Journal of Network and Computer Applications 205 (2022) 103419

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

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

Review

Machine learning techniques in emerging cloud computing integrated paradigms: A survey and taxonomy

Dinesh Soni [*], Neetesh Kumar

Indian Institute of Technology Roorkee, Roorkee, Uttarakhand, India

ARTICLE INFO

Keywords:
Cloud computing
Edge computing
Fog computing
Mist computing
IoT
SDN
Cybertwin
Industry 4.0
Digitaltwin
IIoT
Machine learning (ML)
Integrated computing paradigm

ABSTRACT

Cloud computing offers Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) to provide compute, network, and storage capabilities to the clients utilizing the pay-per-use model. On the other hand, Machine Learning (ML) based techniques are playing a major role in effective utilization of the computing resources and offering Quality of Service (QoS). Based on the customer's application requirements, several cloud computing-based paradigms i.e., edge computing, fog computing, mist computing, Internet of Things (IoT), Software-Defined Networking (SDN), cybertwin, and industry 4.0 have been evolved. These paradigms collaborate to offer customer-centric services with the backend of cloud server/data center. In brief, cloud computing has been emerged with respect to the above-mentioned paradigms to enhance the Quality of Experience (QoE) for the users. In particular, ML techniques are the motivating factor to backend the cloud for emerging paradigms, and ML techniques are essentially enhancing the usages of these paradigms by solving several problems of scheduling, resource provisioning, resource allocation, load balancing, Virtual Machine (VM) migration, offloading, VM mapping, energy optimization, workload prediction, device monitoring, etc. However, a comprehensive survey focusing on multi-paradigm integrated architectures, technical and analytical aspects of these paradigms, and the role of ML techniques in emerging cloud computing paradigms are still missing, and this domain needs to be explored. To the best of the authors' knowledge, this is the first survey that investigates the emerging cloud computing paradigms integration considering the most dominating problem-solving technology i.e., ML. This survey article provides a comprehensive summary and structured layout for the vast research on ML techniques in the emerging cloud computing paradigm. This research presents a detailed literature review of emerging cloud computing paradigms: cloud, edge, fog, mist, IoT, SDN, cybertwin, and industry 4.0 (IIoT) along with their integration using ML. To carry out this study, majorly, the last five years (2017-21) articles are explored and analyzed thoroughly to understand the emerging integrated architectures, the comparative study on several attributes, and recent trends. Based on this, research gaps, challenges, and future trends are revealed.

1. Introduction

Cloud computing is the most promising computing paradigm that brings the concept of "computing utilities" in the market. In cloud computing paradigm, infrastructure, platform, and software services are offered as the "computing utilities" using a pay-per-use model across shared delivery networks, similar to the water, electricity, gas, and telecommunications services (Arockiam et al., 2011). Cloud computing has emerged with several integrated computing and networking paradigms such as edge computing, fog computing, mist computing, Internet of Things (IoT), Software-Defined Networking (SDN), digitaltwin, and industry 4.0. Working with the integration of several cloud computing paradigms and their archetypes is in the trend to meet the next generation computing requirements (Metri and Sarote, 2011). Further, the appliance of Machine Learning (ML) techniques along with integrated cloud computing paradigms is offering new research opportunities to the researchers to meet the future demand of technological advancements (Metri and Sarote, 2011). Best of the authors' knowledge, an extensive study for the integrated computing architectures with machine learning techniques has remained unexplored. On the other hand, a deep study of cloud-driven integrated emerging cloud computing paradigms along with ML techniques is very prominent to the researchers from academia and industry to meet the future computing demands. Therefore, in this paper, we present a systematic literature survey for the integrated, emerging, and cloud computing driven several paradigms with the appliances of ML techniques. Table 1

[*] Corresponding author.
E-mail addresses: d_soni@cs.iitr.ac.in (D. Soni), neetesh@cs.iitr.ac.in (N. Kumar).

https://doi.org/10.1016/j.jnca.2022.103419
Received 21 December 2021; Received in revised form 29 March 2022; Accepted 9 May 2022
Available online 24 May 2022

Table 1
List of important abbreviations.

Abbr	Definition	Abbr	Definition
AI	Artificial Intelligence	GA	Genetic Algorithm
ANN	Artificial Neural Network	GBDT	Gradient Boosted Decision Trees
AdaBoost	Adaptive Boosting	GD	Gradient descent
ACO	Ant Colony Optimization	HMM	Hidden Markov Models
AWS	Amazon Web Services	ICT	Information and Communication Technology
BLE	Bluetooth Low Energy	ITS	Intelligent Transportation System
BNs	Bayesian Networks	IIoT	Industrial Internet of Things
C-RAN	Cloud-Radio Access Network	KNN	k-Nearest Neighbor
CPS	Cyber–Physical System	LR	Linear Regression
CMFL	Communication-Mitigated Federated Learning	ML	Machine learning
DNN	Deep Neural Network	NN	Neural Network
DFMEA	Design Failure Mode and Effects Analysis	PQR	Predicting Query Run-time
DDoS	Distributed Denial of Service	QoS	Quality of Service
DT	Decision Tree	REST	Representational State Transfer
EEMD	Ensemble Empirical Mode Decomposition	RBM	Restricted Boltzmann Machine
eSGD	edge Stochastic Gradient Descent	RL	Reinforcement Learning
FL	Federated Learning	SDN	Software-Defined Networking
FEA	Finite Element Analysis	SM	Smart Manufacturing
F-RAN	Fog-Radio Access Network	QoS	Quality of Service

presents the important abbreviations and their respective definitions used in the survey, in alphabetic order to enhance the readability. Fig. 1 pictorially represents the road-map of this survey paper.

