline results for the more recent ML-based RRM (Section 3).
- We present a comprehensive review of ML as applied to the RRM problem in radar. The discussion will include analysis with regard to the RRM tasks, which include task scheduling, time resource management, target tracking, target classification, spectrum allocation and quality of service (QoS) resource management. We will explain how researchers have used ML techniques for these tasks, and the associated pros and cons of using these techniques (Section 4).
- In addition, for completeness, we review select recent literature on AI specifically ML applications to non-RRM related tasks in radar. Some of the use cases include unmanned aerial vehicle detection, radar-based surveillance, waveform synthesis and recognition, and medical imaging (Section 5).
- Finally, we highlight challenges and discuss potential research directions to ML-based radar. In particular, we make a case for reinforcement learning (RL) and how it will be useful for RRM-based use cases (Section 6).

While the foundation topics cover more traditional benchmark works in Symbolic-AI domain, the latter half of our article reviews state-of-the-art achievements associated with ML-based applications in RRM from public databases such as IEEE Xplore and IET over the duration of last 4–5 years. In particular, we focus on papers from the IEEE International Radar Conference, the IEEE Radar Conference, Asilomar, IEEE Transactions on Aerospace and Electronic Systems, IEEE Aerospace and Electronic Systems Magazine, and IET Transactions on Radar Sonar and Navigation. We hope that our paper will help researchers and professionals in the radar domain identify research gaps and conduct meaningful work in this important area.

2 | WHAT IS RADAR RESOURCE MANAGEMENT?

In simple terms, RRM encompasses the process involved in allocating radar resources to achieve its desired tasks. Some of the most common radar functions are tracking, surveillance, target acquisition, and detection. A radar may perform these tasks by controlling its beam position, dwell time, waveform, and energy [6]. Every radar has a finite resource budget, which may be broadly categorised as time, energy, and processing budgets, where time resources are generic to the tactical task requirements, energy resources are mainly constrained by the available transmitter energy/power, while the computation resources are dependent on the RRM processing unit [44]. All these resource constraints can have an adverse effect on the overall performance of the RRM unit in the radar [45]. The time limitation is especially critical as the radar cannot create an additional timeline.

2.1 | Radar resource management model

A model for RRM will necessarily be complex due to the nature of RRM for MFRs. A general RRM system model is shown in Figure 1. The radar mission is carried out via multiple functions, such as surveillance and tracking. To complete radar functions, one or more tasks per function must be performed; for example, a surveillance function may comprise a single task of monitoring the entire region or it may comprise multiple tasks of monitoring subregions within the specified region of interest. The tasks are prioritised first using a prioritisation algorithm before moving on to task scheduling. The role of scheduling is to coordinate the usage of all radar resources in carrying out RRM so that the system can efficiently fulfil the requirements of as many radar functions as possible.

Task prioritisation is an important factor in RRM, which gives a larger weight to the more urgent tasks as compared to tasks which may be delayed. As a result, lower priority tasks may encounter degraded performance due to fewer available resources, or in the worst case, may be dropped. Task scheduling may be performed simultaneously with the task prioritisation function and the performance can be improved significantly using adaptive techniques which can modify the process in response to changing environments and the target scenario.

2.2 | Radar resource management for networked radars

In RRM for networked radars, where multiple radars are used in an overall cooperative system, coordinated RRM is required

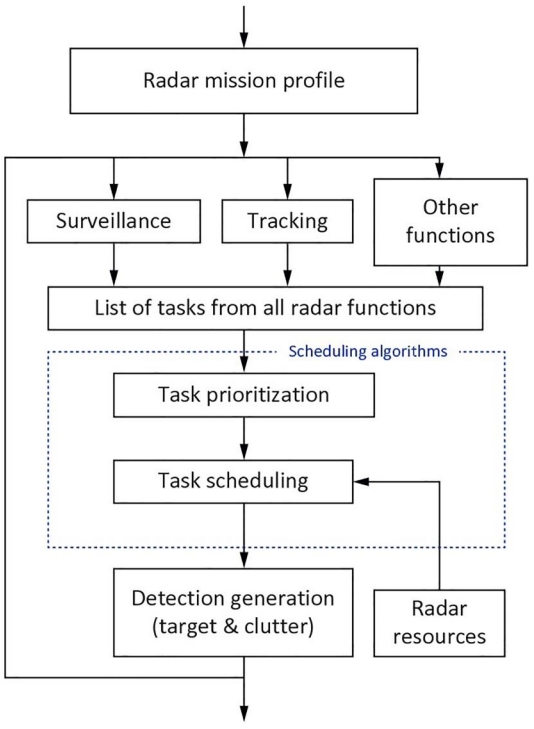

FIGURE 1 A radar resource management (RRM) model [6].

to schedule tracking and surveillance tasks, data processing from all radars and specification setting for distributed tracking. As such, it forms a time-varying multidimensional optimisation problem. It was shown in ref. [6] that the use of coordinated RRM offers the potential for significant performance improvements as compared to independent RRM. In this work, at one extreme, the independent RRM, also referred as Type 0 management, carries out all RRM tasks independently, without the data exchange between radars. Every every radar utilises individual adaptivity, such as fuzzy logic prioritisation [46], adaptive track update intervals, and time-balancing scheduling. On the other hand, coordinated RRM assigns overlapping tracking tasks to the radar with a smaller tracking range in Type 1 management. In Type 2 management, the assignment is done on a look-by-look basis, that is, on each look, the radar with the smallest range is assigned the tracking tasks. Since comparison of target ranges in conducted on every look, Type 2 management is computationally intensive as compared to the Type 1 radar management.

Different forms of distributed tracking are performed to improve the tracking capabilities of the radar [47]. These include: (i) independent tracking, (ii) distributed track fusion, and (iii) distributed track maintenance. As the name suggests, independent tracking involves independent tracking by each radar where tracks are initiated and maintained separately. In distributed track fusion, data fusion is carried out via track-to-track association to remove redundant tracks. Finally, in distributed track maintenance, measurement-to-track data association is conducted for measurements from the radars within the network.

2.3 | Radar resource management for cognitive radar

A cognitive radar is defined as "a radar system that acquires knowledge and understanding of its operating environment through online estimation, reasoning and learning or from databases comprising context information, and exploits this acquired knowledge and understanding to enhance information extraction, data processing and radar management" [5].

As such, RRM for the cognitive radar requires the agile use of resources while getting feedback and learning from the radio environment. A cognitive radar may adapt its degrees of freedom (DoF) in order to optimise its performance. In practice, RRM is performed dynamically in order to obtain the best achievable performance for the current radar tasks. Moreover, multiple tasks may be performed simultaneously and the resource allocation among the tasks can be optimised based on the acquired awareness.

The work in ref. [48] presented an overview of RRM for adaptive and cognitive radar. The techniques discussed represent the increased adaptation to information or knowledge, which can be built upon in the context of a cognitive framework, for example, through stochastic control. It was further discussed that RL techniques can be employed to learn online if the state transition, observation, and reward models are not known for stochastic control.

2.4 | Radar resource management performance metrics

In this sub-section, we will shed light on some of the quantitative measures to evaluate the RRM capabilities of a radar. Since RRM involves many components of radars, therefore, its performance is generally evaluated by overall radar performance. Specifically, the performance is evaluated with respect to the scheduler, detector, and tracker for the two primary MFR functions; detection and tracking. When we talk about the scheduler, the performance can be measured in terms of two different kinds of delays: maximum delay, which is the largest delay of all scheduled beams, and accumulated delay, which is the summation of delays of all scheduled beams. Another measure is ratio of scheduling, which is the ratio of the number of scheduled beams to the total number of beams of the radar mission. Some other metrics are surveillance and tracking occupancies, which are the ratios of the surveillance time and tracking time to the total time respectively.

An MFR's detection capabilities are measured in terms of the probability of detection of a given target, and by the frame time, which is the revisit time of the first detection beam position. The frame time can be defined for a specific region as per the region priority.

Tracker function performance metrics include target indication accuracy (TIA) which measure the error between the true target positions and the estimated track positions for range, azimuth and elevation. Similarly, aggregate TIAs per target is obtained by taking the mean and standard deviation of the target indication accuracies for individual targets. On the other hand, the aggregate TIAs for all targets is obtained by taking the geometric mean of all individual target TIA means. This value represents the performance of the tracker for all targets. The aggregates are measured for range, azimuth, and elevation.

Other measures for measuring RRM performance are track completeness, track occupancy, and frame time [49]. Track completeness is the ratio of the time interval over which any track number is assigned to a target and the total time that the target is in the coverage area of radar. Track occupancy is the fraction of time the radar is transmitting or receiving tracking related signals. Frame time is the duration between surveillance searches in a given spatial region.

One other metric is track continuity which represents the number of track breakups within a specific time period of evaluation for the same known object. Another metric is false track rate measuring the average number of false tracks per unit time, where false tracks are any tracks not associated with a known object. Finally, Single Integrated Air Picture has also been used representing a coherent picture of multiple track metrics, including accuracy, completeness, ambiguity, continuity, timeliness, and cross-platform commonality [50].

3 | SYMBOLIC-AI FOR RADAR RESOURCE MANAGEMENT

This section briefly reviews some of the RRM algorithms which are based on the AI methods with high-level symbolic representations of problems, logic and search. The Symbolic-AI based RRM methods are divided into six categories: (i) Fuzzy Logic Algorithms, (ii) Information Theoretic Methods, (iii) Dynamic programming (DP), (iv) QoS-based Resource Allocation Model (Q-RAM), (v) Waveform-aided algorithms, and (vi) Adaptive Update Rate algorithms.

3.1 | Fuzzy logic algorithms

A fuzzy logic controller is computationally efficient and is, therefore, well suited to perform prioritisation tasks in a radar task scheduler. A fuzzy logic processing unit comprises three steps: (i) fuzzification, (ii) fuzzy rules, and (iii) de-fuzzification. Since there could be conflicting tasks in a radar scheduler, fuzzy logic enables conflict resolution by assigning vague values as target priority factors. In shared resources, fuzzy logic allows a degree of flexibility in the tasks for efficient resource assignment. Multiple research works have proposed the use of fuzzy-based logic for task prioritisation and scheduling in radar. For instance, a decision tree architecture with five fuzzy variables (track quality, hostile, weapon systems, threat, and position) is proposed in refs. [45, 46] for prioritising radar tasks. In another publication, a dynamic fuzzy logic approach is proposed for waveform selection and energy management in a radar system simulation test-bed [51].

3.2 | Information theoretic methods

Established by Claude Shannon [52], the study of information theory has had an enormous impact on science in general; and on communications, signal processing, and control in particular. The principal advantage of the information theoretic approach in sensor management is that it simplifies system design by separating it into two independent tasks: information collection and risk/reward optimisation [53]. It has been suggested in ref. [54] that information theoretic methods can provide gains in a multitude of performance criteria in a direct manner, which makes it suitable for application to multi-function RRM. The work in ref. [55] introduced information theoretic measures relevant to RRM, which have been shown to be suitable for controlling the scheduling of track updates.

A key measure in information theory is entropy, which measures the system's disorder, or an indication of a transition from a stable to a chaotic condition. This indicator can be used for scheduling [56] or resource allocation [57]. In radar systems, the concept of entropy for RRM was first proposed in ref. [58]. The authors utilised the uncertainty aspect for radar systems with time and resource constraints. In particular, the application task was determining the location of targets and updating tracks using a single multifunction phased array radar. The proposed methodology uses a formulated entropy measure to balance resources allocated to each task. In real systems, an adaptive filter is required for more accurate determination of the entropy measure and consequentially more reliable performance.

3.3 | Dynamic programming

Dynamic programming is a popular scheme for resource allocation in multi-stage optimisation problems. The procedure starts with splitting the optimisation problem into a number of sub-problems. Then, a recursive relationship is established for optimality and a decision is taken whether to follow a forward or backward approach to solve the problem. After performing the required calculations, the optimal policy for each stage is found which, in turn, yields the overall optimal policy. In RRM, DP algorithms address both task prioritisation and scheduling simultaneously. In ref. [59], the authors deployed a DP algorithm to minimise target tracking error with phased array radars. A multi-arm bandit problem with hidden Markov models is employed in ref. [60] for optimal beam scheduling in electronically scanned array tracking systems. Another paper [61] proposed a DP-based solution to update the scheduling of search tasks in a phased array radar system. Although DP has been used in the literature extensively for optimisation of radar configurations and parameter dimensions, it is accompanied by high computational complexity which makes its practical implementation difficult.

3.4 | QoS-based Resource Allocation Model algorithms

The Q-RAM is an analytical approach to satisfy multiple simultaneous QoS measures in a resource-constrained environment. Using this model, available resources are appropriately distributed across multiple tasks so that a chosen net utility function is maximised. This approach also allows for a tradeoff between the multiple objectives within the system. In the RRM context, Q-RAM is optimised to maintain an acceptable level of QoS modelled as a cost function. The mathematical formulation is such that a system utility function based on QoS is maximised within the resource constraints. A Q-RAM framework for RRM is presented in ref. [62], consisting of a schedulability envelope, Q-RAM unit and template-based scheduler. The Q-RAM block serves as a resource allocation unit that employs fast convex optimisation to assign parameters to radar tasks, considering factors such as task importance and current utilisation level. Radar QoS optimisation was based on an earlier work on Q-RAM [63] and used initially for adaptive QoS middleware for QoS-based resource allocation and schedulability analysis [64]. A reservation-based task scheduling mechanism is proposed in ref. [65] which guarantees the performance requirements for real-time radars. Another related work is [66] which presents a template-based scheduling algorithm that constructs a set of templates

offline while considering both the timing and power constraints. In ref. [67], a dynamic Q-RAM scheme was presented for a radar tracking application incorporating the physical and environmental factors that influence the QoS of the tasks. The Q-RAM approach presented in ref. [68] showed how time-based constraints could be modelled as a utilisation to enable the use of resource management techniques. Moreover, the optimisation time was shown to be reduced on highly configurable tasks such as those in radar tracking applications.

3.5 | Waveform-aided algorithms

A radar waveform extracts desired information from the illuminated environment in terms of time, frequency, space, polarisation, and modulation. It can be continuous wave or pulsed. In terms of the task scheduling and task prioritisation functions within a radar, intelligent waveform selection improves resource management efficiency. Different waveforms optimise the surveillance, detection, tracking, and classification operations in a radar [6].

A probabilistic data association scheme is introduced in ref. [69], to select optimal waveform parameters that minimise the average total mean square tracking error at each time step. Similarly, another waveform-aided interacting multiple model (IMM) is presented in ref. [70] with the objective of selecting a waveform that decreases the uncertainty for an arbitrary target of interest, based on the maximisation of the expected information obtained about the dynamical model of the target. A beam and waveform-scheduling tracker is proposed in ref. [71], which studied practical methods for achieving a unification of surveillance and tracking for RRM. The method introduced the judicious placement of a permanent virtual agent in the field of view of the radar, which led to the name "Paranoid Tracker".

Other waveform-aided detection, tracking, and classification approaches include [72–74]. An adaptive waveform scheduling approach to detect new targets in the context of finite horizon stochastic DP is proposed in ref. [72], which minimises the time taken to detect new targets with minimal use of radar resources. An algorithm to minimise the tracking errors was proposed by Scala et al. [73]. It is reported in ref. [74] that radar waveforms can be designed to discriminate between targets by maximising the Kullback–Leibler information number that measures of dissimilarity between the observed target and the alternative targets. It is shown that the resulting choice of signal waveforms can produce significant gains in detection performance.

