Neurocomputing 545 (2023) 126327

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Survey paper

Deep neural networks in the cloud: Review, applications, challenges and research directions

Kit Yan Chan [a], Bilal Abu-Salih [b], Raneem Qaddoura [c], Ala' M. Al-Zoubi [d], Vasile Palade [e], Duc-Son Pham [a], Javier Del Ser [f,g], Khan Muhammad [h,*]

[a] School of Electrical Engineering, Computing and Mathematical Sciences, Curtin University, Perth, WA 6102, Australia
[b] School for Information Technology, The University of Jordan, Amman, Jordan
[c] School of Computing and informatics, Al Hussein Technical University, Jordan
[d] School of Science, Technology and Engineering, University of Granada, Granada, Spain
[e] Centre for Computational Science and Mathematical Modelling, Coventry University, Priory Street, Coventry, UK
[f] TECNALIA (Basque Research & Technology Alliance – BRTA), P. Tecnologico, Ed. 700. 48170 Derio (Bizkaia), Spain
[g] University of the Basque Country (UPV/EHU), 48013 Bilbao (Bizkaia), Spain
[h] VIS2KNOW Lab, Department of Applied Artificial Intelligence, School of Convergence, College of Computing and Informatics, Sungkyunkwan University, Seoul 03063, Republic of Korea

ARTICLE INFO

Article history:
Received 31 January 2023
Revised 26 March 2023
Accepted 7 May 2023
Available online 13 May 2023
Communicated by Zidong Wang

Keywords:
Big data
Cloud computing
Deep neural networks
High-performance computing

ABSTRACT

Deep neural networks (DNNs) are currently being deployed as machine learning technology in a wide range of important real-world applications. DNNs consist of a huge number of parameters that require millions of floating-point operations (FLOPs) to be executed both in learning and prediction modes. A more effective method is to implement DNNs in a cloud computing system equipped with centralized servers and data storage sub-systems with high-speed and high-performance computing capabilities. This paper presents an up-to-date survey on current state-of-the-art deployed DNNs for cloud computing. Various DNN complexities associated with different architectures are presented and discussed alongside the necessities of using cloud computing. We also present an extensive overview of different cloud computing platforms for the deployment of DNNs and discuss them in detail. Moreover, DNN applications already deployed in cloud computing systems are reviewed to demonstrate the advantages of using cloud computing for DNNs. The paper emphasizes the challenges of deploying DNNs in cloud computing systems and provides guidance on enhancing current and new deployments.

1. Introduction

Deep neural networks (DNNs) are being developed for numerous pattern recognition applications in a wide range of real-life domains, such as e-commerce, manufacturing, medicine and health, and autonomous vehicles. However, DNNs are computationally demanding, especially in training, because of the large number of parameters required to train them. In general, DNNs have millions of parameters; for example, a popular DNN, AlexNet, has 60 million parameters and another, VGG-16, has 138 million parameters. More recently, it took seven months to train a DNN with 175 billion parameters, developed by OpenAI for natural language processing (NLP) [1]. As such, it is not practical to use a single stand-alone computer to train a large DNN. High performance computational devices are required to train DNNs. Once the DNNs are trained, they are usually run on online mobile devices or smart phones, which are essential in our daily lives. However, DNNs mostly require millions of float-point operations (FLOPs) or more in computations. For example, AlexNet requires 720 million FLOPs and VGG-16 requires 15,300 million FLOPs. Such large DNNs need a lot of storage for their parameters, consume a lot of power in execution, and require a lot of FLOPs for calculations. Therefore, a stand-alone computational device is not powerful enough to deploy a large DNN.

Recently, it has become increasingly common to deploy DNNs using cloud platforms, which are high-performance computing platforms with tremendous speed and memory. Training can be performed within a reasonable time on cloud machine learning

* Corresponding author.
E-mail addresses: kit.chan@curtin.edu.au (K.Y. Chan), b.abusalih@ju.edu.jo (B. Abu-Salih), raneem.qaddoura@htu.edu.jo (R. Qaddoura), ala.m.zoubi@gmail.com, alzoubi@correo.ugr.es (A.M. Al-Zoubi), vasile.palade@coventry.ac.uk (V. Palade), dspham@ieee.org (D.-S. Pham), javier.delser@tecnalia.com (J.D. Ser), khanmuhammad@g.skku.edu (K. Muhammad).

https://doi.org/10.1016/j.neucom.2023.126327
0925-2312/

Nomenclature

List of Acronyms and their Definitions

Acronym	Explanation
1D	One Dimensional
2D	Two Dimensional
3D	Three Dimensional
4G	Fourth Generation
5G	Fifth Generation
6G	Sixth Generation
ARIMA	Autoregressive Integrated Moving Average
AWS	Amazon Web Service
BI	Business Intelligence
BP	Back Propagation
BTS	Base Transceiver Stations
CAs	Combinatorial Auctions
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CUs	Central Units
DL	Deep Learning
DNN	Deep Neural Networks
DQNs	Deep Q-Networks
DRL	Deep Reinforcement Learning
DUs	Distributed Units
SA	Sentiment Analysis
EA	Evolutionary Algorithm
ECS	Amazon Elastic Container Service
FCNN	Fully Connected Neural Network
FER	Facial Emotion Recognition
FLOPs	Float-Point Operations
FWIoU	Frequency-Weighted Intersection over Union
GAN	Generative Adversarial Network
GBT	Gradient Boosting Tree
GKE	Google Kubernetes Engine
GNN	Graph Neural Network
GPS	Global Positioning System
GPU	Graphics Processing Unit
IaaS	Infrastructure as a Service
IoT	Internet of Things
IDSs	Intrusion Detection Systems
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MCC	Mobile Cloud Computing
mIoU	mean Intersection over Union
MIP	Mixed-Integer Programming
ML	Machine Learning
MLP	Multilayer Perceptrons
MPA	Mean Pixel Accuracy
MSE	Mean Square Error
NLP	Natural Language Processing
ONOS	Open Network Operating System
PS	Parameter Server
PSLD	PS Load Distribution
PaaS	Platform as a Service
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
SA	Sentiment Analysis
SaaS	Software as a Service
TELESTO	Multivariate Time Series based Classification Model
TPU	Tensor Processing Unit
WCE	Wireless Capsule Endoscopy

(ML) platforms, such as Amazon Web Services (AWS) Deep Learning and Google Colab. Cloud computing-based centralized servers provide large computational resources, large data storage, high-speed computation, low latency, and high availability. Cloud computing is also used to deploy DNNs for online applications. This manuscript references several recent survey articles related to the deployment of cloud platforms for demanding computations. We divide these survey articles into three categories: security technologies, performance enhancement technologies, and applications.

For security technologies, Yan et al. [2] presented a survey of various technologies encountering malicious attacks. This article also discussed several defense mechanisms against malicious attacks that disrupt cloud servers and services by creating huge amounts of internet traffic. Nita and Mihailescu [3] and Sun et al. [4] presented several recent security technologies, such as encryption, access structure, multiauthority, fine-grained trace mechanism, trust, reputation, and extension of tradition access control. Gai et al. [5] presented a survey, discussing the recent technical fusion of blockchain and clouds, including several cloud-relevant blockchain service models and encryption schemes in blockchain. They also analyzed the performance of a cloud data center in which hardware and software are integrated with blockchain.

For performance enhancement, Xu et al. [6] reviewed several technologies used for managing the performance of virtual machines in cloud infrastructure services. These technologies optimize virtual machine performance in terms of cost, accuracy, effectiveness, and implementation complexity. Pupykina et al. [7] presented several memory management technologies in cloud computing that perform multiple applications, for which users request different quality-of-service levels; these users share the same heterogeneous infrastructures and resources. Wang et al. [8] presented a survey, discussing offloading technologies to optimize offload tasks between cloud and edge systems. These technologies enhance offloading effectiveness in terms of energy consumption, cost, and response time. In addition, several challenges are summarized to provide future research directions and promote edge-cloud markets. Xu et al. [6] presented a survey which provides an overview of computational distribution mechanisms that manage virtual machines in the cloud where requests from multiple users are received. The mechanisms optimize the operational cost, computational accuracy, and complexity. Some challenges and future directions are also outlined. Zhou et al. [9] and Feng et al. [10] provided an overview of cloud resource scheduling technologies that use deep reinforcement learning. These approaches minimize energy consumption to satisfy large demands of user services which are highly dynamic, uncertain, and resilient. However, a general overview of DL is not presented in either survey.

Regarding applications, Khan et al. [11] presented a general overview of mobile cloud architectures, the benefits of cloud computing, and offload decisions for mobile cloud. Applications of mobile cloud computing (MCC), such as mathematical tools, file search, imaging tools, and games, were discussed. Bera et al. [12] provided a survey of cloud computing applications in a smart grid, specifically in the areas of energy management, information management, and security. Cao et al. [13] presented a survey of cloud computing architectures, which provide sensing, computing, con-

trolling, and storing services for cyber-physical systems. Several applications such as building smart grids, intelligent transportation, personalized healthcare, and smart manufacturing were reviewed. However, none of these survey articles focused on developing DNNs in cloud computing systems. Soni and Kumar [14] and Khana et al. [15] presented overviews of machine learning technologies to handle a variety of resource management tasks, such as workload estimation, task and virtual machine (VM) scheduling, resource optimization, and energy minimization. However, only a few deep learning (DL) technologies were covered briefly, and the existing surveys on DL by Saiyeda and Mir [16] and Priya et al. [17], are either too old or too brief.

Recent review articles on cloud computing and deep learning are summarized in Table 1. There is a growing interest in developing cloud computing technologies for DL in many government [18], public [19], private [20], and commercial sectors [21,22]. These technologies use large data storage and super-powerful computational resources for various applications including machine vision, speech recognition, language translation and processing, weather and climate forecasting, bioinformatics, manufacturing automation and defect detection, and drug development. It is important for researchers to integrate DL with high-performance computing. However, Table 1 shows that eleven of the eighteen review articles discuss issues of security [2–5] or performance enhancement [23,7,8,6,9,10]. Seven of the eighteen review articles discuss cloud deployment for demanding computations [11–15,17,16]. In these reviews, descriptions of DL in cloud platforms are either brief or nonexistent [11–15,17]. One of the eighteen review articles presents a more detailed survey of cloud deployment; however the survey was conducted in 2017 which is too old [16]. There are no review articles or recent surveys focusing on the deployment of cloud computing for DNNs. Also no review article has discussed the challenges of deploying DNNs in the cloud and relevant future research directions. This is the motivation for the survey conducted in this article on cloud computing techniques for DNN deployment.

The rest of this article is organized as follows: Section 2 briefly presents several DNN mechanisms, namely the traditional multilayer perceptron (MLP), the recurrent neural network (RNN), convolution neural network (CNN), deep reinforcement learning (DRL), graph neural network (GNN), and optimization of DNNs. The network complexity and the necessity for using cloud computing are discussed. The motivation for deploying DNNs in the cloud is also discussed. Section 3 presents the commonly used cloud computing platforms for deploying DNNs. This information provides guidance for researchers interested in deploying DNNs in the cloud. Section 4 presents several DNN applications implemented in cloud systems, such as in health systems, NLP, business intelligence, anomaly detection, wireless capsule endoscopy (WCE), and mobile-cloud-assisted applications. These applications demonstrate the advantages and effectiveness of deploying DNNs in cloud systems. In addition, researchers may develop cloud-based DNN applications for yet unexplored areas. Section 5 discusses the challenges of the current DNN deployments using cloud computing systems along with prospects for enhancing the current deployments of DNNs on cloud systems. Finally, conclusions are drawn in Section 6.

2. Preliminary: computational complexities in deep learning

A single DNN consists of a large number of parameters, which requires huge amounts of memory space to store. Both training and executing a DNN require a significant amount of time. This section discusses several commonly used DNN architectures, such as MLP, CNNs, and GNNs, which have complex architecture structures and a large number of DNN parameters. This section also highlights the long computational time required to train a DNN. The lengthy computational time is the reason a single stand-alone computer is not practical for training a DNN. Therefore, cloud computing is essential for DNN training.

2.1. Multilayer perceptron

The multilayer perceptron is a commonly used neural network [24–26]. MLP is composed of multiple layers, including an input layer, hidden layers, and an output layer, where each layer contains a set of perception elements known as neurons. Fig. 1 illustrates an MLP with two hidden layers, an input and output layer. In interactions, each node displays a certain amount of bias. The input layer

Table 1
Recent review articles of cloud computing for demanding computations.

Categories	Articles	Survey on cloud computing	DL (or ML) and cloud computing
Security issues	[2]	Defense technologies against malicious attacks to cloud servers	Brief discussion of real-time deployment in ML
	[3,4]	Security technologies in cloud platforms such as encryption, access structure, trust and reputation	Brief discussion of deployment in DL
	[5]	Blockchain services and encryption schemes in cloud platforms	Brief discussion of deployment of DL
Performance enhancement	[23]	DL for solving cloud computing issues such as resource and offloading allocations	Deploying DL in cloud is briefly discussed
	[7]	Cloud memory management for multiple users with demanding computations	Deployment of ML is briefly discussed; no DL discussion
	[8]	Offloading effectiveness enhancement such as energy consumption and response time	Brief description of offloading DL
	[6]	Enhancement technologies for virtual machine performance including cost, accuracy, and computational complexity	Deployment of ML is not discussed
	[9,10]	Cloud resource scheduling using deep reinforcement learning in the cloud	General overview of DL in the cloud not given
Cloud deployment	[11]	Mobile cloud architectures for demanding computations	No discussion of DL applications
	[12]	Cloud computing in a smart grid for energy management and information management	No discussion of DL applications
	[13]	Cyber physical-based cloud computing architectures for sense networks and storing services	Brief description of ML applications
	[14,15]	Machine learning technologies to handle a variety of resource management tasks in cloud	Brief description of DL applications
	[16]	Cloud platforms for DL deployment and challenges	Survey is too old as it was published 5 years ago
	[17]	Effects of using cloud computing and DL on commercial sectors	Brief description of DL applications

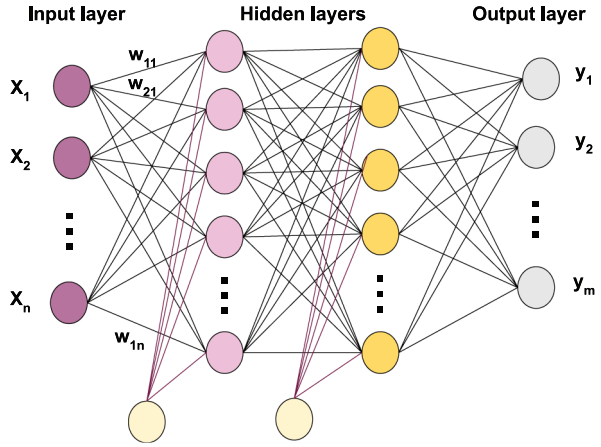

Fig. 1. MLP topology.

contains n input variables $X = \{x_1, x_2, \ldots, x_n\}$ and the output layer contains m output variables $Y = \{y_1, y_2, \ldots, y_m\}$.

The total number of parameters in an MLP can be determined by [27]

$$n \cdot h_1 + \sum_{k=1}^{N_h - 1} h_k \cdot h_{k+1} + h_{N_h} \cdot n \tag{1}$$

where the number of hidden nodes h_i in the ith layer is N_h. Longer computational times are required to optimize an MLP when N_h and h_k are higher. Multiple applications have been developed by applying MLP in the cloud. The study of [28] built a forecasting model using multiple input variables of different types of daily staple food prices to predict the consumer price index of Surabaya using the Amazon Cloud Services environment. The multilayer perceptron algorithm was used to build a prediction system with a hidden layer, epoch, and a number of neurons. Another study [29] used transfer learning based cancer segmentation (TL-CAN-Seg) technology to subdivide cancer-affected areas and store the relevant features in the cloud. To accurately classify areas affected by breast cancer, a novel MLP with an adjusted Levenberg–Marquardt (LM) algorithm was used to learn complicated image patterns and ultimately boost the accuracy of breast cancer diagnosis.

2.2. Recurrent neural network

Compared with MLPs, recurrent neural networks (RNNs) are more effective in processing temporal information, such as text or time series which are correlated sequentially. RNNs introduce new forms of neural processing units in which the output from the previous step in the sequence is used as the input of the current step [30]. The recurrent processing has a hidden state which captures and stores the information in the sequence. This learning capability of RNNs persists, retrieves, and exploits past information to predict. Memory is conferred by virtue of model relationships at different scales in time. The aforementioned hidden state captures information of all previous steps. Therefore, the trained RNN is capable of combining the input sequence and the hidden state to yield the output at a given step of the sequence.

Although RNNs were invented decades ago, computational time and memory storage remain prohibitive, and this drawback has stimulated intense research efforts recently [31]. The number of parameters of a simple RNN is

$$N_h^2 + m \cdot N_h + N_h \cdot n \tag{2}$$

where N_h, m, and n are the dimensions of the hidden, output, and input layers, respectively. When RNNs are deployed, long training and inference times are required to learn sequential data [32]. After deployment, extra time is also required to update RNNs with newly supervised sequential data.

Cloud applications leveraging the deployment of RNNs have mainly relied on natural language processing and time series forecasting tasks. Predictive modeling over naturally sequential data has largely harnessed cloud-deployed RNNs and include workload prediction [33,34], resource usage [35,36], biometric health monitoring [37], and power load forecasting [38], to mention a few. Video processing and/or summarization [39,40] have also been explored via long short-term memory (LSTM) and other recurrent neural architectures deployed on a cloud computing infrastructure in combination with CNNs.

2.3. Convolutional neural network

Fig. 2 shows a CNN consisting of a set of hidden layers, input/output layers, and a fully connected network [41,42]. A hidden layer consists of a convolution layer, activation function, and a pooling layer. In the hidden layer, the convolution layer extracts the input features. An activation function is applied to the convolution output to learn the nonlinear input patterns. The pooling layer combines the activation function outputs into a single value. After several convolutions and pooling operations, useful features are extracted to perform classification. A fully connected neural network (FCNN) uses these useful features to generate N outputs of which each output corresponds to a particular class. For example, the input is classified as the ith class if the value of the ith output is highest.

In general, a CNN consists of millions of network parameters which require a certain amount of time to determine. For example, ShuffleNet has two million parameters [43]; GoogLeNet [44] has 6.4 million parameters; DenseNet has 8–33 million parameters [45]; ResNet has 11.5 million parameters [46]; AlexNet has 62.4 million parameters [47]; VGGNet has 138.4 million parameters [48]; and in particular, ConvNet [48] consists of 133–144 million network weights. More than two weeks are required to train a ConvNet when a system equipped with four NVIDIA Titan Black GPUs is used. OpenAI developed a CNN, namely neural architecture search (NAS), consisting of 175 billion parameters for NLP. Six months are required to train this CNN when 8×P100 in parallel scalings are used [1]. Using a stand-alone computer to train a large CNN is unreasonably time consuming and hence unfeasible; purchasing many computers to train a single CNN is not cost effective.

Examples of CNN deployment in cloud infrastructure to enhance scalability and reduce computational time, were demonstrated in a number of applications including facial emotion recognition [49], business intelligence for trading [20], web spam detection [50], intrusion detection [51], detection of reoccurring anomalies [52,53], classification of protruding lesions [54], image-aware inferencing [55], and intelligent video recording [56].

