Received: 12 April 2022 | Revised: 2 June 2022 | Accepted: 17 June 2022

IET Communications

DOI: 10.1049/cmu2.12447

The Institution of Engineering and Technology WILEY

REVIEW

# A survey on deep reinforcement learning architectures, applications and emerging trends

Surjeet Balhara[1] | Nishu Gupta[2] | Ahmed Alkhayyat[3] | Isha Bharti[4] | Rami Q. Malik[5] | Sarmad Nozad Mahmood[6] | Firas Abedi[7]

[1]Department of Electronics and Communication Engineering, Bharati Vidyapeeth's College of Engineering, New Delhi, India

[2]Department of Electronic Systems, Faculty of Information Technology and Electrical Engineering, Norwegian University of Science and Technology, Gjøvik, Norway

[3]College of Technical Engineering, The Islamic University, Najaf, Iraq

[4]Senior Business Analyst & Solution Architect, SAP Innovation and Technology, Capgemini America Inc., Irving, Texas, USA

[5]Medical Instrumentation Techniques Engineering Department, Al-Mustaqbal University College Hillah, Hillah, Iraq

[6]Department of Computer Engineering Techniques, College of Technical Engineering, Al-Kitab University Kirkuk, Kirkuk, Iraq

[7]Department of Mathematics, College of Education, Al-Zahraa University for Women Baghdad, Baghdad, Iraq

**Correspondence**
Nishu Gupta, Department of Electronic Systems, Faculty of Information Technology and Electrical Engineering, Norwegian University of Science and Technology in Gjøvik, Gjøvik, Norway.
Email: nishugupta@ieee.org

**Abstract**

From a future perspective and with the current advancements in technology, deep reinforcement learning (DRL) is set to play an important role in several areas like transportation, automation, finance, medical and in many more fields with less human interaction. With the popularity of its fast-learning algorithms there is an exponential increase in the opportunities for handling dynamic environments without any explicit programming. Additionally, DRL sophisticatedly handles real-world complex problems in different environments. It has grasped great attention in the areas of natural language processing (NLP), speech recognition, computer vision and image classification which has led to a drastic increase in solving complex problems like planning, decision-making and perception. This survey provides a comprehensive analysis of DRL and different types of neural network, DRL architectures, and their real-world applications. Recent and upcoming trends in the field of artificial intelligence (AI) and its categories have been emphasized and potential challenges have been discussed.

## 1 | INTRODUCTION

During the past decade, artificial intelligence (AI) became an emerging topic envisaging applications almost in every field. AI is applied in different fields, that is, computer vision, navigation, business management etc., in which agent learns on its own by interacting with the environment [1]. AI can efficiently run algorithms to handle and solve many large-scale optimization problems by interacting with different systems. The solid breakthrough in AI is machine learning (ML) in which the agent learns itself without being explicitly programmed. ML needs enormous amount of data for training, but in practical it is difficult to obtain such huge data. This problem is solved by deep learning (DL) and reinforcement learning (RL) which are subsets of ML. Both play significant role in solving various high-level problems in a complex environment. RL is more attractive in solving optimal trial error problems by using model-free learning and model-based methods. It also deals with datasets, data size and predicts the future outcomes based on the input data [2]. DL has become an efficient tool in solving feature extraction problems with various learning patterns from large datasets and also has tremendous applications in various fields such as healthcare [3]. A recent work that focused on application of deep learning in health care is to monitor the diabetes with wearable device and also DL has focused on food industry authors in [4] proposed a food recognition system using DL.

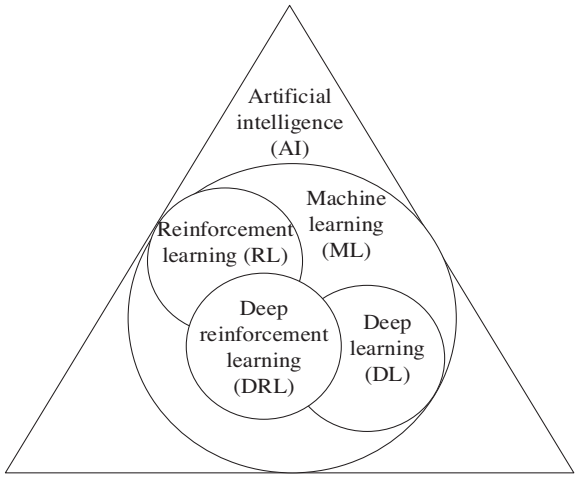


**FIGURE 1** Representation of different applications of AI

Furthermore, it also has applications in traffic prediction, congestion and alleviation which is discussed in [5]. It is inspired by artificial neural network (ANN) that made a benchmark in solving problems like image classification, object detection and time series data prediction by using various neural network like long short-term memory (LSTM), recurrent neural network (RNN), convolutional neural network (CNN) etc. These neural network are similar to neurons in human brain. One can also detect the signals of the neurons using a technique called spike sorting. There are various spike sorting algorithms, which are explained in [6]. Integration of one neural network with the other also be used for better results, one such example is in [7]. DRL is a combination of both DL and RL as shown in Figure 1. In RL, the agent can take decisions periodically by observing the rewards and automatically adjusting its strategy to obtain the optimal policy [8]. RL has applications in the fields of robotics, healthcare, education, games, and video games developed to generalize the agent abilities. The main goal of the agent is to know the capability of the enemies and maximize the coins (i.e. rewards) which are limited in practice [9]. Another subset, DL has a powerful encounter in many areas. It overcomes the drawbacks of RL such as improvement in voice recognition and language translation. Deep neural network (DNN) is the basis for DL working process. It has 2 phases: (i) training and (ii) inference phases. In the training phase, the network learns from the given data and the inference phase is the production phase where the trained model is installed for predicting solutions to real-world problems. Nonlinear transformations and model abstractions are used in the DL technique at a high level for large databases. We can perform even with small datasets [10].

In Table 1, we provide a list of acronyms that may be handy to the researchers to understand the various terminologies used in the context of the presented work.

To understand the complex attention of the task with maximum accuracy, systems use DL algorithms that continuously analyse the data with some logical structure like the human brain and draw conclusions. This combination of DL with RL has led to the development of DRL as shown in Table 2. DNNs contain several connected neurons and each neuron is associated with a weight. As neural network learning is an iterative process, the data increases after each iteration. The function of the neural network is like the human brain which collects and classifies the data according to a specific architecture. To efficiently address

**TABLE 1** Acronyms

| Acronym | Description |
|---|---|
| AI | Artificial intelligence |
| ANN | Artificial neural network |
| CNN | Convolutional neural network |
| DBNN | Deep belief neural network |
| DCNN | Deep convolutional neural network |
| DDPG | Deep deterministic policy gradient |
| DL | Deep learning |
| DNN | Deep neural network |
| DQN | Deep Q network |
| DRL | Deep reinforcement learning |
| DRRN | Deep reinforcement relevance network |
| DSNN | Deep stacked neural network |
| DDQN | Double deep Q network |
| D3QN | Dueling double deep Q network |
| FER | Facial expression recognition |
| FSML | Few short text classifications in meta-learning |
| FMDP | Finance Markov decision process |
| GRU | Gated recurrent unit |
| LIDAR | Light detection and ranging |
| LSTM | Long short-term memory |
| ML | Machine learning |
| MDP | Markov decision process |
| MAML | Model agnostic meta-learning |
| MC | Monte Carlo |
| MIMO | Multiple-input multiple-output |
| NLP | Natural language processing |
| NLSTM | Nested LSTM network |
| NN | Neural network |
| NTM | Neural turing machine |
| NNN | Non-neural network |
| PPO | Proximal policy optimization |
| RNN | Recurrent neural network |
| RvNN | Recursive neural network |
| RL | Reinforcement learning |
| RBM | Restricted Boltzmann machines |
| SUMO | Simulation of urban mobility |
| SARSA | State-action- reward- state-action |
| TCN | Temporal convolution network |
| TRPO | Trust region policy optimization |

**TABLE 2** Development of DRL

| | |
|---|---|
| AI | It is a field of computer science. That aims to solve real-world problems to acquire more success rates by preparing machines to perform the task. Here data plays a prominent role [12]. |
| ML | ML can help the machine to learn on its own without any explicit programming. It provides a balance between agent intelligence and the latest developed web technologies [13]. |
| RL | This is the subset of ML, where the main goal of the RL is to maximize the rewards from the actions by interacting with the environment [14]. |
| DL | A subset of ML, in which agent uses the ANN to learn autonomously from the available large amount of data for classification and regression [15]. |
| DRL | DRL is used as a powerful tool for improving the learning rate of RL algorithms. Best solutions are provided by DRL as it uses a double deep Q network (DDQN) which is the extension of the DQN. DDQNs are used to maximize the Q value from the actions performed by the agent [16]. |

real-world problems and challenges, DRL is used as an emerging tool. For example, the modification that is made to DRL is the development of an algorithm that learns to play Atari 2600 video game at superhuman level range directly from the pixels with deep Q network (DQN) [11].

## 1.1 | Goal

Enhancing the existing surveys on DRL which created a massive impact on different fields, the main motive behind this survey is to explore DRL in-depth and to contribute relevant information regarding the latest upgradation. Moreover, to address queries like 'Is DRL used for automated learning in different environments by adjusting its strategy with various situations with different architectures?', 'Are there any real-time applications of DRL?', 'What are different architectures used in DRL to solve real-world practical problems in various fields?' etc., this survey covers the RL methods with a few of its algorithms, DL and their architectures. It also focuses on the evolution of DRL (which is the integration of DL and RL) along with some real-time applications.