3.6 | Adaptive update rate algorithm

Adaptive selection of sampling time interval has shown to improve the tracking performance of a phased array radar [75]. This is because a high update rate is suitable for a manoeuvering target and a lower update rate is mostly chosen for non-manoeuvering motions. A single update rate is therefore inefficient, and potentially, inadequate, to track targets executing complex manoeuvres. Many researchers have suggested adaptive rate update techniques, such as ref. [76], in which beam scheduling, positioning and detection thresholds are optimised according to the computational load. An IMM model is presented in ref. [77], with dual aims of estimating and predicting the target states and estimating the level of the dynamic process noise. The overall goal is to reduce the number of track updates per unit time. Another work performed optimal scheduling of track updates to minimise the radar energy consumed [78]. Energy minimisation was modelled as a non-linear optimal control problem and optimised to yield a pair of optimal sequences of track update intervals and Signal to Noise ratio (SNR) values.

Symbolic-AI has excellent reasoning capabilities; however, it is difficult to instill learning capabilities into it, which is a key part of human intelligence. Since Symbolic-AI relies on explicit representations and does not take into account implicit knowledge, it struggles, especially when making sense of unstructured data (in the radar environment, for the case of RRM). Towards this end, the AI domains that focus on teaching machines on their own, that is, the ML[1] paradigm is introduced. Since then, ML has shown a huge success in many domains. Recently, there has been an increased trend within the radar research community in using different ML techniques for RRM tasks. In the following section, we review the related literature on ML-based RRM.

4 | MACHINE LEARNING FOR RADAR RESOURCE MANAGEMENT

In this section, we present the core contribution of this work—a review on some of the recent works which performed radar RRM tasks using techniques from ML.

Most RRM tasks include some level of optimisation in order to select current or plan future actions, particularly for a cognitive radar system. However, the associated computational cost may excessively be increased according to the complexity of the RRM task. For example, optimal task selection and scheduling in MFRs with multiple simultaneous tasks at hand in a limited time duration is an NP-hard problem where the complexity increases exponentially orders when multiple timelines are considered. Multiple heuristics have been applied to the problem. While the heuristics significantly reduce the computational time, there is a significant performance gap between the performance of the heuristics and globally optimal solutions. Machine learning can be employed to fill this performance gap while maintaining low complexity. Machine learning has the capability to reduce computational costs in real-time implementation through the use of offline learning, deep neural networks|deep neural network (DNN), online learning, and RL. Moreover, some ML models may be trained

[1]The term "machine learning" is often attributed to Arthur Samuel, who coined the term as "without being explicitly programmed" [198] in 1959.

and used in conjunction with heuristics to produce near optimal performance at the same low complexity levels.

We provide a primer on some of the most prominent ML algorithms, shown in Figure 2, in the Appendix. These algorithms have been used extensively in communications domain, including RRM and signal processing. Conventional ML algorithms are broadly classified as: (i) supervised, (ii) unsupervised, and (iii) RL algorithms. Supervised learning algorithms train a function to learn a mapping from an input to an output via labelled training data. Unsupervised learning, on the other hand, deals with clustering and associative rule mining problems based on unlabelled data. In RL, a combination of exploitation and exploration is used in paradigms such as Markov decision processes (MDPs) to take actions in an environment in order to maximise the cumulative reward. Apart from the three classes, some ML algorithms can learn from unlabelled data in conjunction with small amount of labelled data, called semi-supervised learning.

Deep learning is another paradigm of ML, which has gained massive popularity in scientific computing as its structure and function is said to be based on that of the human brain. Deep learning uses artificial neural networks to perform sophisticated computations on large amounts of data. Deep learning can be supervised, unsupervised, or reinforcement, and it depends mostly on how the neural network (NN) is used. Different NN architectures used for deep supervised learning are particularly useful as a function approximator in deep RL (DRL) as shown in Figure 2. Recent tutorials and overviews of ML algorithms are available for readers unfamiliar with the background and use of ML paradigms [79, 80].

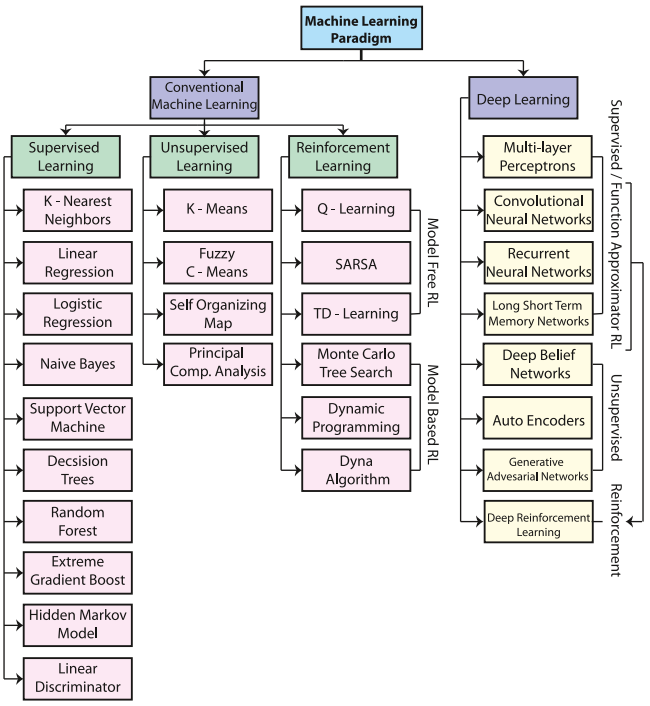

FIGURE 2 The machine learning (ML) paradigm.

Our discussion in this section is divided with respect to the RRM tasks and functions. Specifically, we discuss the recent AI-based literature in RRM in the following RRM task domains: (i) target identification and tracking, (ii) spectrum allocation, (iii) waveform synthesis and selection, (iv) time resource management, (v) task scheduling and parameter selection, and (vi) Q-RAM. An overview of the selected works for this purpose are given in Table 2.

4.1 | Target identification and tracking

We found out numerous recent works on target detection, target tracking, clutter estimation and clutter suppression which employed ML-based techniques. Because efficient RRM is essential to performing real-time identification and tracking, we are including discussion on this function in this section. Coordinated RRM that exploits tracking and sharing data between radars is known to outperform independent RRM tasks at lower track occupancy and frame time [81]. In ref. [82], Bayesian multi-target filtering is performed using Gaussian Mixture probability hypothesis density filter with an Long Short Term Memory Network (LSTM) transition function. The goal of the work is prevention of overestimation of the number of objects by the filter. The network architecture is based on Gaussian multivariate density estimation, and referred to as an Multi-dimensional LSTM (MD-LSTM) network. The architecture is composed of 3 blocks: LSTM, Dense, and Output layers. The trained MD-LSTM model is then used for dynamic probability hypothesis density estimation. For performance validation, state estimation of the proposed model is compared with a benchmark Near-Constant Velocity model. In another experiment, using different probabilities of detection, the impact of missed detection is evaluated. The model prevents the filter from overestimation of objects even in the scenario of false alarms or missed detections [82]. The performance represents a simple simulated scenario necessitating the need for exploring state-of-the-art ML techniques for more complex scenarios with real data.

Another recent work involves Deep Q-Learning (DQL) for target tracking in cognitive radars [83]. The environment is assumed as requiring radar-communication coexistence modelled using a Markov decision process. A single point target is assumed to follow some straight constant velocity trajectory. The states in the MDP include the target position, target velocity, and interference patterns. The authors build a transition probability matrix and a reward matrix over a finite number of training runs. Upon completion of the training runs, Bellman's equation is modelled for the benchmark MDP, while a neural network is trained on the reward given to a particular action. The weights of the deep-Q networks (DQN) are updated to select the considered optimal action which is presumed to give the highest reward. The trained DQN takes current states as the input and estimates the Q value for each potential action, after which it selects the action with the highest Q value. The DQN is shown to outperform the MDP when operating in a frequency band that both have not been

TABLE 2 Summary of literature on machine learning (ML)/artificial intelligence (AI) applications in radar resource management (RRM)

RRM Category	Ref.	RRM Task	ML/AI Technique
Target detection and tracking	[82]	Target tracking	Gaussian mixture probability hypothesis density (GM-PHD) with LSTM model
	[83]	Target tracking	Deep Q networks
	[95]	Clutter identification, target tracking and classification	Survey of AI techniques
	[93]	Tracking and surveillance	Survey of AI techniques
	[85]	High resolution range profiles (HRRP)	Unsupervised and semi-supervised anomaly detection methods
	[87]	Target identification + interference mitigation	Robust principal component analysis + residual over-complete auto-encode
	[93]	Signal detection	LSTM-based deep denoising autoencoder
	[84]	Targets tracking	XG boost supervised learning
	[91]	Clutter estimation	CNN-LSTM estimator
	[92]	Clutter suppression	Online dictionary learning
Spectrum allocation	[96]	Spectrum allocation	RL + LSTM
	[97]	Adaptable bandwidth tracking	Q-learning
	[98]	Spectrum sharing in detection and tracking	Double deep recurrent Q-network (DDRQN)
Waveform synthesis and selection	[100]	Waveform design generation	Deep deterministic policy gradient (DDPG)
	[101]	Waveform synthesis	GAN
	[104]	Waveform selection	Universal learning
	[105]	Frequency + waveform selection	Q-learning
Time resource management	[113]	Time resource management	MCTS + RL
	[114]	Time budget management	Q-learning algorithm with epsilon-greedy policy
Task scheduling and parameter selection	[37]	Antenna selection	CNN classifier
	[108]	Parameter selection, prioritisation, and task scheduling	Branch and bound + NN
	[116]	Dynamic task scheduling	Q-learning
	[112]	Joint task selection and scheduling	MCTS
	[109]	Joint task selection and scheduling	MCTS + policy networks
	[117]	Joint task selection and scheduling	Deep Q-Network
Q-RAM	[121]	Q-RAM	DRL

Abbreviations: CNN, convolutional neural networks; LSTM, Long Short Term Memory; NN, neural network; XG, Extreme Gradient.

trained on, and when the computational complexity increases where MDP becomes inefficient.

The work in ref. [84] performs Extreme Gradient boosting (XGB) supervised learning to outperform the well-known approach of Bayesian filtering in radar target tracking applications. Since a Bayesian tracker requires accurate a priori information for estimation, its performance falls short in unknown environments. The supervised learning model is based on polar coordinates and trained on radar measurements. The loss function is formulated to predict the optimal tree structure and estimate the corresponding leaf values. The performance of the designed XGB filter (XGBF) is compared with the results using particle filtering (PF). It is shown that the XGBF outperforms PF in terms of estimation root mean square error (RMSE) and shows similar performance for 10,000; 20,000; and 40,000 samples. The paper mainly has focussed on the problem of single-target filtering; the inclusion of clutter and multitarget measurement correlation issues would be a challenge, especially if the training data needs to be generated due to unavailability of real data as is the case in this work.

In the area of target identification and detection domain. Bauw et al. addressed the challenges of detecting unusual radar targets using semi-supervised anomaly detection methods (SAD) [85]. High resolution range profiles targets identification by ML has recently received a lot of traction in the radar research community. The authors present an SAD approach which is extension to the earlier proposed Deep Support Vector Data Description model [86]. Unsupervised anomaly detection, even with training pollution, yields reliable results, except for the scenarios of ship detection, since there is a large

variation in shapes and sizes of the vessels. Semi-supervised anomaly detection has the potential to improve the detection results with a smaller number of labelled data points; this is an important benefit since labelling image data is a time consuming and costly process. The work in ref. [87], on the other hand, uses an unfolded robust PCA (RPCA) method for target identification and interference mitigation in radars. The use case discussed in the paper is specifically for radars mounted on autonomous vehicles. The main novelty in this work is introduction of residual overcomplete auto-encoder blocks into the recurrent architecture of unfolded RPCA, which is able to estimate the amplitude and phase of the interference in the environment. The automotive radar interference mitigation data set [88] is used to train the proposed model. The proposed model outperforms selected benchmarks both in terms of the area under the receiver operating characteristic (ROC), as well as the mean absolute error between the range profile amplitudes calculated from label signals and predicted signals.

DL-based techniques have also been proposed for radio signal detection, owing to the fact that the matched filter and likelihood ratio tests become infeasible without a priori information. For instance, in ref. [89], bidirectional LSTM-based denoising encoders were proposed to detect the presence of radar signals in the environment. The bidirectional version of LSTM contains both forward and backward pass, which enables deduction of non-causal information, that is, both forward and backward correlation, providing an improvement in performance [90]. The RNN-based denoising autoencoder outperforms the industry benchmark detectors, such as the energy detector and the time-frequency domain detector in terms of low probability of false alarms and higher area under the ROC.

The ability to discriminate targets from background interference becomes even more important in maritime applications where the sea clutter can only be accurately estimated when there is prior environment information at hand. Convolutional Neural Networks and Auto Encoders (AEs) have shown very high classification accuracy for image-based data sets. To investigate their applicability in detection in sea clutter, the authors of ref. [91] use a hybrid model including CNNs and LSTM to estimate the parameters of the K-plus-noise distribution with low computational complexity. The CNN-LSTM employs the CNN layer for feature extraction and LSTM layer to support sequence prediction. The 1D CNN-LSTM estimator outperforms the $z\log(z)$ algorithm in terms of mean square error (MSE) as well as computational complexity. The work in ref. [92] is also focused on alleviating the issues caused by sea clutter returns which affect the performance during the detection of small targets. Since target detection schemes use an amplitude distribution that require knowledge of specific parameters, inaccuracy in estimation of those parameters yields poor detection results. The authors in the referred work using online dictionary learning which is used for learning sparse representations of signal with faster convergence than dictionary learning. The performance is evaluated using target signal to interference ratio over a large number of range/Doppler maps with signal returns from small vessels. The proposed DL-based sea clutter suppression technique showed marginally better performance than other algorithms in the exo-clutter region.

There have been some survey papers on AI and ML applications on radar surveillance systems worth mentioning. In ref. [93], the authors project big data trajectory as an effective method for improvement in radar surveillance systems. In particular, they focus on methods and use cases for anomaly detection with description on the data sources, data preprocessing framework and tools, data smoothing and the role of sliding windows in ML-based techniques. Two architectures: Lambda and Kappa, are discussed for real-time surveillance. Use case for heat maps in the risk assessment for ship vessels is shown to be very helpful for visual assessment of the situation. Similarly, by clustering areas of interest using hierarchical Density-Based Spatial Clustering of Applications with Noise [94], relationship between objects and ports is shown to be extracted with relative ease. Wrabel et al. [95] present a survey of AI techniques for target surveillance with radar sensors. The focus streams in this work are: (1) clutter identification, (2) target classification, and (3) target tracking. Clutter identification has been done using multiple AI approaches, including Bayes classifier, ensemble methods, k-nearest neighbour (kNN), SVM and neural network models. In addition to these techniques, target classification has also been done using RNNs, CNNs and decision tree models. Similarly, works have been reviewed for target tracking, with ensemble techniques, neural networks, SVMs and RNNs the more popular techniques.

4.2 | Spectrum allocation

We found two recent works, both of which used RL techniques in the radar spectrum allocation domain. In ref. [96], the authors apply RL as a decentralised spectrum allocation approach to avoid mutual interference among automotive radars. Because RL algorithms can learn decision-making policies in unknown environments, it is suitable for this task where radar sensors have limited information about the environment. An LSTM network aggregates observations through time so that the model can learn to choose an optimal sub-band by using both present and past observations. The work assumes that the whole frequency band is divided into non-overlapping sub-bands, and the radar devices are greater than the number of sub-bands.