2.4. Graph neural network

A GNN was developed based on graph representation learning [57–59] which embodies transforming and learning constituents of a graph (nodes and edges) into a low-dimensional continuous space. GNN covers non-Euclidean domains with complex data structures representing relationships between these entities [60], despite using Euclidean 1-D sequences such as texts and 2-D grids such as images [61]. In GNNs, a graph $G = (V, E)$ is the abstraction of an underlying data structure, where V denotes the set of vertices or nodes, and E denotes the edges between them. The relation $(u, v) \in E$ can be either symmetric or asymmetric. Graphs can be

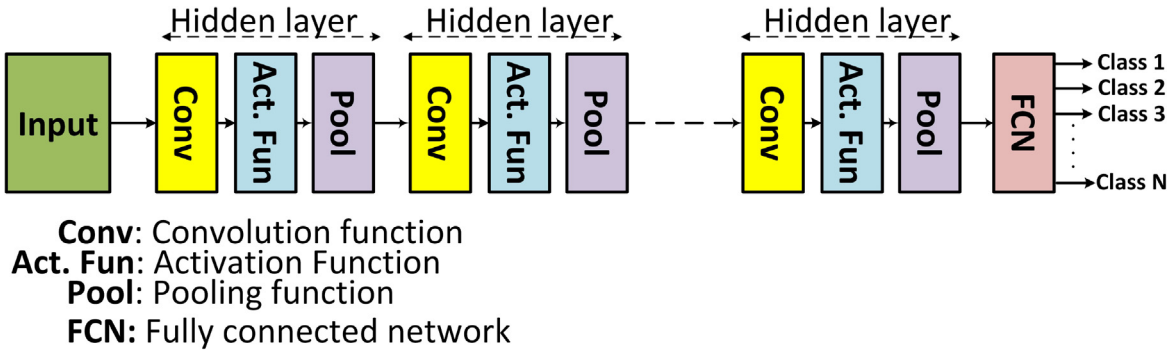

Fig. 2. CNN for classification.

homogeneous, such as social networks, as in Facebook friendships shown in Fig. 3. Nodes and edges can be heterogeneous as in knowledge graphs. Furthermore, graphic topologies or embedded features can change over time.

The number of nodes grows exponentially for GNNs, incurring high computational complexity and memory consumption [62]. For example, ForceNet-Large [63] has 34.8 million parameters; DimeNet++-Large [64] has 10.8 million parameters; SpinConv [65] has 8.9 million parameters; and GemNet-T [66] has 31 million parameters. The Twitter social network is a very large-scale complex graph which contains hundreds of millions of nodes and billions of edges [67,68]. Some large GNNs such as GemNet-XL have several billion parameters [69]. Various current GNN models have only been tested on graphs of very modest size and proved to be inadequate for large-scale graphs which embody complex architectures [70]. Examples of GNN deployment in cloud infrastructure to enable scalable and efficient graph analysis, were demonstrated in a number of applications including recommender systems [71], traffic flow prediction [72], industrial IoT [73,74], privacy preservation [75], and matrix completion [76]. Discussing the sources of latency in training and inference stages of different DNN architectures is important for understanding the optimization of deployment in cloud-based applications. For example, dilated convolutions in CNN can increase the receptive field of the network without increasing the number of parameters or layers, which can help reduce computational complexity and inference latency [77,78]. Randomization-based learning techniques [79] such as echo state network (ESN) can also help reduce latency in training by obviating the need for backpropagation gradients through time, which can be computationally expensive. Pruning techniques in GNNs and CNNs can also help reduce the number of parameters and computations required, which can lead to faster inference times [80,81].It is important to note that different architectures have different prerequisites for training and inference latency, and thus, the techniques used to overcome them may vary. For example, while dilated convolutions may be effective in reducing latency in CNNs, they may not be as effective in other architectures like RNNs or GNNs. Therefore, it is important to consider the specific architecture and its unique sources of latency when developing optimization techniques for cloud-based deployment.

Table 2 summarizes the computational complexity and required training time for commonly used DNNs.

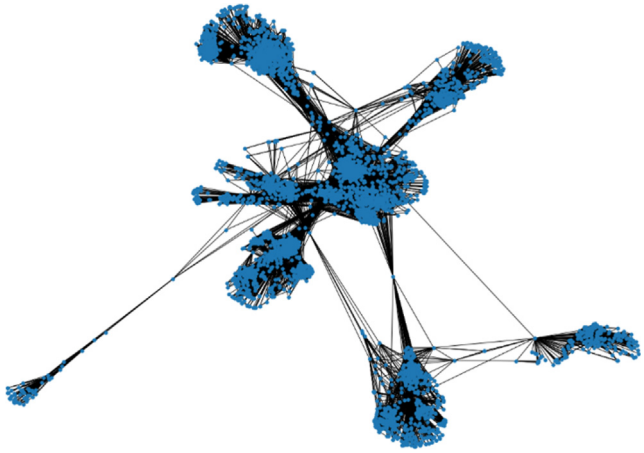

Fig. 3. A symmetric graph of a real anonymous Facebook online social network dataset.This visualization is generated by the second author's own code and is based on the dataset obtained from http://snap.stanford.edu/data/egonets-Facebook.html.

3. Cloud computing architectures for deep learning based applications

DNN structures are complex and require massive numbers of parameters. The time required for training and execution is long. Thus, it is not practical to use a single stand-alone computer to train or deploy a DNN. Cloud computing is a solution for such demanding computations. Cloud computing provides huge amounts of computing power and data storage to users for various DNN implementations and training, which are both computation-

Table 2
Computational complexity in DNNs.

Deep neural networks	Computational complexity
Multilayer perceptron (MLP)	Number of hidden nodes h_i in the ith layer is $N_h : n \cdot h_1 + \sum_{k=1}^{N_h-1} h_k \cdot h_{k+1} + h_{N_h} \cdot n$,Longer computational times are required to optimize MLP when N_h and h_k are larger.
Recurrent Neural Network(RNN)	Number of parameters of a simple RNN is $N_h^2 + m \cdot N_h + N_h \cdot n$,where N_h, m, and n are the dimensions of hidden, output, and input layers, respectively.
Convolutional neural network(CNN)	ShuffleNet has 2 million parameters [43]; GoogLeNet has 6.4 million parameters [44]; DenseNet has 8–33 million parameters [45]; ResNet has 11.5 million parameters [46]; AlexNet has 62.4 million parameters [47]; VGGNet has 138.4 million parameters [48]; ConvNet has 133–144 million network weights [48].
Graph neural network (GNN)	ForceNet-Large has 34.8 million parameters [63]; DimeNet++-Large has 10.8 million parameters [64]; SpinConv has 8.9 million parameters [65]; GemNet-T has 31 million parameters [66].

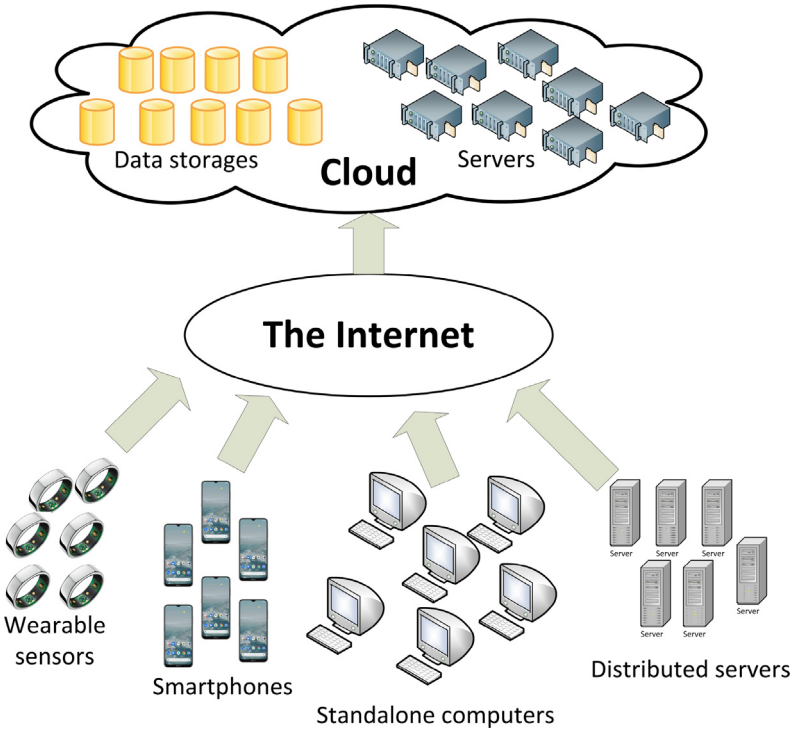

Fig. 4. Cloud data center.

ally demanding. Thus, the cloud benefits users performing intensive applications using DNNs [82].

This section is organized as follows: Section 3.1 presents the structure and architecture of cloud data centers. Section 3.2 introduces the commonly used commercial cloud platforms for DNN deployment; an overview is also presented of public or volunteer cloud computing platforms used to deploy DNNs. Section 3.3 presents the commonly used streaming platforms in the cloud; data-streaming implementations in the cloud are also discussed. This information benefits DL researchers who require powerful, cost-effective, and fast computational platforms to develop DNNs.

3.1. Cloud data centers

Data storage and computations are performed in a cloud data center or on remote clouds such as backhaul and core networks [83]. Fig. 4 illustrates a general cloud-computing architecture consisting of cloud users, internet network providers, and cloud service providers. Computational data are transmitted by users through network providers, and the data are received by servers. Cloud resources are requested to process the data. Users request sufficient access to a shared pool of cloud resources. To satisfy user demands, cloud resources are leveraged to deliver flexible computing capacity and storage [84], thereby supporting cloud providers and making businesses such as Amazon and Google cloud highly profitable [85].

Computational requests from users are shared among distributed cloud platforms with multiple data centers [86]. Resources are shared within a data center or between data centers to facilitate demanding computations. In addition, a distributed cloud can be integrated with a public cloud, hybrid cloud, and edge computing, to further increase computational power. User requests can be allocated to a data center nearby, to reduce data transmission latency. Fig. 5 illustrates the distributed cloud architecture which consists of a distributed cloud and many sub-clouds. A central controller in the distributed cloud allocates computational tasks to sub-clouds based on resource availability. The workload allocations are dependent on the physical limitations of the sub-clouds, such as distributed units (DUs) and central units (CUs). The sub-cloud performs the work by directing it from a single node to a multi-node edge site.

3.2. DL in the cloud

A main advantage of cloud computing is providing dynamic computing resources, and this is particularly important for numerous DL workloads with varying levels of computational requirements for different tasks and datasets. Many cloud providers offer such services, for example Amazon EC2 auto scaling[1] or Microsoft Azure scale up (increased capacity) and scale out (multiple instances).[2] Cloud providers facilitate DL workloads on limited cloud resources.

3.2.1. Parameter server

The approach of parameter servers (PSs) [87] is developed to scale distributed ML contexts in cloud data centers [88]. PSs have been implemented on many DL platforms, such as tensorflow[3] and mxnet,[4] to train DNNs.

Server failures are still possible in cloud data centers, and hence, servers are not fully reliable. Learning task terminations in a cloud environment have to be preempted by appropriate job-sharing and backup [88]. The PS framework consists of a server group and worker group. The server group has a manager and server nodes. The worker group has a task scheduler and worker nodes. The group has access to DNN training. The shared parameters are represented as vectors of (key, value) using consistent hashing. Working information from nodes is pushed to the server and global

[1] https://aws.amazon.com/ec2/autoscaling/
[2] https://learn.microsoft.com/en-us/azure/app-service/manage-scale-up
[3] https://www.tensorflow.org/tutorials/distribute/parameter_server_training
[4] https://mxnet.apache.org/versions/1.8.0/api/faq/distributed_training

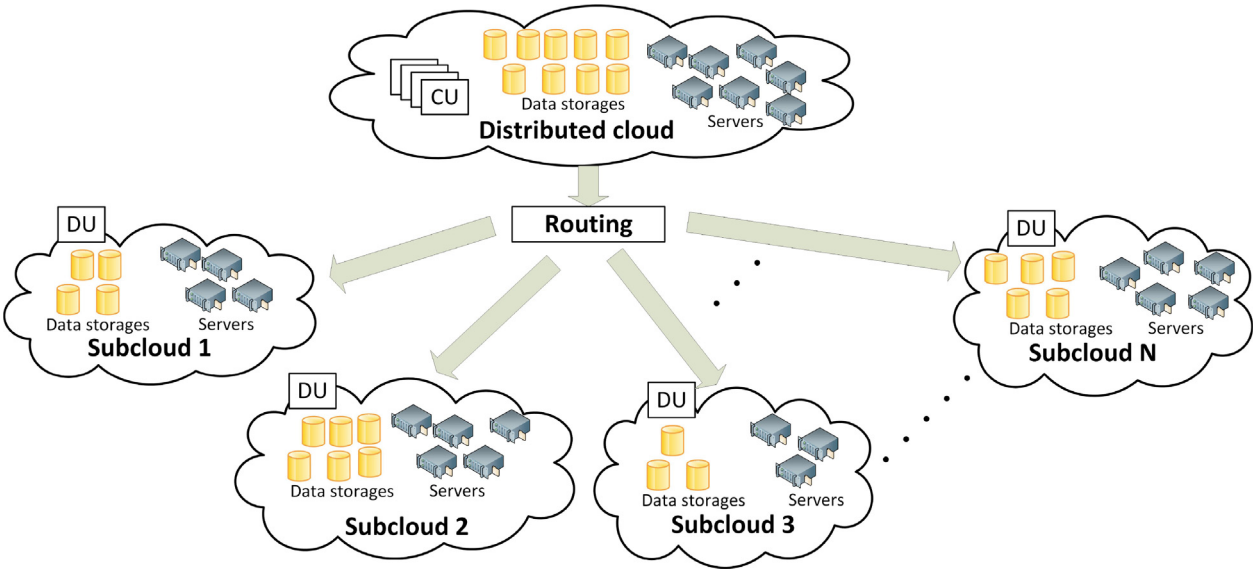

Fig. 5. Cloud system with distributed architecture.

information is then pulled from the server to each node. The framework supports asynchronous tasks and dependency by invoking a call. The framework also facilitates flexible consistency when the algorithm is not sensitive to data inconsistencies. Hence, reliability in PS can be improved.

The PS architecture is more suitable for heterogenous production data centers and public clouds compared with other approaches such as AllReduce[5] because the computing clusters are usually connected to a large and dynamic pool of resources. However, the original PS framework has several limitations including lack of elasticity, imbalance, and static parameter assignments. Extra available resources cannot be integrated into training tasks which have been started. Many workloads assigned to each node do not have optimal capacities. Approaches have been proposed to overcome these limitations to empower cloud computing. For example, Harlap et al. [89] proposed an elastic PS framework called Proteus to scale up training on public clouds. The framework uses three transition stages to dynamically assign PSs and workers, as a cost-saving measure, when transient revocable resources are available. To improve elasticity, Litz [90] introduced logical executors which map physical nodes and allow an executor state to control individual applications. This approach also uses micro-tasks to establish dependency, thereby performing micro-task dispatch accordingly.

3.2.2. Advanced learning frameworks

Advanced learning frameworks have been proposed to improve system reliability and performance for training DNNs in the cloud. The DL-driven framework DL2 [91] uses a combination of supervised learning and reinforcement learning to schedule workloads and dynamically resize resources allocated to jobs. The DNN is first trained offline to learn the resource allocation from past decisions, and reinforcement learning is then used to train the DNN. More recently, Chen et al. [92] proposed a dynamic PS load distribution scheme, called PSLD, which uses the exploitation–exploration strategy. The scheme consists of three stages. First, information is gathered on each PS. Second, profiling is performed by requesting that each worker measure the performance of each PS. Based on the gathered information, superior and low-performing PSs can be identified. Third, a PS is selected from either the superior set (exploitation) or the entire PS set (exploration) in a probabilistic manner, considering the communication time between workers and workloads. PSLD also performs dynamic PS scaling using the information of redistributing gradients to PSs. The PSLD requests workers to update information at each iteration. Wang et al. [93] proposed a lightweight method called elastic parameter server (EPS) which allocates and deallocates resources to jobs dynamically. The approach attempts to improve resource utilization and training speed. Two heuristic scheduling modes, namely incoming job scheduling and running job scheduling, are used to improve scalability. These two modes update selected workers for scheduled jobs and remove workers who have no access to extra resources.

Recent frameworks are more specific with regard to DL workloads and private clouds. For example, Hu et al. [94] studied practical resource scaling issues on AWS and Huawei clouds. They found that only a small number of computationally intensive training jobs monopolize the system's resource pool; jobs in another queue require significant amounts of waiting time. To reduce the waiting time, a concept of training progress based on integer programming was proposed to perform the optimization. Such an approach is particularly useful for cloud services, such as Google Kubernetes Engine (GKE),[6] Amazon Elastic Container Service (ECS), Red Hat OpenShift Container Planform, and Huawei ModelArts. Menouer et al. [95] proposed a scheduling strategy specifically developed so that cloud providers can select optimal computing nodes to execute submitted Kubernetes containers, which is based on a multi-criterion strategy involving node computing information, such as utilization of CPU, memory, disk, power consumption, number of running containers, and container size. The multi-criterion strategy aggregates all criteria in a single rank. Performance evaluation was conducted with respect to computing time, power consumption, waiting time, makespan, and power consumption.

3.3. Data streaming for the cloud

In many sensor networks and control systems such as driverless cars or smart grids, real-time data streaming is essential for DL. Real-time measures or data have to be captured to perform recognition or for decision making. Otherwise, system performance and

[5] https://github.com/baidu-research/tensorflow-allreduce

[6] https://kubernetes.io/

Table 3
Cloud data-streaming platforms.

Cloud data-streaming platforms/ Providers	Links	Deployment of DNNs
Spark streaming/ Spark parallel data analysis system	spark.apache.org/streaming	Create data ingestion and streaming pipelines for deploying DNNs and running distributed SQL queries
Apache Flink/ German Stratosphere	flink.apache.org	Machine learning APIs and infrastructures that simplify the building of pipelines for DNN deployments
Google's Cloud Dataflow/ Integration of Flink, Spark, and Google's Cloud [100]	beam.apache.org	Deploy batch and streaming data processing pipelines to simplify large-scale data dynamics for DNN deployments
Amazon Kinesis/ Amazon Cloud Service [100]	aws.amazon.com/kinesis	Ingest streaming data using Kinesis Data Streams and use DNNs to emit the results to AWS data stores such as Lambda, DynamoDB
Azure Event Hubs/ Azure	azure.microsoft.com/en-us/services/event-hubs/	Use dynamic data pipelines to stream big data and use DNNs as user-defined functions to perform real-time data analysis
IBM stream analytics/ IBM	ibm.com/analytics/us/en/technology/stream-computing	Build streams flow and pipeline to analyze data using IBM Watson and use DNNs to perform real-time predictions

safety levels decline because the measures or data are not the most up-to-date versions. As data volumes captured in each sample can be large, DNNs cannot be retrained using a stand-alone computer. Therefore, data streaming is essential in a cloud. Section 3.3.1 introduces several cloud data-streaming platforms for research purposes and Section 3.3.2 discusses some recent data-streaming approaches for cloud systems.

3.3.1. Cloud data streaming platforms

In academic or government sectors, cloud data-streaming platforms have been used to analyze data captured by sensor networks. For example, a geophysical sensor network has been developed by the Southern California Earthquake Center to perform geospatial data analysis [96]. This sensor network is installed with thousands of sensors to capture data continuously at a high sampling rate. The captured geospatial data are used to study climate change and develop earthquake and inland flooding forecasting systems. The Geodesy Advancing Geosciences and EarthScope (GAGE) global positioning system (GPS) network uses data from more than a thousand GPS sensors to study seismic hydrological properties in North America [97]. The US National Science Foundation-funded developed a sensor network at multiple worldwide locations to study climate change and carbon cycling [98]. A sensor network was developed by the City of Chicago array of things (AoT)[7] to capture city-related data such as environmental temperature, humidity, surface vibrations, and magnetic fields along with atmospheric information such as carbon monoxide and air particles. The captured data are analyzed for future city development and planning.

Several other cloud data-streaming platforms for research purposes are reported in [99]. These data-streaming platforms are provided by open-source communities and are summarized in Table 3.

3.3.2. Data-streaming approaches

As many DNNs are implemented in time-varying environments, they must continuously learn or be retrained on newly captured data. A data-streaming approach is developed to determine when streaming data are required to update DNN parameters [101]. The approach decides whether to update DNNs by trading off performance and training costs. The approach was implemented on TensorflowOnSpark for three online learning workloads, thereby reducing the overall elapsed time. Ashfahani et al. [102] developed a data-streaming approach to modify the network structure with respect to newly captured data. Network nodes can be grown and pruned depending on time-varying dynamics to increase network performance while minimizing network complexity. When the approach was tested on commonly used datasets, the DNNs developed by Ashfahani et al. [102] outperformed those developed by existing approaches in terms of network complexity and performance.