## 1.2 | Methodology

We go along with the systematic literature review guidelines in [17] to carry out this review. Bibliometric analysis is used in this approach. In this study, state-of-the-art research was supported by many databases. The publication time interval from 2018 to 2020 was considered. The predominant stages of the study were as follows.

### 1.2.1 | Search strategy

First, we began by searching the databases such as IEEE, Springer etc. with the common keywords, 'Machine learning', 'Deep learning', 'Reinforcement learning', 'Artificial intelligence', 'Deep learning architectures', 'Deep reinforcement learning', 'Deep reinforcement learning architectures' and 'Applications'. To achieve the consistency of the search results, similar keywords were used in all the databases during the time interval of 2018–2020.

### 1.2.2 | Material collection

The data is collected from the standard literature databases considering only journals in the English language and conferences. From the above-mentioned search terms, a total of 250 publications were identified from 2018 to 2022 in this review.

### 1.2.3 | Screening

In screening, duplicate articles are removed and the resultant 200 unique articles were sent to the next stage.

### 1.2.4 | Inclusion

The next step after screening is the inclusion step, where the importance of the screened articles is investigated. As a result, 140 articles were shortlisted for further consideration. In a further step, the authors read the entire articles (140) out of which 105 qualified for final review in this study. The present research database consists of 105 articles and all these articles have contributed to carry out this study.

### 1.2.5 | Exclusion

- Articles not in the English language.
- Full articles that are unavailable on the digital library.
- Research that does not address the eligibility.
- All types of Thesis work.
- Brief research papers.

## 1.3 | Organization of the paper

The organization of the paper is shown in Figure 2.

Section 2 presents the background in the context of RL and its algorithms. Section 3 presents a detailed survey and qualitative analysis of deep learning techniques and various neural network that pave the fundamentals of AI. Section 4 talk s about DRL and its applications. In Section 5, open issues and challenges have been discussed. Finally, the survey concludes in Section 6.

# 2 | BACKGROUND

John MC Carthy in the 1990′s defined AI as "AI is the science and engineering of making intelligent machines, especially


**FIGURE 2** Organization of the paper

intelligent computer programs". The word AI is used when the functioning of one human brain is associated with another human brain in the tasks like learning, problem solving etc. During the 21st century, AI is an important area of research in several fields. The scope for AI has grown exceedingly as the machine's intelligence with ML has created an intense impact on various fields like business, finance, computer vision etc. There are many subcategories of AI one of them is ML which has become a great research topic at present [17]. The two important phases in ML are training and testing. In the training phase, the algorithms are trained to achieve the right output. In ML, some algorithms are used for learning the training set to procure a model. These algorithms can be classified into two groups; neural network (NN) algorithms and non-neural network (NNN) algorithms. CNN, RNN, DNN and other neural network architectures come under NN algorithms whereas NNN algorithms deal with support vector machine, k-means, naive Bayes etc. [18]. ML is further classified into four categories, one amongst them is RL. RL deals with sequential decision-making that plots situations to actions by maximizing the reward within the bounds of its environment. The agent (learner) is not directly trained on which action to be taken at every step rather it should follow trial and error to point out the action for producing maximum reward [19]. One of the recent advancements made in ML is DL. It has extended extreme developments in the areas of layer design and network structure. The training of DL algorithms requires a large amount of data. DL has achieved great success in recent times and many DL techniques are being introduced every year. With these advancements, many changes took place in different fields including image classification, computer vision, object detection, NLP etc., which acquired a prominent breakthrough with the development of many network such as Le net, Google net, VGG net, Dense net [20]. For solving path planning and decision-making problems in RL, Q-learning and state-action-reward-state-action (SARSA) are used for single-agent applications but in multi-agent it is difficult to solve this problem. Hence, Shangfeizhern proposed "Improved Deep Deterministic Policy Gradient Algorithm" in DRL [21]. In wireless communication, the development of DRL provided solutions to routing problems in heavy traffic areas with "router selection MDP" instead of traditional MDP. This paper discusses the working of DL with some of its architectures and DRL and their real-world applications [22].

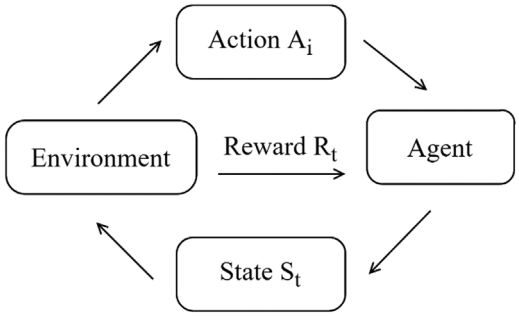


**FIGURE 3** Working of RL

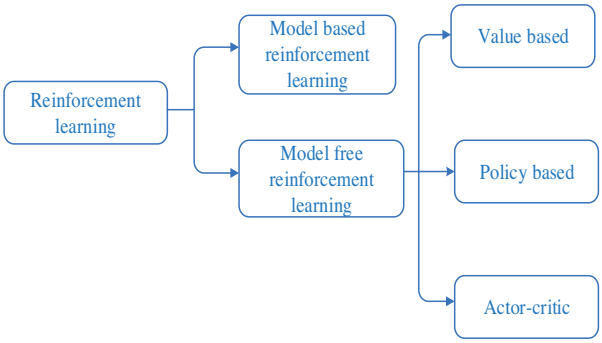


**FIGURE 4** Classification of RL

## 2.1 | Reinforcement learning

The integral of RL is to learn the active reciprocal action between the agent and the environment in which the agent operates. If the action performed by the agent motivates the environment to provide a positive reward, then the agent's propensity to execute that action is reinforced. The important terms in RL are agent, environment, action, state, and reward. The agent is the learner and the environment is the one with which the agent interacts to map situations into actions. Action is the decision taken by the agent to perform the task. State refers to the situation restored by the environment after performing a task. The reward is the feedback from the environment which is shown in Figure 3 [23, 24].

In RL, the agent does not have any prior knowledge about which action to perform. It learns by maximizing the rewards. An agent gets a reward or penalty based on its performance

**TABLE 3** Comparison of various model-free RL algorithms

| Algorithm | Description | ON policy | OFF policy |
|---|---|---|---|
| Q-learning | It is a value-based state-action-reward-state RL algorithm used for finding the optimal action selection policy using a Q-function [28]. | | ✓ |
| SARSA | It is a value-based state-action-reward-state-action (SARSA) RL algorithm. It is a modified Q-learning algorithm used for the Markov decision process (MDP) [29]. | ✓ | |
| DQN | A deep-Q network is an actor-critic algorithm used for learning discrete actions and also learns from replay buffer with previous experiences [30]. | | ✓ |
| DDPG | Deep deterministic policy gradient is an actor-critic algorithm used in the generation of temporal difference (TD) [31]. | | ✓ |
| PPO | Proximal policy optimization is an actor-critic algorithm that uses only confined updates to the policy for maintaining the stability of the learning process [31]. | ✓ | |
| TRPO | Trust region policy optimization is an actor-critic algorithm with two network one for estimating the policy and the other for estimating the advantages of the function [32]. | ✓ | |
| PEBL | Pessimistic ensembles for offline deep reinforcement learning is built on double deep Q-learning (DDQ) and uncertainty is estimated using a multi-headed bootstrap approach to calculate an effective pessimistic value penalty [36]. | ✓ | |

RL algorithms are divided into both model based and model free as shown in Figure 4.

by considering the feedback from the environment. The agent updates the action policy until it reaches the optimum policy. Policy illustrates the action to be taken from the state. Another paradigm of RL is trial and error. Agent observes the environment at each time step $T$ in a state $S_t$, and receives a reward or penalty after every action based on current policy [25–27]. Popular model-free RL algorithms are stated in Table 3.

## 2.2 | Conventionally used RL algorithms

### 2.2.1 | Markov decision process

Markov decision process (MDP) is the mathematical framework to determine an environment in RL that is used for making sequential decisions. In MDP, the agent observes the state '$S_t$' interacts with the environment and takes a random action '$A_t$' which maximizes the next state and receives feedback from the environment as a reward '$R_t$' at each time step $T$. Whenever the agent shifts from state '$S_t$' to next state '$S_{t+1'}$, based on the transition probability '$p$' a set of action, state and reward is generated from MDP. MDP depends only on the present but not on past variables. It consists of 5 elements in processing an RL problem which is mentioned below:

(i) An environment with which the agent interacts.
(ii) A set of states during the agent's interaction with the environment.
(iii) An agent within the environment performs all possible actions.
(iv) Depending upon the action taken by the agent an immediate reward is gained.
(v) The discount factor for calculating further rewards is associated with each transition [33].

In partial observable MDP, the agent's state is not fully connected. It maintains the present and past states in memory of the belief state. In this, the agent interacts with the environment partially for making optimal policy [34].

### 2.2.2 | Q learning

It is a model-free RL algorithm. The core idea behind the Q-learning algorithm is to impart the agent about the strategy to be taken under different environment conditions for taking optimal action in the Markov domain. The basis of Q-learning is a Q table in which rows and columns represent the values of state and actions and chooses the action according to Q value [35].