The RL-based spectrum allocation works as follows. First, the processes the signal from last step and constructs the current observation. Then, the transmitter Q-network selects a sub-band by aggregating historical observations. The reward generated from receiver terminal acts as a guidance for the transmitter Q-network to choose a better sub-band selection policy. The proposed algorithm is evaluated against benchmark decentralised spectrum allocation methods such as the random and myopic policy. Results show superior performance in terms of success rate in different traffic density scenarios. However, the Q-network is trained and tested in a simulated

environment with relatively simple scenario model to show the feasibility of the proposed approach. The real world environments could be complex necessitating the modelling, which can better represent the practical scenarios.

We have already established Q-learning as a strong candidate for resource management-based problems due to its inherent ability to find the optimal action-value function without needing any model for the environment. In ref. [97], the authors model the radar-communication bandwidth allocation problem as an MDP and then apply policy iteration to determine the optimal policy. To mitigate interference between the radar and communication networks, the MDP and Q-learning-based model learns the time-frequency spectral occupancy pattern of the interference. Simulations were carried out in three interference environment scenarios: (i) constant interference, (ii) intermittent interference with a high transmission probability; and (iii) intermittent interference with a low transmission probability. Not only did the radar learn the interference pattern in frequency, but was also able to trade signal-to-interference-plus-noise ratio (SINR) for increased bandwidth in situations when the target was close to the radar. The work investigated only five subbands; a higher number will increase the size of the state space exponentially resulting in more complex training taking much longer times. Policy-based DRL techniques, where the objective is to learn a set of parameters, far fewer than the state space, can resolve this issue as has been shown in the extended work [98].

The work in ref. [98] applies non-linear value function approximation via DRL to address dynamic non-cooperative coexistence between a cognitive pulsed radar and a communications system. The DRL-based approach allows the radar to vary the bandwidth and centre frequency of its linear frequency modulator which improves both target detection and spectral efficiency. In particular, the authors use the DQL algorithm, which is extended to a Double deep recurrent Q-network (DDRQN), which is shown to further improve on the stability and policy iteration of the DQL approach. This work is an extension to an earlier approach for radar waveform selection using MDPs [97]. Unlike the model in ref. [97], which takes a default action when the state transition model is unspecified for a situation, the proposed model utilises an estimated function value to perform a more informed action. The performance of the proposed algorithm is done with policy iteration and sense-and-avoid (SAA) algorithms through experiments performed on a software defined radio). The published results exhibit faster convergence and better learning in new scenarios as compared to the benchmark schemes.

4.3 | Waveform synthesis and selection

Waveform optimisation is one of they key features of cognitive radar with adaptive transmitters and receivers. Choosing waveforms from a library or codebook (CB) of pre-defined waveforms can be done for a specific or multiple radar tasks simultaneously. The core of waveform optimisation is in exploitation of the DoF due to any form of diversity, for instance, spatial diversity, beam patterns, frequency diversity, code diversity, and polarisation.

Optimal waveform selection may be carried out using a NN-based framework as done in ref. [99], in which the authors analysed eclipsing, blind velocity, clutter, propagation, and jamming factors for a radar. Optimal waveform parameters were estimated using a non-linear NN model.

The discussion in ref. [100] provides a generic overview on application of neural networks and ML in the development of cognitive radars with a motive to reduce the computational cost for real-time implementation. The paper discusses a use case scenario where RL is employed to generate waveforms with a 26 dB power spectral density (PSD) notch. The problem is to place the notch within the radar bandwidth such that it minimises the interference from jammers and other communicating devices. The idea is to select a set of phases, which selects waveforms to create a notch within the PSD of the created signal. The non-linear optimisation problem of phase selection is solved using the Deep deterministic policy gradient (DDPG) algorithm. Deep deterministic policy gradient is, essentially, an actor-critic model to generate training inputs and rewards as per the quality of the NN outputs. The actor produces a set of phases for evaluation by the simulation environment. The environment performs a discrete Fourier transform and calculates the value of the formulated objective function. This value is the reward for the action and is fed to the critic NN. The design of the critic NN is such that it takes state and action as inputs in different layers and outputs the Q-value which is back-propagated to the action input layer to obtain an error for the action. The DDPG RL algorithm alleviates the need of a labelled dataset via a simple environment simulator. The generated 26 dB PSD notch may not be sufficient for real-world deployment, however it can be increased via fine-tuning of the model parameters. Importantly, this RL approach eliminated the need for a large amounts of labelled data that are not available for the training of models prior to deployment of a radar.

Another very important aspect of radar waveform optimisation is the synthesis of novel radar waveforms with a desirable ambiguity function (AF) shape and constant modulus property. Since there are a limited number of code sequences available in radar code families, it causes problems when operating MFRs or multiple input multiple output-based communication systems. As discussed in Section 2, GAN-based approaches are now widely used to generate realistic synthetic data which could improve training results in DL applications. In ref. [101], GAN-based NN structures are used to generate realistic waveforms from a training set of already existing waveforms. In particular, a Wasserstein GAN [102] structure is developed for complex-valued input data. The model is trained on Frank and Oppermann codes to synthesise new waveforms which yield high autocorrelation, identical AFs, and low cross-correlation with the existing codes. The AF diagrams of the synthesised waveforms show a high similarity with the training dataset waveforms. With negligible cross-correlations of the GAN generated waveforms with the training dataset, it clearly shows that GANs can generate

realistic and distinct radar waveforms. The synthesised waveforms were also constrained to possess a constant modulus for efficient amplifier use.

Since radar signals exhibit temporal correlation, applying memory-based learning algorithms on partial state information to learn waveform selection strategies improves radar performance as compared to the benchmark minimum expected mean-squared tracking error [103]. The work in ref. [104] builds a model of the radar environment using a context tree which is further utilised to select waveforms in a signal-dependent target channel. The authors formulate a Lempel-Ziv-based waveform selection algorithm which is a cost optimal solution for a finite-order Markov target channel. The universal learning algorithm maintains estimates of transition probabilities of observing a particular state given the current context information. The context tree is updated by traversing backwards over the previously observed outcomes. During each step, an action is selected by either exploiting known information about rewards or exploring new actions. The objective functions focus on target detection accuracy as well as maximising the mutual information. The universal learning approach yields higher average SINR and lower RMSE as compared to benchmark schemes. Since the universal learning algorithm is highly complex, the authors considered a limited size waveform catalogue and state-space discretisation to maintain tractability; this may effect the performance in more practical scenarios.

Another waveform synthesis scheme [105], specifically for anti-jamming radars, examines RL-based joint adaptive frequency hopping and pulse-width allocation on anti-jamming as the current anti-jamming strategies (using frequency hopping and pulse width allocation) often have difficulty in adapting their policy to complex and unpredictable environments. Like in other RL works described, the objective function is modelled as an MDP. *Q*-Learning is utilised to learn the optimised radar anti-jamming policy in partial information environment situations. The reward function value is a quantified version of radar anti-jamming functions and encompasses both RRM tasks, that is, frequency hopping and pulse width allocation. The *Q*-learning based joint optimisation algorithm is compared with a benchmark random hopping strategy, which chooses a random frequency from the band during every instance [106]. The *Q*-learning based strategy yields higher average reward at different hopping cost and number of transmitted pulse values.

4.4 | Time resource management/task scheduling and parameter selection

Scheduling multiple tasks in a limited time budget is one of the most critical RRM tasks in MFRs. Time being a limited resource, needs to be carefully allocated to different tasks based on some priority assignment. Such an optimisation problem where the objective is to minimise the number of dropped and delayed tasks is an NP-hard problem [7]. The branch and bound (B&B) scheme is known to yield the optimal solution for this problem [107]; however, the computational complexity of the B&B algorithm increases exponentially with the number of tasks to be scheduled.

Shaghaghi et al. in their seminal work on ML in RRM studied parameter selection, prioritisation, and scheduling within RRM domain for multichannel radars [108]. To overcome the complexity issue, in this work, the authors trained a value network consisting of a DNN with the data obtained by running the B&B algorithm offline. Essentially, the trained DNN estimates the value of the nodes of the search tree, thereby speeding up the B&B process by eliminating the nodes which are far from optimal solution. The DNN-based solution converges to a near optimal solution while significantly reducing the computational burden. In order to make the algorithm more robust to a estimation error, a scaling factor is introduced, where choosing a sufficiently high scaling factor means that fewer nodes are eliminated from the search tree. The scheduling performance in that case is found to be very close to B&B approach, however, with a slight increase in computational burden in terms of node visits.

To further reduce the computational time while also providing near optimal results, the authors implemented Monte Carlo Tree Search (MCTS) in ref. [109], where along with the dominance rules of B&B, DNN as a policy network was used to focus the search on more promising branches in a tree structure. The MCTS is combined with a DNN using the popular AlphaGo and AlphaZero approaches [110, 111]. At every tree node, a priority distribution is created from a policy network trained by supervised learning on ideal solutions obtained by the B&B method. Although the methodlogy is somewhat to similar to ref. [108], there are some marked distinctions. For instance, the algorithm in ref. [108] requires a fixed number of tasks, while in this work, the input state focuses on the next tasks to be scheduled. This enables an arbitrary number of active input tasks. Unlike [108] which employs value functions, here a policy network, modelled by a 7-layer DNN, is utilised. Simulation results show that the average cost approaches the optimal B&B performance as the number of Monte Carlo rollouts are increased. As compared to benchmark schemes, near optimal performance is achieved along with orders of magnitude lower computational complexity than B&B method.

Although the works in refs. [108, 109] produce near optimal results with reduced computational burden, however, both require training data to be generated via offline execution of BnB, which again needs large computational time. Furthermore, different training data would be needed corresponding to different problem size and/or task distributions. Above all, both the methods do not provide the radar with the capability to adapt to a dynamic environment.

Towards this end, the same authors developed an approximate algorithm based on the MCTS method, where the cognitive scheduler is trained with the data from radar interactions with the environment [112]. An RL model is used to train the policy network under multiple constraints, such as non-homogeneous channels, blocked channels, and periodic tasks. The purpose of policy network is to reduce

the width of the MCTS search. Each RRM task has an associated start time, completion time deadline, and dropping cost. It is assumed that tasks may be executed differently on different channels. The Q-function value provides an estimate of the expected utility obtained if a particular action is taken at a given node. Statistics obtained from running MCTS are used to train network parameters, which are then adjusted to minimise the cross-entropy loss. The policy network has a depth of seven layers, with the first four being convolutional and the last three fully connected. The proposed MCTS + policy network model outperforms the benchmark algorithms by giving lower average cost and also lower task drop rate.

Building on the supervised learning approach of Shaghaghi et al. [109] and RL work in ref. [112], Gaafar et al. propose a modified MCTS solution for the task scheduling problem in order to find an effective low complexity solution [113]. The modified MCTS is further complemented with an RL-based model which can learn using a reward-based mechanism without the need of a large set of training data. The first modification to the classical MCTS algorithm is to not allow the revisit to the states that have all solutions branching from it already visited. Secondly, task sorting is based on starting time, therefore earlier tasks are chosen with higher probability.

The utility function is composed of three conflicting factors: (i) supporting task selection based on earlier start times, (ii) supporting task selection based on known low costs, and (iii) exploration of tasks with smaller number of visits. In the RL-based approach, a DNN is trained to learn the best values for the probability vector for all tasks. The MCTS yields actions with better solutions, but the DNN then guides the MCTS towards better search policies using training data. This DNN guided MCTS system from ref. [113] is shown as Figure 3. Using a DNN with 5 layers as the DNN structure, the results show reduced average validation cost, a lower percentage of dropped tasks, and near optimal average cost.

The algorithm however suffered from long online time for training as well as for scheduling a single problem. Moreover, this work did not consider adaptability to major changes in the environment as the probability distribution of task features was considered to be fixed in the training and testing phases of the algorithm. Eventually, the algorithm needs to train itself for an even longer time if the task distribution and environment change abruptly. The robustness of the approach to possible differences between the probability distribution of task features in the two phases is unclear.

Another recent work is [114] focussing on adaptive revisit interval selection (RIS) in MFRs as a time management problem formulated as an MDP with unknown state transition probabilities and reward distributions. The reward function is proposed to minimise tracking load|transfer learning (TL) while keeping the track loss probability as an optimisation constraint. A Q-learning algorithm with an epsilon-greedy policy is used to solve this problem. The objective of minimising time budget and tracking losses is reflected in the immediate and cumulative rewards of the agent. The performance of the proposed algorithm is compared with a benchmark

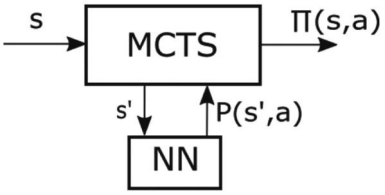

FIGURE 3 Neural network (NN) guided monte carlo tree search (MCTS) model [113].

prediction error covariance matrix (PECM)-based solution [115]. The comparison is made using mean and peak tracking loss and position prediction error values. In both the metrics, the proposed RL-based scheme outperforms the benchmark significantly, solidifying the usefulness of RL-based algorithms in RIS-related time management RRM tasks. The results, however, suggest that the learning speed will decreases with an increase in the size of the state space - a major issue for tabular-based RL methods. Moreover, Q-learning struggles in a non-stationary environment, which is specifically the case of RRM for cognitive radar.

Another recent work utilises Q-learning for dynamic task scheduling in MFRs [116]. First, an MDP is created for the MFR network executing tasks with dropped task ratio being the evaluation criterion. After that, the state-action space is designed for Q-learning algorithm. The action selection is performed by taking in consideration both the immediate and future reward, the new states would yield. The Q-learning based scheme clearly shows lower percentage of dropped tasks in comparison with benchmark first come first executed method. The method, however, has not taken into account the delay cost, an important metric in radar task scheduling problems.

In a more recent work [117], a Deep Q-Network agent is developed and tested using two different reward schemes and the performance is shown to be better than EST for the case of queues with overlapping tasks only. For the other case of queues with non-overlapping tasks, the performance of EST is better than the DQN agent. Moreover, DQN results are shown for only four tasks, which is a small number when considering practical scenarios.

We found a few recent papers from the databases explored that exploit ML techniques on RRM-based task scheduling and parameter selection. The first study is ref. [37] in which the authors employ a CNN for direction of arrival (DoA) estimation in phased array radar antenna systems. The problem is modelled as a multi-class classification where each class designates a different subarray. Without apriori knowledge of the target location, feature maps are extracted to train the CNN from the covariance samples of the received array signal. Training data is created with subarrays that yield lowest minimal bound on the MSE. The CNN model which is chosen to have a depth of nine layers in this work does not rely on antenna geometry for optimal selection of antenna subarrays. The CNN-based structure provides 32% better classification than earlier SVM-based models [118] and also 72% more accurate DoA estimation.

Our survey has clearly shown the popularity of RL-based algorithms in scheduling optimisation problems of RRM as: (i) it does not require external training data to learn as is needed in supervised learning, (ii) it reduces computational time while producing near optimal results, (iii) it has the potential to adapt to the dynamic environments.

4.5 | QoS-based Resource Allocation Model

QoS-based Resource Allocation Model was formally introduced in ref. [119] and its application for Symbolic-AI-based RRM has already been discussed in Section 3.4. The objective with Q-RAM is to maximise the utility of a set of radar tasks over operational parameters, such as waveform, dwell time, and tracking filter, while meeting the resource constraints. For dynamic environments, Q-RAM is computationally inefficient as it must repetitively recompute the operational parameters in resource allocation frames, which imposes a limit on the reaction time of the algorithm as well. Towards this end, the continuous double auction parameter selection algorithm is proposed in refs. [7, 120] which enables the solution from the previous time step to be adapted to the current time step without a complete re-computation of the resource allocation, hence reducing computation for dynamic RRM problem.