Li et al. [103] proposed an incremental high-order DL model to adapt online data which are captured at a very high sampling rate. The approach first transforms the data in a vector space into a high-order tensor space to reduce adaptation time. The first-order approximation is then developed to avoid parameter incrementation which is time consuming but common in iterative methods. Hence, DNNs are more efficient at adapting to time-varying environments and satisfying real-time requirements compared with existing iterative methods. Pratama et al. [104] proposed a novel fuzzy neural network which automatically embeds fuzzy rules from data streams. A simplification procedure was used to merge redundant hidden layers to prevent uncontrollable growth in the network size. Experimental results showed that the approach is effective at controlling network size and maintaining network performance. Nguyen et al. [105] also developed a sensor network to capture maritime data to enhance maritime traffic levels, safety, and security in real time. A deep recurrent neural network integrated with streaming data was developed to control fishing activities, detect smuggling and transshipment, and forecast maritime pollution. Latent variable modeling and data streaming were combined to capture key components in maritime dynamics. The approach was effective at analyzing noisy and irregular time-sampling data in maritime environments.

When imposing data streaming on the design of DNNs in the cloud, low inference latencies are required for the DNNs to predict in real time. To respond to actions in real time, the serving latency can be improved in two ways.[8] First, smaller DNNs can be designed or accelerators can be used to boost DNNs. Second, data features can be stored in a low latency data storage location. Those data features can be used to perform offline precomputing predictions to reduce the response time for real-time predictions. To further enhance DNN responses and accuracy, incremental training can be used to adapt to newly streamed data [106]. After deploying DNNs for a certain period of time, the predictions of the DNNs are not as accurate as they used to be; the environments change gradually such that past data do not fully indicate the current environment which is time-varying. A DNN can be trained on newly streamed data, the patterns of which are not included in the previous training. Integrating data streaming with incremental training is essential to enhance DNN performance. Based on the newly streamed data, incremental training updates the DNNs at regular intervals. Model artifacts from a popular publicly available DNN can also be used to adapt to the

[7] arrayofthings.github.io

[8] https://cloud.google.com/architecture/minimizing-predictive-serving-latency-in-machine-learning

newly streamed data. A new DNN can be updated without training from scratch.

4. Applications of DNNs in the cloud

Cloud-based DNNs have been deployed in various applications. This section presents several DNN cloud applications including NLP, business intelligence (BI), cybersecurity, anomaly detection, travel, wireless capsule endoscopy, and mobile-cloud-assisted implementations. These applications are presented in subsections which are followed by tables summarizing the application content. The summaries help identify the research challenges involved in applying cloud systems to DNNs, as discussed in Section 5.

4.1. Natural language processing

Natural language processing is an important application in DL for automatically manipulating natural language such as text and speech. NLP can be used in medicine, security, economics, and agriculture, and relies on the availability of accurate textual material and related tools [107]. Sentiment analysis (SA) is an NLP task that measures the sentiment of the text. Medhat et al. [108] defined SA as a computational approach for processing sentiments, subjectivity, and opinions within the text. Feldman et al. [109] described SA as opinion mining, seeking the authors' opinion on particular entities. Many DL approaches have been proposed to solve problems related to SA [110–112]. The utilization of cloud computing and DL is essential to the processing of huge amounts of text data in SA, and some attempts have been made to utilize cloud computing resources for SA on social networks [18].

Sinnott et al. [113] proposed a cloud-based sentiment approach based on DNN for solving sentiment prediction problems in relation to opinions expressed on the social network Twitter. A DNN was implemented using cloud resources to analyze Twitter big data. These resources helped in data processing, analysis, distribution, and storage of large-scale data. Their approach outperformed other approaches with an accuracy of 80%. An SA cloud-based system was proposed for facial emotion recognition (FER) of medical patients [49]. In this system, a CNN was used to capture facial expressions and classify the emotions as disgusted, happy, angry, sad, neutral, or surprised. Furthermore, the FER model was implemented on a cloud-based GPU platform to increase processing speed. The proposed framework could measure and identify the intensity of patient's pain based on the FER system.

A recommendation system for SA was proposed by integrating DNN in the cloud such that an RNN recommends places based on the user's current location while analyzing the reviews of these places simultaneously. The accuracy of the recommendation system was improved by having RNN learn the collected reviews. Similar to this approach, stock price prediction through SA was proposed by Mohan et al. [19]. This approach gathers volatile information including various economic and political factors, investor sentiment, and leadership change to perform predictions. Mohan et al. [19] created a dataset by collecting five years of daily stock prices from S&P500 companies and nearly 265,000 financial news articles. As the data size was very large, cloud resources were employed to train their DL-based prediction model, whereby more accurate stock price predictions were achieved compared with other classical prediction methods.

The rapid development of social media and certain websites with critical reviews of products serve as a huge collection of resources for customers worldwide. Reviews of essential services and products are useful resources when making business decisions [111]. The data contains information regarding customers' opinions about services and products, opinion sentiment, and market changes. Ghorbani et al. [114] applied a model combining long short-term memory (LSTM) with CNN to determine word polarity in a cloud environment. The proposed model utilizes DL algorithms and word embedding techniques for feature representation, thereby achieving more than 89% accuracy. Raza et al. [115] employed LSTM and RNN for sentiment prediction using a cloud review dataset. The dataset contains consumer reviews of the cloud provider services. The experiments show that the proposed approach with RNN and LSTM achieved 95% accuracy in predicting customer opinion.

The aforementioned approaches demonstrate that the utilization of cloud computing and DL is very effective for various resource-intensive NLP applications, such as SA. Table 4 summarizes the DL deployed for NLP using cloud computing mechanisms, covering the characteristics of the deployments including network models, loss functions, specific task, evaluation metrics, datasets and state-of-the-art methods, and the corresponding performances.

4.2. Business intelligence

Business intelligence helps organizations analyze and process business data through a decision support system, in an attempt to produce sustainable, stable, and productive businesses [117]. BI deployed in the cloud enhances production for available big BI data. Many services on the cloud are made accessible to BI specialists by cloud providers such as Amazon's Redshift hosting BI data warehouse, Google's BigQuery data analytics service, IBM's Bluemix cloud platform, and Amazon's Kinesis data processing service [118]. Deploying BI in the cloud has attracted much attention from researchers. Balachandran and Prasad [118] have discussed the benefits, challenges, and risks of deploying cloud storage and cloud computing for BI. Cloud deployment improves BI processes because cloud services are cost-efficient and easily available; cloud services also provide fast deployment and ease of integration. In addition, cloud BI-based approaches have been discussed for improving the decision-making strategies of business owners.

Solutions for general-purpose BI services are also provided based on cloud computing and DL. Prasomphan [20] proposed a chatbot service to provide customer support information, training schedules, customer reservations, and virtual assistant services whereby CNNs are used to understand and respond to customer questions and comments. Moreno et al. [119] proposed a BI and data-driven cloud architecture which uses cloud services to analyze big data using a prescriptive rather than a descriptive approach. The proposed architecture uses big data to help modern market-oriented organizations determine new marketing insights, create new products and services for potential customers, and produce new lines of business.

Certain domains of BI have also been integrated with cloud computing and DL. Mohan et al. [19] considered time-series data from news articles regarding daily stock prices of 500 companies. They also used DNNs to predict stock prices in the cloud environment, whereby DNNs were used by decision-makers to infer real-time stock prices. Autoregressive integrated moving average (ARIMA), RNN, and LSTM were incorporated into generate models and the performance was evaluated. Juarez and Afli [120] studied the stock market prices based on a framework which was integrated with DL and cloud computing for web applications. The framework analyzes news articles from financial publications and predicts whether stock prices will increase or decrease to advise users to buy or sell in the stock market. Khan et al. [117] developed a model based on DL to perform forecasting for products using real-time organization data collected from the market. A DL model implemented in the cloud was used to predict weekly, monthly, and quarterly product demands based on these real-time data.

Table 4
Deploying DLs in cloud for NLP.

Applications	Network models	Lossfunctions	Evaluationmetrics	Datasets	Performance
Twitter-based sentiment analysis [113]	DNN	Not reported	Accuracy	55 million Tweets	Compared with other approaches, their approach is 80% more accurate.
Facial emotion recognition (FER) [49]	CNN	Error function	Probability	FER2013	Patients receive easy yet accurate treatment.
Recommendation system [116]	RNN	Not reported	Accuracy and F1 measure	Movie and restaurant datasets	With the help of recursive methods and *Z*-values, the proposed system enables users to identify precise location.
Stock price prediction [19]	RNN and LSTM	MAE	MAPE	Five years of daily stock prices and 265,000 financial news articles	In comparison with other classical prediction methods, DL improves stock price accuracy.
Polarity in a cloud environment [114]	LSTM and CNN	Not reported	Accuracy	Not reported	The proposed model has been demonstrated to be more than 89% accurate.
Reviews sentiment [115]	LSTM and RNN	Cross-entropy	Accuracy, precision, recall, and F1 score	Cloud review dataset (6259 reviews)	The approach typically achieves an accuracy of 95%.

Table 5
Deploying DLs in the cloud for business intelligence.

Applications	Network models	Loss functions	Evaluation metrics	Datasets	Performance
Future sale/product demand forecasting [117]	DeepAR	Not reported	Accuracy	Sales, inventory, and calendar data	92.38% accuracy for store demand forecasting
Chatbot in trading system for SMEs [20]	CNN	Not reported	Accuracy	Cornell_Movie-Dialogs_Corpus dataset	DNN accuracy for chatbot learning is better than traditional models
Stock price prediction [19]	ARIMA, RNN, and Facebook Prophet	Not reported	MAPE	Daily stock prices for S&P500 companies for five years, more than 265,000 financial news articles	Better results achieved with RNN; higher correlation between textual information and stock price direction
Stock price prediction [120]	RNN and sequential model	Sigmoid	Accuracy	Google News articles from financial publications	46% accuracy for the 2505 test set, 43% accuracy for the 3521 test set, and 63% accuracy for the 3789 test set

Table 5 summarizes the DL deployed in cloud-computing mechanisms for BI.

4.3. Cybersecurity

Cyberspace has evolved at a dramatic pace because of the augmented connectivity of a vast volume of electronic devices causing an urgent need to develop sophisticated technologies to ensure space security. Therefore, cybersecurity is essential for safeguarding systems to avoid potential digital attacks [121]. Furthermore, the continuous propagation of big data can be considered an opportunity to detect and deter digital threats [122]. This can be attained by incorporating DL techniques such as CNN, deep belief networks, generative adversarial networks (GANs), RNNs, deep Q-networks (DQNs), and autoencoders [123,121]. Incorporating DL techniques with cloud computing, offers automatic intelligence capability for the detection and prevention of cyber threats, such as phishing, spamming, and spoofing.

Gupta et al. [124] developed a DNN to detect and prevent cybersecurity threats in healthcare systems. The DNN predicts cyberattacks on dataflows by analyzing the data being transferred between clouds and detecting the changes which occur in the associated metadata. This is accomplished by using a hierarchy of cooperating DL models in an edge computing environment to reduce training time. Chai et al. [125] proposed a DNN to detect meaningful phishing websites. Specifically, three modalities of content namely webpage text, navigation content, and visual content of images were used as DNN inputs.

To stop social spammers, Abu-Salih et al. [126] developed a systematic and effective approach. The notions of spammers' behavior and topics of interest were built to extract consolidated features which were used as the inputs of ML- and DL-based classifiers. The utility of Abu-Salih et al.'s [126] approach was validated by cloud computing services including the Australian Pawsey cloud-based services. Approaches based on DL and cloud services were also proposed to detect spammers, phishers, and similar categories of cyber threats [127–130,50]. Abdullayeva et al. [127] developed an autoencoder DL-based approach to detect advanced persistent threats which target confidential and personal data in the cloud. The utility of the proposed model was validated over a large volume of data in the cloud environment. Makkar et al. [50] proposed a framework to detect web spam at three layers of services, namely, data collection, edge computing, and cloud computing. The developed system architecture outperformed the previously developed cloud-based spam detection schemes.

Intrusion and threat discovery from very large-scale event and log datasets point to another important research direction. Technologies have been developed by implementing DL in cloud computing. Sethi et al. [131] proposed a DQN by incorporating cloud infrastructures; the approach achieved higher accuracy in detecting attacks. Hossain et al. [132] proposed a log data reduction technique for forensic analysis. The developed system was able to offer a reduction factor of 4.6 to 19, displaying capability for analyzing around 1 million events per second. DeepLog [133] is another DL approach based on an LSTM technique which is designed to parse

Table 6
Deploying DLs in the cloud for cybersecurity.

Applications	Network models	Loss functions	Evaluation metrics	Datasets	Performance
Security in healthcare [124]	Stacked autoencoder[1]	MSE	Accuracy, precision, recall, and F1 score	UNSW-BOT-IoT dataset[2]	Shorter computational time is achieved with improved performance
Detecting phishing websites [125]	Multi-modal hierarchical attention mechanism (MMHAM)[3]	Cross-entropy	Accuracy, precision, recall, and F1 score	DMOZ[4]	More effective and accurate for detecting phishing websites
Detecting social spam [126]	Multi-layer feed-forward neural network	Quadratic	Average precision, recall, and F1 score	Social manually-labeled Twitter dataset	Higher model utility was validated by cloud computing services including Australian Pawsey cloud-based services
Detecting advanced persistent threat [127]	Autoencoder DNN	Reconstruction loss	Accuracy	Malware training sets [136]	Higher model utility was validated through a large volume of data in the cloud environment
Detecting web spam [50]	LSTM and CNN	Categorical cross-entropy	Accuracy	Manually collected dataset	A reduction in response time of up to 70% when transferring computation of DNN model to the edge
Intrusion detection [131]	Deep reinforcement learning	Not reported	Accuracy and area under curve	UNSW-NB15[5]	Higher performance achieved in identifying new and complex attacks
Intrusion detection [134]	RCNN	Not reported	Accuracy and precision	DARPA IDS[6] and CSE-CIC-IDS2018[7]	More efficient threat classifier to protect cloud network layers
Intrusion detection in multi-cloud environments [51]	CNN	Cross-entropy	Accuracy, detection rate, and precision	NSL-KDD[8]	Higher performance in multi-cloud IoT environment

[1] Autoencoders are a subcategory of unsupervised ANNs, used to minimize the dimensionality of the data when a nonlinear function defines how dependent and independent characteristics relate to one another. [2] The BoT-IoT dataset was developed in the UNSW Canberra Cyber Range Lab by constructing a realistic network environment. The traffic on the network is a mix of regular and botnet activity. The source files for the dataset are offered in a variety of forms, such as original pcap files, produced argus files, and csv files [137]. The dataset can be accessed via the following link: https://research.unsw.edu.au/projects/bot-iot-dataset (accessed 28/11/2022) [3] The hierarchical attention mechanism, which was first developed for document categorization, expands the classic attention mechanism by making use of the hierarchical structures of the document [138] [4] A multilingual open-content repository of WWW links: https://dmoz-odp.org/ (accessed 29/11/2022) [5] A dataset for network intrusions is UNSW-NB15 comprising nine different attacks, including worms, backdoors, DoS attacks, and fuzzers. Raw network packets are included in the collection. A total of 175,341 records make up the training set, whereas 82,332 records from the attack and normal types make up the testing set [139] [6]https://www.ll.mit.edu/r-d/datasets/1999-darpa-intrusion-detection-evaluation-datase (accessed 28/11/2022) [7]https://www.unb.ca/cic/datasets/ids-2018.html (accessed 28/11/2022) [8]https://www.unb.ca/cic/datasets/nsl.html (accessed 28/11/2022).

large volumes of log data to detect cyberattacks. The system could detect five out of six attacks in the VAST Challenge 2011 dataset.

Intrusion detection systems (IDSs) have also incorporated cloud computing and DL for large-scale events and log data [134,51,135]. Thilagam and Aruna [134] developed a convolution-based RNN to build an IDS in the cloud network environment. The Ant Lion optimization algorithm was embedded into LSTM and CNN to build a threat classifier to protect cloud network layers. Two datasets namely DARPA IDS[9] and CSE-CIC-IDS2018[10] were used to evaluate model performance. Selvapandian et al. [51] built a CNN to detect intrusions in a multi-cloud IoT environment. CNN with minimal features was developed to address the vulnerabilities caused by network complexity and open broadcast characteristics of IoT networks. The performance of the CNN was verified using the NSL-KDD dataset[11] which simulates intrusions in a multi-cloud IoT environment.

Table 6 summarizes the DL deployed in cloud computing mechanisms for cybersecurity.

4.4. Anomaly detection

Similar to humans who are capable of detecting anomalies signaling potential hazards or beneficial opportunities, DNNs have recently been deployed on the cloud to perform anomaly detection [140]. Wu et al. [141] developed a model based on GNN to detect anomalies in industrial IoT which embodies an evolving interconnectivity between sensors, instruments, and other IoT devices. The evolving inter-connectivity is distributed and managed by multi-cloud systems. Scheinert et al. [52] developed a multivariate time-series-based classification model (TELESTO) to detect recurring anomalies which are deployed for infrastructure as a service (IaaS) in the cloud environment. The TELESTO model transforms multivariate time series into graphs in spatial and temporal dimensions. A novel graph CNN architecture is used to detect recurring anomalies using graphic classifiers. Detecting and localizing anomalies using GNNs were also studied in the distributed cloud [21,142,143,22]. In addition, the complexity introduced by state-of-the-art distributed smart resources such as electrical vehicles, smart heating and cooling systems, and other internet-connected devices can be mitigated by designing GNNs in the cloud [144] to model the collective behavior of interconnected assets.

Cloud-based GNNs were also implemented for anomaly detection applications. Lee et al. [145] developed a GNN-based method for winner determination in multi-unit combinatorial auctions (CAs) running on the cloud. An augmented bipartite bid–item graph-based GNN was developed to learn a continuous probability map which indicates the probability of each bid belonging to the designated optimal allocation. The approach resolved the combinatorial auction problems in cloud networks. Gao et al. [146] developed a GNN to tackle the low-utilization-rate problem for cloud computing resources. The approach of Gao et al. benefited from several variations of GNN architectures constructed based on homogeneous and heterogeneous graphs, making predictions of cloud computing load more accurate. Rafiq et al. [147] examined the utility of a GNN by modeling the complex relationship between network traffic features. A topology-aware knowledge defined networking (KDN) system based on GNN was proposed to predict optimal paths for service function chaining (SFC) deployment and traffic steering. The utility of Rafiq et al.'s model was validated

[9] https://www.ll.mit.edu/r-d/datasets/1999-darpa-intrusion-detection-evaluation-dataset
[10] https://www.unb.ca/cic/datasets/ids-2018.html
[11] https://www.unb.ca/cic/datasets/nsl.html

Table 7
Deploying DLs in the cloud for anomaly detection.

Applications	Network model	Loss function	Evaluation metrics	Datasets	Performance
Detect recurring anomalies in cloud services [52]	Graph CNN	Cross-entropy	Accuracy, recall, precision, and F1 score	Manually labeled dataset	Better detection of recurring anomalies that benefit from multi-cloud environment
Winner determination in multi-unit auctions [145]	GNN	Cross-entropy	Execution time complexity	A manual collection of 500 bids	Smaller resource consumption, revenue loss, time complexity, and higher user satisfaction
Anomaly detection in system logs [133]	LSTM	Cross-entropy	Precision, recall, and F measure	HDFS log [155] and OpenStack log (manually collected)	The system is able to detect five out of six attacks in the VAST Challenge 2011 dataset using CloudLab [156]
Detecting low-utilization-rate problems in the cloud [146]	GNN	Not reported	Micro-precision	Various enterprise-based datasets	More accurate cloud computing workflow loads
Anomaly detection in surveillance networks [53]	CNN with LSTM	Not reported	Area under the curve (AUC) and runtime complexity	UCF-Crime[1] and UCFCrime2 Local[2] datasets	Higher functionality within complex surveillance scenarios in cloud environment

[1]:https://www.kaggle.com/datasets/odins0n/ucf-crime-dataset (accessed 28/11/2022) [2]:https://www.kaggle.com/datasets/lamnguyenvu98/ucfcrime2local (accessed 28/11/2022).

by a complete animated environment embodying the open network operating system (ONOS),[12] OpenStack,[13] and open-source MANO (OSM).[14] GNNs embedded into the cloud have proven to be effective in tackling other cloud-based problems including job dispatching [148], mobile app recommendation in edge computing [149], parallel computing [150], connected autonomous vehicles [151], and runtime performance prediction [152].