### 2.2.3 | Temporal difference

Sutton proposed TD algorithm in 1998. It is a model-free RL algorithm which is the combination of both Monte Carlo (MC) algorithm and dynamic programming technology that is used for solving forecast problems in RL, which are in time series. In the TD algorithm, the learning process uses the current action and immediate state to estimate the current state. It aims to maximize the reward by adjusting the strategy continuously while interacting with the environment [36].

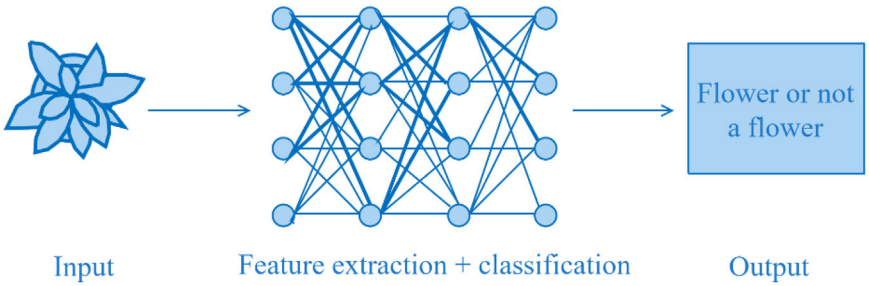


**FIGURE 5** Working of DL

### 2.2.4 | Monte Carlo

This is a model-free method that repeatedly estimates the value function at each state and the average return for each state–action pair is recorded. Knowledge of transition probabilities is not required for this method. This method learns the value function by executing the set of trials for every state and averages the rewards from the state [37].

## 2.3 | Challenges with reinforcement learning

Safety is an important parameter while considering system operations during the learning phase. In RL, due to the limited availability of data in the real world, algorithms are trained with a limited number of patterns during the learning phase. RL algorithms have many practical real-world problems with large and continuous state and action spaces. In many cases of RL direct training is not possible. In this case, an off-policy and off-line training system is used where training is done by the recent iterations of the algorithms [38].

# 3 | DEEP LEARNING

DL has acquired eminence because of its large training data and output accuracy. The main goal of DNNs in DL is to imitate the behaviour of the human brain to solve complex problems in a real-time environment [37, 38]. In DL, there is a fully connected structure of the deep neural network. This structure is very deep [39]. As conventional ML, DL is also split into two frames, unsupervised and supervised. Unsupervised learning deals with unlabelled data. Collection of unlabelled data is very easy while training is difficult. Restricted Boltzmann machines (RBMs) perform the training of unlabelled data. RBMs and auto encoders are the two learning models for unsupervised learning. Supervised learning deals with labelled data sets for training and building of system model as this system model learns the relationship between input and output. CNN and RNN are the two supervised learning models [40]. In DL, CNN is the most popular neural network [41]. Figure 5 illustrates the working of DL. In the below case the designer feds a large number of different types of flowers to the database. DNN classifies these flowers based on their colour, shape and size. Later, if any flower is given as input, then it produces the output based on the feature extraction.

A neural network is an interconnected structure with several neurons. Each neuron gets an input, undergoes a process, and produces the output. Every neuron of the output layer performs the sum of the input values that are received from the input neurons and generates the output with the help of nonlinear transformation functions. With back propagation, corrections in the network are made using stochastic gradient descent by considering the derivatives of errors at each neuron [42]. There are two approaches for training the data, supervised and unsupervised. Supervised training is used for solving the problems of regression and classification whereas unsupervised training is used for solving probabilistic distribution problems. There are two types of neural network models, discriminative and generative. In the discriminative model, the data transfer takes place from bottom to top i.e., from the input it flows via hidden layers and produces output. This model is used in supervised training of data whereas in the generative model the data flow takes place in the opposite direction, that is, top-down. This model is used in unsupervised training of data [43].

## 3.1 | Feed forward neural network

It is the simplest form of ANN. In this type of ANN, the data is transferred in the forward direction from input to output through hidden layers. It does not have cycles or loops in the network. It is a generally used neural network. In this type of ANN, one data is independent of other data. Here, the output depends upon the present input. It does not consider the past data for the calculation of present data. This mean that the old data is erased when the new data is entered [44]. Feed forward neural network is not able to understand sequential data since the understanding of sequential data requires knowledge of the previous data. RNN solves this problem.

## 3.2 | Recurrent neural network

RNN is a class of neural network that is used for processing temporal sequence data which may include a text or video. RNNs can remember the previous data and uses that data to generate the output for the next node as shown in Figure 6.

It consists of a feedback loop in its memory cell which depends on time but remembering of previous data is limited to a certain period. Long-term sequential data capturing is not possible in RNN. To overcome this problem, we use LSTM

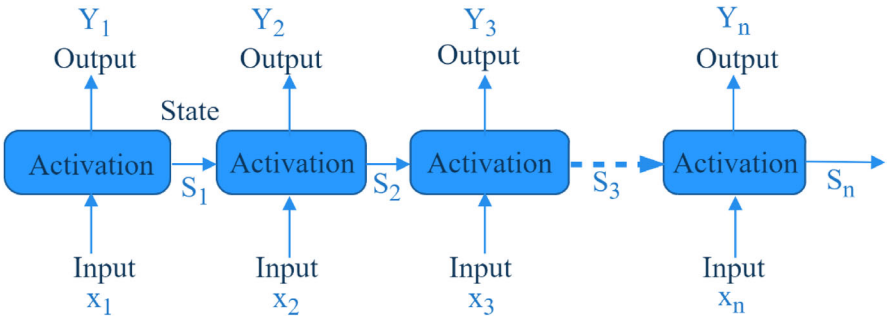


**FIGURE 6** Information flow in RNN

[45]. One more problem with RNN is gradient vanishing and gradient exploding which is observed in reverse propagation.

## 3.3 | Long short-term memory

The main aim for enabling the LSTM is to make RNN remember the previous data for a short or long time based on the importance of the event. LSTM is a subclass of RNN. In RNN, we have come across gradient vanishing and gradient exploding problems LSTM overcomes these problems. Another advantage of LSTM is it remembers the previous data for long time as RNN remembers it for a limited time only and forecasts the information to a certain distance. If the distance limit between the two sequences of data is reached then RNN cannot predict the next data from previous data. Hence, LSTM solves this problem [46]. It stores and memorizes the data even after 1000 gap intervals between the data. LSTM has long-term memory. This long-term memory is otherwise called an LSTM cell and was proposed by Hochreiter and Schmidhuber in 1997 for long-term dependencies. Later Schmidhuber in 2000 and Gers in 2001 modified it. LSTM has three gates to store the data they are forget gate, input gate and output gate. The function of the forget gate is to give instructions to the cell state about which information is to be stored and which is to be forgotten. If the output of forget gate is zero then the cell state multiplies zero to the matrix position and if the output is one then the information is stored in the cell state. Input gate sends the specified selected information or data to the cell state from the data stored in the forget gate and saves the information in it. Hence, it is also known as save vector. Output gate is also known as focus vector and is responsible for determining which data should be forwarded to the next state among all the data that is entered and saved in the input gate. Mainly LSTM network is classified into two types LSTM dominated network and integrated LSTM network [47, 48]. Tables 4 and 5 describe different types of LSTM dominated network and integrated LSTM network.

## 3.4 | Convolutional neural network

CNNs are the class of DNNs. They work similarly to the functioning of the visual cortex in the brain. CNN processes the data which is in the form of matrices such as an image. CNN became more significant due to its feature extraction application. If CNN receives 2D structure data, it processes it and produces the high-level abstraction with the help of pooling functions and a set of moving filters. The fundamental CNN architecture is built up with a convolutional, pooling and fully connected layers. The convolutional and pooling layers are responsible for feature extraction and the fully connected layer is responsible for classification or regression. The convolution layer contains the set of training data and the set of learning filters. Each filter is termed as kernel, this kernel is of small size (height, width) later they extend through the depth of the input. In pooling layer, down sampling operation is performed along the width and height of the input. Therefore, input volume is alleviated without the data loss. This pooling is of two types: average pooling and max pooling. Fully connected layer performs the final required operation (regression or classification). This layer gets the input from the previous convolution layers merges all the inputs and produces the single output vector [50, 51]. CNNs are used for processing the structural data which may include image feature extraction and non-linear function estimator. The architecture of CNN is shown in Figure 7. Before CNN, computer vision with handicraft features such as HAAR, a local binary pattern is used but it does not automatically learn the features during the training period, which will be done by CNN [52].

CNNs have wide range of applications in the areas of image classification, video and image recognition, and NLP [53]. Advancement of CNN is kervolutional neural network [55]. Different types of CNN models are shown in Table 6.