In the recent literature on RRM, there is one paper that employs ML on QoS-based resource allocation model for intelligent decision making in radar systems [121]. This work used a DRL model where a NN agent predicts a sequence of desirable task configurations without requiring all configurations in the resource utility space. Specifically, the agent learns to output the task configuration that yields the highest difference quotient of utility to resource with the input configuration. The reward for each action is a direct function of the above utility-resource-quotient realised from that particular action. The agent is modelled using a single-worker advantage actor critic network [122]. The RL trained agent is quite successful in choosing task configurations that yield a high utility for the given resources. In particular, 97%–99% of the Q-RAM performance is achieved with 120,000 training steps.

The true advantage of this RL-based technique is in the reduced computational complexity. In mathematical terms, the improvement is by a factor of $\log c$, where c is the number of possible configurations per task. The performance reported in ref. [121] does not appear to be any better than ref. [7] in dynamic environments; however, it has shown the potential of RL-based RRM in overload situations without extreme computational complexity. In real world applications, the number of configurations per task can become impractically high where the RL agent can be trained via the Wolpertinger algorithm [123]. Moreover, the RL agent-based approach can be easily integrated into existing Q-RAM implementations and enables the "self-learning" capability required for a cognitive radar system.

5 | MACHINE LEARNING FOR NON-RRM TASKS—SELECTED LITERATURE SURVEY

Although the scope of this survey paper is a comprehensive review for ML applications in RRM domain, for the sake of completeness, we include a brief review of some of the recent work involving ML/AI for non-RRM tasks. In ref. [124], the authors provide a review of ML applications to synthetic aperture radars (SARs) operating in the millimetre wave (mmWave) frequency band. The authors present an overview of SAR technology, followed by recent applications of mmWave SAR. In particular, the authors elaborate on the use of CNNs for land cover classification from polarimetric SAR images [125], a RPCA model for artefact removal in mm-Wave automotive SAR [126], and SAR-GAN which is based on synthetic data creation from SAR image inputs [127].

In ref. [128], the authors use CNNs to classify low probability of intercept signals. The authors employ TL on pre-trained CNN architectures to reduce the required training time. VGG-16 [129] along with TL not only shows high classification accuracy, but also reduces training time which enables electronic warfare data processing in near real-time. The reduced size of autonomous aircrafts, such as drones, makes it difficult to detect them over longer distances. Robust classification methods are required to ensure a high discriminating ability for characterisation of movements.

The authors of ref. [130] use the raw time series, spectrogram, deep latent vectors and covariance data to design an AI-based global model using forward connected NNs. Even though the dataset distribution skewed the distribution between object classes, the designed model performs multi-class classification with high precision and recall values. Similarly, SAR imaging of moving targets is done using RPCA in ref. [131]. A phase-space reflectivity matrix is constructed for single-channel SAR systems and a rank-1 RPCA eliminates the requirement for singular value decomposition. Moving target detection is done by reshaping the images as 2D matrices. The alternating direction method of multipliers method reaches good performance in fewer iterations as compared to the proximal gradient descent method.

A DL model is designed in ref. [132] for autonomous vehicles working on a JRC architecture to improve safety and efficiency by utilising both radar detection and data communication functions. A Double Deep Q-Network with experience replay memory and DNN are used to avoid the slow convergence of the conventional Q-learning algorithm. Tracking load|transfer learning is also used to accelerate the learning process in unknown environments. The combined approach reduces the mis-detection probability and also requires a smaller number of training iterations to converge to an optimal policy. Another work performs radar target reconstruction using a topology of linear chains of CNNs, where the input radar signals are learnt layer by layer through the designed network [133]. Synthetic aperture radar images from the SAR Ship Detection Dataset [134] are used as the input to

the designed model. During the learning process, a direct map from the radar echo to the reflectivity function is obtained. The final outcome is a DL network that recovers scanning target information with superior performance in target reconstruction and anti-noise capability.

Machine learning has found its way in radar-based classification tasks, for instance, in ref. [135], where the authors apply DL models to classify micro-Doppler radar time series. Deep temporal architectures on time-frequency representations are developed along with information geometry on Riemannian manifolds through symmetric positive definite NNs. The developed model is tested on drone classification tasks and compared with second-order models on robustness due to a lack of data. The authors of ref. [136] present an architecture for cognitive radar that aims to perform non-cooperative target classification. The authors use a three-layer model from an earlier work [137] comprising knowledge-based, rule-based, and skill-based layers. Feature formation and recognition form the lower layers at the receiver, while target identification is performed at the knowledge-based layer. Target recognition is performed using a CNN-based architecture, while auto-encoders are used for change detection. Markov decision processes are also used to evaluate the target classification task using this technique. The authors conclude that consistent performance requires consistent behaviour generation on all three abstraction layers.

The work in ref. [138] is a slightly different application of NNs in detection and classification of changes in multi-temporal SAR COSMO-SkyMed images. The DL model in this work is based on pulse coupled neural networks, that perform change detection, while change type identification is done by a standard Multilayer perceptron NN. Both models are efficient in that they deliver high accuracy while requiring processing time in the order of seconds.

Other interesting use cases include ML/AI applications on sensor data fusion [139]. The authors present multiple examples along with the associated challenges. Some challenges yet to be addressed in this domain include development of contextual models, TL for coordination from one situation to another, a first-principles understanding for estimation and explainability, and novel distributed architectures for multi-sensor design architectures.

Chandrasekar studies the application of AI in weather radar tasks, in particular, quantitative precipitation estimation, precipitation classification, and nowcasting [140]. Quantitative precipitation estimation is performed using a hybrid DNN model with the first model concept for ground-based radar rainfall estimation, and the second model for space-based radar rainfall estimation. The ML-based models demonstrate superior performance as compared to the conventional fixed-parameter radar rainfall relations. For precipitation classification, an RF classifier first identifies the individual hydrometer type, followed by an RF regressor deployed for hail size estimation. Finally, for precipitation nowcasting, a CNN-based model is proposed as a superior alternative to conventional radar nowcasting approaches. Since data labelling is a laborious process, GANs and deep convolutional AEs are used in ref. [141] to create synthetic range-Doppler maps of Frequency-Modulated Continuous-Wave radars. Using the synthetic data to train a detector system yields a performance improvement of more than 10% as compared to the smaller real dataset.

Discussing some recognition tasks, unknown radar emitters are recognised using Markov chains and LSTM models in different environments of known and unknown emitters [142]. Long Short Term Memory Network models are shown to be highly robust with respect to missing syllables. The LSTM entropic open-set (EOS) shows higher classification accuracy. Both EOS and cross entropy LSTMs work well for corrupted datasets. For fast decision making, MC is the preferred option. Waveform classification using a complex-valued Fourier synchrosqueezing transform algorithm shows superior performance than the normally employed Choi-Williams distribution (CWD) method [143]. Results show that the transform followed by a CNN-based recognition system shows better classification results even in low SNR regimes.

DNNs may also be used for unsupervised heterogeneous change detection using the underlying cross-domain dissimilarity measure for strengthening approaches based in image transformations [144]. Different SAR image datasets are used for experimentation with low values of κ, showing that the DL-based model has a strong potential for change detection in SAR datasets.

6 | CHALLENGES AND FUTURE RESEARCH DIRECTIONS

The literature presented above has shown the potential of ML techniques in replacing general, explicitly designed RRM models with appropriate models learnt from data. However, these methods generally require large amount of training data, which mostly would not be practical for radars. Moreover, the training data may not represent all the environments in which the radar may be required to operate. Another important concern is the adaptability to continuously changing environments for cognitive RRM. Towards this end, RL algorithms have been applied wherein the optimal policy can be learnt by continuously interacting with an environment. Policy-based RL methods using DL are specifically found to be more suitable due to: (i) the better convergence, (ii) the ability to learn stochastic policies, and (iii) the effectiveness in continuous action space. However, online learning in RL can be slow and can lead to the same radar type having different learning states and hence performances. In addition, validation of the online learning process should be ensured in order to avoid the risk of the radar learning an undesirable unpredictably leading to poor performance.

In this section, we briefly touch upon some of the challenges which are worth investigating in the domain of this work. On the back of the stated challenges, we then discuss some promising research directions for ML enthusiasts working in RRM domain.

6.1 | Challenges in machine learning applications to radar resource management

6.1.1 | Lack of radar specific machine learning models

From our literature review, observation, and experience, we have concluded that the most pressing issue is non-availability of task specific models for radar function optimisation. The normal practice has been to apply ML models which have been developed for other well known applications; like image processing; without major structural modifications to RRM related tasks. Unless we have ML models that are designed specifically for RRM tasks, the benchmark models alone will not be able to capture the fast varying environment trends.

6.1.2 | Lack of robust artificial intelligence models

Ensuring that the ML models are robust enough to yield meaningful insights is a major issue in all ML application domains. For high stake functions, like remote surgeries and self-driving cars, applying ML models may be a risky option. Similarly in radar RRM, applying the model in a simulated environment with controlled settings may do more harm than good. The approach is not viable as ML cannot be operationalised in real world settings. Ideally, an RRM ML model should be capable of withstanding adverse and unknown test environments in the rigorous testing phase.

6.1.3 | Homogeneous testing environments

We have observed that most of the ML-related work in RRM is conducted on environments that are essentially homogeneous. The data used to train the models does not effectively (indeed, in many cases cannot) capture the dynamic nature intrinsic to radar. The incident power density and clutter environment changes rapidly and any trained ML model is expected to adapt to these changing scenarios for it to be deemed reliable. In other RRM use cases also, such as target detection, radar cross section for targets of practical interest fluctuates rapidly, so a single simulation based data set is insufficient for the desired performance. As an analogy, just like self driving cars are not yet commercialised with full extent due to the inability of ML models to guarantee full proof reliability in unknown road situations, similarly, ML-based models will find it hard to be the sole decision makers in mission critical radar tasks.

6.1.4 | Varying characteristics of radar bands

Yet another challenge is the diverse behaviour of radar on different frequency bands. For instance, the range resolutions for X band and L band are quite distinctive. Training a dataset that is generated from a multi channel radar which operates on both the channel becomes a challenge. Robust ML models are required to tackle the unique interference behaviour of each band. Literature that has considered multi-channel radar operations often considers homogeneous properties across channels, which is far from reality. The next step requires going beyond this idealistic situation and developing robust models that are indifferent to channel properties fluctuations. Since simply increasing the data set would cause over-fitting, the correct approach is to modify existing techniques to create RRM specific models.

6.1.5 | Security issues of machine learning systems

The ML systems, specifically those employing online learning methods, are systematically vulnerable to a new type of cybersecurity attack, which can manipulate these systems by altering their behaviour to serve a malicious end goal [145]. As ML systems are integrated into RRM, these ML attacks represent an emerging and systematic vulnerability with the potential to have significant effects on the security. The regulating bodies should require compliance programs to adopt a set of best practices in securing systems against ML attacks, including considering attack risks and surfaces when deploying online learning systems, adopting IT-reforms to make attacks difficult to execute, and creating attack response plans.

6.2 | Promising research avenues

6.2.1 | Addressing dataset limitation issues

We have observed ML multiple issues with the datasets used to train the ML models. One is the limitation of data set, and the other is the homogeneity within the dataset. To address the small dataset size, it is necessary to augment the data set with simulations that are close to real complex EM environment which are needed to apply TL-based techniques. The existing data augmentation methods merely focus on the manipulation of original data samples, while neglecting practical operating considerations. Since environmental conditions significantly impact the radar echo signals, therefore, combining EM scattering computation with measurement data may be used to improve the quality of the generated data. At the same time, it is essential that the simulation data generated must cater to complicated scenarios which are expected in real RRM optimisation tasks.

6.2.2 | Novel learning architectures

There is a margin for exploration in novel learning methodologies, such as few/zero shot learning, which extracts discriminative features out of small training samples for better accuracy and

performance. Zero-shot learning techniques focus on learning attributes of the semantic layer which are used to predict classes in inference stage. This enables prediction of unknown classes even without associated training data. Not only do these techniques enable rare class identification, but also help in reduced data management and complexity costs. Since many RRM tasks occasionally experience unknown scenarios, for instance, classification of an unknown clutter distribution, zero and few shot learning can help generalise new tasks of limited supervised experience mimicking human ability to acquire knowledge through generalisation and analogy [146].

Lately, TL for RL algorithms has been widely studied [147, 148], where the experience gained in learning to perform source task is transferred to improve learning performance in a related, but different target task. In the context of DRL, TL has recently shown to be effective in various applications, such as handling environment changes [149] and general game learning [150]. Tracking load|transfer learning can be used for RRM in a cognitive radar for fast adaptability to dynamic environments, where the training time can significantly be reduced on variations of target environments by warm-starting the policy learnt by a DRL agent on the source environment.

Another approach that has gained recent traction is the model-based RL in which the agent creates a model to understand the environment. By learning a model of the environment, the agent can predict the next state and reward prior to taking actions. Since RRM is a complex process involving simultaneous optimisation of multiple tasks, predicting the dynamics of the environment allows transferability to other goals and tasks. Through optimised trajectory for least cost, the training iterations required are many orders lower than those required for model free RL. This, consequently, is highly beneficial for online learning models.

6.2.3 | Multiple channel optimisation

When RRM algorithms assign resources to different tasks, the process becomes computationally intensive when multiple lines are considered simultaneously. When multiple channels are included as DoF along with the other constraints, the optimisation complexity becomes exponentially hard. It will be interesting for researchers to investigate the approaches for search space reduction and evaluate different considerations that may be taken for more reliable application of ML algorithms.

6.2.4 | Explainable machine learning algorithms

The black box nature of DNN algorithms poses severe challenges in practical application of mission critical radar decisions, such as target identification, optimal resource allocation during multiple target detection etc. Fundamental questions about the algorithms such as: what the inner structure of the model does with the inputs, how is the model able to detect and classify targets etc. Remain unanswered. Explainable ML is crucial for someone at the helm of decision making to credibly believe the ML models when deploying highly critical strategies, for example, anti-jamming measures, and multi weapon deployment. Integration of recent advancements, like post-hoc explanations [151], visualisation of latent presentation [152], interpretable model design [153], embedding interdisciplinary knowledge like human brain cognitive science [154], and combination of symbolism with connectionism [155] with the existing techniques will allow causal reasoning and interpretability to the results.

7 | CONCLUSION

Efficient RRM is one of the critical optimisation problems in radar, due to the large number of executable tasks within a finite resource budget. We presented a comprehensive literature review on the AI applications in RRM domain, providing a brief overview of Symbolic-AI-based RRM and a complete review of ML use cases in RRM. Because of the high complexity of the multi-dimensional optimisation problem, ML provides a suitable model for real-time RRM. We have presented methodologies, design metrics, and KPIs used in the review studies. We also surveyed and segregated the existing works on the basis of the RRM use cases, that is, target detection and tracking, spectrum allocation, waveform synthesis and selection, time resource management, task scheduling and parameter selection, and Q-RAM. In addition to the literature review, the paper also presents and elaborates on the key challenges within the ML applied to RRM domain. We also provided some interesting research avenues that will hopefully encourage the researchers to develop robust novel solutions that are more suited to RRM functionalities and can adapt well to dynamic radar environments.

AUTHOR CONTRIBUTIONS

Umair Hashmi: Conceptualisation; Formal analysis; Investigation; Methodology; Validation; Writing – original draft; Writing – review & editing. **Sunila Akbar**: Formal analysis; Writing – original draft; Writing – review & editing. **Ravi Adve**: Conceptualisation; Formal analysis; Methodology; Supervision; Writing – review & editing. **Peter W. Moo**: Project administration; Writing – review & editing. **Jack Ding**: Project administration; Writing – review & editing.