Real-time and instant anomaly detection have recently attracted the attention of researchers for their importance in overall surveillance and management. Waseem et al. [53] proposed a real-time anomaly detection method for surveillance scenarios using efficient CNN and B-LSTM. Waseem et al. [153] developed an attention residual LSTM for effective anomaly detection. Ullah et al. [154] also proposed a two-stream neural network for anomaly detection in surveillance videos, which was assisted by an AI of Things (AIoT) paradigm. This research reflects an important trend; however, more investigation is required with test-bed simulations and deployments in diverse environments, such as fog, snow, sand, and rain.

Table 7 summarizes the DL deployed in cloud computing mechanisms for anomaly detection.

4.5. Travel

In the tourism and hospitality industries, AI accommodates and processes relevantly large datasets which are generated by both consumers and service providers. For example, one of the large relevant data islands in tourism is collected from GPS applications [157] which are commonly integrated with social data [158], IoT data [159], and web traffic data [160]. This massive dataset is processed within the context of "smart" tourism industry which provides intelligent and personalized services for tourists [161]. Sophisticated intelligent techniques are required to analyze such heterogeneous, multifaceted, and distributed datasets [162]. Recently, Cepeda and Domingo [163] proposed a DNN-based recommendation system which advises personalized tourist activities/attractions in smart cities. The proposed architecture consists of three technical layers, namely device, fog, and cloud layers. The device layer is used to collect tourist sensor activities including search and visit-planning data. The fog layer is responsible for digesting and dispatching activities which are collected from edge networks. The top layer is the cloud layer which conducts intelligent analytics. In particular, a multi-classification DNN model was developed in the cloud and used to provide personalized and real-time recommendations for tourist attractions. Incorporation of cloud computing reduces data processing time.

DNNs were also proposed to understand tourists' behavior and improve their experiences [164,165,161,166]. Piccialli et al. [165] developed a DNN for path prediction to monitor and predict occupancy of available rooms. Visitors' behavioral data was collected by IoT and cloud services thereby providing effective real-time predictions, offering tourists improved visiting routes. Chang et al. [166] proposed a multiple-CNN model which is integrated with the Word2Vec model to analyze hotel reviews collected from "TripAdvisor" to extract semantic and syntactic relations. The model extracts hidden patterns, understands consumer booking behavior, and furnishes strategic recommendations to decision-makers. Kontogianni et al. [161] developed a DNN for smart tourism. The framework encompasses two modules: (i) image labeling of objects and landscapes and (ii) collaborative filtering based on a DNN architecture. The proposed framework is effective in accommodating large-scale distributed tourism datasets.

Applying DNN technology in transportation and traffic domains has also attracted the attention of research and industrial communities with respect to population growth, climate change, air pollution, and other relevant urgent matters [167]. This is also reinforced by recent developments in intelligent transportation systems which are based on large-scale and distributed sensor data, including traffic density and speed, road cameras, and public transport transponders [168]. DNN technologies are crucial in the development of various solutions in support of intelligent transportation domains such as transportation safety planning [169], traffic speed prediction [170], collision prediction [171], and parking occupancy prediction [172]. However, the sophistication of the developed DNNs and the abundant heterogeneous datasets require high-speed and high-performance computing. Therefore, cloud computing offers a solution for accommodating the computational demand of DNNs, thereby resolving the latency and bandwidth limitations of intelligent transportation systems [173]. Chen et al. [174] developed a DNN-based traffic flow detection system using cloud and fog computing services. In particular, three modules were embedded into the cloud-based system, namely a vehicle detection module, vehicle tracking module, and vehicle counting module. Each module was integrated using cloud and edge computing facilities to allow high-speed data processing.

Iqbal et al. [175] proposed an integrated system which combines an intelligent decision-making scheme with cloud computing to benefit intelligent transportation systems. The DNN was trained using large-scale traffic congestion datasets; the proposed integrated system improves traffic multimedia data transmission.

[12] https://opennetworking.org/onos/
[13] https://www.openstack.org/
[14] https://osm.etsi.org/

Table 8
Deploying DLs in cloud for travel.

Applications	Network model	Loss function	Evaluation metrics	Datasets	Performance
Tourist attraction recommendations in smart cities [163]	MLP	Binary cross-entropy	Loss, accuracy, precision, recall and F1-score	Survey collection	Higher modeling efficiency and less data processing time
Path forecasting in an IoT system [165]	Encoder-Decoder DNN	Categorical cross-entropy	Accuracy	5,200 manually collected visitor paths	More effective real-time predictions for visitor behaviors
To analyze hotel reviews and responses [166]	CNN	Cross-entropy	Recall, precision, and F1-Score	113,685 hotel reviews	Higher utility and effectiveness in extracting hidden patterns for consumer booking
Promoting tourism personalized services [161]	ANN	Mean-squared error	Accuracy and loss	Movielens dataset[1]	More effective in accommodating large-scale and distributed tourism datasets
Traffic flow detection	CNN	Sum of squared errors	Accuracy, precision, and runtime	UA-DETRAC[2]	More effective data processing
Ambulance tracking in video surveillance [175]	ANN	Not reported	Runtime	Manually-collected multimedia dataset	Higher utility for transmitting multimedia data
Intelligent traffic congestion-avoidance system [176]	ANN	Not reported	Case study	Manually-collected vehicle movement data	More effective data analytics for traffic data

[1] This dataset includes 100,000 user ratings for 9000 films:https://www.kaggle.com/datasets/grouplens/movielens-20m-dataset (accessed 28/11/2022) [2] The dataset used in vehicle detection and tracking contains 8250 manually-labeled vehicles and 1.21 million target object frames:https://detrac-db.rit.albany.edu/ (accessed 28/11/2022).

Thejaswini et al. [176] proposed a DNN which was trained on traffic data at the edge. The proposed DNN establishes a "digital twin", for which the data were collected from roadside cameras. The DNN can also be retrained by newly collected data.

Table 8 summarizes the DL deployed in cloud computing mechanisms for anomaly detection.

4.6. Remote medical diagnosis: wireless capsule endoscopy as a use case

Over the past two years, an exponential growth has been observed in wireless capsule endoscopy (WCE). WCE approaches are comparable to conventional endoscopy in visualization of the interior of humans for diagnosis. These technologies were initially developed in 2000 and received endorsement in 2001 following clinical trials by the Food and Drug Administration. The technologies have better portability and distinct applications in systemic biologic delivery [177] and health services [178,179].

In WCE, a patient swallows a pill-sized capsule that contains a camera for recording and monitoring gastrointestinal tracks. The camera attached to the capsule head captures the stream and proceeds to transmit the data to a portable image recording device. This device is fixed to a human body containing an antenna array with a few leads. The time to expel the capsule from the body requires approximately 72 h, with the initial 8 h being the most significant for visual representation of the gastrointestinal tract [180]. During the initial 8 h, about 50,000 frames are captured depending on the capture rate of the underlying capsule. Inspecting all captured frames to identify abnormalities is time consuming. In addition, this approach is not preferable because of the significant amount of redundancy in the frames. The capsule often stops due to food particles, affecting capsule flow and greatly increasing the chance of capturing redundant frames. In traditional endoscopy, the patient must visit the hospital for the endoscopy set to be inserted to enable the transfer of the video data to the diagnostic system. The enormous amount of captured video data requires inordinate efforts and time for physical inspection. A practitioner requires an average of two hours to inspect about 50,000 frames for only one patient.

To solve these issues, a patient's phone uses a light DNN to perform lightweight processing of the generated data during the diagnostic process; the application is deployed in the cloud to identify and discard redundant frames during the WCE procedure [181]. The system is illustrated in Fig. 6(a). In addition, sharing massive amounts of data with specialists and doctors in remote areas for diagnosis is quite challenging because observation and analysis time is excessive. Filtering methods such as those proposed in [182,183] discard and eliminate non-informative and redundant frames [184,185]. Similarly, other techniques such as segmentation [186] and anomaly detection [187–189] in WCE frames need automation for enhanced analysis. Technologies have been proposed for WCE data detection, classification, and segmentation.

To illustrate the concept with an example, the sequence of WCE frames acquired from the capsule is F_t with $t = 1, 2, 3, \ldots, N_T$, where N_T is the total number of frames. The approach eliminates redundant frames and classifies the remaining WCE frames as informative or non-informative. The frames are converted from RGB into a COC color-based model to compute the integral image. Similarly, features such as inertia moments, curvature map, and multi-scale contrast are calculated whereby the saliency scores obtained from these features are normalized. Depending on the available resources such as the patient's smartphone battery, important information is then sent to the health center and cloud systems for further analysis, whereas the non-informative part is discarded [190]183.

Table 9 summarizes the DL deployed in cloud computing mechanisms for WCE.

4.7. Mobile-cloud-assisted applications

A mobile cloud-assisted application is an emerging technology with a wide range of applications. MCC mainly focuses on the capability of offloading tasks towards cloud servers to extend the system lifetime. MCC also reduces the computational burden of mobile devices such as smartphones, tablets, and iPads. An extensive trace-driven assessment is performed to ensure an efficient offloading inference engine and to reduce resource constraints from smartphones with far lower processing power than conventional approaches [195]. During the WCE procedure, analysis and sharing of huge amounts of data require instant removal of redundant and irrelevant frames to prioritize the video before using DNNs. However, video prioritization in WCE is quite challenging and is likely to be infeasible when resources and computational power are limited. Therefore, MCC is incorporated to provide storage, massive computation, and software services at low cost. This mechanism is illustrated in Fig. 6(b). Using this strategy, light-

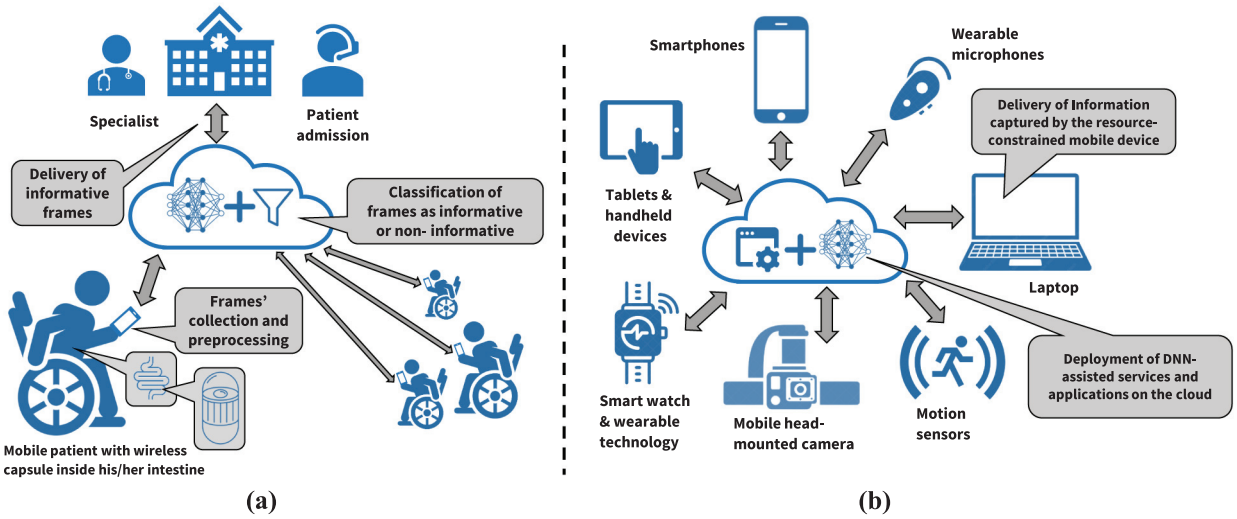

Fig. 6. WCE procedure for personalized and efficient healthcare [183]. (a) The patient's smartphone is used to process the frames, thereby discarding redundant information using a lightweight DNN redundancy removal network. (b) Mobile-cloud-assisted mechanism that assists the WCE in reducing computational power.

Table 9
Deploying DLs in the cloud for wireless capsule endoscopy.

Applications	Network model	Loss function	Evaluation metrics	Datasets	Performance
Bleeding areas segmentation	DFCA-Net (withRes2Net101 as backbone) [179]	Joint loss function	Mean pixel accuracy (MPA), mean intersection over union (mIoU)	GI bleeding image dataset	mIoU: 86.858%, MPA: 95.211%
Image segmentation [187]	CNN	Not reported	Sensitivity and accuracy	Largest WCE dataset with 440,000 images	Accuracy: 88.5%, sensitivity: 84.6%
Polyp recognition [191]	Stacked sparse autoencoder with image manifold constraint	Not reported	Overall recognition accuracy (ORA)	3000 WCE images from 35 patients' WCE videos	ORA: 98%
Bleeding detection [192]	DCNN	Not reported	F1 score	10,000 WCE images	F1 score: 0.9955
Anomaly detection and localization	WCENet [193]	Categorical cross-entropy loss	Accuracy, receiver operating characteristic (ROC), frequency-weighted intersection over union (FWIoU), average dice score (ADS)	KID dataset	Accuracy: 98%, ROC: 99%, FWIoU: 81%, ADS: 56%
Detection of erosions and ulcerations [194]	DCNN	Not reported	ROC	5360 WCE images	ROC: 0.958
Detection and classification of protruding lesions [54]	DCNN	Not reported	AUC, sensitivity, specificity	30,584 WCE images of protruding lesions from 292 patients	AUC: 0.911, Sensitivity: 90.7%, specificity: 79.8%

weight processing can be performed at the edge whereas the resource-hungry activities are performed in the cloud which has powerful computational resources.

Yang et al. [196] presented an offloading service for resource-constrained mobile devices. They combined numerous resources including memory, CPUs, and bandwidth to reserve mobile resources. Miettinen et al. [197] claimed that energy efficiency is significant for mobile devices, and MCC saves energy through offloading strategies. Hsieh et al. [198] showed the strengths of MCC in telemedicine. They advised that the combination of mobile technologies and cloud computing is necessary because high-speed data delivery is required for mobile teleconsultants and big data centers. In addition, patient data are securely stored in and retrieved from the cloud. Fortino et al. [199] proposed a framework to develop cloud-assisted body sensor applications. This framework consists of a multi-tier architecture integrating data streams from body sensors, cloud computing, and middleware. The large-scale sensor data are processed and shared among users in mobile devices and the cloud, to ensure security and privacy issues in mobile-cloud assisted applications.

Table 10 summarizes the DL deployed in cloud computing mechanisms for mobile-cloud-assisted applications.

5. Challenges and future directions

As discussed in Section 4, many applications requiring big data analysis and high-performance computing have been deployed using cloud-based DNNs, benefiting from the large computational and data storage resources of cloud systems. DNNs are powerful tools for pattern recognition. However, they present several research challenges including energy consumption, training and execution time, data security, and cloud interoperability. This section discusses some of these challenges and suggests future research directions for those who are interested in performing research on both DNNs and cloud computing. Fig. 7 summarizes these challenges and directions graphically.

Table 10
Deploying DLs in mobile-cloud assisted applications.

Applications	Network model	Loss function	Evaluation metrics	Datasets	Performance
Video summarization	DNN [181]	Not reported	F measure	Selected videos from Gastrolab and WCE Video Atlas datasets	F measure: 0.82
Video summarization	DNN [190]	Not reported	Precision, recall, and F measure	Selected videos from Gastrolab and WCE Video Atlas datasets	Precision: 0.85, recall: 0.84, F measure: 0.85
Image-aware inferencing [55]	IF-CNN	Custom-loss	Processing latency and accuracy	ILSVRC2012	Accuracy: 92.21%, latency: 159 ms
DNN tuning [200]	DNNTune	Not reported	Speed and energy	ImageNet, PASCAL VOC, Penn Tree Bank (PTB), MNIST	1.66× speedup and 15× energy saving compared with mobile- and cloud-only approach
Intelligent video recording [56]	Faster R-CNN	Not reported	Detection rate	PASCAL VOC 2007	4 FPS
LiDAR data classification and reproduction [201]	Multi-faceted CNN (MFC)	Not reported	Accuracy and kappa score	KITTI and MLS	Improvement in accuracy: 1.7%, improvement in kappa: 2.2%
Efficient training and inferencing [202]	JointDNN	Not reported	Latency and energy consumption	Raw data (image, videos)	Reduction in latency and energy: 18–22×

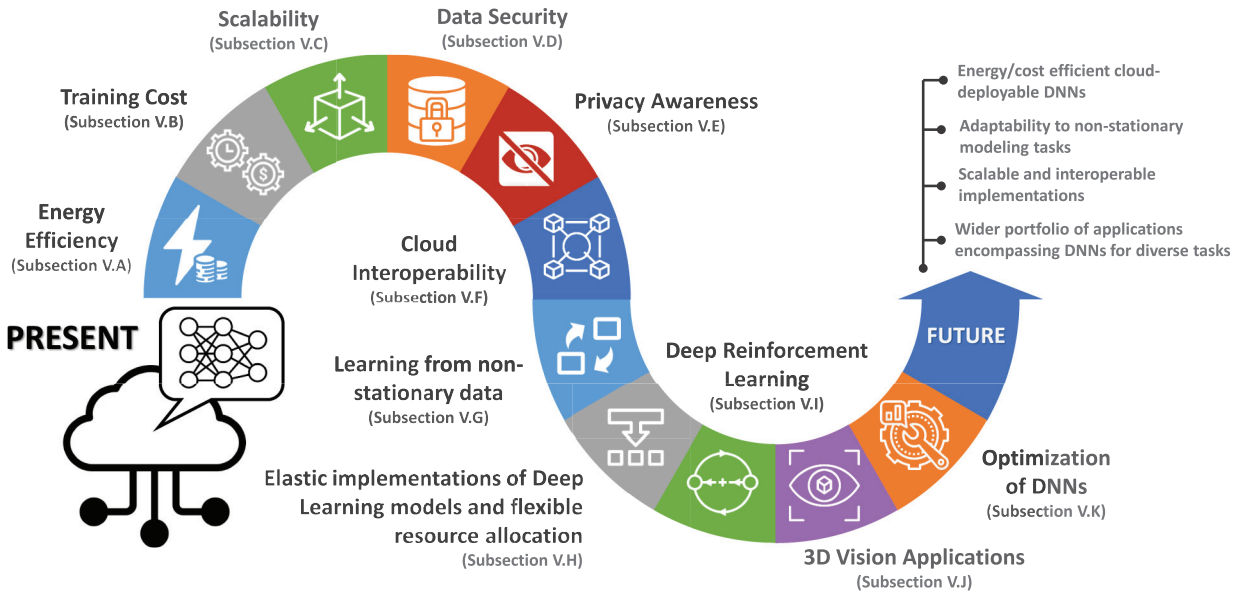

Fig. 7. Summary of challenges and research directions discussed in this section.

5.1. Energy efficiency

Developing a DNN requires a certain amount of training time. For example, a CNN was developed by OpenAI for NLP. It took six months to train this CNN using P100×8 parallel scaling [1]. The carbon footprint of developing this CNN is the same as that required during the lifetime of running five cars. Large amounts of energy are required by DNNs. Future research can focus on developing techniques to increase the efficiency of cloud data centers through green IT [203] by reducing carbon emissions due to energy consumption to maximize energy savings. Ideas need to be explored on the use of renewable energy sources, smart grids, and efficient energy data storage and computation for data centers. At the hardware-component level, various techniques can be developed to minimize the energy consumption of servers and maximize memory and storage in data centers. At the network level, techniques can be developed to increase the efficiency of data exchanges between nodes in data centers. In the long term, hardware technologies need to be investigated and developed to build green data centers.