## 3.5 | Deep belief neural network

Deep belief neural network (DBNN). It is the advancement of the CNN wherein only the top layer learns the input whereas in DBNN, every layer learns the input. It consists of many hidden layers, connected to each other. DBNNs are arranged by several RBM's. It has two layers; one is visible and the other is hidden layer [60]. The connection is only between the layers but units inside the layers are not connected. Hence, this is not an intra-connection. The connection is both, undirected and directed connection. Undirected connection is established between the first and second layers whereas the remaining layers

**TABLE 4** Different types of LSTM-dominated network

| Network | Author | Year | Problems addressed and enhancements proposed |
|---|---|---|---|
| Bi-directional LSTM network | Gravesand Schmidhuber | 2005 | Adopted by Han, Wu, Jiang and Davin in 2017, Yu, Xu, and Zhang in 2018, Thoreau and Reczko in 2007 for predicting protein localization [49]. |
| Stacked LSTM network | Fernandez, Graves, and Schmidhuber | 2007 | Adopted by Du, Zhang, Nguyen and Han in 2017 for solving problems in vehicle-to-vehicle communication. |
| | | | Adopted by Sutskever, Vinyals and Le in 2014 to perform English-to-French translation task. |
| | | | Adopted by Saleh, Hossny and Nahavandi in 2018 for constructing a deep-stacked LSTM network for predicting the capacity of road users [49]. |
| Multi-dimensional LSTM (MDLSTM) | Graves, Fernandez, and Schmidhuber | 2007 | Adopted by Graves et al. for dealing air freight database. |
| | | | Adopted by Li, Mohamed, Zweig, and Gong in 2016 for constructing MDLSTM Network by using time-frequency LSTM cells [49]. |
| Grid LSTM network | Kalchbrenner, Danihelka and Graves | 2015 | Adopted by Li and Sainath in 2017 for lowering complexity in the computations [49]. |
| Convolutional LSTM network | Shi et. al | 2015 | Wei, Zhou, Sankaranarayanan, Sengupta and Samet in 2018 adopted Conv LSTM for solving tweet count prediction, a spatiotemporal sequence forecasting problem [49]. |
| Depth gated LSTM network | Yao, Cohn, Vylomova, Duh and Dyer | 2015 | Used by Yao et. al for performing the Chinese-to-English machine translation task. Adopted by Zhang, Chen et al. in 2015 and proposed algorithms for training DGLSTM and achieved better results [49]. |
| Gate feedback LSTM network | Chun, Gulcehre, Cho and Bengio | 2015 | Used by Chung, Gulcehre Cho, and Bengio for solving problems of learning multiple adaptive time scales [49]. |
| Tree-structured LSTM | Zhu, Sobhani and Gu, Tai, Socher, Manningo | 2015 | For solving the problem that occurred in chain structured LSTM, that is, the combining of words and phrases in NLP, chain structured LSTM exhibit poor properties. |
| | | | Later, Tai et al. in 2015 proposed two tree structured LSTM architecture [49]. |
| Graph LSTM network | Liang, Shen, Xiang et. al | 2016 | Adopted by Liang, Shen and Feng et al. in 2016 had evidenced the superiority of graph LSTM. |
| | | | network on four databases, that is, fashion dataset, PASCAL-Person part data set, ATR data set, and Horse–Cow parsing data set. |
| | | | Adopted by Liang et al. in 2017 and extended LSTM for learning the multilevel graph structures [49]. |
| Nested LSTM network (NLSTM) | Moniz and Krueger | 2018 | Used in the character-level language processing of NLSTM network [49]. |
| DRL-LSTM | S. Lakshmi Durga et al. | 2022 | LSTM neural network has been implemented for efficient allocation of channels with the help of deep DRL model. |

**TABLE 5** Various types of integrated LSTM network

| Network | Author | Problems solved |
|---|---|---|
| Neural turing machine (NTM) | Graves, Wayne and Danihelka | Adopted by Graves et al. in 2014 for constructing the neural network. |
| | | Xie and Shi in 2018 improved the training speed by introducing read write mechanism for NTM [49]. |
| DBNN-LSTM (combination of DBNN and LSTM) | Vohra, Goel and Sahao | Advancement made to RNN-LSTM. It deals with larger duration time [49]. |
| Multiscale LSTM network | Cheng et al. | He used this network for learning the traffic pattern of the internet [49]. |
| C-LSTM | Zhou, Wu, Zhang and Zhou | This network used in document modelling [49]. |
| LSTM-in-LSTM network | Song, Tang, Xiao, Wu, Zhang | These networks are used for fine grained textual representation of images [49]. |
| CFCC-LSTM | Yang et al. | Combination of fully connected LSTM, CFCLSTM and CNN. This was used for handling spatial information [49]. |

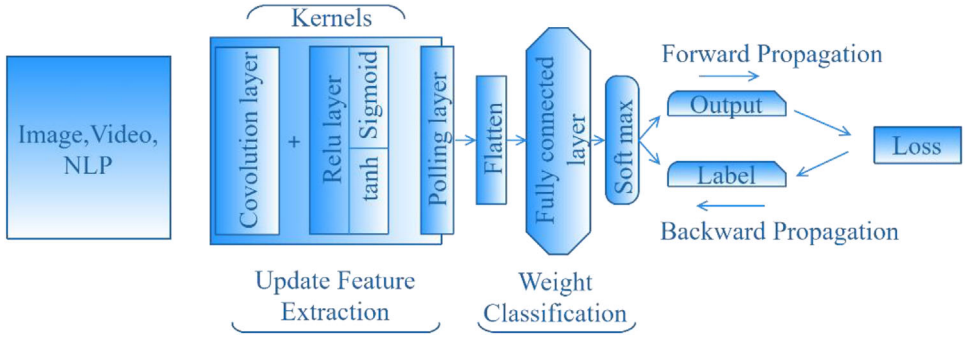


**FIGURE 7** Convolution neural network architecture

**TABLE 6** Different types of CNN models

| Model | Developed group | Application |
|---|---|---|
| Alex Net | Alex Krizheusky, Geoffrey Hinton | Object recognition [55] |
| ZF Net | Mathew Zeiler, Rob Fevgue | Image recognition [56] |
| VGG Net | Visual Geometry Group | Visual recognition [57] |
| Google Net | Christian Szegedy | Speech recognition [57] |
| Residual Net | Kaiming | Image recognition [57] |
| Squeeze Net | Researchers of Deep scale, University of California, Berkeley | Computer vision [58] |
| Mobile Net | Howard et al. | Pattern recognition [55] |
| Dense Net | Gao et al. | Image recognition [59] |

have directed connections. DBNNs are mostly used in recognition of video, image and in capturing the moving data but it is rarely used today due to drawbacks like lack of intra-connection between the units inside the layer [61, 62].

## 3.6 | Deep stacked neural network

Deep stacked neural network (DSNN) are also named as deep convex network. DSNN are designed by the combination of modules present in the network. These modules are generally used by the DSNN. It contains input, output and hidden layer. The output of one layer is given as input to the next layer. DSNN is used for reducing complexity in the classification. Each module is trained separately. The working of DSNN is found to be better as compared to DBNN [63].

## 3.7 | Gated recurrent unit

Gated recurrent unit (GRU) is the extension of LSTM and has a similar structure [54]. It also has gates in its architecture but unlike LSTM it has only two gates: reset gate and update gate. The reset gate specifies the amount of information that must be forgotten. Working of the update gate is like the working of input and forget gate in LSTM. It specifies the amount of data

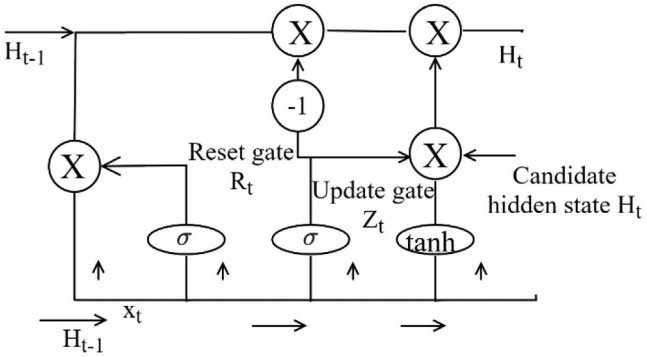


**FIGURE 8** Gated recurrent unit network

that must be passed to the next stage. The architecture of GRU is shown in Figure 8 [64].

There are primarily two gates in a GRU as opposed to three gates in an LSTM cell. The first gate is the Reset gate ($R_t$) and the other one is the Update gate ($Z_t$). At each timestamp 't', it takes an input $X_t$ and the hidden state $H_{t-1}$ from the previous timestamp 't−1′. It outputs a new hidden state $H_t$ which is passed to the next timestamp.

## 3.8 | Recursive neural network

Recursive neural network (RvNN) signifies pyramidal structure and classifies the outcome based on the compositional vectors. RvNNs are the structures that are used for processing the data which is in orbitary form like a tree. It takes recursive data as input. The main aim of this structure is to produce the output of fixed-width where the input is of variable size. Successful application of RvNNs is in the area of NLP [64]. The comparison of various DL architectures is listed in Table 7.