ACKNOWLEDGEMENT

This work was funded by the Defence Research and Development Canada (DRDC) under contract number W7714-207097.

CONFLICT OF INTEREST

The author declares that there is no conflict of interest that could be perceived as prejudicing the impartiality of the research reported.

DATA AVAILABILITY STATEMENT

Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

ORCID

Umair Sajid Hashmi https://orcid.org/0000-0001-8704-7132

REFERENCES

1. Richards, M.A. (ed.) Principles of Modern Radar: Basic Principles. Institution of Engineering and Technology (2010). Radar, Sonar & Navigation
2. Moo, P., DiFilippo, D.: Multifunction RF systems for naval platforms. Sensors J. 18(7), 2076 (2018). https://doi.org/10.3390/s18072076
3. Charlish, A., et al.: The development from adaptive to cognitive radar resource management. IEEE Aero. Electron. Syst. Mag. 35(6), 8–19 (2020). https://doi.org/10.1109/maes.2019.2957847
4. Miranda, S.L.C., et al.: Comparison of scheduling algorithms for multifunction radar. IET Radar, Sonar Navig. 1(10), 414–424 (2007). https://doi.org/10.1049/iet-rsn:20070003
5. Charlish, A., et al.: Cognitive radar management. Novel Radar Techniq. Appl. 2, 157–193 (2017)
6. Moo, P., Ding, Z.: Adaptive Radar Resource Management. Academic Press (2015)
7. Charlish, A., Woodbridge, K., Griffiths, H.: Phased array radar resource management using continuous double auction. IEEE Trans. Aero. Electron. Syst. 51(3), 2212–2224 (2015). https://doi.org/10.1109/taes.2015.130558
8. Noyes, S.P.: Calculation of next time for track update in the MESAR phased array radar. IET Conf. Proc. (1), 2 (1998)
9. Sinha, A., et al.: Track quality based multitarget tracking algorithm. Sig. Data Proc. Small Targ. 6236, 623609 (2006). International Society for Optics and Photonics
10. Llinas, J., et al.: Revisiting the JDL Data Fusion Model II (2004)
11. Charlish, A., Katsilieris, F.: Array Radar Resource Management. Institution of Engineering and Technology (2017)
12. Orman, A.J., et al.: Scheduling for a multifunction phased array radar system. Eur. J. Oper. Res. 90(1), 13–25 (1996). https://doi.org/10.1016/0377-2217(95)00307-x
13. Shaghaghi, M., Adve, R.S., Ding, Z.: Multifunction cognitive radar task scheduling using Monte Carlo tree search and policy networks. IET Radar, Sonar Navig. 12(10), 1437–1447 (2018). https://doi.org/10.1049/iet-rsn.2018.5276
14. Haykin, S.: Cognitive radar: a way of the future. IEEE Signal Process. Mag. 23(1), 30–40 (2006). https://doi.org/10.1109/msp.2006.1593335
15. Masood, U., Farooq, H., Imran, A.: A machine learning based 3D propagation model for intelligent future cellular networks. In: 2019 IEEE Global Communications Conference (GLOBECOM), pp. 1–6 (2019)
16. Ibrahim, M., et al.: Embracing complexity: agent-based modeling for HetNets design and optimization via concurrent reinforcement learning algorithms. IEEE Trans. Netw. Service Manag. 18(4), 4042–4062 (2021). https://doi.org/10.1109/tnsm.2021.3121282
17. Hashmi, U.S., et al.: Towards real-time user QoE assessment via machine learning on LTE network data. In: 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall), pp. 1–7 (2019)
18. Naveed, M.H., et al.: Assessing deep generative models on time series network data. IEEE Access 10, 64601–64617 (2022). https://doi.org/10.1109/access.2022.3177906
19. Yu, D., Deng, L.: Deep learning and its applications to signal and information processing [exploratory DSP]. IEEE Signal Process. Mag. 28(1), 145–154 (2011). https://doi.org/10.1109/msp.2010.939038
20. Bishop, C.M.: 'Pattern Recognition and Machine Learning: All "Just the Facts 101" Material'. Information Science and Statistics. Private Limited, Springer India (2013)
21. Brewka, G.: Artificial intelligence—a modern approach by stuart Russell and Peter norvig, prentice Hall. Series in artificial intelligence, Englewood cliffs, NJ. Knowl. Eng. Rev. 11(1), 78–79 (1996). https://doi.org/10.1017/s0269888900007724
22. DARPA Contract to Apply Machine Learning to the Radio Frequency Spectrum. (2018). ref. No. 151/2018
23. DARPA.: Behavior Learning for Adaptive Electronic Warfare (BLADE) (2010). (DARPA-BAA-10-79)
24. DARPA.: Adaptive Radar Countermeasures (ARC) (2012). (DARPA-BAA-12-54)
25. Huang, G., et al.: Specific emitter identification based on nonlinear dynamical characteristics. Can. J. Electr. Comput. Eng. 39(1), 34–41 (2016). https://doi.org/10.1109/cjece.2015.2496143
26. Zhang, M., et al.: Neural networks for radar waveform recognition. Symmetry 9(5), 75 (2017). https://doi.org/10.3390/sym9050075
27. Chierchia, G., et al.: SAR image despeckling through convolutional neural networks. In: 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 5438–5441 (2017)
28. Wang, J., et al.: Ground target classification in noisy SAR images using convolutional neural networks. IEEE J. Sel. Top. Appl. Earth Obs. Rem. Sens. 11(11), 4180–4192 (2018). https://doi.org/10.1109/jstars.2018.2871556
29. Ma, Y., et al.: SAR target recognition based on transfer learning and data augmentation with LSGANs. In: 2019 Chinese Automation Congress (CAC), pp. 2334–2337 (2019)
30. Wang, Z., et al.: SAR target detection based on SSD with data augmentation and transfer learning. Geosci. Rem. Sens. Lett. IEEE 16(1), 150–154 (2019). https://doi.org/10.1109/lgrs.2018.2867242
31. Furukawa, H.: Deep Learning for Target Classification from SAR Imagery: Data Augmentation and Translation Invariance. CoRR (2017). abs/1708.07920
32. Zhao, Z., et al.: Discriminant deep belief network for high-resolution SAR image classification. Pattern Recogn. 61, 686–701 (2017). https://doi.org/10.1016/j.patcog.2016.05.028
33. Metcalf, J., Blunt, S.D., Himed, B.: A machine learning approach to cognitive radar detection. In: 2015 IEEE Radar Conference (RadarCon), pp. 1405–1411 (2015)
34. Liu, Z., et al.: Moving target indication using deep convolutional neural network. IEEE Access 6, 65651–65660 (2018). https://doi.org/10.1109/access.2018.2877018
35. Kang, L., et al.: Reinforcement learning based anti-jamming frequency hopping strategies design for cognitive radar. In: 2018 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), pp. 1–5 (2018)
36. Wang, L., et al.: Reinforcement learning-based waveform optimization for MIMO multi-target detection. In: 2018 52nd Asilomar Conference on Signals, Systems, and Computers, pp. 1329–1333 (2018)
37. Elbir, A.M., Mishra, K.V., Eldar, Y.C.: Cognitive radar antenna selection via deep learning. IET Radar, Sonar Navig. 13(6), 871–880 (2019). https://doi.org/10.1049/iet-rsn.2018.5438
38. Zappone, A., Di Renzo, M., Debbah, M.: Wireless networks design in the era of deep learning: model-based, AI-based, or both? IEEE Trans. Commun. 67(10), 7331–7376 (2019). https://doi.org/10.1109/tcomm.2019.2924010
39. Zhang, C., Patras, P., Haddadi, H.: Deep learning in mobile and wireless networking: a survey. IEEE Commun. Surv. Tutor. 21(3), 2224–2287 (2019). https://doi.org/10.1109/comst.2019.2904897
40. Jiang, C., et al.: Machine learning paradigms for next-generation wireless networks. IEEE Wireless Commun. 24(2), 98–105 (2017). https://doi.org/10.1109/mwc.2016.1500356wc
41. Luong, N.C., et al.: Applications of deep reinforcement learning in communications and networking: a survey. IEEE Commun. Surv. Tutor. 21(4), 3133–3174 (2019). https://doi.org/10.1109/comst.2019.2916583
42. Luong, N.C., et al.: Radio resource management in joint radar and communication: a comprehensive survey. IEEE Commun. Surv. Tutor. 23(2), 780–814 (2021). https://doi.org/10.1109/comst.2021.3070399
43. Lang, P., et al.: A Comprehensive Survey of Machine Learning Applied to Radar Signal Processing (2020)
44. Ding, Z.: A survey of radar resource management algorithms. In: 2008 Canadian Conference on Electrical and Computer Engineering, pp. 001559–001564 (2008)
45. Miranda, S.L.C., et al.: Phased array radar resource management: a comparison of scheduling algorithms. In: Proceedings of the 2004

IEEE Radar Conference (IEEE Cat. No. 04CH37509), pp. 79–84 (2004)
46. Miranda, S.L.C., et al.: Fuzzy logic approach for prioritisation of radar tasks and sectors of surveillance in multifunction radar. IET Radar, Sonar Navig. 1(10), 131–141 (2007). https://doi.org/10.1049/iet-rsn:20050106
47. Bar-Shalom, Y.: Multitarget-multisensor Tracking: Advanced Applications (1990)
48. Charlish, A., et al.: The development from adaptive to cognitive radar resource management. IEEE Aero. Electron. Syst. Mag. 35(6), 8–19 (2020). https://doi.org/10.1109/maes.2019.2957847
49. Moo, P.W., Ding, Z.: Coordinated radar resource management for networked phased array radars. IET Radar, Sonar Navig. 9(11), 1009–1020 (2015). https://doi.org/10.1049/iet-rsn.2013.0368
50. Blair, W.D.: Multitarget tracking metrics for SIAP systems. In: International Conference On Info Fusion-Fusion08. (2008)
51. Stoffel, A.P.: Heuristic energy management for active array multifunction radars. In: Proceedings of IEEE National Telesystems Conference - NTC '94, pp. 71–74 (1994)
52. Shannon, C.E.: A mathematical theory of communication. Bell Syst. Tech. J. 27(3), 379–423 (1948). https://doi.org/10.1002/j.1538-7305.1948.tb01338.x
53. Hero, A.O., et al.: Foundations and Applications of Sensor Management. Springer (2008)
54. Kreucher, C., Hero, A.O., Kastella, K.: A comparison of task driven and information driven sensor management for target tracking. In: Proceedings of the 44th IEEE Conference on Decision and Control, pp. 4004–4009 (2005)
55. Charlish, A., Woodbridge, K., Griffiths, H.: Information theoretic measures for MFR tracking control. In: 2010 IEEE Radar Conference, pp. 987–992 (2010)
56. Gan, H.S., Wirth, A.: Comparing deterministic, robust and online scheduling using entropy. Int. J. Prod. Res. 43(10), 2113–2134 (2005). https://doi.org/10.1080/00207540412331333405
57. Christodoulou, S., Ellinas, G., Aslani, P.: Entropy-based scheduling of resource-constrained construction projects. Autom. ConStruct. 18(7), 919–928 (2009). https://doi.org/10.1016/j.autcon.2009.04.007
58. Berry, P.E., Fogg, D.A.B.: On the use of entropy for optimal radar resource management and control. In: 2003 Proceedings of the International Conference on Radar (IEEE Cat. No.03EX695), pp. 572–577 (2003)
59. La Scala, B.F., Moran, B.: Optimal target tracking with restless bandits. Digit. Signal Process. 16(5), 479–487 (2006). special Issue on DASP 2005. https://doi.org/10.1016/j.dsp.2006.04.008
60. Krishnamurthy, V., Evans, R.J.: Hidden Markov model multiarm bandits: a methodology for beam scheduling in multitarget tracking. IEEE Trans. Signal Process. 49(12), 2893–2908 (2001). https://doi.org/10.1109/78.969499
61. Wintenby, J.: 'Resource Allocation in Airborne Surveillance Radar', PhD dissertation, Chalmers University of Technology, Sweden, (2003)
62. Ghosh, S.: 'Scalable QoS-Based Resource Allocation', PhD dissertation, Dept of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, (2004)
63. Lee, C., et al.: On quality of service optimization with discrete QoS options. In: Proceedings of the Fifth IEEE Real-Time Technology and Applications Symposium, pp. 276–286 (1999)
64. Ghosh, S., et al.: Integrated QoS-aware resource management and scheduling with multi-resource constraints. R. Time Syst. 33(1-3), 7–46 (2006). https://doi.org/10.1007/s11241-006-6881-0
65. Kuo, C.F., Kuo, T.W., Chang, C.: Real-time digital signal processing of phased array radars. IEEE Trans. Parallel Distr. Syst. 14(5), 433–446 (2003). https://doi.org/10.1109/tpds.2003.1199062
66. Shih, C.S., et al.: Scheduling real-time dwells using tasks with synthetic periods. In: RTSS 2003. 24th IEEE Real-Time Systems Symposium, 2003, pp. 210–219 (2003)
67. Ghosh, S., et al.: Adaptive QOS optimizations with applications to radar tracking. In: Proceedings of the 10th International Conference on Real-Time and Embedded Computing Systems and Applications, Gothenburg, Sweden (2004)
68. Hansen, J., et al.: Resource management for radar tracking. In: 2006 IEEE Conference on Radar, pp. 8 (2006)
69. Kershaw, D.J., Evans, R.J.: Waveform selective probabilistic data association. IEEE Trans. Aero. Electron. Syst. 33(4), 1180–1188 (1997). https://doi.org/10.1109/7.625110
70. Howard, S., Suvorova, S., Moran, B.: Optimal policy for scheduling of Gauss-Markov systems. In: Proceedings of the Seventh International Conference on Information Fusion, pp. 888–892. Citeseer (2004)
71. Suvorova, S., Howard, S.D., Moran, W.: Beam and waveform scheduling approach to combined radar surveillance & tracking — the paranoid tracker. In: 2006 International Waveform Diversity & Design Conference, pp. 1–5 (2006)
72. La Scala, B.F., Moran, W., Evans, R.J.: Optimal adaptive waveform selection for target detection. In: 2003 Proceedings of the International Conference on Radar (IEEE Cat. No.03EX695), pp. 492–496 (2003)
73. Scala, B.L., Rezaeian, M., Moran, B.: Optimal adaptive waveform selection for target tracking. In: 2005 7th International Conference on Information Fusion, vol. 1, pp. 6 (2005)
74. Sowelam, S.M., Tewfik, A.H.: Waveform selection in radar target classification. IEEE Trans. Inf. Theor. 46(3), 1014–1029 (2000). https://doi.org/10.1109/18.841178
75. Benoudnine, H., et al.: Fast adaptive update rate for phased array radar using IMM target tracking algorithm. In: 2006 IEEE International Symposium on Signal Processing and Information Technology, pp. 277–282 (2006)
76. van Keuk, G., Blackman, S.S.: On phased-array radar tracking and parameter control. IEEE Trans. Aero. Electron. Syst. 29(1), 186–194 (1993). https://doi.org/10.1109/7.249124
77. Koch, W.: Adaptive parameter control for phased-array tracking. In: Signal and Data Processing of Small Targets 1999, vol. 3809, pp. 444–455. International Society for Optics and Photonics (1999)
78. Hong, S.M., Jung, Y.H.: Optimal scheduling of track updates in phased array radars. IEEE Trans. Aero. Electron. Syst. 34(3), 1016–1022 (1998). https://doi.org/10.1109/7.705920
79. Sarker, I.H.: Machine learning: algorithms, real-world applications and research directions. Sprin. J. SN Comput. Sci. 2(160), 160 (2021). https://doi.org/10.1007/s42979-021-00592-x
80. Dong, S., Wang, P., Abbas, K.: A survey on deep learning and its applications. Comput. Sci. Rev. 40, 100379 (2021). https://doi.org/10.1016/j.cosrev.2021.100379
81. Moo, P., Ding, Z.: Coordinated Radar Resource Management for Networked Phased Array Radars. Technical Memorandum DRDC Ottawa (2013)
82. Schlangen, I., Jung, S., Charlish, A.: A non-Markovian prediction for the GM-PHD filter based on recurrent neural networks. In: 2020 IEEE Radar Conference (RadarConf20), pp. 1–6 (2020)
83. Kozy, M., et al.: Applying deep-Q networks to target tracking to improve cognitive radar. In: 2019 IEEE Radar Conference (RadarConf), pp. 1–6 (2019)
84. Deng, J., et al.: Supervised learning based online filters for targets tracking using radar measurements. In: 2020 IEEE Radar Conference (RadarConf20), pp. 1–6 (2020)
85. Bauw, M., et al.: From unsupervised to semi-supervised anomaly detection methods for HRRP targets. In: 2020 IEEE Radar Conference (RadarConf20), pp. 1–6 (2020)
86. Ruff, L., Robert, A.: Vandermeulen and Nico Garnitz and Alexander Binder and Emmanuel Maller and Klaus-Robert Maller and Marius Kloft. Deep semi-supervised anomaly detection (2020)
87. Ristea, N.C., et al.: Automotive radar interference mitigation with unfolded robust PCA based on residual overcomplete auto-encoder blocks. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 3203–3208 (2021)
88. Ristea, N.C., Anghel, A., Ionescu, R.T.: Fully convolutional neural networks for automotive radar interference mitigation. In: 2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall), pp. 1–5 (2020)