This is also being actively investigated from multiple modeling perspectives, from weight sparsification to temporal neural processing (e.g. spiking neural networks), with the latter better suited for low-consumption neuromorphic hardware. However, cloud implementations of deep learning models have not incorporated such advances yet. In general, performance is affected and degraded when the design of the deep learning model embraces such approaches to reduce energy consumption. The race towards Green Machine Learning (also referred to as Green Artificial Intelligence [204,205]) should bring new visions to this issue, so that cloud systems leveraging deep learning-based pipelines can stop being a threat to sustainability.

5.2. Training cost

Although the cloud resource is large and effective in speeding up the training process of DNNs, a cost-effective way to reduce training cost is to perform the DNN training on a free cloud provider, such as Google Colab. However, these computational resources are limited and are constrained by the number of required running hours. Renting a cluster of virtual machines is extremely costly. For example, training a model such as $\text{Transformer}_{\text{big}}$ using P100×8 requires at least 289 USD; training an NLP model such as $\text{BERT}_{\text{base}}$ using V100×4 requires at least 3751 USD; training an NLP model

such as NAS using TPUv2×1 requires at least 44,055 USD [1]. Effective parallel programming makes full use of the computational resources of cloud cluster multi-nodes [206].

For example, Li et al. [207] proposed a population-based searching algorithm using an asynchronous parallel framework which consists of a controller and workers. The controller generates various architectures of DNNs; the workers train and evaluate DNNs. Workers require a significant amount of time to train DNNs; they return the network performance to the controller and the controller sends newly updated DNNs to perform another iteration of training until a DNN is generated with good performance.

This asynchronous parallel approach which generates a set of DNNs is more effective than the parallel process at computing a single DNN. The approach improves computational efficiency. Hence, the computational cost can be reduced. The development of a cost-effective parallel computing framework is envisaged as a research direction for integrating DNNs and cloud computing.

5.3. Scalability

In general, deep-learning models of realistic complexity require a relatively higher inference latency than other shallow learning approaches. This may not yield issues in latency-insensitive (i.e. delay-tolerant) applications, when the rate at which the model is queried is low enough not to saturate the computing resources of the cloud system in which it is deployed. However, this may not be the case in deep learning-powered services that are concurrently accessed by several users, or in applications that demand a fast, near-real-time response from the cloud model. Under these circumstances, one can replicate and launch on demand containerized implementations of such deep learning models. However, this workaround does not scale nicely in terms of resources. Therefore, implementing a deep-learning model in the cloud can be troublesome if it is done without bearing in mind the inference working regime at which the model will operate.

A promising direction is to perform neural architecture search (NAS [208,209]) driven by objectives tightly coupled with the hardware characteristics of the cloud system. Unfortunately, most NAS exercises considering complexity as one of the optimization goals, also consider deployment-agnostic measures of complexity, such as the number of trainable parameters. An interesting research direction would be to explore different methodologies to get such measures of performance closer to the specifics of the cloud system so that the neural architecture search yields optimized models capable of scaling up more efficiently than their off-the-shelf, non-optimized containerized counterparts.

5.4. Data security

In cloud computing, data security is a main concern in applying DNNs or cloud computing in many applications, such as healthcare systems and banking systems that involve private, personal, and confidential data. For example, patient data are used to train DNNs in healthcare systems [210–212]. Online patient data are transmitted to the cloud which is embedded with a DNN to perform data analysis. In addition, DNNs need to be adjusted online for dynamic environments because additional data transmissions are required. However, local storage and data transmissions do not fully satisfy the data security issues. To solve the data security issues, blockchain is used to keep data private and secure [213]. Blockchain is deployed in the cloud-based IoT architecture which is engaged in DNNs, providing access and storage control for private data to exchange between nodes in the cloud and the outside network. For example, blocks are created to store patient identifiers in the health blockchain systems. Each patient's data are also encrypted and authenticated, whereby a pointer is created. The blockchain contains patient data captured from wearable sensors and smartphones. These patient data are stored in a database which can be accessed by the DNN in the cloud for patient data analysis. Similar blockchain systems can be implemented in other applications such as banking systems which require high data privacy and security. The development of a blockchain system integrated with DNNs and cloud servers is another future research direction.

Further along these lines, different attacks have been studied to discover characteristics of the training data by querying deep learning models, exploiting the fact that such models tend to memorize much of the characteristics of the training data. Property inference, feature reconstruction and membership inference are some of the attack methodologies that can successfully leak private information from a trained deep learning model [214,215]. Cloud implementations of these models can further exacerbate the chances of being attacked, revealing private information that can be sensitive in some applications (e.g. medical diagnosis). Further research is required to reduce the exposure of deep learning models deployed on the cloud to these attacks.

5.5. Privacy awareness

Privacy preservation is a critical concern in deploying DNNs on cloud computing platforms. Differential privacy [216,217], homomorphic computing [218], and federated learning [219] are three promising approaches for addressing this challenge. Differential privacy aims to provide strong privacy guarantees by adding noise to the data before sharing it with a third party. This approach has been successfully applied to various DNN architectures, including CNNs and LSTMs [220–223]; however, it also comes with a trade-off between privacy and accuracy. Homomorphic computing enables computations on encrypted data without decrypting it, thereby preserving the privacy of the data. However, homomorphic computing is currently limited in terms of scalability and performance, which precludes its applicability to large-scale DNNs [224,218,225]. Federated learning is a distributed machine learning approach that enables DNN training on local data without sharing it with a central server. This approach has gained significant attention because of its ability to address privacy concerns while leveraging the collective intelligence of data across multiple devices [226]. However, federated learning also poses new challenges such as hierarchical federated learning [227], explainability in federated learning [228], new methodologies for model aggregation in federated environments, and proper scheduling of the local updates and deliveries upstream depending on the availability of local resources. Although the aforementioned approaches hold promise for addressing privacy concerns in cloud-based DNNs, they also bring new challenges that need to be addressed in future research.

5.6. Cloud interoperability

Deep-learning techniques have become the preferred effective method for processing and analyzing big data [229]. To ensure scalability, DNNs can be partitioned across large-scale clusters of machines to distribute training and inference [230]. The training dataset can be massive with various big data features which require efficient multi-cloud computing resources to accommodate the propagation, storage, and analysis of real-time and heterogeneous datasets [231]. Multi-cloud computing has made great strides in fulfilling the needs of big data. However, immaturity of various cloud services has resulted in integration issues, technical incompatibilities, and operational complexity [232,233]. Furthermore, the massive amount of generated data as well as the need for real-time data processing have conflated this issue. In fact, 75% of 572 cybersecurity professionals surveyed by Fortinet

reported that the skills and knowledge to integrate multi-cloud solutions are inadequate [234]. These challenges can be mitigated by ensuring cloud interoperability [235,236].

Cloud interoperability refers to the process of seamlessly deploying, migrating, and configuring application workloads across various multi-cloud environments [237], whereby organizations can easily access, manipulate, exchange, or share information and use functionalities across various cloud service providers [238]. Interoperability depicts a perfect solution because it enables multi-cloud multi-vendor cloud environments to share and access their data and interact with each other. To ensure cloud interoperability, semantic technologies such as knowledge graphs can be incorporated to enable semantic representation of different hardware and software resources as well as server configuration across heterogeneous multi-cloud environments. For example, the underlying structure of knowledge graphs can be used to provide a formal representation of different multi-cloud services which can pave the way to data integration, unification, and information sharing [239]. Despite a few attempts to address the interoperability in multi-cloud environments [240–242], this remains an open research area where more approaches are needed to construct adequate and coherent knowledge graphs.

5.7. Learning from non-stationary data: retraining efficiency and adaptation

When data are produced continuously over time, the knowledge captured by the model may become obsolete because of the non-stationary, time-varying nature of the modeling task [243,244]. This is often the case in scenarios where the source that produces the information is subject to exogenous phenomena that imprint the changes on the data, such as temperature sensors monitoring a machine in industrial prognosis. When this task variability holds, a straightforward strategy is to adapt the model by retraining it with the most recent data at the cloud end. This approach, however, echoes the scalability issues mentioned previously, as the training times required by deep learning models can exhaust the responsiveness of the cloud system. To avoid this, training can be done incrementally, that is, by updating the trainable parameters only using the most recent data instances received by the model. However, the so-called catastrophic forgetting issue of deep neural networks refers to incremental updates becoming so dominant in the knowledge captured by the model, that the model forgets about relevant patterns observed in the past, overfitting and considering primarily the recent information. More elaborated continual learning approaches [245] for cloud-deployed models should be in the research agenda in years to come.

5.8. Elastic implementations of deep learning models and flexible resource allocation

The layered structure of deep neural networks should be a suitable framework for developing flexible models that can be redistributed on-the-fly over different parts of the communication setup, from the edge to the cloud. However, in our experience the literature often approaches this allocation statically, without considering the availability of resources in the different devices involved in the communication between the user(s) and the cloud system. In practice, available resources vary, either due to intermittent communications links, battery depletion of mobile devices, or the need for addressing other computing tasks that have priority over the neural processing task allocated to the device. To accommodate this variability, elastic deep Learning implementations (in terms of their capability to be distributed over different remote subsystems) and flexible/intelligent resource allocation strategies should be further investigated and put to practice. Reinforcement learning is a promising path to consider for the intelligent management of flexible models [246].

5.9. Deep reinforcement learning

Deep neural networks also prevail in the reinforcement learning of the behavioral policy of an agent which interacts with the environment toward the fulfillment of a given goal [247]. DNN-based reinforcement learning provides good function approximators for high-dimensional data, when the state-space or actions are too large to be modeled using traditional methods based on look-up tables, such as Q learning [248,249]. Hence, deep reinforcement learning (DRL) is effective at mapping states or state–action pairs to values. For example, neural architectures comprising convolutional layers can be used for reinforcement learning based on visual data such as autonomous driving [250] and robotic navigation [251]. In DRL, the environment is assumed to deliver a scalar measure of reward in response to every new action of the agent. The computation of this reward is environment or task specific, whereas the specific procedure by which the value for every state–action pair is learned and exploited in the policy. This distinguishes between different architectures existing in the DRL panorama such as deep Q learning and actor–critic networks [252,253].

When implementing DRL in a cloud platform, delivering the observations of the agent to the cloud and receiving the action therefrom can be satisfied if, (1) the timing constraints of the reinforcement learning task are able to accommodate the bandwidth requirements, communication, and processing latencies upstream; (2) the architecture of the DRL is able to remain private or is trained with the observations of different agents deployed over several albeit interrelated environments. Satisfying the two conditions is challenging because DRL requires buffering information, such as the replaying experience, which stabilizes the convergence of the agent's learning process. Such advances have important implications in terms of the bandwidth and latency incurred by data communicating with the cloud. Consequently, delegating the agent's learning process to a cloud must be decided by a manifold of factors, including preprocessing observations on the edge and adjusting the agent's learning schedule to alleviate bandwidth and timing requirements. Resolving these conditions can be a future research direction.

5.10. 3D vision applications

Driverless autonomous vehicles which require DNNs for object detection and navigation have not been fully implemented in cloud systems [254]. Implementation of driverless autonomous vehicles requires high-performance and high-power computations. Accurate and fast driving decisions are necessary. Higher computation power is required because DNNs are updated when vision data are newly captured, necessitating the integration of DNNs and cloud computing. DNNs perform the decisions and cloud systems provide the computational resources and data storage. In addition, high-speed data transmissions are required because data for driverless autonomous vehicles are 3-D and data exchanges between cloud and cars are large. Similarly, high-power computations and high-speed data exchanges are necessary for 3-D video games and small-sized object detection [255]. The emerging 6G communication technology together with cloud-based DNNs have the potential to provide a practical solution for implementing 3-D vision analysis in real time.

5.11. Optimization of DNNs

Backpropagation methods are used to optimize DNNs. However, only local optima of DNNs are found because DNNs with large

numbers of parameters and complex searching landscapes are highly nonlinear. To overcome this problem, heuristic methods such as evolutionary algorithms (EAs) have been employed [256,257]. EAs have been used to optimize hyperparameters [256,258–261], network weights [256,257], network configurations [257], and network architectures [262–265]. However, the computational time of heuristic algorithms is impractically long [264,266]. The unreasonably long computational time can be explained by the evolutionary mechanism whereby a large population of individuals evolve through many generations. Each individual represents a DNN structure. In each generation, the fitness of each individual is evaluated. The DNN parameters are optimized by backpropagation, which requires very long computational times [267]. Many articles do not report the computational time of DNN training in EAs [268–273]. It is impractical to develop DNNs using EAs locally on standalone machines or remotely on a cloud machine [267]. Although early research has been suggested on integrating metaheuristic algorithms with cloud platforms to speed up the training [274], recent evolutionary deep learning networks are associated with fewer than thousand weights [275]. A practical cloud platform has not been developed for metaheuristic algorithm implementation or gradient-independent heuristic search algorithms. This development can be a future research direction.

6. Conclusion

DNNs have been implemented and deployed in many real-life application domains. However, DNNs are computationally demanding and consist of thousands or even millions of parameters which enable the learning during training; they also require millions of FLOPs to execute in prediction mode. It is not feasible to deploy DNNs on single stand-alone computers or run them on mobile devices for many applications which involve the storage and analysis of big data. There is currently noticeably growing interest in deploying DNNs in cloud computing systems. This review article first presented the motivation for DNN deployment and training in cloud systems, thereafter discussing the computational complexity of the commonly used DNNs including MLP, CNN, RNN, and GNN, which involve large numbers of parameters and FLOPs. The article also presented an extensive overview of public or volunteer cloud computing platforms that have developed and deployed DNNs. This information can be used by researchers or software engineers to select the most appropriate cloud computing platform for their applications involving DNNs. The article also provided an overview of some application areas such as NLP, BI, cybersecurity, anomaly detection, and travel, which have recently benefited from DNN deployments in cloud computing. This overview first illustrated the advantages and the effectiveness of deploying DNNs in cloud systems, and subsequently outlined the main challenges encountered when deploying DNNs in the cloud. Future directions were also proposed to enhance current deployments using cloud computing systems with DNNs. It is expected that this review article will serve as a useful guide for researchers and developers who are interested in deploying DNNs on cloud computing platforms.

CRediT authorship contribution statement

Kit Yan Chan: Conceptualization, Project administration, Supervision, Writing - original draft, Writing - review & editing. **Bilal Abu-Salih:** Resources, Writing - original draft, Writing - review & editing. **Raneem Qaddoura:** Writing - original draft, Writing - review & editing. **Ala' M. Al-Zoubi:** Writing - original draft, Writing - review & editing. **Vasile Palade:** Project administration, Writing - review & editing. **Duc-Son Pham:** Writing - original draft, Writing - review & editing. **Javier Del Ser:** Project administration, Writing - original draft, Writing - review & editing. **Khan Muhammad:** Project administration, Writing - original draft, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

J. Del Ser acknowledges funding support from the Basque Government through the EGIA project (KK-2022/00119) and the Consolidated Research Group MATHMODE (IT1456-22).

References

[1] E. Strubell, A. Ganesh, and A. McCallum, Energy and policy considerations for deep learning in nlp, in: Annual Meeting of the Association for Computational Linguistics, 2019.
[2] Q. Yan, F.R. Yu, Q.X. Gong, J.Q. Li, Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: a survey, some research issues, and challenges, IEEE Communications Surveys and Tutorials 18 (1) (2016) 602–622.
[3] S.L. Nita and M.I. Mihailescu, On artificial neural network used in cloud computing security – a survey, in Proceedings of the International Conference of Electronics, Computers and Artificial Intelligence, 2018.
[4] P.J. Sun, Privacy protection and data security in cloud computing: a survey, challenges, and solutions, IEEE Access, vol. 7, pp. 147 420–147 452, 2019.
[5] K. Gai, J. Guo, L.H. Zhu, S. Yu, Blockchain meets cloud computing: a survey, IEEE Transactions on Neural Networks and Learning Systems 22 (3) (2022) 2009–2030.
[6] F. Xu, F.M. Liu, H. Jin, and A.V. Vasilakos, Managing performance overhead of virtual machines in cloud computing: A survey, state of the art, and future directions, Proceedings of the IEEE, vol. 102, no. 1, pp. 11–31, 2014.
[7] A. Pupykina and G. Agosta, Survey of memory management techniques for HPC and cloud computing, IEEE Access, vol. 7, pp. 167 351–167 373, 2019.
[8] B. Wang, C.C. Wang, W.W. Huang, Y. Song, X.Y. Qin, A survey and taxonomy on task offloading for edge-cloud computing, IEEE Access 8 (2020) 186 080–186 101.
[9] G. Zhou, W. Tian, R. Buyya, Deep reinforcement learning-based methods for resource scheduling in cloud computing: A review and future directions, Journal of Cloud Computing 11 (2022), paper number 3.
[10] Y. Feng and F. Liu, Resource management in cloud computing using deep reinforcement learning: A survey, in Proceedings of the 10th Chinese Society of Aeronautics and Astronautics Youth Forum, 2023, pp. 635–643.
[11] A.R. Khan, M. Othman, S.A. Madani, S.U. Khan, A survey of mobile cloud computing application models, IEEE Communications Surveys and Tutorials 16 (1) (2014) 393–413.
[12] S. Bera, S. Misra, J.J.P.C. Rodrigues, Cloud computing applications for smart grid: a survey, IEEE Transactions on Parallel and Distributed Systems 26 (5) (2015) 1477–1494.
[13] K. Cao, S.Y. Hu, Y. Shi, A.W. Colombo, S. Karnouskos, X. Li, A survey on edge and edge-cloud computing assisted cyber-physical systems, IEEE Transactions on Industrial Informatics 17 (11) (2021) 7806–7819.
[14] D. Soni, N. Kumar, Machine learning techniques in emerging cloud computing integrated paradigms: A survey and taxonomy, Journal of Network and Computer Applications 205 (2022).
[15] T. Khana, W.H. Tiana, R. Buyya, Machine learning (ml)-centric resource management in cloud computing: A review and future directions, Journal of Network and Computer Applications 204 (2022).
[16] A. Saiyeda, M.A. Mir, Cloud computing for deep learning analytics:a survey of current trends and challenges, International Journal of Advanced Research in Computer Science 8 (2) (2017).
[17] P.S. Priya, P. Malik, A. Mehbodniya, V. Chaudhary, A. Sharma, and S. Ray, The relationship between cloud computing and deep learning towards organizational commitment, in: Proceedings of the 2nd International Conference on Innovative Practices in Technology and Management, 2022.
[18] F. Benedetto and A. Tedeschi, Big data sentiment analysis for brand monitoring in social media streams by cloud computing, in: Sentiment Analysis and Ontology Engineering. Springer, 2016, pp. 341–377.
[19] S. Mohan, S. Mullapudi, S. Sammeta, P. Vijayvergia, and D.C. Anastasiu, Stock price prediction using news sentiment analysis, in: 2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService). IEEE, 2019, pp. 205–208.

[20] S. Prasomphan, Improvement of chatbot in trading system for smes by using deep neural network, in: 2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA). IEEE, 2019, pp. 517–522.

[21] D. Scheinert, A. Acker, L. Thamsen, M.K. Geldenhuys, and O. Kao, Learning dependencies in distributed cloud applications to identify and localize anomalies, in: Proceedings of IEEE/ACM International Workshop on Cloud Intelligence, 2021, pp. 7–12.

[22] M.A. Elsayed, M. Zulkernine, Predictdeep: Security analytics as a service for anomaly detection and prediction, IEEE Access 8 (2020) 45 184–45 197.

[23] F. Jauro, H. Chiroma, A.Y. Gital, M. Almutairi, S.M. Abdulhamid, J.H. Abawajy, Deep learning architectures in emerging cloud computing architectures: Recent development, challenges and next research trend, Applied Soft Computing Journal 96 (2020), paper number 106582.

[24] J.-F. Chen, Q.H. Do, H.-N. Hsieh, Training artificial neural networks by a hybrid pso-cs algorithm, Algorithms 8 (2) (2015) 292–308.

[25] A.A. Heidari, H. Faris, S. Mirjalili, I. Aljarah, M. Mafarja, Ant lion optimizer: theory, literature review, and application in multi-layer perceptron neural networks, Nature-Inspired Optimizers (2020) 23–46.