# 4 | DEEP REINFORCEMENT LEARNING

In DRL, the agent interacts with the environment and learns the optimal policy for mapping states to corresponding actions. This solves many real-world issues [70]. Here, the agent must be

**TABLE 7** Year and application of various DL techniques

| Year | Neural network | Application | Working |
|---|---|---|---|
| 1990–1995 | RNN (Recurrent neural network) | Speech recognition, handwriting recognition [65] | This is the basic architecture compared to other deep architectures. The recurrent network feed back into prior layer. Feedback allows RNN to maintain the memory of past inputs and model problems in time. |
| 1995–2000 | LSTM (Long short -term memory) | Natural language text compression, handwriting recognition, Speech recognition, gesture recognition, image captioning. [66] | In LSTM instead of neuron based neural architecture, The concept of memory cell was introduced. Memory cell consists of three gates forget gate, input gate, and output gate. |
| 1995–2000 | CNN (Convolution neural network) | Image recognition, document analysis, NLP, decoding facial recognition, understanding climate [66] | It was inspired biologically from animal visual cortex. It is a multilayer neural network. CNN is made of several layers used for implementation, feature extraction and classification. |
| 2005–2010 | DBNN | Image recognition, information retrieval, natural language understanding, predicts failures [67] | DBNN is network architecture, which include novel training algorithm. The input layer represents raw sensory inputs and hidden layer learns abstract representation of inputs whereas the output layer implements network classification. |
| 2010–2015 | DSNN (Deep stacked neural network) | Information retrieval, continuous speech recognition [67] | It is also called as deep convex network (DCN). It consists of set of modules, each of which is sub network in the overall hierarchy of the DSNN. |
| 2010–2015 | GRU (Gated recurrent unit) | Natural language text compression, handwriting recognition, speech recognition, gesture recognition, image captioning [68] | GRU is the simplification of LSTM. GRU has two gates an update gate and reset gate. The update gate indicates how much of the previous cell data is maintained. Whereas a reset gate defines how to incorporate the new input with the previous cell data. |
| 2015–2020 | RvNN (Recursive neural network) | Image and sentence deconstruction [69] | It is used for processing the data of variable lengths, and processing data structure inputs. It enables the users to not only identify constitutes of input data but also determine quantitative relationship between them. |

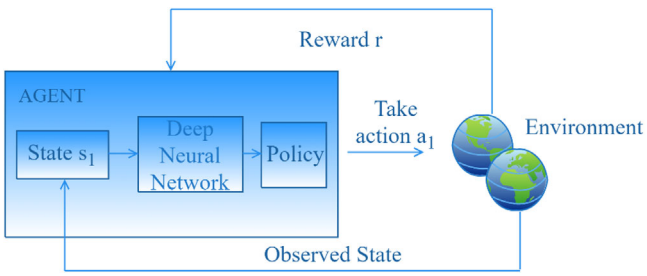


**FIGURE 9** Working of DRL

capable of sensing the present input state and its corresponding actions. With the rapid development of social networking sites, privacy and storage have become a serious issue as the transfer of a huge amount of data to the cloud has become difficult. Solutions to this problem are provided by DRL [71]. This technique is successful in the fields of robotics, finance, healthcare, videogames etc. Many previous unsolved problems were solved by this model [72]. To extract high dimensional observation features in RL, DRL is used where DNN is applied to it [73]. To increase robustness for easy convergence against overestimating value function, a DDQN and dueling architectures are used [74]. The agent performs action by observing the environment accordingly and output is achieved by employing a technique of deep neural network for approximating Q value for each state as shown in Figure 9.

## 4.1 | Deep reinforcement learning architectures

### 4.1.1 | Deep Q network

In DQN, we use DNN to estimate Q value. This network is used to satisfy the Bellman equation [21]. It is directly drawn from the Q-learning technique and it has several advantages when compared with the Q network. For example, in video games, learning of the optimal operation to achieve the best score through the screen images is performed by DQN. In this, you learn by collecting the information consisting of 'on what screen', 'with which operation', 'how the score will change' and 'what will be the next screen' by this analogy an algorithm is constructed. For any pair ($s$, $a$) with the current state '$s$' and the action '$a$', you will receive reward '$r$' and the next state '$s_1$'. Functioning of DQN is shown in Figure 10. One can train the model to the maximum level if sufficient data is available. It is impossible to collect the data for all possible pairs ($s$, $a$).

Experience replay and target network are the two main features of DQN that led to its success. State transitions, rewards and actions are stored in the experience replay which are essential in performing Q-learning and subdivide them into mini-batches for updating the neural network from previous memories. However, target network is based upon the generated target Q value the loss for every action during training process will be figured out. A huge action space problem in

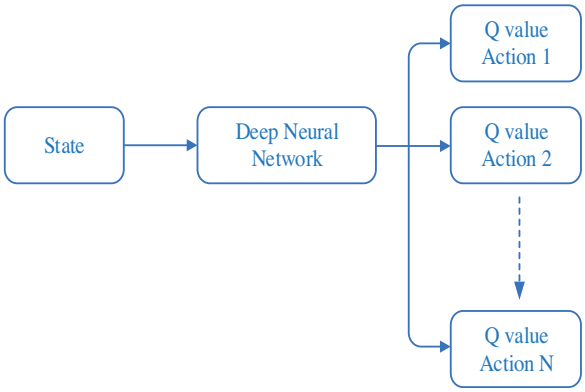


**FIGURE 10** Deep Q network

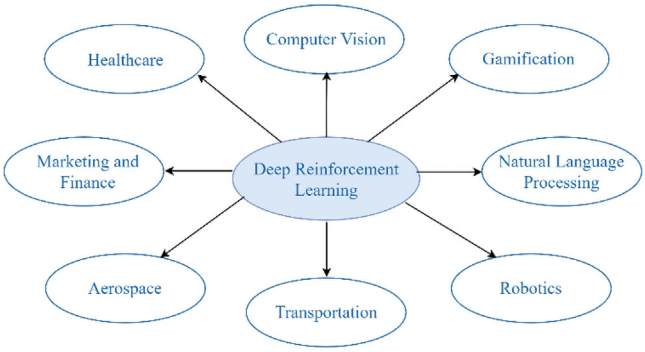


**FIGURE 11** Real-time applications of DRL

DQN has been overcome by using a deep reinforcement relevance network [75, 76]. DQN uses the TD algorithm for error loss calculation.

### 4.1.2 | Double deep Q network

DDQN was introduced by "Hado Van Hasselt". It is used to minimize the overestimation problem by decomposing the max operation in the target to action selection. It is the combination of both DNN and DQN. This method came into existence to overcome the problem of overestimation of Q values in previously discussed models. As we know that the best possibility for the next state is the action with a higher Q value but the Q value accuracy depends on what we tried, what outcome we have got, and what will be the next state for this trial. At the beginning of the experiment, we do not have enough Q values to estimate the best possibility. At this stage, as there are fewer Q values to estimate choosing, the highest Q value from the limited values may lead you to a false action towards the target. To overcome this problem, DDQN is used. Here, we use two DQNs, one for the selection of the Q value and the other calculates the target Q value for choosing that specific action using the target network. This DDQN helps to minimize the overestimation of the Q values which helps in reducing the training time [75, 77, 101].

### 4.1.3 | Dueling Q network

Dueling Q network is used to solve the problems in the DQN model by using two network, current network, and target network. The current network approximates the Q value. On the other hand, target network selects the next best action and performs the action chosen by the target. In some cases, it is not required to approximate the value of each action. For this, we use a dueling Q network. In some gaming settings, we choose left (or) right movement when a collision occurs whereas in some cases it is necessary to know which action is to be taken. We design a single Q network architecture which is referred to as a dueling network. Instead of using a single sequence following the convolution layer, we are using two sequences. These two sequences are used to separate estimation values, advantage function, and finally combine both, producing a single Q value. Thus, the output of the dueling network is the Q function which is trained with many existing algorithms including DDQN and SARSA [75, 78]. The advancement of the dueling deep Q network is dueling double deep Q network (D3QN) [79].

### 4.1.4 | Deep reinforcement relevance network

A new architecture is designed for handling action and state spaces in sequential text-based games. Deep reinforcement relevance network (DRRN) for natural language action space is based on DQN. This is used to learn the importance of the state and action in text-based games. This was designed to extract the meaning rather than memorizing text. As in DRL, it just performs the task of (i) designing both the action and states using neural network and (ii) computing the Q function in continuous space whereas in DRRN, it has a pair of neural network. One is used for state encapsulation, other for action encapsulation and these both are combined by the interaction function. While DRRN converges much faster than the other, it achieves a higher average reward and performs better in capturing relevance between state and action [80].

## 4.2 | Applications of deep reinforcement learning

Some real time applications of DRL are shown in Figure 11.

### 4.2.1 | Gamification

DRL has significant applications in games. After applying DRL to games like cart pole and mountain car, the masters have tested the gaming strategy and they found that it repeatedly won the game. In this case, the designers faced the challenges that came across during the gaming process and analysed every possible scenario of gaming by learning from wins,

losses and draw over sometimes. The designer feeds the neural network with thousands of rules and scenarios. The game itself starts with random play. After each play, the system analyses the result and sets the parameters of the neural network to become the strongest player. With the help of deep neurons, it makes its move very dynamically by increasing its gaming power [81].

#### 4.2.2 | Aerospace

DRL is extensively used in Aerospace. There have been a lot of research works with applications of AI in this area, like the works from Dario Izzo@ESA, Robert Furfaro@Arizona etc. [82]. The authors' focus is on evolutionary optimization, tree searches and machine learning, including deep learning and reinforcement learning as the key technologies and drivers for current and future research in the field. The article [83] proposes a CNN that is able to cope with changes in illumination, cloud coverage and landscape features, which are introduced by the fact that the different images are taken over successive satellite passages at the same region.

#### 4.2.3 | Robotics

DRL is applied in robotics for navigation of the mobile robot in an unfamiliar environment by avoiding the obstacles to reach the desired destination autonomously with an RGB-D camera by using a DDQN algorithm. Navigation of the mobile robot to the desired destination without using any maps is done by asynchronous deterministic policy gradients with light detection and ranging (LIDAR) and the commands will be provided to the mobile robot for avoiding obstacles by estimating the Q value from the DQN to know the depth of the image by using RGB-D sensor [79, 84]. DRL is also used in solving flocking control problems in multi robotic systems in complex environments using an algorithm called multi-agent DDPG, which helps the multi-robot system in performing a flocking task with greater convergence speed [85].