89. Nuhoglu, M.A., Alp, Y.K., Akyon, F.C.: Deep learning for radar signal detection in electronic warfare systems. In: 2020 IEEE Radar Conference (RadarConf20), pp. 1–6 (2020)
90. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Trans. Signal Process. 45(11), 2673–2681 (1997). https://doi.org/10.1109/78.650093
91. Kerbaa, T.H., et al.: CNN-LSTM based approach for parameter estimation of K-clutter plus noise. In: 2020 IEEE Radar Conference (RadarConf20), pp. 1–6 (2020)
92. Giovanneschi, F., Rosenberg, L., Cristallini, D.: Online dictionary learning techniques for sea clutter suppression. In: 2020 IEEE Radar Conference (RadarConf20), pp. 1–6 (2020)
93. Dästner, K., et al.: Machine learning support for radar-based surveillance systems. IEEE Aero. Electron. Syst. Mag. 36(7), 8–25 (2021). https://doi.org/10.1109/maes.2020.3001966
94. Pedregosa, F., et al.: Scikit-learn: Machine Learning in Python (2018)
95. Wrabel, A., Graef, R., Brosch, T.: A survey of artificial intelligence approaches for target surveillance with radar sensors. IEEE Aero. Electron. Syst. Mag. 36(7), 26–43 (2021). https://doi.org/10.1109/maes.2021.3065069
96. Liu, P., et al.: Decentralized automotive radar spectrum allocation to avoid mutual interference using reinforcement learning. IEEE Trans. Aero. Electron. Syst. 57(1), 190–205 (2021). https://doi.org/10.1109/taes.2020.3011869
97. Selvi, E., et al.: Reinforcement learning for adaptable bandwidth tracking radars. IEEE Trans. Aero. Electron. Syst. 56(5), 3904–3921 (2020). https://doi.org/10.1109/taes.2020.2987443
98. Thornton, C.E., et al.: Deep reinforcement learning control for radar detection and tracking in congested spectral environments. IEEE Trans. Cogn. Commun. Netw. 6(4), 1335–1349 (2020). https://doi.org/10.1109/tccn.2020.3019605
99. Huizing, A.G., Spruyt, J.A.: Adaptive waveform selection with a neural network (radar performance optimisation). In: 92 International Conference on Radar, pp. 419–421 (1992)
100. Smith, G.E., et al.: Neural networks & machine learning in cognitive radar. In: 2020 IEEE Radar Conference (RadarConf20), pp. 1–6 (2020)
101. Saarinen, V., Koivunen, V.: Radar waveform synthesis using generative adversarial networks. In: 2020 IEEE Radar Conference (RadarConf20), pp. 1–6 (2020)
102. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein GAN (2017)
103. Kershaw, D.J., Evans, R.J.: Optimal waveform selection for tracking systems. IEEE Trans. Inf. Theor. 40(5), 1536–1550 (1994). https://doi.org/10.1109/18.333866
104. Thornton, C.E., Buehrer, R.M., Martone, A.F.: Waveform Selection for Radar Tracking in Target Channels with Memory via Universal Learning (2021)
105. Ailiya, Yi, W., Yuan, Y.: Reinforcement learning-based joint adaptive frequency hopping and pulse-width allocation for radar anti-jamming. In: 2020 IEEE Radar Conference (RadarConf20), pp. 1–6 (2020)
106. Axelsson, S.R.J.: Analysis of random step frequency radar and comparison with experiments. IEEE Trans. Geosci. Rem. Sens. 45(4), 890–904 (2007). https://doi.org/10.1109/tgrs.2006.888865
107. Land, A.H., Doig, A.G.: An automatic method of solving discrete programming problems. Econometrica 28(3), 497–520 (1960). https://doi.org/10.2307/1910129
108. Shaghaghi, M., Adve, R.S.: Machine learning based cognitive radar resource management. In: 2018 IEEE Radar Conference (RadarConf18), pp. 1433–1438 (2018)
109. Shaghaghi, M., Adve, R.S., Ding, Z.: Multifunction cognitive radar task scheduling using Monte Carlo tree search and policy networks. IET Radar, Sonar Navig. 12(10), 1437–1447 (2018). https://doi.org/10.1049/iet-rsn.2018.5276
110. Silver, D., et al.: Mastering the game of go with deep neural networks and tree search. Nature 529(7587), 484–489 (2016). https://doi.org/10.1038/nature16961
111. Silver, D., et al.: Mastering the game of Go without human knowledge. Nat 550(7676), 354–359 (2017). https://doi.org/10.1038/nature24270
112. Shaghaghi, M., Adve, R.S., Ding, Z.: Resource management for multifunction multichannel cognitive radars. In: 2019 53rd Asilomar Conference on Signals, Systems, and Computers, pp. 1550–1554 (2019)
113. Gaafar, M., et al.: Reinforcement learning for cognitive radar task scheduling. In: 2019 53rd Asilomar Conference on Signals, Systems, and Computers, pp. 1653–1657 (2019)
114. Pulkkinen, P., et al.: Time budget management in multifunction radars using reinforcement learning. In: 2021 IEEE Radar Conference (RadarConf21), pp. 1–6 (2021)
115. Daeipour, E., Bar.Shalom, Y., Li, X.: Adaptive beam pointing control of a phased array radar using an IMM estimator. In: Proceedings of 1994 American Control Conference - ACC '94, vol. 2, pp. 2093–2097 (1994)
116. Xu, L., Zhang, T.: Reinforcement learning based dynamic task scheduling for multifunction radar network. In: 2020 IEEE Radar Conference (RadarConf20), pp. 1–5 (2020)
117. George, T., Wagner, K., Rademacher, P.: Deep Q-network for radar task-scheduling problem. In: 2022 IEEE Radar Conference (RadarConf22), pp. 1–5 (2022)
118. Joung, J.: Machine learning-based antenna selection in wireless communications. IEEE Commun. Lett. 20(11), 2241–2244 (2016). https://doi.org/10.1109/lcomm.2016.2594776
119. Rajkumar, R., et al.: A resource allocation model for QoS management. In: Proceedings Real-Time Systems Symposium, pp. 298–307 (1997)
120. Charlish, A.: Autonomous Agents for Multi-Function Radar Resource Management (2011)
121. Durst, S., Braggenwirth, S.: Quality of service based radar resource management using deep reinforcement learning. In: 2021 IEEE Radar Conference (RadarConf21), pp. 1–6 (2021)
122. Mnih, V., et al.: Asynchronous methods for deep reinforcement learning. In: Balcan, M.F., Weinberger, K.Q. (eds.) Proceedings of the 33rd International Conference on Machine Learning. vol. 48 of Proceedings of Machine Learning Research, pp. 1928–1937. PMLR, New York (2016)
123. Dulac-Arnold, G., et al.: Reinforcement learning in large discrete action spaces. CoRR (2015). abs/1512.07679
124. Sengupta, A., et al.: A review of recent advancements including machine learning on synthetic aperture radar using millimeter-wave radar. In: 2020 IEEE Radar Conference (RadarConf20), pp. 1–6 (2020)
125. Ma, Y., Li, Y., Zhu, L.: Land cover classification for polarimetric SAR image using convolutional neural network and superpixel. Prog. Electromagn. Res. B 83, 111–128 (2019). https://doi.org/10.2528/pierb18112104
126. Shan, W., et al.: Successive stripe artifact removal based on robust PCA for millimeter wave automotive radar image. In: 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pp. 1391–1394 (2019)
127. Wang, P., Patel, V.M.: Generating high quality visible images from SAR images using CNNs. In: 2018 IEEE Radar Conference (RadarConf18), pp. 0570–0575 (2018)
128. Lay, B., Charlish, A.: Classifying LPI signals with transfer learning on CNN architectures. In: 2020 Sensor Signal Processing for Defence Conference (SSPD), pp. 1–5 (2020)
129. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition (2015)
130. Barbaresco, F., Brooks, D., Adnet, C.: Machine and deep learning for drone radar recognition by micro-Doppler and Kinematic criteria. In: 2020 IEEE Radar Conference (RadarConf20), pp. 1–6 (2020)
131. Thammakhoune, S., et al.: Moving target imaging for synthetic aperture radar via RPCA. In: 2021 IEEE Radar Conference (RadarConf21), pp. 1–6 (2021)
132. Hieu, N.Q., et al.: Transferable Deep Reinforcement Learning Framework for Autonomous Vehicles with Joint Radar-Data Communications (2021)
133. Pei, J., et al.: Scanning radar target reconstruction using deep convolutional neural network. In: 2020 IEEE Radar Conference (RadarConf20), pp. 1–6 (2020)

134. Li, J., Qu, C., Shao, J.: Ship detection in SAR images based on an improved faster R-CNN. In: 2017 SAR in Big Data Era: Models, Methods and Applications (BIGSARDATA), pp. 1–6 (2017)
135. Brooks, D., et al.: Deep learning and information geometry for drone micro-Doppler radar classification. In: 2020 IEEE Radar Conference (RadarConf20), pp. 1–6 (2020)
136. Brüggenwirth, S., et al.: Cognitive radar for classification. IEEE Aero. Electron. Syst. Mag. 34(12), 30–38 (2019). https://doi.org/10.1109/maes.2019.2958546
137. Rasmussen, J.: Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models. IEEE Trans. Syst. Man. Cybern. SMC-13(3), 257–266 (1983). https://doi.org/10.1109/tsmc.1983.6313160
138. Benedetti, A., et al.: Using neural networks for change detection and classification of COSMO-SkyMed Images. In: 2020 IEEE Radar Conference (RadarConf20), pp. 1–5 (2020)
139. Blasch, E., et al.: Machine learning/artificial intelligence for sensor data fusion - opportunities and challenges. IEEE Aero. Electron. Syst. Mag. 36(7), 80–93 (2021). https://doi.org/10.1109/maes.2020.3049030
140. Chandrasekar, V.: AI in weather radars. In: 2020 IEEE Radar Conference (RadarConf20), pp. 1–3 (2020)
141. de Oliveira, M.L.L., Bekooij, M.J.G.: Generating synthetic short-range FMCW range-Doppler maps using generative adversarial networks and deep convolutional autoencoders. In: 2020 IEEE Radar Conference (RadarConf20), pp. 1–6 (2020)
142. Apfeld, S., Charlish, A.: Recognition of unknown radar emitters with machine learning. IEEE Trans. Aero. Electron. Syst. 57(6), 1–4447 (2021). https://doi.org/10.1109/taes.2021.3098125
143. Kong, G., Koivunen, V.: Radar waveform recognition using fourier-based synchrosqueezing transform and CNN. In: 2019 IEEE 8th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), pp. 664–668 (2019)
144. Anfinsen, S.N., et al.: Unsupervised heterogeneous change detection in radar images by cross-domain affinity matching. In: 2020 IEEE Radar Conference (RadarConf20), pp. 1–6 (2020)
145. Eykholt, K., et al.: Robust physical-world attacks on deep learning visual classification. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1625–1634 (2018)
146. Wang, Y., et al.: Generalizing from a few examples: a survey on few-shot learning. ACM Comput. Surv. 53(3), 1–34 (2020). https://doi.org/10.1145/3386252
147. Taylor, M.E., Stone, P.: Transfer learning for reinforcement learning domains: a survey. J. Mach. Learn. Res. 10(56), 1633–1685 (2009)
148. Madden, M.G., Howley, T.: Transfer of experience between reinforcement learning environments with progressive difficulty. Artif. Intell. Rev. 21(3/4), 375–398 (2004). https://doi.org/10.1023/b:aire.0000036264.95672.64
149. Chalmers, E., et al.: Context-switching and adaptation: brain-inspired mechanisms for handling environmental changes. In: 2016 International Joint Conference on Neural Networks (IJCNN), pp. 3522–3529 (2016)
150. Banerjee, B., Stone, P.: General game learning using knowledge transfer. In: The 20th International Joint Conference on Artificial Intelligence, pp. 672–677 (2007)
151. Ribeiro, M.T., Singh, S., Guestrin, C.: "Why should I trust you?": explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '16, pp. 1135–1144. Association for Computing Machinery, New York (2016)
152. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., et al. (eds.) Computer Vision – ECCV 2014, pp. 818–833. Springer International Publishing, Cham (2014)
153. Letham, B., et al.: Interpretable classifiers using rules and Bayesian analysis: building a better stroke prediction model. Ann. Appl. Stat. 9(3) (2015). https://doi.org/10.1214/15-aoas848
154. Lake, B.M., Salakhutdinov, R., Tenenbaum, J.B.: Human-level concept learning through probabilistic program induction. Science 350(6266), 1332–1338 (2015). https://doi.org/10.1126/science.aab3050
155. Shanahan, M., et al.: An Explicitly Relational Neural Network Architecture (2020)
156. Vapnik, V.N.: Statistical Learning Theory. Wiley-Interscience (1998)
157. Yekkehkhany, B., et al.: A comparison study of different kernel functions for SVM-based classification of multi-temporal polarimetry SAR data. Int. Arch. Photogram. Rem. Sens. Spatial Inf. Sci. 40(2), 281–285 (2014). https://doi.org/10.5194/isprsarchives-xl-2-w3-281-2014
158. Bakker, J.: 'Intelligent Traffic Classification for Detecting DDoS Attacks Using SDN/OpenFlow', 1–142 (2017)
159. Box, G.E., Tiao, G.C.: Bayesian Inference in Statistical Analysis, vol. 40. John Wiley & Sons (2011)
160. Quinlan, J.R.: Induction of decision trees. Mach. Learn. 1(1), 81–106 (1986). https://doi.org/10.1007/bf00116251
161. Karatsiolis, S., Schizas, C.N.: Region based support vector machine algorithm for medical diagnosis on Pima Indian diabetes dataset. In: 2012 IEEE 12th International Conference on Bioinformatics Bioengineering (BIBE), pp. 139–144 (2012)
162. Burrows, W.R., et al.: CART decision-tree statistical analysis and prediction of summer season maximum surface ozone for the Vancouver, Montreal, and Atlantic regions of Canada. J. Appl. Meteorol. Climatol. 34(8), 1848–1862 (1995). https://doi.org/10.1175/1520-0450(1995)034<1848:cdtsaa>2.0.co;2
163. Breiman, L.: Random forests. Mach. Learn. 45(1), 5–32 (2001). https://doi.org/10.1023/a:1010933404324
164. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '16, pp. 785–794. Association for Computing Machinery, New York (2016)
165. Adebayo, S.: How the Kaggle Winners Algorithm XGBoost Algorithm Works (2020)
166. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE 77(2), 257–286 (1989). https://doi.org/10.1109/5.18626
167. Fisher, R.A.: The use of multiple measurements in taxonomic problems. Ann. Eugen. 7(2), 179–188 (1936). https://doi.org/10.1111/j.1469-1809.1936.tb02137.x
168. Alpaydin, E.: Introduction to Machine Learning Cambridge. MIT Press, Mass (2004)
169. Tan, P., Steinbach, M., Kumar, V.: Introduction to Data Mining. Pearson Addison Wesley 8, 13 (2006). Citado
170. Guthikonda, S.: Kohonen Self-Organizing Maps. Wittenberg University. Wittenberg University, Wittenberg (2005)
171. Hashmi, U.S., Darbandi, A., Imran, A.: Enabling proactive self-healing by data mining network failure logs. In: 2017 International Conference on Computing, Networking and Communications (ICNC), pp. 511–517 (2017)
172. Bengio, Y.: Learning Deep Architectures for AI. Now Publishers Inc (2009)
173. Vincent, P., et al.: Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J. Mach. Learn. Res. 11(12) (2010)
174. Kingma, D.P., Welling, M.: Auto-encoding Variational Bayes. International Conference on Learning Representations (2013)
175. Goodfellow, I., et al.: Generative adversarial networks. Commun. ACM 63(11), 139–144 (2020). https://doi.org/10.1145/3422622
176. Sutton, R., Barto, A.: Reinforcement Learning: An Introduction', pp. 1 (1998)
177. Kaelbling, L.P., Littman, M.L., Moore, A.W.: Reinforcement learning: a survey. J. Artif. Intell. Res. 4, 237–285 (1996). https://doi.org/10.1613/jair.301
178. Sutton, R.S.: Learning to predict by the methods of temporal differences. Mach. Learn. 3(1), 9–44 (1988). https://doi.org/10.1007/bf00115009