[26] Y.-S. Park, S. Lek, Artificial neural networks: multilayer perceptron for ecological modeling, Developments in environmental modelling, vol. 28, ch. 7, Elsevier, 2016, pp. 123–140.

[27] R. Qaddoura, M. Al-Zoubi, H. Faris, I. Almomani, et al., A multi-layer classification approach for intrusion detection in iot networks based on deep learning, Sensors 21 (9) (2021) 2987.

[28] S. Zahara, S. Sugianto, Prediksi indeks harga konsumen komoditas makanan berbasis cloud computing menggunakan multilayer perceptron, JOINTECS (Journal of Information Technology and Computer Science) 6 (1) (2021) 21–28.

[29] G.S.B. Jahangeer, T.D. Rajkumar, Cloud storage based diagnosis of breast cancer using novel transfer learning with multi-layer perceptron, International Journal of System Assurance Engineering and Management (2022) 1–13.

[30] D. Mandic, J. Chambers, Recurrent neural networks for prediction: learning algorithms, architectures and stability, Wiley, 2001.

[31] Z.C. Lipton, J. Berkowitz, and C. Elkan, A critical review of recurrent neural networks for sequence learning, arXiv preprint arXiv:1506.00019, 2015.

[32] R. Pascanu, T. Mikolov, and Y. Bengio, On the difficulty of training recurrent neural networks, in: Proceedings of International Conference on Machine Learning, 2013, pp. 1310–1318.

[33] P. Yazdanian, S. Sharifian, E2lg: a multiscale ensemble of lstm/gan deep learning architecture for multistep-ahead cloud workload prediction, The Journal of Supercomputing 77 (2021) 11 052–11 082.

[34] Y.S. Patel, J. Bedi, MAG-D: A multivariate attention network based approach for cloud workload forecasting, Future Generation Computer Systems (2023).

[35] S. Ouhame, Y. Hadi, A. Ullah, An efficient forecasting approach for resource utilization in cloud data center using cnn-lstm model, Neural Computing and Applications 33 (2021) 10 043–10 055.

[36] N. Tran, T. Nguyen, B.M. Nguyen, G. Nguyen, A multivariate fuzzy time series resource forecast model for clouds using lstm and data correlation analysis, Procedia Computer Science 126 (2018) 636–645.

[37] H.L. Leka, Z. Fengli, A.T. Kenea, A.T. Tegene, P. Atandoh, and N.W. Hundera, A hybrid cnn-lstm model for virtual machine workload forecasting in cloud data center, in: 2021 18th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP). IEEE, 2021, pp. 474–478.

[38] Z. Ding, J. Wang, Y. Cheng, and C. He, Alice: A lstm neural network based short-term power load forecasting approach in distributed cloud-edge environment, in: Journal of Physics: Conference Series, vol. 1624, no. 5. IOP Publishing, 2020, p. 052017.

[39] T. Hussain, K. Muhammad, W. Ding, J. Lloret, S.W. Baik, V.H.C. de Albuquerque, A comprehensive survey of multi-view video summarization, Pattern Recognition 109 (2021).

[40] K. Muhammad, T. Hussain, J. Del Ser, V. Palade, V.H.C. De Albuquerque, Deepres: A deep learning-based video summarization strategy for resource-constrained industrial surveillance scenarios, IEEE Transactions on Industrial Informatics 16 (9) (2019) 5938–5947.

[41] K. Fukushima, S. Miyake, T. Ito, Neocognitron: A neural network model for a mechanism of visual pattern recognition, IEEE Transactions on Systems, Man and Cybernetics 13 (5) (1983) 826–834.

[42] A. Krizhevsky, I. Sutskever, and G. Hinton, Imagenet classification with deep convolutional neural network, in: Proceedings of the Conference of Neural Information Processing Systems, 2012, pp. 1106–1114.

[43] X. Zhang, X. Zhou, M. Lin, and J. Sun, Shufflenet: An extremely efficient convolutional neural network for mobile devices, in: Proceedings of The IEEE Computer Vision and Pattern Recognition Conference, 2018.

[44] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, Going deeper with convolutions, in: Proceedings of The IEEE Computer Vision and Pattern Recognition Conference, 2015.

[45] G. Huang, Z. Liu, L. van der Maaten, and K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of The IEEE Computer Vision and Pattern Recognition Conference, 2018.

[46] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, Microsoft Research, Technical Report (2015) [Online]. Available:https://arxiv.org/pdf/1512.03385.pdf.

[47] A. Krizhevsky, I. Sutskever, and G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proceedings of Conference on Neural Information Processing Systems. IEEE, 2012.

[48] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Proceedings of the International Conference on Learning Representations, 2015.

[49] A.D. Torres, H. Yan, A.H. Aboutalebi, A. Das, L. Duan, and P. Rad, Patient facial emotion recognition and sentiment analysis using secure cloud with hardware acceleration, in: Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications. Elsevier, 2018, pp. 61–89.

[50] A. Makkar, U. Ghosh, P.K. Sharma, Artificial intelligence and edge computing-enabled web spam detection for next generation iot applications, IEEE Sensors Journal 21 (22) (2021) 25 352–25 361.

[51] D. Selvapandian, R. Santhosh, Deep learning approach for intrusion detection in iot-multi cloud environment, Automated Software Engineering 28 (2) (2021) 1–17.

[52] D. Scheinert and A. Acker, Telesto: A graph neural network model for anomaly classification in cloud services, in: Proceedings of International Conference on Service-Oriented Computing. Springer, 2020, pp. 214–227.

[53] W. Ullah, A. Ullah, I.U. Haq, K. Muhammad, M. Sajjad, S.W. Baik, CNN features with bi-directional LSTM for real-time anomaly detection in surveillance networks, Multimedia Tools and Applications 80 (11) (2021) 16 979–16 995.

[54] H. Saito, T. Aoki, K. Aoyama, Y. Kato, A. Tsuboi, A. Yamada, M. Fujishiro, S. Oka, S. Ishihara, T. Matsuda, et al., Automatic detection and classification of protruding lesions in wireless capsule endoscopy images based on a deep convolutional neural network, Gastrointestinal endoscopy 92 (1) (2020) 144–151.

[55] G. Shu, W. Liu, X. Zheng, J. Li, If-cnn: Image-aware inference framework for cnn with the collaboration of mobile devices and cloud, IEEE Access 6 (2018) 68 621–68 633.

[56] C.-H. Chen, C.-R. Lee, W.C.-H. Lu, Smart in-car camera system using mobile cloud computing framework for deep learning, Vehicular Communications 10 (2017) 84–90.

[57] B. Abu-Salih, M. Al-Tawil, I. Aljarah, H. Faris, P. Wongthongtham, K.Y. Chan, A. Beheshti, Relational learning analysis of social politics using knowledge graph embedding, Data Mining and Knowledge Discovery (2021) 1–40.

[58] P. Cui, X. Wang, J. Pei, W. Zhu, A survey on network embedding, IEEE Transactions on Knowledge and Data Engineering 31 (5) (2018) 833–852.

[59] W.L. Hamilton, R. Ying, and J. Leskovec, Representation learning on graphs: Methods and applications, in: Proceedings of IEEE Data Engineering Bulletin, 2017.

[60] M.M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, P. Vandergheynst, Geometric deep learning: going beyond euclidean data, IEEE Signal Processing Magazine 34 (4) (2017) 18–42.

[61] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, M. Sun, Graph neural networks: A review of methods and applications, AI Open 1 (2020) 57–81.

[62] F. Frasca, E. Rossi, D. Eynard, B. Chamberlain, M. Bronstein, and F. Monti, Sign: Scalable inception graph neural networks, in: International Conference on Machine Learning, 2020.

[63] W. Hu, M. Shuaibi, A. Das, S. Goyal, A. Sriram, J. Leskovec, D. Parikh, and C.L. Zitnick, Forcenet: A graph neural network for large-scale quantum calculations, in: Proceedings of The Conference on Neural Information Processing Systems, 2021.

[64] J. Klicpera, S. Giri, J.T. Margraf, and S. Günnemann, Fast and uncertainty-aware directional message passing for non-equilibrium molecules, in: Proceedings of The Conference on Neural Information Processing Systems, 2020.

[65] M. Shuaibi, A. Kolluru, A. Das, A. Grover, A. Sriram, Z. Ulissi, and C.L. Zitnick, Rotation invariant graph neural networks using spin convolutions, in: Proceedings of The Conference on Neural Information Processing Systems, 2021.

[66] J. Klicpera, F. Becker, and S. Günnemann, Gemnet: Universal directional graph neural networks for molecules, in: Proceedings of The Conference on Neural Information Processing Systems, 2021.

[67] 2020.[Online]. Available: https://blog.twitter.com/engineering/en_us/topics/insights/2020/graph-ml-at-twitter.

[68] A. Sriram, A. Das, B.M. Wood, S. Goyal, and C.L. Zitnick, Towards training billion parameter graph neural networks for atomic simulations, in: Proceedings of The International Conference on Learning Representations, 2022.

[69] J. Gasteiger, F. Becker, and S. Gunnemann, Gemnet: Universal directional graph neural networks for molecules, in: Proceedings of The Conference on Neural Information Processing Systems, 2022.

[70] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, S.Y. Philip, A comprehensive survey on graph neural networks, IEEE Transactions on Neural Networks and Learning Systems 32 (1) (2020) 4–24.

[71] R. Ying, R. He, K. Chen, P. Eksombatchai, W.L. Hamilton, and J. Leskovec, Graph convolutional neural networks for web-scale recommender systems, in: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, 2018, pp. 974–983.

[72] L. Ge, Y. Jia, Q. Li, X. Ye, Dynamic multi-graph convolution recurrent neural network for traffic speed prediction, Journal of Intelligent & Fuzzy Systems no. Preprint (2023) 1–14.

[73] R. Yumlembam, B. Issac, S.M. Jacob, and L. Yang, Iot-based android malware detection using graph neural network with adversarial defense, IEEE Internet of Things Journal, 2022.
[74] Y. Li, S. Xie, Z. Wan, H. Lv, H. Song, Z. Lv, Graph-powered learning methods in the internet of things: A survey, Machine Learning with Applications 11 (2023).
[75] S. Wang, Y. Zheng, X. Jia, Secgnn: Privacy-preserving graph neural network training and inference as a cloud service, IEEE Transactions on Services Computing (2023).
[76] R. v. d. Berg, T.N. Kipf, and M. Welling, Graph convolutional matrix completion, arXiv preprint arXiv:1706.02263, 2017.
[77] S. Pratiher, S. Chattoraj, D. Nawn, M. Pal, R.R. Paul, H. Konik, and J. Chatterjee, A multi-scale context aggregation enriched mlp-mixer model for oral cancer screening from oral sub-epithelial connective tissues, in: 2022 30th European Signal Processing Conference (EUSIPCO). IEEE, 2022, pp. 1323–1327.
[78] Y. Li, K. Li, X. Liu, Y. Wang, L. Zhang, Lithium-ion battery capacity estimation–a pruned convolutional neural network approach assisted with transfer learning, Applied Energy 285 (2021).
[79] P.N. Suganthan, R. Katuwal, On the origins of randomization-based feedforward neural networks, Applied Soft Computing 105 (2021).
[80] T. Le-Cong, H.J. Kang, T.G. Nguyen, S.A. Haryono, D. Lo, X.-B.D. Le, and Q.T. Huynh, Autopruner: transformer-based call graph pruning, in: Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2022, pp. 520–532.
[81] J. Poyatos, D. Molina, A.D. Martinez, J. Del Ser, F. Herrera, Evoprunedeeptl: An evolutionary pruning model for transfer learning based deep neural networks, Neural Networks 158 (2023) 59–82.
[82] D.T. Hoang, C. Lee, D. Niyato, P. Wang, A survey of mobile cloud computing: architecture, applications, and approaches, Wireless Communications and Mobile Computing 13 (2013) 1587–1611.
[83] Q. Zhang, L. Cheng, R. Boutaba, Cloud computing: State-of-the-art and research challenges, Journal of Internet Services and Applications 1 (1) (2010) 7–18.
[84] T. Dillon, C. Wu, and E. Chang, Cloud computing: Issues and challenges, in: Proceedings of the IEEE International Conference on Advanced Information Networking and Applications, 2010.
[85] J. Novet. (2021) How amazon's cloud business generates billions in profit. [Online]. Available: https://www.cnbc.com/2021/09/05/how-amazon-web-services-makes-money-estimated-margins-by-service.html.
[86] What is distributed cloud? 2022. [Online]. Available: https://www.windriver.com/solutions/learning/distributed-cloud.
[87] A. Smola and S. Narayanamurthy, An architecture for parallel topic models, Proceedings of the VLDB Endowment, vol. 3, no. 1–2, pp. 703–710, 2010.
[88] M. Li, D.G. Andersen, J.W. Park, A.J. Smola, A. Ahmed, V. Josifovski, J. Long, E.J. Shekita, and B.-Y. Su, Scaling distributed machine learning with the parameter server, in: Proceedings of Symposium on Operating Systems Design and Implementation, 2014, pp. 583–598.
[89] A. Harlap, A. Tumanov, A. Chung, G.R. Ganger, and P.B. Gibbons, Proteus: agile ML elasticity through tiered reliability in dynamic resource markets, in: Proceedings of the Twelfth European Conference on Computer Systems, 2017, pp. 589–604.
[90] A. Qiao, A. Aghayev, W. Yu, H. Chen, Q. Ho, G.A. Gibson, and E.P. Xing, Litz: Elastic framework for high-performance distributed machine learning, in: Proceedings of Annual Technical Conference, 2018, pp. 631–644.
[91] Y. Peng, Y. Bao, Y. Chen, C. Wu, C. Meng, W. Lin, Dl2: A deep learning-driven scheduler for deep learning clusters, IEEE Transactions on Parallel and Distributed Systems 32 (8) (2021) 1947–1960.
[92] Y. Chen, Y. Peng, Y. Bao, C. Wu, Y. Zhu, and C. Guo, Elastic parameter server load distribution in deep learning clusters, in: Proceedings of the 11th ACM Symposium on Cloud Computing, 2020, pp. 507–521.
[93] S. Wang, A. Pi, X. Zhou, Elastic parameter server: Accelerating ML training with scalable resource scheduling, IEEE Transactions on Parallel and Distributed Systems 33 (5) (2021) 1128–1143.
[94] L. Hu, J. Zhu, Z. Zhou, R. Cheng, X. Bai, and Y. Zhang, An optimal resource allocator of elastic training for deep learning jobs on cloud, arXiv preprint arXiv:2109.03389, 2021.
[95] T. Menouer, KCSS: kubernetes container scheduling strategy, The Journal of Supercomputing 77 (5) (2021) 4267–4293.
[96] N. Nakata, J.P. Chang, J.F. Lawrence, P. Boue, Body wave extraction and tomography at long beach california with ambient noise interferometry, Journal of Geophysical Research: Solid Earth 120 (2) (2015) 1159–1173.
[97] D.A. Philips, C.P. Santillan, L.M. Wang, R.W. King, W.M. Szeliga, T. Melbourne, M. Floyd, T.A. Herring, Plate boundary observatory and related networks: GPS data analysis methods and geodetic products, Reviews of Geophysics 54 (2016) 759–808.
[98] T. Cowles, J. Delaney, J. Orcutt, R. Weller, The ocean observatories initiative: sustained ocean observing across a range of spatial scales, Marine Technology Society Journal 44 (6) (2010) 54–64.
[99] I. Foster, D.B. Gannon, Cloud Computing For Science And Engineering, The MIT Press, 2017.
[100] T. Akidau, R. Bradshaw, C. Chambers, S. Chernyak, R.J. Fernandez-Moctezuma, R. Lax, S. McVeety, D. Mills, F. Perry, E. Schmidt, and S. Whittle, The dataflow model: a practical approach to balancing correctness, latency, and cost in massive scale, unbounded, out-of-order data processing, Proceedings of the VLDB ENdowment, vol. 8, no. 12, pp. 1792–1803, 2015.
[101] W. Rang, D.L. Yang, D.Z. Cheng, Y. Wang, Data life aware model updating strategy for stream-based online deep learning, IEEE Transactions on Parallel and Distributed Systems 32 (10) (2021) 2571–2581.
[102] A. Ashfahani, Autonomous deep learning: Incremental learning of deep neural networks for evolving data streams, in: Proceedings of the International Conference on Data Mining Workshops, 2019, pp. 83–90.
[103] Y.L. Li, M. Zhang, W. Wang, Online real-time analysis of data streams based on an incremental high-order deep learning model, IEEE Access 6 (2018) 77 615–77 623.
[104] M. Pratama, W. Pedrycz, G.I. Webb, Online real-time analysis of data streams based on an incremental high-order deep learning model, IEEE Transactions on Fuzzy Systems 28 (7) (2020) 1315–1328.
[105] D. Nguyen, R. Vadaine, G. Hajduch, R. Garello, and R. Fablet, A multi-task deep learning architecture for maritime surveillance using ais data streams, in: Proceedings of IEEE 5th International Conference on Data Science and Advanced Analytics, 2018, pp. 331–340.
[106] S.S. Zhang, J.W. Liu, and X. Zuo, Adaptive online incremental learning for evolving data streams, 2022. [Online]. Available: https://arxiv.org/abs/1805.04754.
[107] D.W. Otter, J.R. Medina, J.K. Kalita, A survey of the usages of deep learning for natural language processing, IEEE Transactions on Neural Networks and Learning Systems 32 (2) (2021) 604–624.
[108] W. Medhat, A. Hassan, H. Korashy, Sentiment analysis algorithms and applications: A survey, Ain Shams Engineering Journal 5 (4) (2014) 1093–1113.
[109] R. Feldman, Techniques and applications for sentiment analysis, Communications of the ACM 56 (4) (2013) 82–89.
[110] R. Obiedat, L. Al-Qaisi, R. Qaddoura, O. Harfoushi, A. Al-Zoubi, An intelligent hybrid sentiment analyzer for personal protective medical equipments based on word embedding technique: The COVID-19 era, Symmetry 13 (12) (2021) 2287.
[111] D.A. Al-Qudah, A.-Z. Ala'M, P.A. Castillo-Valdivieso, H. Faris, Sentiment analysis for e-payment service providers using evolutionary extreme gradient boosting, IEEE Access 8 (2020) 189 930–189 944.
[112] R. Obiedat, O. Harfoushi, R. Qaddoura, L. Al-Qaisi, A. Al-Zoubi, An evolutionary-based sentiment analysis approach for enhancing government decisions during COVID-19 pandemic: The case of jordan, Applied Sciences 11 (19) (2021) 9080.
[113] R.O. Sinnott and S. Cui, Benchmarking sentiment analysis approaches on the cloud, in: Proceedings of 2016 IEEE 22nd International Conference on Parallel and Distributed Systems, 2016, pp. 695–704.
[114] M. Ghorbani, M. Bahaghighat, Q. Xin, F. Özen, Convlstmconv network: a deep learning approach for sentiment analysis in cloud computing, Journal of Cloud Computing 9 (1) (2020) 1–12.
[115] M.R. Raza, W. Hussain, and J.M. Merigó, Long short-term memory-based sentiment classification of cloud dataset, in: Proceedings of IEEE Innovations in Intelligent Systems and Applications Conference, 2021, pp. 1–6.
[116] G. Preethi, P.V. Krishna, M.S. Obaidat, V. Saritha, and S. Yenduri, Application of deep learning to sentiment analysis for recommender system on cloud, in: Proceedings of International conference on computer, information and telecommunication systems. IEEE, 2017, pp. 93–97.
[117] M.A. Khan, S. Saqib, T. Alyas, A.U. Rehman, Y. Saeed, A. Zeb, M. Zareei, E.M. Mohamed, Effective demand forecasting model using business intelligence empowered with machine learning, IEEE Access 8 (2020) 116 013–116 023.
[118] B.M. Balachandran, S. Prasad, Challenges and benefits of deploying big data analytics in the cloud for business intelligence, Procedia Computer Science 112 (2017) 1112–1122.
[119] C. Moreno, R.A.C. González, E.H. Viedma, Data and artificial intelligence strategy: A conceptual enterprise big data cloud architecture to enable market-oriented organisations, International Journal of Interactive Multimedia and Artificial Intelligence 5 (6) (2019) 7–14.
[120] C. Juarez and H. Afli, Online news analysis on cloud computing platform for market prediction. in Collaborative European Research Conference, 2020, pp. 125–140.
[121] P. Dixit and S. Silakari, "Deep learning algorithms for cybersecurity applications: A technological and status review," Computer Science Review, vol. 39, paper number 100317, 2021
[122] M.M. Alani, Big data in cybersecurity: a survey of applications and future trends, Journal of Reliable Intelligent Environments (2021) 1–30.
[123] P. Podder, S. Bharati, M. Mondal, P.K. Paul, and U. Kose, Artificial neural network for cybersecurity: a comprehensive review, arXiv preprint arXiv:2107.01185, 2021.
[124] L. Gupta, T. Salman, A. Ghubaish, D. Unal, A.K. Al-Ali, R. Jain, Cybersecurity of multi-cloud healthcare systems: A hierarchical deep learning approach, Applied Soft Computing 118 (2022), paper number 108439.
[125] Y. Chai, Y. Zhou, W. Li, Y. Jiang, An explainable multi-modal hierarchical attention model for developing phishing threat intelligence, IEEE Transactions on Dependable and Secure Computing 19 (2) (2022) 790–803.
[126] B. Abu-Salih, D.A. Qudah, M. Al-Hassan, S.M. Ghafari, T. Issa, I. Aljarah, A. Beheshti, and S. Alqahtan, An intelligent system for multi-topic social spam detection in microblogging, arXiv preprint arXiv:2201.05203, 2022.
[127] F.J. Abdullayeva, Advanced persistent threat attack detection method in cloud computing based on autoencoder and softmax regression algorithm, Array 10 (2021), paper number 100067.
[128] A. Vadariya and N.K. Jadav, A survey on phishing URL detection using artificial intelligence, in: Proceedings of International Conference on Recent