#### 4.2.4 | Transportation

Today traffic congestion is a serious issue in many metropolitan cities. To avoid congestion in traffic and to provide a smooth flow of traffic, transportation plays a significant role by reducing the delay, improving the traffic flow and optimizing fuel efficiency. Many transportation systems perform some dynamic actions which may route an agent from source to destination with the best route by reducing the delays in time and adjusting the traffic signals with the low delay. The vehicles based upon the environment and communication facilities have the ability to make decisions. Smart vehicles automatically slow the vehicle when there is traffic congestion, this is possible by providing vehicles with onboard radar which uses DQN to optimize the real-time traffic control policies [86].To solve many real-time problems and to provide better navigation when compared with some traditional routing algorithms, DRL-based real-time navigation and vehicle routing method is proposed by using simulation of urban mobility (SUMO) for training DNN to reroute vehicles to the destination in the real-time complex environment [87].

#### 4.2.5 | Marketing and finance

Due to the ambiguity in the financial data, there occurs fluctuations in the stock market. Therefore, prediction of stock market is a challenging task [88]. DRL is applied to finance in improving the profit in the finance market using some algorithms that enables the agent to learn how to achieve profits in any sector of the market. The major task of DRL is to collect the data for designing the model in the finance market with low delay and with less expensive training. To achieve this, the finance Markov decision process (FMDP) is used. It is the same as the traditional Markov decision process in which the agent goes through different states and performs actions. Depending on the action taken, it receives the reward. The FMDP provides security during buying and selling of the stocks [89, 90].

#### 4.2.6 | Computer vision

Computer vision is used for visual perception tasks with CNN for feature extraction, object detection, speech recognition and image classification. In 2009, imagNet was developed to classify the images automatically in predefined classes. Object detection is very difficult to track the exact location of an object compared with the image classification [91, 92]. DBNN architecture is tested on many speech recognition applications including google-voice input speech, you tube speech [93]. To overcome the imbalanced data classification problem in ML, a DRL-based model was proposed in which the problem was first converted into MDP and later solved by a DQN [94].

#### 4.2.7 | Healthcare

AI technologies are realistically altering and empowering the healthcare system. At present RL and DL have been extensively used to determine and discover innovative healthcare applications and services namely, medical imaging [95]. DRL also focuses on lung cancer as much of the global population is suffering from lung tumours; and on providing solutions to computer-aided diagnosis. Value-based DL models including DQN and hierarchical DRL models are used in the treatment of lung cancer and its diagnosis [96–98].

In addition to these, DRL is also applied to the fields like cyber security [99] ariel network, that is, drones [100], trading and autonomous vehicles [101–104].

## 5 | OPEN ISSUES AND CHALLENGES

Some of the open issues and challenges in the field of ML and DRL are as follows.

### 5.1 | Vulnerability to attacks

Though DRL performance is better compared to the other learning techniques and is capable of defending some attacks but is still defenceless to many attacks. During the training phase, though we train many adversarial samples to the DRL network, there are still new samples created for newly trained samples to decept the DRL network. To avoid this, the training of DRL algorithms must be robust. For example, in NLP, to provide security and vulnerability to the system with increasing success rate and reducing word replacement rate, a novel black-box attack model and white-box attack model were proposed. Black box attack model is based on differential evolution algorithm whereas the white-box attack model is based on a factor called the coefficient of variation which improves the replacement rate of the word. Black-box attack model has noticeable input and output relationship but it is deficit of clarity. White-box attack model has the noticeable relationship features in behaviour between the input and output. As these two models are less robust, adversarial attack is used to restore the text. Though the white-box attack model improves the replacement rate it is poor in performance compared with the black-box model. In future, to increase the robustness and security of these two models we can first pre-process the text before entering the classifier.

### 5.2 | Inaccuracy in weather forecasting

To sense the climatic conditions and to predict the changes in the weather we use different weather monitoring systems. However, the predictions made by these systems are not accurate enough and occur some delays in the results. The existing weather monitoring systems take present weather conditions as input and predict the future state. To provide accurate prediction, instead of taking the present weather conditions the system should be designed in such a way that it takes the past conditions and stores the data in cloud storage rather than external storage with the help of some wireless technology.

For mountain and slope lands, when compared with traditional weather prediction models LSTM based temporal convolution network (TCN) forecast the weather conditions for different surfaces over a long time (i.e., up to 12 h). Besides this, multiple-input multiple-output (MIMO) models are used to predict the weather conditions, which produces a better mean-square error but the output is not accurate. Therefore, producing accurate weather predictions in mountain areas and slopes is still a challenging task.

### 5.3 | Improvement in the data quality

A major issue in the recognition of facial expression is that it requires huge data during the training phase of neural network but there is a storage problem in facial expression recognition (FER) in terms of quality as well as quantity because of the presence of small-scale data sets. Due to this, the entire expression is not captured into a single dataset as a result, imbalanced distribution takes place. Consider in the case of navigation, the inputs given to the agent are high dimensional images during the learning phase. The agent undergoes several interactions with the environment. These interactions increase when the environment is complex. As the number of interactions increases, rewards increases, which results in data inefficiency due to the requirement of large training time. Due to this, there is poor performance in navigation.

### 5.4 | Employing transfer learning

After performing one assignment instead of training the learning models once again for another similar assignment we can transfer the knowledge of the previous assignment to the next similar assignment. This can alleviate the cost of training models and one can solve upcoming similar problems. A transfer learning model was proposed to overcome the classification problem during the training phase and to improve the classification accuracy by using some DL models (i.e., Squeeze net, Densenet, mobile net, Resnet) and also to overcome the image segmentation problem with the use of U net model. A CNN-based NGG 16 architecture was proposed to overcome the problem of image classification of high-dimensional datasets with less feature extraction time. In transfer learning, there is another approach for classifying the emotion perception from images based on visual features.

### 5.5 | Network segmentation using ML

The main aim of network segmentation is to allocate the network to different resources for better performance. This network segmentation dependent on ML has many advantages. Firstly, ML is made to learn the network allocation process based on service demands and resource allocation strategy. Now it is aware of network allocation. With the help of transfer learning, knowledge on network allocation in one environment is shared to the network allocation in another environment which results in increasing the speed of the process. During different training phases, segmenting images with poor contrast and low resolution is a challenging task in 2D and 3D segmentation process. To address the above issue, a novel deep convolutional neural network (DCNN) was proposed for μCT (micro-computed tomography) images. DCNN is successfully implemented into different types of μCT images (2D and 3D) for extracting features more superior when compared with the traditional ML approach. It is successfully implemented in 2D

segmentation but in 3D it exhibits low performance which is still a drawback.

### 5.6 | Meta learning

The main issue with DL models is that when massive data is fed during the training phase for classification it takes too much time to select the correct sample. To address this issue, meta-learning was proposed for fast adaptation to new learning tasks from a few trained datasets. A novel approach for text classification named model agnostic meta-learning (MAML) framework and for short text classification, few short text classifications in meta-learning (FSML) were proposed to learn transferable features in the discrete text. The problem with meta-learning is that a text classification error is exhibited during the training phase of heterogeneous data.

## 6 | CONCLUSION

In this era of AI, DRL has emerged as a prominent application of ML. The success of DRL is augmented with the use of different DL models. Applications of DRL are deepening its roots day by day, finding its applications in diversified fields like aerospace, healthcare, medical, automotive, finance, robotics, chemistry, AI toolkits, Bots, videogames, aerodynamics etc. This survey contributes to the overview of the DL models and evolution of DRL and its architectures with a few real-world applications. We conclude our work with few open issues and challenges in the area of ML and DRL for future research. There is an immense scope for DRL due to its learning behaviour and hence, the possibilities of its application are immeasurable. The survey also concludes that a larger skill force will be required to cater to these applications with specific knowledge in different industrial sectors, especially in healthcare, automotive, smart city and intelligent transportation.

### CONFLICT OF INTEREST

The authors state no conflict of interest to any person, whosoever.

### FUNDING INFORMATION

The authors received no specific funding for this work.

### ORCID

*Nishu Gupta* https://orcid.org/0000-0002-1568-368X
*Ahmed Alkhayyat* https://orcid.org/0000-0002-0962-3453