179. Chaslot, G., et al.: Monte-carlo tree search: a new framework for game AI. In: AIIDE (2008)
180. Larson, R.: Dynamic programming with reduced computational requirements. IEEE Trans. Automat. Control 10(2), 135–143 (1965). https://doi.org/10.1109/tac.1965.1098129
181. Sutton, R.S.: Dyna, an integrated architecture for learning, planning, and reacting. SIGART Bull 2(4), 160–163 (1991). https://doi.org/10.1145/122344.122377
182. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature 521(7553), 436–444 (2015). https://doi.org/10.1038/nature14539
183. Werbos, P.: 'Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences', Ph D dissertation, Harvard University, (1975)
184. Hinton, G.E.: Deep belief networks. Scholarpedia 4(5), 5947 (2009). https://doi.org/10.4249/scholarpedia.5947
185. Jeong, C.M., Jung, Y.G., Lee, S.J.: Deep belief networks based radar signal classification system. J. Ambient Intell. Hum. Comput., 1–13 (2018). https://doi.org/10.1007/s12652-018-0774-7
186. Liu, S., et al.: Radar emitter recognition based on SIFT position and scale features. IEEE Trans. Circuits Syst. II: Express Br. 65(12), 2062–2066 (2018). https://doi.org/10.1109/tcsii.2018.2819666
187. Ciresan, D.C., et al.: High performance convolutional neural networks for image classification. In: Twenty-second International Joint Conference on Artificial Intelligence (2011)
188. Lecun, Y., et al.: Gradient-based learning applied to document recognition. Proc. IEEE 86(11), 2278–2324 (1998). https://doi.org/10.1109/5.726791
189. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. 25, 1097–1105 (2012)
190. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. CoRR (2015). abs/1409.1556
191. Szegedy, C., et al.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: Thirty-first AAAI Conference on Artificial Intelligence (2016)
192. He, K., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2015)
193. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT press (2016)
194. Gers, F.A., Schmidhuber, J., Cummins, F.: Learning to forget: continual prediction with LSTM. In: 1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470), vol. 2, pp. 850–855 (1999)
195. Arulkumaran, K., et al.: Deep reinforcement learning: a brief survey. IEEE Signal Process. Mag. 34(6), 26–38 (2017). https://doi.org/10.1109/msp.2017.2743240
196. Mnih, V., et al.: Human-level control through deep reinforcement learning. Nature 518(7540), 529–533 (2015). https://doi.org/10.1038/nature14236
197. Khan, A.A., Adve, R.S.: Centralized and distributed deep reinforcement learning methods for downlink sum-rate optimization. IEEE Trans. Wireless Commun. 19(12), 8410–8426 (2020). https://doi.org/10.1109/twc.2020.3022705
198. Koza, J.R., et al. (eds.) Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming, pp. 151–170. Springer Netherlands, Dordrecht (1996)

How to cite this article: Hashmi, U.S., et al.: Artificial intelligence meets radar resource management: a comprehensive background and literature review. IET Radar Sonar Navig. 17(2), 153–178 (2023). https://doi.org/10.1049/rsn2.12337

APPENDIX A

Primer on machine learning

A.1 | Conventional machine learning

A.1.1 | Supervised learning

These models use a data set with a set of features (or observations) and the labels/classes for those set of features. For instance, for image data, the observations may be images of animals with the classes being the name of the animal. The goal of these algorithms is to take the observed data as input and predict the output value (in the case of regression), or class (in the case of classification problems). In using neural networks to achieve these goals, the process involves optimising the Neural network (NN) parameters by training the model with a sufficient number of labelled data samples.

One issue of importance is that of overfitting, which is caused by the model memorising the training data instead of learning the more general mapping from features to labels. This generally occurs by training a model with a large number of parameters such that it can fit nearly any dataset. The overfitting can be addressed by reducing the complexity of the learning model. For example, in neural networks, the overfitting is combated with the use of fewer layers, fewer neurons per layer, or regularisation techniques like dropout.

Below, we briefly discuss some of the most commonly used supervised learning algorithms:

(i) K-nearest neighbour: K-nearest neighbour is one of the fundamental supervised ML algorithms used in both regression and classification problems. K-nearest neighbour is based on feature similarity measures to estimate the value or label of a test data point. Without making any assumptions on the distribution of vectors representing the data points, it uses distance calculation measures and the variable k, which represents the number of closest neighbouring data points to be considered to make any decision. For instance, in classification-based problems, for a test data point, the mode class for the k nearest neighbours is associated with the data point. Choosing an appropriate value of k is important in obtaining reliable results. Some of the prominent distance measures used in kNN are the cosine similarity and Euclidean distance.

(ii) Linear Regression: For a labelled dataset, linear regression fits a linear relationship between the features of the given data points and the labels. Simple linear regression refers to the use of a single input variable, while multiple linear regression deals with multiple input variables. It is assumed that the effect of independent variables on the value of label is additive in nature. In linear regression, the distances or residuals between a straight line and all the data observations are minimised via least squares (not maximum likelihood) estimation, thereby eliminating the assumption of the errors to be normally distributed. These errors or residuals are however assumed to be statistically independent and homoscedastic.

(iii) Logistic Regression: Logistic regression is another commonly used algorithm for classification problems. The

algorithm employs a logistic function to estimate the label of unknown data points. The algorithm scores each test data point on the probability of falling into a particular class. Unlike linear regression, the hypothesis in logistic regression limits the cost function between the values of 0 and 1. Instead of MSE measure in the loss function of linear regression, logistic regression uses the log-loss function along with gradient descent algorithm to optimise the model parameters. This technique faces the drawback of being unable to handle a large number of categorial variables and more complex relationships between the features.

(iv) Support Vector Machine (SVM): Proposed in 1998 [156], SVM is a discriminative classifier defined by a separating hyperplane, which is the decision boundary separating the classes. Support vector machines can separate the classes, which cannot be separated by a linear classifier, by using the kernel induced feature space. Using a kernel function it transforms data from input space into a high-dimensional feature space in which it searches for a separating hyperplane, so, that non-linear data can also be separated using hyperplane in high dimensional space. The aim is to select a hyperplane which maximises the distance between support vectors (data points closest to the hyperplane) and the hyperplane. Kernel selection is a critical task in SVMs that depends on the nature of the input dataset. If the dataset is linearly separable, a linear kernel may be used, otherwise polynomial and RBF-based SVM classifiers show better performance [157].

(v) Naive Bayes (NB): The Naive Bayes (NB) algorithm is a classification technique that finds the class label of an observation from the set of features with the assumption of independence among predictors [158]. In simple terms, the NB algorithm assumes that there is no effect of any feature other than the particular feature under consideration. It is based on Bayes' theorem which calculates the probability of an event occurring using a priori knowledge of the environment. Mathematically, the conditional probability is expressed as:

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)}. \tag{1}$$

In the given equation, H is a hypothesis, E represents new evidence, $P(E|H)$ is the posterior probability of occurrence of E conditioned on hypothesis H, and $P(H|E)$ is the probability of a test data sample belonging to a class. Due to its simple design, the algorithm may be trained well using a small number of training data points [159].

(vi) Decision Tree (DT): The idea behind DT is to train models that can predict the class or value of the target variable using decision rules inferred from the training data. A DT has a flowchart-like structure with the root at the tip and the tree splits into branches/edges on a condition. The terminal node which does not split further is the decision/leaf of the DT. The paths from root to leaves represent the classification rules. Decision trees models in which the target variable is categorical in nature are called classification trees, whereas those in which the target variable can take continuous values are known as regression trees. DTs are simple to implement and have intuitive knowledge expression. Some commonly used DT classifiers are ID3 [160], C4.5 [161], and Classification and Regression Tree [162].

(vii) Random Forest (RF): DTs are known to be prone to overfitting in situations when the tree is particularly deep. Random forests mitigate this problem by using a collection of randomly created DTs whose results are aggregated into a single final result [163]. This process of using multiple trees in a single process is called ensemble learning. While DTs may give importance to a particular feature set, RFs are robust as features are chosen randomly during the training. Because of this, RFs are capable of limiting overfitting without a substantial increase in the error due to bias. Random forests classifiers have the following procedure: (i) each data point is passed through every tree in RF; (ii) each tree gives a classification result; (iii) the data point is classified as the label which has most votes. Although RFs are more accurate than DTs, they are not easy to interpret, have a longer training time and also not suitable for high dimensional data.

(viii) Extreme Gradient Boosting (XGBoost): Extreme gradient boosting, like RF also combines multiple DT, but in this case, ensemble learning is used which aggregates multiple weak models (or trees), iterates these weak learners by assigning weights to each tree which yields a final model [164]. Instead of random selection, each model is an improvement over the previous iteration. While RF builds each model independently and applies majority rule at the end, gradient boosting builds one tree at a time and uses ensembling along the way to improve the shortcomings of existing weak learners. The aim here is to simultaneously reduce two losses: training loss and regularisation loss given by:

$$obj(\theta) = L(\theta) + \omega(\theta). \tag{2}$$

Here L represents training loss and ω, regularisation loss. Extreme gradient Boost is considered as the state-of-the-art in gradient boosted trees and commonly considered as the top algorithm in many Kaggle competitions [165].

(ix) Hidden Markov Model (HMM): A HMM is a statistical model used to describe the evolution of observable events which depend on multiple non-observable internal factors. An HMM is based on a MC where each state generates one out of multiple visible observations. Hidden Markov Model, like other Markov models, obeys the memoryless property, which basically indicates that the conditional probability distribution of future states only depends on current states and is independent of all past states [166]. HMMs are used in multiple ML-based applications, such as speech and handwriting recognition.

(x) Linear Discriminant Analysis (LDA): Linear Discriminant Analysis (LDA), a technique first developed in 1936 [167], is a dimensionality reduction technique used in preprocessing and classification tasks. Linear Discriminant Analysis projects features from a higher dimensional space onto a lower-dimensional space to avoid the effects of the curse of dimensionality and efficient ML modelling. Linear Discriminant Analysis models assume a Gaussian distribution of variables and use Bayes' theorem to estimate probabilities of

class association. Some notable extensions to LDA include quadratic discriminant analysis, flexible discriminant analysis, and regularised discriminant analysis.

A.1.2 | Unsupervised learning

In contrast to supervised learning, unsupervised learning algorithms work with unlabelled datasets. This means that there are no labels or output values associated with data points. The basic aim of these algorithms is to find hidden patterns and structures within the datasets. Additionally, unsupervised learning can also provide insights such as correlation or the inter-dependence between different features. Most common applications of unsupervised learning include clustering and data aggregation [168]. Below, we briefly discuss some of the most commonly used unsupervised learning algorithms:

(i) K-means: K-Means is the most popular unsupervised learning algorithm and a partition clustering technique that clusters the given data in K clusters where K is the user-specified number of clusters [169]. The idea is to find K centres, called cluster centroids for each cluster. On seeing a new example, the algorithm assigns the closest cluster to which the example belongs based on a distance measure such as Euclidean distance. Then a new centroid for each cluster is obtained. The centroid re-assignments and updates are repeated until the position of the centroid does not change significantly. The optimal cluster centroid is the one which minimises the sum of squares error (SSE) of the clustered dataset, mathematically given by

$$c_k = \frac{1}{m_k} \sum_{x \in C_k} x_k, \tag{3}$$

where C_k denotes a cluster association, c_k denotes the cluster centroid, x_k are the data point coordinates and m_k are the number of data points within a cluster. The elbow method is commonly used to determine a suitable value of K which is the point when SSE decrease becomes linear as K is incremented.

(ii) Fuzzy C-Means (FCM): This is another clustering technique similar to K-means, with the only difference being that in K-means, a data point either belongs to a set or not, whereas in Fuzzy C-Means (FCM), each data point has a degree of association with every cluster between 0 and 1. The sum of association weights with all classes is 1 for each data point [169]. Similar to K-means, FCM also minimises the SSE via updating centroids and assigning each data point to the closest centroid calculated by

$$c_j = \frac{\sum_{i=1}^{n} w_{ij}^{p} x_i}{\sum_{i=1}^{n} w_{ij}^{p}}. \tag{4}$$

Here w_{ij} is the association weight for data point x_i belonging to class j and p is the rate of weight.

(iii) Self-Organizing Map (SOM): Kohonen self-organising maps are an unsupervised neural network structure that perform dimension reduction and clustering through the mapping of node weights to conform to the input data presented to the network [170]. SOMs are composed of two layers: (i) input layer, and (ii) map layer. SOMs are represented using a 2-dimensional structure of neurons where every neuron has two properties: its position in the map (x, y co-ordinates) and its CB vector. The CB vectors are dimensionally equal to the input data.