Trends in Machine Learning, IoT, Smart Cities and Applications, 2021, pp. 9–20.
[129] L. Tang, Q.H. Mahmoud, A survey of machine learning-based solutions for phishing website detection, Machine Learning and Knowledge Extraction 3 (3) (2021) 672–694.
[130] G. Li, P. Sharma, L. Pan, S. Rajasegarar, C. Karmakar, N. Patterson, Deep learning algorithms for cyber security applications: A survey, Journal of Computer Security 29 (5) (2021) 447–471.
[131] K. Sethi, R. Kumar, N. Prajapati, and P. Bera, Deep reinforcement learning based intrusion detection system for cloud infrastructure, in: Proceedings of the IEEE International Conference on Communication Systems and Networks, 2020, pp. 1–6.
[132] M.N. Hossain, J. Wang, O. Weisse, R. Sekar, D. Genkin, B. He, S.D. Stoller, G. Fang, F. Piessens, and E. Downing, Dependence-preserving data compaction for scalable forensic analysis, in: Proceedings of the 27th Security Symposium, 2018, pp. 1723–1740.
[133] M. Du, F. Li, G. Zheng, and V. Srikumar, Deeplog: Anomaly detection and diagnosis from system logs through deep learning, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 2017, pp. 1285–1298.
[134] T. Thilagam, R. Aruna, Intrusion detection for network based cloud computing by custom rc-nn and optimization, ICT Express 7 (4) (2021) 512–520.
[135] Y. Shen, E. Mariconti, P.A. Vervier, and G. Stringhini, Tiresias: Predicting security events through deep learning, in: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, 2018, pp. 592–605.
[136] M. Ramilli, Malware training sets: a machine learning dataset for everyone, Marco Ramilli Web Corner, 2016.
[137] N. Koroniotis, N. Moustafa, E. Sitnikova, B. Turnbull, Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset, Future Generation Computer Systems 100 (2019) 779–796.
[138] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, Hierarchical attention networks for document classification, in: Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies, 2016, pp. 1480–1489.
[139] N. Moustafa and J. Slay, Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set), in: 2015 military communications and information systems conference (MilCIS). IEEE, 2015, pp. 1–6.
[140] L. Bergman and Y. Hoshen, Classification based anomaly detection for general data, in: Proceedings of International Conference on Learning Representations, 2020.
[141] Y. Wu, H.-N. Dai, and H. Tang, Graph neural networks for anomaly detection in industrial internet of things, IEEE Internet of Things Journal, 2021.
[142] H. Wang, Z. Wu, H. Jiang, Y. Huang, J. Wang, S. Kopru, and T. Xie, Groot: An event-graph-based approach for root cause analysis in industrial settings, in: Proceedings of IEEE/ACM International Conference on Automated Software Engineering, 2021.
[143] A. Protogerou, S. Papadopoulos, A. Drosou, D. Tzovaras, I. Refanidis, A graph neural network method for distributed anomaly detection in iot, Evolving Systems 12 (1) (2021) 19–36.
[144] F. Fusco, B. Eck, R. Gormally, M. Purcell, and S. Tirupathi, Knowledge-and data-driven services for energy systems using graph neural networks, in: Proceedings of IEEE International Conference on Big Data. IEEE, 2020, pp. 1301–1308.
[145] M. Lee, S. Hosseinalipour, C.G. Brinton, G. Yu, and H. Dai, A fast graph neural network-based method for winner determination in multi-unit combinatorial auctions, IEEE Transactions on Cloud Computing, 2020.
[146] M. Gao, Y. Li, and J. Yu, Workload prediction of cloud workflow based on graph neural network, in: Proceedings of International Conference on Web Information Systems and Applications, 2021, pp. 169–189.
[147] A. Rafiq, T.A. Khan, M. Afaq, and W.-C. Song, Service function chaining and traffic steering in sdn using graph neural network, in: Proceedings of IEEE International Conference on Information and Communication Technology Convergence, 2020, pp. 500–505.
[148] Z. Yu, W. Liu, X. Liu, and G. Wang, Drag-jdec: A deep reinforcement learning and graph neural network-based job dispatching model in edge computing, in: Proceedings of IEEE/ACM 29th International Symposium on Quality of Service. IEEE, 2021, pp. 1–10.
[149] T. Liang, X. Sheng, L. Zhou, Y. Li, H. Gao, Y. Yin, L. Chen, Mobile app recommendation via heterogeneous graph neural network in edge computing, Applied Soft Computing vol. 103, paper number 107162 (2021).
[150] A. Said, S.-U. Hassan, S. Tuarob, R. Nawaz, M. Shabbir, DGSD: Distributed graph representation via graph statistical properties, Future Generation Computer Systems 119 (2021) 166–175.
[151] S. Chen, J. Dong, P. Ha, Y. Li, S. Labi, Graph neural network and reinforcement learning for multi-agent cooperative control of connected autonomous vehicles, Computer-Aided Civil and Infrastructure Engineering 36 (7) (2021) 838–857.
[152] Y. Gao, X. Gu, H. Zhang, H. Lin, and M. Yang, "Runtime performance prediction for deep learning models with graph neural network," Technical Report MSR-TR-2021-3. Microsoft, Tech. Rep., 2021.
[153] W. Ullah, A. Ullah, T. Hussain, Z.A. Khan, S.W. Baik, An efficient anomaly recognition framework using an attention residual lstm in surveillance videos, Sensors 21 (8) (2021) 2811.
[154] W. Ullah, A. Ullah, T. Hussain, K. Muhammad, A.A. Heidari, J. Del Ser, S.W. Baik, V.H.C. De Albuquerque, Artificial intelligence of things-assisted two-stream neural network for anomaly detection in surveillance big video data, Future Generation Computer Systems 129 (2022) 286–297.
[155] W. Xu, L. Huang, A. Fox, D. Patterson, and M. Jordan, "Online system problem detection by mining patterns of console logs," in: 2009 ninth IEEE international conference on data mining. IEEE, 2009, pp. 588–597.
[156] D. Duplyakin, R. Ricci, A. Maricq, G. Wong, J. Duerig, E. Eide, L. Stoller, M. Hibler, D. Johnson, K. Webb et al., The design and operation of {*CloudLab*}", in: 2019 USENIX annual technical conference (USENIX ATC 19), 2019, pp. 1–14.
[157] X. Zhou, C. Xu, B. Kimmons, Detecting tourism destinations using scalable geospatial analysis based on cloud computing platform, Computers, Environment and Urban Systems 54 (2015) 144–153.
[158] K. Vassakis, E. Petrakis, I. Kopanakis, J. Makridis, and G. Mastorakis, Location-based social network data for tourism destinations, in: Proceedings of Big data and innovation in tourism, travel, and hospitality, 2019, pp. 105–114.
[159] W. Wang, N. Kumar, J. Chen, Z. Gong, X. Kong, W. Wei, H. Gao, Realizing the potential of the internet of things for smart tourism with 5g and ai, IEEE Network 34 (6) (2020) 295–301.
[160] S.Y. Park, B. Pan, Identifying the next non-stop flying market with a big data approach, Tourism Management 66 (2018) 411–421.
[161] A. Kontogianni, E. Alepis, C. Patsakis, Promoting smart tourism personalised services via a combination of deep learning techniques, Expert Systems with Applications vol. 187, paper number 115964 (2022).
[162] M. Mariani, R. Baggio, Big data and analytics in hospitality and tourism: a systematic literature review, International Journal of Contemporary Hospitality Management no. 1, paper number 232 (2021).
[163] J.C. Cepeda-Pacheco, M.C. Domingo, Deep learning and internet of things for tourist attraction recommendations in smart cities, Neural Computing and Applications (2022) 1–19.
[164] J. Guerra-Montenegro, J. Sanchez-Medina, I. La na, D. Sanchez-Rodriguez, I. Alonso-Gonzalez, and J. Del Ser, "Computational intelligence in the hospitality industry: A systematic literature review and a prospect of challenges," Applied Soft Computing, vol. 102, p. 107082, 2021.
[165] F. Piccialli, F. Giampaolo, G. Casolla, V.S. Di Cola, K. Li, A deep learning approach for path prediction in a location-based iot system, Pervasive and Mobile Computing vol. 66, paper number 101210 (2020).
[166] Y.-C. Chang, C.-H. Ku, C.-H. Chen, Using deep learning and visual analytics to explore hotel reviews and responses, Tourism Management 80 (2020).
[167] G. Díaz, H. Macià, V. Valero, J. Boubeta-Puig, F. Cuartero, An intelligent transportation system to control air pollution and road traffic in cities integrating cep and colored petri nets, Neural Computing and Applications 32 (2) (2020) 405–426.
[168] P. Arthurs, L. Gillam, P. Krause, N. Wang, K. Halder, A. Mouzakitis, A taxonomy and survey of edge cloud computing for intelligent transportation systems and connected vehicles, IEEE Transactions on Intelligent Transportation Systems (2021).
[169] Q. Cai, M. Abdel-Aty, Y. Sun, J. Lee, J. Yuan, Applying a deep learning approach for transportation safety planning by using high-resolution transportation and land use data, Transportation Research Part A: Policy and Practice 127 (2019) 71–85.
[170] J. Wang, R. Chen, Z. He, Traffic speed prediction for urban transportation network: A path based deep learning approach, Transportation Research Part C: Emerging Technologies 100 (2019) 372–385.
[171] X. Wang, J. Liu, T. Qiu, C. Mu, C. Chen, P. Zhou, A real-time collision prediction mechanism with deep learning for intelligent transportation system, IEEE Transactions on Vehicular Technology 69 (9) (2020) 9497–9508.
[172] S. Yang, W. Ma, X. Pi, S. Qian, A deep learning approach to real-time parking occupancy prediction in transportation networks incorporating multiple spatio-temporal data sources, Transportation Research Part C: Emerging Technologies 107 (2019) 248–265.
[173] A. Paranjothi, M.S. Khan, S. Zeadally, A survey on congestion detection and control in connected vehicles, Ad Hoc Networks 108 (2020).
[174] C. Chen, B. Liu, S. Wan, P. Qiao, Q. Pei, An edge traffic flow detection scheme based on deep learning in an intelligent transportation system, IEEE Transactions on Intelligent Transportation Systems 22 (3) (2020) 1840–1852.
[175] M.M. Iqbal, M.T. Mehmood, S. Jabbar, S. Khalid, A. Ahmad, G. Jeon, An enhanced framework for multimedia data: Green transmission and portrayal for smart traffic system, Computers & Electrical Engineering 67 (2018) 291–308.
[176] R.S. Thejaswini and S. Rajaraajeswari, A real-time traffic congestion-avoidance framework for smarter cities, in: Proceedings of AIP Conference Proceedings, vol. 2039, no. 1, paper number 020009, 2018.
[177] S. Sarker, B. Wankum, T. Perey, M.M. Mau, J. Shimizu, R. Jones, B.S. Terry, A novel capsule-delivered enteric drug-injection device for delivery of systemic biologics: A pilot study in a porcine model, IEEE Transactions on Biomedical Engineering 69 (6) (2022) 1870–1879.
[178] S. Sangodoyin, E.M. Ugurlu, M. Dey, M. Prvulovic, A. Zajić, Leveraging on-chip transistor switching for communication and sensing in neural implants and gastrointestinal devices, IEEE Transactions on Biomedical Engineering 69 (1) (2021) 377–389.
[179] S. Li, P. Si, Z. Zhang, J. Zhu, X. He, N. Zhang, Dfca-net: Dual feature context aggregation network for bleeding areas segmentation in wireless capsule endoscopy images, Journal of Medical and Biological Engineering 42 (2) (2022) 179–188.

[180] M.R. Basar, F. Malek, K.M. Juni, M.S. Idris, M.I.M. Saleh, Ingestible wireless capsule technology: A review of development and future indication, International Journal of Antennas and Propagation vol. 2012, paper number 807165 (2012).
[181] I. Mehmood, M. Sajjad, S.W. Baik, Mobile-cloud assisted video summarization framework for efficient management of remote sensing data generated by wireless capsule sensors, Sensors 14 (9) (2014) 17 112–17 145.
[182] R. Hamza, K. Muhammad, Z. Lv, F. Titouna, Secure video summarization framework for personalized wireless capsule endoscopy, Pervasive and Mobile Computing 41 (2017) 436–450.
[183] K. Muhammad, S. Khan, N. Kumar, J. Del Ser, S. Mirjalili, Vision-based personalized wireless capsule endoscopy for smart healthcare: Taxonomy, literature review, opportunities and challenges, Future Generation Computer Systems 113 (2020) 266–280.
[184] K. Muhammad, M. Sajjad, M.Y. Lee, S.W. Baik, Efficient visual attention driven framework for key frames extraction from hysteroscopy videos, Biomedical Signal Processing and Control 33 (2017) 161–168.
[185] K. Muhammad, J. Ahmad, M. Sajjad, S.W. Baik, Visual saliency models for summarization of diagnostic hysteroscopy videos in healthcare systems, SpringerPlus vol. 5, paper number 1495 (2016).
[186] R. Shrestha, S.K. Mohammed, M.M. Hasan, X. Zhang, K.A. Wahid, Automated adaptive brightness in wireless capsule endoscopy using image segmentation and sigmoid function, IEEE Transactions on Biomedical Circuits and Systems 10 (4) (2016) 884–892.
[187] J.-Y. He, X. Wu, Y.-G. Jiang, Q. Peng, R. Jain, Hookworm detection in wireless capsule endoscopy images with deep learning, IEEE Transactions on Image Processing 27 (5) (2018) 2379–2392.
[188] X. Wu, H. Chen, T. Gan, J. Chen, C.-W. Ngo, Q. Peng, Automatic hookworm detection in wireless capsule endoscopy images, IEEE Transactions on Medical Imaging 35 (7) (2016) 1741–1752.
[189] Y. Yuan, J. Wang, B. Li, M.Q.-H. Meng, Saliency based ulcer detection for wireless capsule endoscopy diagnosis, IEEE Transactions on Medical Imaging 34 (10) (2015) 2046–2057.
[190] I. Mehmood, M. Sajjad, S.W. Baik, Video summarization based tele-endoscopy: a service to efficiently manage visual data generated during wireless capsule endoscopy procedure, Journal of medical systems 38 (9) (2014) 1–9.
[191] Y. Yuan, M.Q.-H. Meng, Deep learning for polyp recognition in wireless capsule endoscopy images, Medical physics 44 (4) (2017) 1379–1389.
[192] X. Jia, M.Q.-H. Meng, A deep convolutional neural network for bleeding detection in wireless capsule endoscopy images, 38th annual international conference of the IEEE engineering in medicine and biology society (EMBC), IEEE 2016 (2016) 639–642.
[193] S. Jain, A. Seal, A. Ojha, A. Yazidi, J. Bures, I. Tacheci, O. Krejcar, A deep cnn model for anomaly detection and localization in wireless capsule endoscopy images, Computers in Biology and Medicine 137 (2021).
[194] T. Aoki, A. Yamada, K. Aoyama, H. Saito, A. Tsuboi, A. Nakada, R. Niikura, M. Fujishiro, S. Oka, S. Ishihara, et al., Automatic detection of erosions and ulcerations in wireless capsule endoscopy images based on a deep convolutional neural network, Gastrointestinal endoscopy 89 (2) (2019) 357–363.
[195] X. Gu, K. Nahrstedt, A. Messer, I. Greenberg, D. Milojicic, Adaptive offloading for pervasive computing, IEEE Pervasive Computing 3 (3) (2004) 66–73.
[196] K. Yang, S. Ou, H.-H. Chen, On effective offloading services for resource-constrained mobile devices running heavier mobile internet applications, IEEE Communications Magazine 46 (1) (2008) 56–63.
[197] A.P. Miettinen and J.K. Nurminen, Energy efficiency of mobile clients in cloud computing, in: Proceedings of 2nd USENIX Workshop on Hot Topics in Cloud Computing, 2010.
[198] J.-C. Hsieh, A.-H. Li, C.-C. Yang, Mobile, cloud, and big data computing: contributions, challenges, and new directions in telecardiology, International Journal of Environmental Research and Public Health 10 (11) (2013) 6131–6153.
[199] G. Fortino, D. Parisi, V. Pirrone, G. Di Fatta, Bodycloud: A SaaS approach for community body sensor networks, Future Generation Computer Systems 35 (2014) 62–79.
[200] C. Xia, J. Zhao, H. Cui, X. Feng, J. Xue, Dnntune: Automatic benchmarking dnn models for mobile-cloud computing, ACM Transactions on Architecture and Code Optimization (TACO) 16 (4) (2019) 1–26.
[201] B. Kumar, G. Pandey, B. Lohani, S.C. Misra, A multi-faceted cnn architecture for automatic classification of mobile lidar data and an algorithm to reproduce point cloud samples for enhanced training, ISPRS journal of photogrammetry and remote sensing 147 (2019) 80–89.
[202] A.E. Eshratifar, M.S. Abrishami, M. Pedram, JointDNN: An efficient training and inference engine for intelligent mobile cloud computing services, IEEE Transactions on Mobile Computing 20 (2) (2019) 565–576.
[203] A. Katal, S. Dahiya, T. Choudhury, A survey on cloud computing in energy management of the smart grids, Cluster Computing (2021).
[204] W. Pedrycz, Towards green machine learning: challenges, opportunities, and developments, Journal of Smart Environments and Green Computing 2 (4) (2022) 163–174.
[205] R. Schwartz, J. Dodge, N.A. Smith, O. Etzioni, Green ai, Communications of the ACM 63 (12) (2020) 54–63.
[206] H. Jin, Q. Song, and X. Hu, Auto-keras: An efficient neural architecture search system, in: ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2019, pp. 1946–1956.
[207] A. Li, O. Spyra, S. Perel, V. Dalibard, M. Jaderberg, C.J. Gu, D. Budden, T. Harley, and P. Gupta, A generalized framework for population based training, in: Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2019.
[208] T. Elsken, J.H. Metzen, F. Hutter, Neural architecture search: A survey, The Journal of Machine Learning Research 20 (1) (2019) 1997–2017.
[209] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, X. Chen, X. Wang, A comprehensive survey of neural architecture search: Challenges and solutions, ACM Computing Surveys (CSUR) 54 (4) (2021) 1–34.
[210] H. Li, P. Wu, Z. Wang, J. f. Mao, F.E. Alsaadi, and Z.N. Y, "A generalized framework of feature learning enhanced convolutional neural network for pathology-image-oriented cancer," Computers in Biology and Medicine, vol. 151A, 2022, 106265.
[211] H. Li, N.Y. Zeng, P.S. Wu, K. Clawson, Cov-net: A computer-aided diagnosis method for recognizing covid-19 from chest x-ray images via machine vision, Expert Systems with Applications 207 (2022).
[212] P. Wu, Z. Wang, B. Zheng, H. Li, F.E. Alsaadi, N. Zeng, Aggn: Attention-based glioma grading network with multi-scale feature extraction and multi-modal information fusion, Computers in Biology and Medicine 152 (2023).
[213] I. Makdoom, M. Abolhasan, H. Abbas, W. No, Blockchains adoption in iot: the challenges, and a way forward, Journal of Network and Computer Applications 125 (2019) 251–279.
[214] H. Hu, Z. Salcic, L. Sun, G. Dobbie, P.S. Yu, X. Zhang, Membership inference attacks on machine learning: A survey, ACM Computing Surveys (CSUR) 54 (11s) (2022) 1–37.
[215] M. Jegorova, C. Kaul, C. Mayor, A.Q. O'Neil, A. Weir, R. Murray-Smith, S.A. Tsaftaris, Survey: Leakage and privacy at inference time, IEEE Transactions on Pattern Analysis and Machine Intelligence (2022).
[216] M. Abadi, A. Chu, I. Goodfellow, H.B. McMahan, I. Mironov, K. Talwar, and L. Zhang, Deep learning with differential privacy, in: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, 2016, pp. 308–318.
[217] C. Dwork, "Differential privacy: A survey of results," in Theory and Applications of Models of Computation: 5th International Conference, TAMC 2008, Xi'an, China, April 25-29, 2008. Proceedings 5. Springer, 2008, pp. 1–19.
[218] S. Meftah, B.H.M. Tan, C.F. Mun, K.M.M. Aung, B. Veeravalli, V. Chandrasekhar, Doren: toward efficient deep convolutional neural networks with fully homomorphic encryption, IEEE Transactions on Information Forensics and Security 16 (2021) 3740–3752.
[219] Z. Zhong, W. Bao, J. Wang, X. Zhu, X. Zhang, Flee: A hierarchical federated learning framework for distributed deep neural network over cloud, edge, and end device, ACM Transactions on Intelligent Systems and Technology (TIST) 13 (5) (2022) 1–24.
[220] R. Gupta, I. Gupta, D. Saxena, A.K. Singh, A differential approach and deep neural network based data privacy-preserving model in cloud environment, Journal of Ambient Intelligence and Humanized Computing (2022) 1–16.
[221] J. Vasa, A. Thakkar, Deep learning: Differential privacy preservation in the era of big data, Journal of Computer Information Systems (2022) 1–24.
[222] M.H. Rahman, M.M. Mowla, S. Shanto, Differential privacy enabled deep neural networks for wireless resource management, Mobile Networks and Applications 27 (5) (2022) 2153–2162.
[223] J. Xiong, H. Zhu, Real-time trajectory privacy protection based on improved differential privacy method and deep learning model, Journal of Cloud Computing 11 (1) (2022) 1–15.
[224] F. Gava, L.M. Bayati, A scalable algorithm for homomorphic computing on multi-core clusters, 2022 21st International Symposium on Parallel and Distributed Computing (ISPDC), IEEE (2022) 57–64.
[225] S. Meftah, B.H.M. Tan, K.M.M. Aung, L. Yuxiao, L. Jie, B. Veeravalli, Towards high performance homomorphic encryption for inference tasks on cpu: An mpi approach, Future Generation Computer Systems 134 (2022) 13–21.
[226] J.A. Alzubi, O.A. Alzubi, A. Singh, M. Ramachandran, Cloud-iiot-based electronic health record privacy-preserving by cnn and blockchain-enabled federated learning, IEEE Transactions on Industrial Informatics 19 (1) (2022) 1080–1087.
[227] Z. Liu, Z. Gao, J. Wang, Q. Liu, J. Wei, et al., Ppefl: An edge federated learning architecture with privacy-preserving mechanism, Wireless Communications and Mobile Computing 2022 (2022).
[228] J. Fiosina, "Interpretable privacy-preserving collaborative deep learning for taxi trip duration forecasting," in Smart Cities, Green Technologies, and Intelligent Transport Systems: 10th International Conference, SMARTGREENS 2021, and 7th International Conference, VEHITS 2021, Virtual Event, April 28–30, 2021, Revised Selected Papers. Springer, 2022, pp. 392–411.
[229] B. Jan, H. Farman, M. Khan, M. Imran, I.U. Islam, A. Ahmad, S. Ali, G. Jeon, Deep learning in big data analytics: a comparative study, Computers & Electrical Engineering 75 (2019) 275–287.
[230] J. Dean, G.S. Corrado, R. Monga, K. Chen, M. Devin, Q.V. Le, M.Z. Mao, M. Ranzato, A. Senior, P. Tucker et al., Large scale distributed deep networks, in: Proceedings of Advances in Neural Information Processing Systems, 2012.
[231] B. Abu-Salih, P. Wongthongtham, D. Zhu, K.Y. Chan, A. Rudra, Social Big Data Analytics: Practices, Techniques, and Applications, Springer Nature, 2021.
[232] A.A. Taha, W. Ramo, H.H.K. Alkhaffaf, Impact of external auditor–cloud specialist engagement on cloud auditing challenges, Journal of Accounting & Organizational Change 17 (3) (2021) 309–331.