### REFERENCES

1. Ejaz, M.M., Tang, T.B., Lu, C.-K.: Autonomous visual navigation using deep reinforcement learning: An overview. In: *IEEE SCOReD Conference (SCOReD)*. Malaysia, pp. 294–299 (2019)
2. Han, M., Zhao, J., Zhang, X., Shen, J., Li, Y.: The reinforcement learning method for occupant behavior in building control: A review. Energy Built Environ. 2(2), 137–148 (2020)
3. Bari Antor, M., Jamil, A.H.M., Mamtaz, M., Monirujjaman Khan, M., Aljahdali, S., Kaur, M., Singh, P., Masud, M.: A comparative analysis of machine learning algorithms to predict Alzheimer's disease. J. Healthcare Eng. 2021, 9917919 (2021)
4. Salim, N.O.M., Zeebaree, S.R.M., Sadeeq, M.A.M., Radie, A.H., Shukur, H.M., Rashid, Z.N.: Study for food recognition system using deep learning. J. Phys.: Conf. Ser. 1963(1), 012014 (2021)
5. Kumar, N., Raubal, M.: Applications of deep learning in congestion detection, prediction and alleviation: A survey. Transp. Res. Part C: Emerging Technol. 133, 103432 (2021)
6. Hussein, H.A., Zeebaree, S.R.M., Sadeeq, M.A.M., Shukur, H.M., Alkhayyat, A., Sharif, K.H.: An investigation on neural spike sorting algorithms. In: International Conference on Communication & Information Technology (ICICT). Iraq, pp. 202–207 (2021)
7. Aliwy, A., Abbas, A., Alkhayyat, A.: NERWS: Towards improving information retrieval of digital library management system using named entity recognition and word sense. Big Data Cognit. Comput. 5(4), 59 (2021)
8. Luong, N.C., Hoang, D.T., Gong, S., Niyato, D., Wang, P., Liang, Y.-C., Kim, D.: Applications of deep reinforcement learning in communications and networking: A survey. IEEE Commun. Surv. Tutorials 21(4), 3133–3174 (2019)
9. Kanagawa, Y., Kaneko, T.: Rogue-gym: A new challenge for generalization in reinforcement learning. In: IEEE CoG Conf. (CoG). pp. 1–8 (2019)
10. Arulkumaran, K., Deisenroth, M.P., Brundage, M., Bharath, A.A.: Deep reinforcement learning: A brief survey. IEEE Signal Proc Mag. 34(6), 26–38 (2017)
11. Wang, C., Zhang, Q., Tian, Q., Li, S., Wang, X., Lane, D., Petillot, Y., Wang, S.: Learning mobile manipulation through deep reinforcement learning. Sensors. 20(3), 939 (2020)
12. Ge, Y.: A survey on big data in the age of artificial intelligence. In: 6th IEEE International ICCSS Conference (ICCSS). China, pp. 72–77 (2019)
13. Chen, L., Chen, P., Lin, Z.: Artificial Intelligence in Education: A Review. IEEE Access 8, 75264–75278 (2020)
14. Keneshloo, Y., Shi, T., Ramakrishnan, N., Reddy, C.K.: Deep reinforcement learning for sequence-to-sequence models. IEEE Trans. Neural Networks Learn. Syst. 31(7), 2469–2489 (2019)
15. Lei, L., Tan, Y., Zheng, K., Liu, S., Zhang, K., Shen, X.: Deep reinforcement learning for autonomous internet of things: model, applications and challenges. IEEE Commun. Surv. Tutorials 22(3), 1722–1760 (2020)
16. Muñoz, G., Barrado, C., Çetin, E., Salami, E.: Deep reinforcement learning for drone delivery. Drones 3(3), 72 (2019)
17. Cioffi, R., Travaglioni, M., Piscitelli, G., Petrillo, A., De, F.: F: Artificial intelligence and machine learning applications in smart production: Progress, trends, and directions. Sustainability 12(2), 492 (2020)
18. Xue, M., Yuan, C., Wu, H., Zhang, Y., Liu, W.: Machine learning security: Threats, countermeasures, and evaluations. IEEE Access 8, 74720–74742 (2020)
19. Cao, W., Yan, Z., He, Z., He, Z.: A comprehensive survey on geometric deep learning. IEEE Access 8, 35929–35949 (2020)
20. Jiao, L., Zhao, J.: A survey on the new generation of deep learning in image processing. IEEE Access 7, 172231–172263 (2019)
21. Zheng, S., Liu, H.: Improved multi-agent deep deterministic policy gradient for path planning-based crowd simulation. IEEE Access 7, 147755–147770 (2019)
22. Jiang, F., Dashtipour, K., Hussain, A.: A survey on deep learning for the routing layer of computer network. In: IEEE UCET Conference (UCET). UK, pp. 1–4 (2019)
23. Yang, T., Zhao, L., Li, W., Zomaya, A.Y.: Reinforcement learning in sustainable energy and electric systems: A survey. Annu. Rev. Control 49, 145–163 (2020)
24. Ruan, A., Shi, A., Qin, L., Xu, S., Zhao, Y.: A reinforcement learning based Markov-decision process (MDP) implementation for SRAM FPGAs. IEEE Trans. Circuits Syst. II Express Briefs 67(10), 2124–2128 (2019)
25. Gupta, A., Chaurasiya, V.K.: Reinforcement learning based energy management in wireless body area network: A survey. In: IEEE CICT Conference (CICT). India, pp. 1–6 (2019)
26. Varghese, V., Mahmoud, N., Qusay, H.: A survey of multi-task deep reinforcement learning. Electronics 9(9), 1363 (2020)

27. Mehta, D.: State-of-the-art reinforcement learning algorithms. Int. J. Eng. Res. Technol. 8(12), 6 (2019)
28. Zhou, X., Wu, P., Zhang, H., Guo, W., Liu, Y.:.: Learn to navigate: Cooperative path planning for unmanned surface vehicles using deep reinforcement learning. IEEE Access 7, 165262–165278 (2019)
29. Xiali, I., Zhengyu, L., Wang, S., Wei, Z., Wu, L.: A reinforcement learning model based on temporal difference algorithm. IEEE Access 7, 121922–121930 (2018)
30. Siddiqi, U.F., Sait, S.M., Uysal, M.: Deep Q-learning based optimization of VLC systems with dynamic time-division multiplexing. IEEE Access 8, 120375–120387 (2020)
31. Zeng, F., Wang, C., Ge, S.S.: A survey on visual navigation for artificial agents with deep reinforcement learning. IEEE Access 8, 135426–135442 (2020)
32. Ahmad, R., Soltani, M.D., Safari, M., Srivastava, A., Das, A.: Reinforcement learning based load balancing for hybrid liFi WiFi Networks. IEEE Access 8, 132273–132284 (2020)
33. Veres, M., Moussa, M.: Deep learning for intelligent transportation systems: A survey of emerging trends. IEEE Trans. Intell. Transp. Syst. 21(8), 3152–3168 (2019)
34. Nian, R., Liu, J., Huang, B.: A review on reinforcement learning: Introduction and applications in industrial process control. Comput. Chem. Eng. 139, 106886 (2020). https://doi.org/10.1016/j.compchemeng.2020.106886
35. Gupta, R.: A survey on machine learning approaches and its techniques. In: 2020 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS). India, (2020)
36. Smit, J., Ponnambalam, C.T., Spaan, M.T., Oliehoek, F.A.: PEBL: Pessimistic ensembles for offline deep reinforcement learning. In: Robust and Reliable Autonomy in the Wild Workshop at the 30th International Joint Conference of Artificial Intelligence, Canada, (2021)
37. Nguyen, T.T., Nguyen, N.D., Nahavandi, S.: Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications. IEEE Trans. Cybern. 50(9), 3826–3839 (2020)
38. Pamina, J., Raja, B.: Survey on deep learning algorihms. Int. J. Emerging Technol. Innovative Eng. 5(1), 6 (2019)
39. Wang, F., et al.: Deep learning for edge computing applications: A state-of-the-art survey. IEEE Access 8, 58322–58336 (2020)
40. Ma, X., et al.: A survey on deep learning empowered IoT applications. IEEE Access 7, 181721–181732 (2019)
41. Krishna, S.T., Kalluri, H.K.: Deep learning and transfer learning approaches for image classification. Int. J. Recent Technol. Eng. 7(5S4), 427–432 (2019)
42. Otter, D.W., Medina, J.R., Kalita, J.K.: A survey of the usages of deep learning for natural language processing. IEEE Trans. Neural Networks Learn. Syst. 32(2), 604–624 (2020)
43. Shrestha, A., Mahmood, A.: Review of deep learning algorithms and architectures. IEEE Access 7, 53040–53065 (2019)
44. Hemeida, A.M., Hassan, S.A., Mohamed, A.-.A.A., Alkhalaf, S., Mahmoud, M.M., Senjyu, T., E-D, A.B.: Nature- Inspired algoritms for feed forward nneural netwok classifiers: A survey of one decade of research. Ain Shams Eng. J. 11(3), 659–675 (2020)
45. Huang, B., et al.: Signal frequency estimation based on RNN. In: 2020 Chinese Control And Decision Conference (CCDC). China, (2020)
46. Aditi, M.K., Poovammal, E.: Image classification using a hybrid LSTM-CNN deep neural network. Int. J. Eng. Adv. Technol. 8(6), 1342–1348 (2019)
47. Minaee, S., Abdolrashidi, A., Su, H., Bennamoun, M., Zhang, D.: Biometric recognition using deep learning: A survey. ArXiv (2019)
48. Yu, Y., Si, X., Hu, C., Zhang, J.: A review of recurrent neural networks: LSTM cells and network architectures. Neural Comput. 31(7), 1235–1270 (2019)
49. Yu, Y., et al: A review of recurrent neural networks: LSTM cells and network architectures. Neural Comput. 31(7), 1235–1270 (2019)
50. Akinyelu, A., Blignaut, P.: Convolutional neural network-based methods for eye gaze estimation: A survey. IEEE Access 8, 142581–142605 (2020)
51. Chen, J., Ran, X.: Deep learning for edge computing applications: A review. IEEE Access 107, 1655–1674 (2019)
52. Grigorescu, S., et al.: A survey of deep learning techniques for autonomous driving. J. Field Rob. 37(3), 362–386 (2020)
53. Pedrycz, W., Chen, S.-M.: Deep Learning, Concepts and Architectures. Springer, Cham, Switzerland (2019)
54. Lakshmi Durga, S., Rajeshwari, C., Hamed Allehaibi, K., Gupta, N., Nammas Albaqami, N., et al.: Deep reinforcement learning-based long short-term memory for satellite iot channel allocation. Intell. Autom. Soft Comput. 33(1), 1–19 (2022)
55. Wang, C., Yang, J., Xie, L., Yuan, J.: Kervolutional neural networks. In: IEEE CVPR Conference. Virtual, pp. 31–40 (2020)
56. Alom, M.Z., et al.: A state-of-the-art survey on deep learning theory and architectures. Electronics 8(3), 292 (2019)
57. Alom, M.Z., et al: The history began from alexnet: A comprehensive survey on deep learning approaches. arXiv preprint arXiv 1803.01164 (2018)
58. Shu, M.: Deep learning for image classification on very small datasets using transfer learning. *Creative Components*. 345, (2019). https://lib.dr.iastate.edu/creativecomponents/345
59. Lee, H.J., et al.: Real-time vehicle make and model recognition with the residual SqueezeNet architecture. Sensors 19(5), 982 (2019)
60. Chen, Z., Liu, G.: DenseNet+ inception and its application for electronic transaction fraud detection. In: IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS). pp. 2551–2558 (2019). https://doi.org/10.1109/HPCC/SmartCity/DSS.2019.00357
61. Rizk, Y., et al: Deep belief networks and cortical algorithms: A comparative study for supervised classification. Appl. Comput. Inf. 15(2), 81–93 (2019)
62. Zhang, S., Zhang, S., Wang, B., Habetler, T.G.: Deep learning algorithms for bearing fault diagnostics – A comprehensive review. IEEE Access 8, 29857–29881 (2020)
63. Yenumaladoddi Jayasimha, R., Reddy, V.S.: A robust face emotion recognition approach through optimized SIFT features and adaptive deep belief neural network. J. Appl. Secur. Res. 16(3), 1–22 (2020)
64. Dargan, S., et al: A survey of deep learning and its applications: A new paradigm to machine learning. Arch. Comput. Methods Eng. 27, 1–22 (2019)
65. Pouyanfar, S., et al.: A survey on deep learning: Algorithms, techniques, and applications. ACM Comput. Surv. 51(5), 1–36 (2018)
66. Choudhary, T., et al.: A comprehensive survey on model compression and acceleration. Artif. Intell. Rev. 53(7), 5113–5155 (2020)
67. Dargan, S., Kumar, M., Ayyagari, A.: Survey of deep learning and its applications: A new paradigm to machine learning. Arch. Comput. Methods Eng. 27(4), 1071–1092 (2020)
68. Choi, J., et al.: Convolutional neural network technology in endoscopic imaging: Artificial intelligence for endoscopy. Clin. Endosc. 53(2), 117–126 (2020)
69. Zhou, R., Liu, F., Gravelle, C.W.: Deep learning for modulation recognition: A survey with a demonstration. IEEE Access 8, 67366–67376 (2020)
70. Rezk, N.M., et al.: Recurrent neural networks: An embedded computing perspective. IEEE Access 8, 57967–57996 (2020)
71. Sadr, H., Pedram, M.M., Teshnehlab, M.: Multi-view deep network: A deep model based on learning features from heterogeneous neural networks for sentiment analysis. IEEE Access 8, 86984–86997 (2020)
72. Nelson, V.V., Mahmoud, Q.H.: A survey of multi-task deep reinforcement learning. Electronics 9(9), 1363 (2020)
73. Zhan, Y., Zhang, J.: An incentive mechanism design for efficient edge learning by deep reinforcement learning approach. In: IEEE INFOCOM 2020-IEEE Conference on Computer Communications. Virtual, (2020)
74. Park, S., et al.: Adaptive real-time offloading decision-making for mobile edges: Deep reinforcement learning framework and simulation results. Appl. Sci. 10(5), 1663 (2020)
75. Wang, C., et al.: Learning mobile manipulation through deep reinforcement learning. Sensors 20(3), 939 (2020)