Self-Organising Map training is carried out in three distinct phases [171]:

(i) Initialisation: In the first step, each CB vector is assigned a value for every dimension using random initialisation, or linear initialisation in which the CB vector is formed by the eigenvectors associated with the two greatest eigenvalues.
(ii) Coarse Training: This is the first phase of Self-Organising Map training which is characterised by a higher neighbourhood radius and learning rate with a fewer number of epochs.
(iii) Fine Training: This training round contains a higher number of epochs and a relatively small radius around the best matching neuron.

(iv) Principal Component Analysis (PCA): Principal Component Analysis is an unsupervised exploratory data analysis and dimensionality reduction algorithm. The main target of PCA is to determine vectors, referred to as principal components, that explain the largest amount of variance in the features of the dataset. Principal Component Analysis allows representation of data as linear combination of the principal components. Principal Component Analysis is also useful for noise reduction as a large number of features can be represented using a small number of PCs. While LDA is a supervised algorithm which maximises separation between different classes within the dataset, PCA is independent of class labels and maximises the variance using features in the dataset.

(i) Auto Encoders (AEs): Autoencoders are an unsupervised form of neural networks used for representation learning. The input data is compressed to a low-dimensional representation of the original input data. An autoencoder is composed of three components, namely encoder, code and decoder. The encoder is a form of NN that is employed to learn a compressed representation of the raw data. The decoder is another NN which attempts to recreate the input from the compressed data given by the encoder. Auto Encoders are often used to learn a compact representation of the data for dimension reduction [172]. There are further extensions to the AE models, such as the Denoising Auto-Encoder (DAE) used for initialising DNN weights [173], and Variational Auto-Encoders (VAEs) which generate virtual examples from a target data distribution [174].

(ii) Generative Adversarial Networks (GANs): First introduced in 2014 [175], GAN or GANs comprise two competing agents, typically neural networks, referred to as the generator and discriminator. The generator network attempts to transform a noise vector, $\mathbf{z}$, to fit the probability distribution of the input data, whereas the discriminator attempts to

accurately distinguish the generated data from the real data. The loss convergence of the generator and discriminator terminates the training period. Essentially, the two networks are jointly involved in a two player min-max game up until the discriminator fails to distinguish between the real data and the generated data, a point denoted by attainment of a Nash equilibrium. Mathematically, this process is expressed as:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data(x)}}[\log(D(x))] + \mathbb{E}_{z\sim p_z(z)}[1-\log(D(G(z))].$$

Here $\mathbf{x} \sim p_{data}(\mathbf{x})$ denotes that the input data vector $\mathbf{x}$ follows the probability density function, $p_{data}(\mathbf{x})$, log $(D(\mathbf{x}))$ is a measure of the predicted output of the discriminator for x_i, $\log(D(G(\mathbf{z}))$ is the output of the discriminator on the GAN generated data $G(\mathbf{z})$. The aim is to maximise the ability of the discriminator to identify real data from generator produced data, so we maximise this value, whereas the generator part of the equation tries to minimise the discriminator's ability to correctly classify real and fake data. This translates to maximising the first term and minimising the second term of the given equation.

A.1.3 | Reinforcement learning (RL)

Reinforcement learning algorithms refer to the category of algorithms in which an agent learns in *an interactive environment* using trial and error and feedback from its own actions and experiences [176, 177]. Specifically, it comprises environment, state, reward, value, and policy components. Markov decision processes provide the mathematical framework to describe the RL environment, which comprises a finite set of states $\mathscr{S}$, action set $\mathscr{A}$, reward function $R(s)$, and a transition model. Reinforcement learning uses rewards as feedback for positive and negative behaviour.

Essentially, RL methods learn from both good and bad outcomes in the training process without external labelled training data. A value function evaluates a state (or an action taken in a state) for all states, from which a policy can be derived. To determine the optimal policy, the agent explores new states or chooses actions that maximise its reward from the already known information. This dilemma is commonly known as the tradeoff between exploration versus exploitation. The key benefits of an RL model are of not requiring prelabelled data and its adaptability to varying environments as it trains itself on the fly via the specific reward function of the problem. The agent can use a single prediction of the next reward and next state via the model, or it can ask the model for the expected next reward, or the full distribution of next states and next rewards. Accordingly, RL techniques can be categorised as "model-free" or "model-based".

(i) Model-Free RL: A model-free RL estimates the optimal policy without using or estimating the dynamics (transition and reward functions) of the environment. Model-free methods primarily rely on learning through experience without requiring the model of the environment, and in many cases they can be applied to simulated experience just as well as to real experience.

The model-free RL works as follows:

(i) The agent experiences the real environment,
(ii) Learn the value function (and/or the policy) from experience.

Model-free methods are much simpler than the model-based methods as they are not affected by biases in the design of the model. Some of the common model-free RL algorithms are described below:

(a) Q-Learning: Q-Learning is a Temporal Difference (TD) learning methodology in which the reward at a time instance is the combination of the immediate reward and discounted rewards in the future. In Q-learning, the Q function is maximised for each state. Using an epsilon greedy approach, the Q-values are updated, leading to a Q function $Q(s,a)$ value for every state $s \in \mathscr{S}$ and $a \in \mathscr{A}$. Once the Q-table is obtained, for each state, the action associated with the maximum Q-value is then selected. Importantly, Q-learning is restricted to the cases where the states and actions are discrete, that is, $\mathscr{S}$ and $\mathscr{A}$ are finite discrete sets.

(b) State–action–reward–state–action (*SARSA*): While Q-Learning (QL) is an off-policy algorithm, SARSA is an on-policy technique which takes the maximum possible reward of the new state and ignores the current policy being used. SARSA learns a near optimal policy by considering the reward if we follow a given policy to the next state. Since SARSA learns the action selection while learning, it receives a higher average reward per trial as compared to QL.

(c) Temporal Difference learning: Temporal Difference is an unsupervised learning technique in which the agent aims to predict the expected value of a variable present at the end of a sequence of states [178]. The learning is performed by bootstraping on the current estimate of the value function. While TD performs random sampling from the environment, like Monte Carlo methods, the difference is that TD adjusts predictions before the final outcome is known. The simplest version of TD is TD (0) in which the value function is updated at each step and the value of next state and reward is evaluated as well.

(ii) Model-Based RL: Model-based, as the name implies, has an agent trying to understand its environment and creating a model for it based on its interactions with this environment. In such a system, preferences take priority over the consequences of the actions, that is, the greedy agent will always try to perform an action that will get the maximum reward irrespective of what that action may actually cause.

The iterative cycle of model-based RL works as follows:

(i) The agent experiences the real environment,
(ii) A model is learnt from experience,
(iii) The simulated model is used to plan, which allows to estimate the policy (and/or value) function without directly interacting with the real environment.
(iv) Use this learnt policy to take real actions.

The model-based methods are easy to train with SL and can be very data efficient by achieving a better policy with

fewer environmental interactions as compared to model-free methods; however, the policy learnt can only be as good as the model. Moreover, they can be combined with model-free methods by using model-free method in the key model update step. Some of the common model-based RL algorithms are described below:

(a) Monte Carlo Tree Search: When discussing RL algorithms, it is pertinent that we include discussion about an algorithm used in decision processes, called MCTS. Monte Carlo Tree Search consists of four phases: (i) Selection, (ii) Expansion, (iii) Rollout/Simulation, and (iv) Backpropagation [179]. In the selection step, a tree node is selected that has the highest probability of winning. The selection function is run recursively until a leaf node is reached. Here the exploration-exploitation tradeoff takes place in an effort to select the best move and also explore unknown moves. In the expansion state, more nodes are created to increase options in the "move" step. The simulation state involves running the RL algorithm on every child node to evaluate how to progress in the tree. Backpropagation is the last step in which each tree node that was traversed during the game is updated using the reward accrued. Although MCTS has the advantage of not requiring domain knowledge, it is memory intensive as the tree grows after a few iterations.

(b) Dynamic programing: Dynamic programming is called model-based as it uses the model's predictions or distributions of next state (i.e. the state transition matrix) as well as a reward in order to calculate optimal actions. Dynamic programming, an important branch of operations research, was first introduced by Bellman in 1951 [180] to enable multi-stage decision making. First, policy evaluations or predictions are performed to compute the state-value function for an arbitrary policy. After the policy evaluation, policy improvement is done to find the optimal policy through convergence of an iterative policy process. Value iteration is obtained if the Bellman optimality equation is turned into an update rule.

(c) Dyna Algorithm: The Dyna algorithm, first introduced by Sutton [181], aims at integrated learning and planning by learning a model from experience. The aim is to learn and plan a value function or policy from real and simulated experiences. The learning distribution is used for model learning, while simulated experience is used for planning updates. The term search control is used to refer to the process to select states and actions for the simulated experiences generated by the model. Planning is achieved applying RL to the simulated experiences. Dyna-Q, which is an example of Dyna algorithms, combines both RL and model learning, such that planning is done by one-step tabular Q-planning and learning is one-step tabular Q-learning.

A.2 | Deep Learning (DL)

A.2.1 | Deep neural networks (DNN)

The optimisation of network parameters to yield optimal performance is a computationally intensive process. To tackle this high computational complexity and enable real-time network optimisation, learning-based approaches may be employed whereby the input and output of a resource allocation algorithm are fed into a DNN [182]. The DNN then learns the non-linear mapping between the two and approximates it for any new condition. DNN-enabled resource optimisation in wireless networks provides two advantages: (i) real-time optimisation is possible due to significantly lower computational requirements to retrieve the near-optimal parameters from a trained model; and (ii) model training can be performed offline by running the optimisation algorithm on a simulated data set, hence the training process does not interfere with the network's operation. In this section, we discuss some of the DL algorithms presented in Figure 2.

A.2.2 | Multi-layer perceptrons (MLPs)

A NN consists of layers that have perceptrons connected with each other through some weights. All inputs at a perceptron are weighted, aggregated and passed through an activation function, such as sigmoid, softmax and rectified linear unit. An Multilayer perceptron (MLP) is the standard form of forward neural network in which the perceptron's output is fed directly to the next perceptron. An MLP is capable of separating data into classes which are not linearly separable as long as it has sufficient depth. The training algorithm for an MLP is the gradient descent optimisation algorithm, also known as backpropagation [183].

A.2.3 | Deep belief networks (DBNs)

A DNN faces the issue of vanishing gradients with increasing depth of the DNN. This can be resolved by using a Deep Belief Network (DBN) which employs a divide and conquer approach. A restricted Boltzman machine (RBM) is first trained and then all the trained networks are combined using fine-tuning of the entire network. So in essence, a DBN is composed of stacked RBMs [184]. Each RBM has a single hidden and visible layer. Once stacked with a DBN, an RBM's hidden layer is not visible to its visible layer, but to the visible layer of the next RBM. In radar, DBN has been used to radar emitter recognition and classification [185, 186].

A.2.4 | Convolutional neural networks (CNNs)

A CNN is a DL algorithm especially used for image processing and classification tasks. A CNN is successfully able to capture the spatial and temporal dependencies through application of pre-filtering layers which are convolutional and pooling layers. In the convolution layer, input image is processed using convolution operation which extracts the features while compressing the information. In the pooling layer, the output from previous layers is combined into a single perceptron using maximum or average value [187]. The compressed feature information is fed to another MLP structure which yields the final classification result. Some famous CNN architectures include LeNet-5 [188], AlexNet [189], VGGNet [190], GoogleNetv4 [191], and ResNet [192].

A.2.5 | Recurrent neural networks (RNNs)

Recurrent Neural Networks are specifically designed to model time-series or sequential data, such that there is sequential correlation between data samples. Each layer produces an output via recurrent connections between hidden units [193]. Recurrent Neural Networks may have different activation functions for each layer. Each neuron in a recurrent layer combines different weights with the current input but the intermediate vector from the previous step. This will induce correlation among computations which is beneficial to exploit temporal correlations in the input time series data.

A.2.6 | Long short term memory networks (LSTMs)

Recurrent Neural Networks suffer from short-term memory, that is, for long sequences, they are not able to carry information from earlier steps. Long Short Term Memory Networks on the other hand are well-suited to longer data sequences since they allow for lags between events in a time series. Long Short Term Memory Networks have feedback connections that regulate the flow of information and through blocks, called gates, learn to retain important data patterns within sequences [194]. An LSTM consists of four gates, namely forget, memory, information and output. Each gate has a fully connected NN with a sigmoid activation function, except for the information gate, which is activated through a `tanh` function.

A.2.7 | Auto encoders (AEs)

Autoencoders are an unsupervised form of neural networks used for representation learning. The input data is compressed to a low-dimensional representation of the original input data. An autoencoder is composed of three components, namely encoder, code and decoder. The encoder is a form of NN that is employed to learn a compressed representation of the raw data. The decoder is another NN which attempts to recreate the input from the compressed data given by the encoder. Auto Encoders are often used to learn a compact representation of the data for dimension reduction [172]. There are further extensions to the AE models, such as the DAE used for initialising DNN weights [173], and VAEs which generate virtual examples from a target data distribution [174].

A.2.8 | Generative adversarial networks (GANs)

First introduced in 2014 [175], GAN or GANs comprise two competing agents, typically neural networks, referred to as the generator and discriminator. The generator network attempts to transform a noise vector, $\mathbf{z}$, to fit the probability distribution of the input data, whereas the discriminator attempts to accurately distinguish the generated data from the real data. The loss convergence of the generator and discriminator terminates the training period. Essentially, the two networks are jointly involved in a two player min-max game up until the discriminator fails to distinguish between the real data and the generated data, a point denoted by attainment of a Nash equilibrium. Mathematically, this process is expressed as:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data(x)}}[\log(D(x))] + \mathbb{E}_{z\sim p_z(z)}[1-\log(D(G(z))].$$

Here $\mathbf{x} \sim p_{data}(\mathbf{x})$ denotes that the input data vector $\mathbf{x}$ follows the probability density function, $p_{data}(\mathbf{x})$, $\log(D(\mathbf{x}))$ is a measure of the predicted output of the discriminator for x_i, $\log(D(G(\mathbf{z}))$ is the output of the discriminator on the GAN generated data $G(\mathbf{z})$. The aim is to maximise the ability of the discriminator to identify real data from generator produced data, so we maximise this value, whereas the generator part of the equation tries to minimise the discriminator's ability to correctly classify real and fake data. This translates to maximising the first term and minimising the second term of the given equation.

A.2.9 | Deep reinforcement learning (DRL)

Deep RL is a framework which combines DL with RL [195]. Like RL, the learning procedure is adaptive in the sense that agents interact with the environment, take actions and receive feedback on the result of those actions. Traditional RL techniques become infeasible when full knowledge about the environment for estimation of action-value is not available, or when the number of possible states and actions grow infinitely large. Neural networks solve this dilemma in DRL to estimate the action-value and/or the policy functions. The NN is trained to optimise its parameters, so as to select actions that yield the best future return. In the following, we describe a commonly used method of DRL, Deep-Q Learning:

Deep Q-Learning (DQL): Deep RL networks can be broadly classified as DQN and policy gradient networks. Like DRL, in DQN, DL is combined with principles of Q-learning such that the state action value function $(Q(s,a))$ is determined by a DNN [196]. In DQL, the agent targets augmentation of the cumulative future reward. Unlike traditional Q-learning which follows a linear model for the Q-function, in DQL, NN best universal approximators estimate the corresponding Q-function from the state action pair. Deep Q-Learning uses experience replay, which is the act of storing and replaying game states, in order to train the learning model in small batches and consequentially to avoid any skewing of the dataset distributions of different states.

Policy gradient methods, on the other hand, use the trial-and-error approach to determine an action policy based on the rewards. By trial-and-error, these approach uses gradient-based methods to improve the policy. Crucially, these methods can also be applied to continuous state and action spaces. Once trained, the policy determines the actions to be taken (given a state) [197].