[233] Z. Zhang, C. Wu, D.W. Cheung, A survey on cloud interoperability: taxonomies, standards, and practice, ACM SIGMETRICS Performance Evaluation Review 40 (4) (2013) 13–22.
[234] H. Schulze, Cloud security report, Fortinet, Report, 2021. [Online]. Available: https://www.fortinet.com/content/dam/fortinet/assets/analyst-reports/ar-cybersecurity-cloud-security.pdf.
[235] C. Ramalingam, P. Mohan, Addressing semantics standards for cloud portability and interoperability in multi cloud environment, Symmetry 13 (2) (2021) 317.
[236] 2020.[Online]. Available: https://insightaas.com/cloud-interoperability-and-portability-necessary-or-nice-to-have/.
[237] R. Ranjan, The cloud interoperability challenge, IEEE Cloud Computing 1 (2) (2014) 20–24.
[238] A. Romasanta and J. Wareham, Fair data through a federated cloud infrastructure: Exploring the science mesh, in: Proceedings of European Conference on Information Systems, 2021.
[239] B. Abu-Salih, Domain-specific knowledge graphs: A survey, Journal of Network and Computer Applications 185 (2021).
[240] H. Mezni, M. Sellami, S. Aridhi, F.B. Charrada, Towards big services: a synergy between service computing and parallel programming, Computing 103 (11) (2021) 2479–2519.
[241] O. Adedugbe, E. Benkhelifa, R. Campion, F. Al-Obeidat, A.B. Hani, U. Jayawickrama, Leveraging cloud computing for the semantic web: review and trends, Soft Computing 24 (8) (2020) 5999–6014.
[242] I. Grangel-González, F. Lösch, and A. ul Mehdi, Knowledge graphs for efficient integration and access of manufacturing data, in: Proceedings of 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), vol. 1. IEEE, 2020, pp. 93–100.
[243] G. Ditzler, M. Roveri, C. Alippi, R. Polikar, Learning in nonstationary environments: A survey, IEEE Computational Intelligence Magazine 10 (4) (2015) 12–25.
[244] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, G. Zhang, Learning under concept drift: A review, IEEE transactions on knowledge and data engineering 31 (12) (2018) 2346–2363.
[245] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, T. Tuytelaars, A continual learning survey: Defying forgetting in classification tasks, IEEE transactions on pattern analysis and machine intelligence 44 (7) (2021) 3366–3385.
[246] T. Khan, W. Tian, G. Zhou, S. Ilager, M. Gong, R. Buyya, Machine learning (ml)–centric resource management in cloud computing: A review and future directions, Journal of Network and Computer Applications (2022).
[247] R.S. Sutton, A.G. Barto, Reinforcement learning: An introduction, MIT press, 2018.
[248] V. François-Lavet, P. Henderson, R. Islam, M.G. Bellemare, and J. Pineau, An introduction to deep reinforcement learning, arXiv preprint arXiv:1811.12560, 2018.
[249] Y. Li, Deep reinforcement learning: An overview, arXiv preprint arXiv:1701.07274, 2017.
[250] B.R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A.A. Al Sallab, S. Yogamani, P. Pérez, Deep reinforcement learning for autonomous driving: A survey, IEEE Transactions on Intelligent Transportation Systems (2021) 4909–4926.
[251] F. Zeng, C. Wang, and S.S. Ge, "A survey on visual navigation for artificial agents with deep reinforcement learning," IEEE Access, vol. 8, pp. 135 426–135 442, 2020.
[252] K. Arulkumaran, M.P. Deisenroth, M. Brundage, A.A. Bharath, Deep reinforcement learning: A brief survey, IEEE Signal Processing Magazine 34 (6) (2017) 26–38.
[253] K. Shao, Z. Tang, Y. Zhu, N. Li, and D. Zhao, A survey of deep reinforcement learning in video games, arXiv preprint arXiv:1912.10944, 2019.
[254] A. Gupta, A. Anpalagan, L. Guan, A.S. Khwaja, Deep learning for object detection and scene perception in self-driving cars:survey, challenges, and open issues, Array vol. 10, paper number 100057 (2021).
[255] N.Y. Zeng, P.S. Wu, Z.D. Wang, H. Li, W.B. Liu, X.H. Liu, A small-sized object detection oriented multi-scale feature fusion approach with application to defect detection, IEEE Transactions on Instrumentation and Measurement 71 (2022).
[256] N. Rodríguez-Barroso, A.R. Moya, J.A. Fernández, E. Romero, E. Martínez-Cámara, and F. Herrera, Deep learning hyper-parameter tuning for sentiment analysis in twitter based on evolutionary algorithms, in: 2019 Federated Conference on Computer Science and Information Systems (FedCSIS). IEEE, 2019, pp. 255–264.
[257] S.R. Young, D.C. Rose, T.P. Karnowski, S.-H. Lim, and R.M. Patton, Optimizing deep learning hyper-parameters through an evolutionary algorithm, in: Proceedings of the Workshop on Machine Learning in High-Performance Computing Environments, 2015, pp. 1–5.
[258] R. Tanabe and A. Fukunaga, Success-history based parameter adaptation for differential evolution, in: Proceedings of IEEE Congress on Evolutionary Computation, 2013, pp. 71–78.
[259] M. Wu, W. Su, L. Chen, Z. Liu, W. Cao, K. Hirota, Weightadapted convolution neural network for facial expression recognition in human-robot interaction, IEEE Transactions on Systems, Man, and Cybernetics - Systems 51 (3) (2021) 1473–1484.
[260] M. Gong, J. Liu, A. Qin, K. Zhao, K.C. Tan, Evolving deep neural networks via cooperative coevolution with backpropagation, IEEE Transactions on Neural Networks and Learning Systems 32 (1) (2021) 420–434.
[261] M.R. Chen, B.P. Chen, G.Q. Zeng, K. Lu, P. Chu, An adaptive fractional-order bp neural network based on extremal optimization for handwritten digits recognition, Neurocomputing 391 (2020) 260–272.
[262] Z. Lu, G. Sreekumar, E. Goodman, W. Banzhaf, K. Deb, V.N. Boddeti, Neural architecture transfer, IEEE Transactions on Pattern Analysis and Machine Intelligence 43 (2021).
[263] B. Ma, X. Li, Y. Xia, Y. Zhang, Autonomous deep learning: A genetic dcnn designer for image classification, Neurocomputing 379 (2020) 152–161.
[264] F. Mattioli, D. Caetano, A. Cardoso, E. Naves, and E. Lamounier, An experiment on the use of genetic algorithms for topology selection in deep learning, Journal of Electrical and Computer Engineering, no. Article ID 3217542, 2019.
[265] B. Wang, Y. Sun, B. Xue, and M. Zhang, A hybrid ga-pso method for evolving architecture and short connections of deep convolutional neural networks, in: Proceedings of the 16th Pacific Rim International Conference on Artificial Intelligence, 2019.
[266] M. Suganuma, S. Shirakawa, and T. Nagao, A genetic programming approach to designing convolutional neural network architectures, in: Proceedings of Twenty-Seventh International Joint Conference on Artificial Intelligence, 2017, pp. 5369–5373.
[267] A.D. Martinez, J. Del Ser, E. Villar-Rodriguez, E. Osaba, J. Poyatos, S. Tabik, D. Molina, F. Herrera, Lights and shadows in evolutionary deep learning: Taxonomy, critical methodological analysis, cases of study, learned lessons, recommendations and challenges, Information Fusion 67 (2021) 161–194.
[268] T. Serizawa and H. Fujita. (2021) Optimization of convolutional neural network using the linearly decreasing weight particle swarm optimization. [Online]. Available: https://arxiv.org/abs/2001.05670.
[269] J. Fregoso, C.I. Gonzalez, G.E. Martinez, Optimization of convolutional neural networks architectures using pso for sign language recognition, Axioms vol. 10, number paper 139 (2021).
[270] Z. Fouad, M. Alfonse, M. Roushdy, A.-B.M. Salem, Hyper-parameter optimization of convolutional neural network based on particle swarm optimization algorithm, Bulletin of Electrical Engineering and Informatics 10 (6) (2021) pp.
[271] X. Chen, Y. Sun, M. Zhang, D. Peng, Evolving deep convolutional variational autoencoders for image classification, IEEE Transactions on Evolutionary Computation 25 (5) (2021).
[272] T. Zhang, C. Lei, Z. Zhang, X.-B. Meng, C.L.P. Chen, As-nas: Adaptive scalable neural architecture search with reinforced evolutionary algorithm for deep learning, IEEE Transactions on Evolutionary Computation 25 (5) (2021).
[273] L. Xie and A. Yuille, Genetic cnn, in: Proceedings of IEEE International Conference on Computer Vision, 2017.
[274] S. Pimminger, S. Wagner, W. Kurschl, and J. Heinzelreiter, Optimization as a service: On the use of cloud computing for metaheuristic optimization, in: Proceedings of the International Conference on Computer Aided Systems Theory, 2013, p. 348–355.
[275] A.D. Martinez, J.D. Ser, E. Villar-Rodriguez, E. Osaba, J. Poyatos, S. Tabik, D. Molina, F. Herrera, Lights and shadows in evolutionary deep learning taxonomy, critical methodological analysis, cases of study, learned lessons, recommendations and challenges, Information Fusion 67 (2021) 161–194.

Kit Yan Chan received the Ph.D. degree in computing from London South Bank University, London, U.K., in 2006. He is a Senior Lecturer in the School of Electrical Engineering, Computing and Mathematical Science, Curtin University, Perth, WA, Australia. He was a Full Time Researcher in Hong Kong Polytechnic University (2004–2009) and Curtin University (2009–2013). His research interests include artificial intelligence, machine learning and their applications to new product development, underwater acoustic communications, image/video quality evaluations etc. He has published more than 100 journal papers and several books and serves as an associate editor for several reputed journals.

Bilal Abu-Salih is an Assistant Professor at The University of Jordan and an Adjunct Professor at the Curtin University, Australia. He holds a PhD in Information Systems (with a focus on Social Big Data Analytics) from Curtin University. He worked with cross-disciplinary funded research projects which are related to data analytics, machine learning, data mining of social media, big data analysis, etc. Those projects are involved with academic research and also involved software development and industrial implementation. Bilal's research interests include Social Big Data, Social Trust, Machine Learning/Data Mining, Knowledge Graphs, NLP, and Information Retrieval.

Raneem Qaddoura is an Assistant Professor at Al Hussein Technical University. She earned her Ph.D. in computer science in machine learning and data mining. Raneem Qaddoura combines both academic and industrial experience of more than 15 years. She is also an active research member of the Evolutionary and Machine learning group (Evo-ML.com), which focuses on evolutionary algorithms, machine learning, and their applications for solving fundamental problems in different areas. Her current research involves evolutionary computation, deep learning, data classification, and clustering.

Ala' M. Al-Zoubi received the B.Sc. degree in software engineering from Al-Zaytoonah University, in 2014, and the M.Sc. degree in web intelligence from The University of Jordan, Jordan, in 2017. He is currently pursuing the Ph.D. degree with the School of Science, Technology and Engineering, University of Granada. He worked as a Teacher and a Research Assistant on several projects funded by The University of Jordan. During his graduate studies, he has published several publications in well-recognized journals and conferences. His research interests include evolutionary computation, machine learning, and security in social network analysis and other research fields. He is a member of two research groups, such as the JISDF Research Group, that focus on bridge the gap between the academic and industry mechanisms in security and the Evo-ML Research Group, where the group focuses on evolutionary algorithms, machine learning, and their applications for solving important problems in different areas.

Vasile Palade (Senior Member, IEEE) received the Ph.D. degree from the University of Galati, Galati, Romania, in 1999. He joined Coventry University, Coventry, UK, in 2013, after working for several years as a Lecturer with the Department of Computer Science, University of Oxford, Oxford, UK. He has authored 200 papers in journals and conference proceedings as well as several books. He is currently a Professor of Artificial Intelligence and Data Science in the Centre for Data Science at Coventry University. His research interests include the area of machine learning and applications, deep learning and neural networks, various nature inspired optimization algorithms, computer vision, and natural language processing. Prof. Palade has delivered keynote talks and chaired international conferences on machine learning and applications. He is an Associate Editor for several reputed journals.

Duc-Son Pham (Senior Member, IEEE) received the Ph. D. degree from the Curtin University of Technology, in 2005. He is currently a Senior Lecturer with the Discipline of Computing, Curtin University, Perth, Western Australia. His current research interests include sparse learning theory, large-scale data mining, convex optimization, and advanced deep learning with applications to computer vision and image processing. He was a recipient of the Young Author Best Paper Award 2010 for a publication in IEEE Transactions on Signal Processing.

Javier Del Ser (Senior Member, IEEE) received his first PhD in Telecommunication Engineering (Cum Laude) from the University of Navarra, Spain, in 2006, and a second PhD in Computational Intelligence (Summa Cum Laude, Extraordinary Prize) from the University of Alcala, Spain, in 2013. He has held several positions as a Professor and a Researcher at different institutions of the Basque Country. Currently he is a Research Professor in Data Analytics and Optimization at TECNALIA (Spain) and an adjunct professor at the University of the Basque Country (UPV/EHU). His research interests gravitate on the use of Artificial Intelligence for data modeling and optimization problems arising in a diverse range of application fields such as Energy, Transport, Telecommunications, Health and Industry, among others. In these fields he has published more than 400 scientific articles, co-supervised 15 Ph. D. Thesis, edited 6 books, co-authored 9 patents and participated/led more than 50 research projects. He has also been involved in the organization of various national and international conferences, has chaired three international workshops, and serves as an associate editor in a number of indexed journals, including Information Fusion, Swarm and Evolutionary Computation and IEEE Transactions on Intelligent Transportation Systems.

Khan Muhammad (Senior Member, IEEE) received the Ph.D. degree in digital contents from Sejong University, Republic of Korea, in February 2019. He was an Assistant Professor with the Department of Software, Sejong University, from March 2019 to February 2022. He is currently the Director of the Visual Analytics for Knowledge Laboratory (VIS2KNOW Laboratory) and an Assistant Professor (Tenure-Track) with the Department of Applied AI, School of Convergence, College of Computing and Informatics, Sungkyunkwan University, Seoul, Republic of Korea. His research interests include intelligent video surveillance, medical image analysis, information security, and multimedia data analysis. He has registered ten patents and has contributed over 200 papers in peer-reviewed journals and conference proceedings in his areas of research. He is an associate editor/editorial board member of more than 14 journals. He is among the highly cited researchers in 2021 and 2022 according to the Web of Science (Clarivate).