76. Sabry, M., Khalifa, A.: On the reduction of variance and overestimation of deep Q-learning. arXiv preprint arXiv:1910.05983 (2019)
77. Chen, S.Y.-.C., et al.: Variational quantum circuits for deep reinforcement learning. IEEE Access 8, 141007–141024 (2020)
78. Li, Y., Qi, F., Wang, Z., Yu, X., Shao, S.: Distributed edge computing offloading algorithm based on deep reinforcement learning. IEEE Access 8, 85204–85215 (2020). https://doi.org/10.1109/ACCESS.2020.2991773
79. Hu, Z., Wan, K., Gao, X., Zhai, Y., Wang, Q.: Deep reinforcement learning approach with multiple experience pools for UAV's autonomous motion planning in complex unknown environments. Sensors (Basel) 20(7), 1890 (2020, March 29). https://doi.org/10.3390/s20071890
80. Wen, S., Zhao, Y., Yuan, X.: Path planning for active SLAM based on deep reinforcement learning under unknown environments. Intell. Serv. Rob. 13, 263–272 (2020). https://doi.org/10.1007/s11370-019-00310-w
81. Ruan, X., et al.: Mobile Robot Navigation Based on Deep Reinforcement Learning. In: Chinese Control and Decision Conference (CCDC). China, (2019)
82. Dai, Y., et al.: A survey on dialog management: Recent advances and challenges. arXiv preprint arXiv 2005.02233 (2020)
83. Zhang, H., Zhou, A., Lin, X.: Interpretable policy derivation for reinforcement learning based on evolutionary feature synthesis. Complex Intell. Syst. 6(3), 741–753 (2020)
84. Izzo, D., Märtens, M. Pan, B.: A survey on artificial intelligence trends in spacecraft guidance dynamics and control. Astrodyn 3, 287–299 (2019)
85. Märtens, M., Izzo, D., Krzic, A., et al.: Super-resolution of PROBA-V images using convolutional neural networks. Astrodyn 3, 387–402 (2019)
86. Kirtas, M., et al.: Deepbots: A webots-based deep reinforcement learning framework for robotics. In: IFIP International Conference on Artificial Intelligence Applications and Innovations. Greece, (2020)
87. Zhu, P., et al.: Multi-robot flocking control based on deep reinforcement learning. IEEE Access 8, 150397–150406 (2020)
88. Gregurić, M., et al.: Application of deep reinforcement learning in traffic signal control: An overview and impact of open traffic data. Appl. Sci. 10(11), 4011 (2020)
89. Koh, S., et al.: Real-time deep reinforcement learning based vehicle navigation. Appl. Soft Comput. 96, 106694 (2020)
90. Kulaglic, A., Ustundag, B.B.: Stock price prediction using predictive error compensation wavelet neural networks. CMC-Comput. Mater. Continua 68(3), 3577–3593 (2021)
91. Chakraborty, S.: Capturing financial markets to apply deep reinforcement learning. arXiv preprint arXiv 1907.04373 (2019)
92. Li, Y., Ni, P., Chang, V.: Application of deep reinforcement learning in stock trading strategies and stock forecasting. Computing 1305–1322 (2019)
93. Feng, X., et al.: Computer vision algorithms and hardware implementations: A survey. Integration 69, 309–320 (2019)
94. Rodriguez-Ramos, A., et al.: Adaptive inattentional framework for video object detection with reward-conditional training. IEEE Access 8, 124451–124466 (2020)
95. Alam, M., et al.: Survey on deep neural networks in speech and vision systems. Neurocomputing 417, 302–321 (2020)
96. Lin, E., Chen, Q., Qi, X.: Deep reinforcement learning for imbalanced classification. Appl. Intell. 50(8) (2020)
97. Naeem, M., Paragliola, G., Coronato, A.: A reinforcement learning and deep learning based intelligent system for the support of impaired patients in home treatment. Expert Syst. Appl. 168, 114285 (2021)
98. Liu, Z., et al.: Deep reinforcement learning with its application for lung cancer detection in medical Internet of Things. Future Gener. Comput. Syst. 97, 1–9 (2019)
99. Lu, M., et al.: Application of reinforcement learning to deep brain stimulation in a computational model of Parkinson's disease. IEEE Trans. Neural Syst. Rehabil. Eng. 28(1), 339–349 (2019)
100. Nguyen, T.T., Reddi, V.J.: Deep reinforcement learning for cyber security. IEEE Trans. Neural Networks Learn. Syst. (2021)
101. Ahmad, T.A., Koubaa, A., Mohamed, N.A., Ibrahim, H.A., Ibrahim, Z.F., Kazim, M., Ammar, A., Benjdira, B., Khamis, A.M., Hameed, I.A., Casalino, G.: Drone deep reinforcement learning: A review. Electronics 10, 999 (2021)
102. Millea, A.: Deep reinforcement learning for trading—A critical survey. Data 6, 119 (2021)
103. Zhang, Y., Li, X., Li, X.: Reinforcement learning cropping method based on comprehensive feature and aesthetics assessment. IET Image Process. 16, 1415–1423 (2022). https://doi.org/10.1049/ipr2.12420
104. Wei, H., Zhao, W., Ai, Q., Zhang, Y., Huang, T.: Deep reinforcement learning based active safety control for distributed drive electric vehicles. IET Intell. Transp. Syst. 16, 813–824 (2022). https://doi.org/10.1049/itr2.12176
105. Tian, C., Shaik, S., Wang, Y.: Deep reinforcement learning for shared control of mobile robots. IET Cyber-Syst. Robot. 3(4), 315–330 (2021). https://doi.org/10.1049/csy2.12036


**How to cite this article:** Balhara, S., Gupta, N., Alkhayyat, A., Bharti, I., Malik, R.Q., Mahmood, S.N., Abedi, F.: A survey on deep reinforcement learning architectures, applications and emerging trends. IET Commun. 1–16 (2022). https://doi.org/10.1049/cmu2.12447