As an overview, edge computing deals with local data, and avoids data uploading to the cloud. It processes the data directly on the edge devices, and the devices are connected with nearby sensors or gateways (Gupta et al., 2017; Kumar et al., 2017; Wang et al., 2018b). Fog computing provides the facility to communicate, store, and compute data locally utilizing edge computing infrastructure that reduces the bandwidth, latency, and energy requirements (Sideratos et al., 2015; Ferreira et al., 2017; Shin et al., 2014). Mist computing processes the data on the network at the extreme edge that contains several micro-controllers and sensors. It harvests the connected resources by using the sensor's computation and communication capabilities (Asif-Ur-Rahman et al., 2018; Byers and Wetterwald, 2015; La et al., 2019). The Internet of Things (IoT) can be realized in the form of sensors, actuators, mobile phones, etc. that can interact and communicate with each other via local or internet-based connections (Kafi et al., 2013; Qin et al., 2016). The major applications of the IoT can be found in smart cities, transportation (Kumar et al., 2021), health system and agriculture (Jakkula and Cook, 2010). The Software-Defined Networking (SDN) is a new computer networking paradigm (Kim et al., 2017). SDN and Network Function Virtualization (NFV) offer network virtualization in cloud computing. Further, SDN programs the switches in a novel way and makes fine-grain traffic forwarding decisions in mega-scale data centers (Ahmed et al., 2017). A CyberTwin (CT) or digitaltwin is an identical 3D digital replica of a real-world location or object (Damjanovic-Behrendt and Behrendt, 2019). It is a key enabler of technology for smart manufacturing utilizing cloud infrastructure for the deployment and maintenance (Yu et al., 2019). Industrial Internet of Things (IIoT 4.0) is an emerging technology focusing on the automation of applications in industrial and machine-to-machine communication utilizing Artificial Intelligence (AI) and cloud computing technologies (Boyes et al., 2018; Souri et al., 2019). The study of integration among emerging cloud computing paradigms becomes crucial for understanding future technological trends. As cloud computing has been integrated directly or indirectly with a wide range of disciplines and applications to offer virtual services in the form of infrastructure, platform, and software (Arockiam et al., 2011). Further, an appropriate integration among different cloud computing paradigms offers the complementary advantages of the several integrating technologies to enhance the performance of the application and meet the user QoS requirements (Buyya et al., 2010).

The utilization of ML techniques in the cloud integrated computing paradigms is in the trend to meet several QoS requirements. To address the dynamic scheduling in the cloud based cyber–physical production system, Artificial Neural Network (ANN) and other soft computing techniques are being used for optimizing several QoS parameters (Ding et al., 2019). For workload management in cloud based databases, ML techniques such as nearest neighbor and classification tree are being used for Predicting Query Run-time (PQR) tree modeling (Gupta et al., 2008). The PQR learning enhances the application's scalability by making efficient resource allocation decisions. Unified Reinforcement Learning (URL) technique provides real-time provisioning of auto-configuration of VMs (Xu et al., 2012). Further, ANN and Linear Regression (LR) are utilized in adaptive resource provisioning to satisfy future resource demands (Islam et al., 2012). Support Vector Machine (SVM), ANN, and LR are also used in designing prediction models for cloud resource provisioning (Bankole and Ajila, 2013). The Support Vector Regression (SVR) provides an optimal resource allocation and load balancing by reducing service response time (Huang et al., 2013). The distributed Deep Neural Network (DNN) based architectures are also utilized to reduce communication cost, energy, and response time (Ogden and Guo, 2018). Furthermore, ANN, SVM, Random Forest (RF), and Decision Tree (DT) techniques are commonly used to improve the accuracy in Smart Grid (SG) (Alcaraz et al., 2011), Intelligent Transportation Systems (ITS) and Smart Manufacturing (SM) domains (Lahouar and Slama, 2015; Ponz et al., 2015; Monedero et al., 2012; Chakraborty et al., 2011). ARIMA (Autoregressive Integrated Moving Average) and SVR improve performance of live Virtual machine (VM) migration (Patel et al., 2016). On the same line, ANN and LR are also used for resource prediction during VM migration, while Reinforcement Learning (RL) and SVM are used for VM management. In mobile edge computing and storage technology, an anomaly detection method with DT integration using Deep Learning (DL) is proposed to enable better offloading strategies. The SVM, K-Nearest Neighbor (KNN), and RF are the frequently used algorithms for IoT device identification (Tuama et al., 2016), whereas Gaussian Mixture Model (GMM) and KNN offer maximum accuracy (Huynh et al., 2015). With this discussion, it is inferred that ML techniques are essentially in trend with the emerging cloud computing paradigms which need to be investigated thoroughly.

1.1. Google trends and yearly publication statistics

Fig. 2 shows the statistics for the number of articles used in the study considering each domain of emerging technology paradigms. To get a view of Google trends for continuing research in the domain of emerging cloud computing integrated paradigms, a study is carried out based on recent five years (2017–21) Google Trends data, and a pictorial representation of the data is shown in Fig. 3.

As an outcome, it is observed that edge, fog, and mist are in top trends for the integration with other cloud computing paradigms. This study explores the research trends based on the state of the art for the several cloud computing paradigms i.e., edge computing, fog computing, mist computing, IoT, SDN, digitaltwin, and industry 4.0. Further, Fig. 4 also presents yearly (2017–21) publication statistics from the Web of Science database. Based on this study, it is observed that the majority of articles appeared from the edge computing and IoT paradigms with the integration of other cloud computing paradigms.

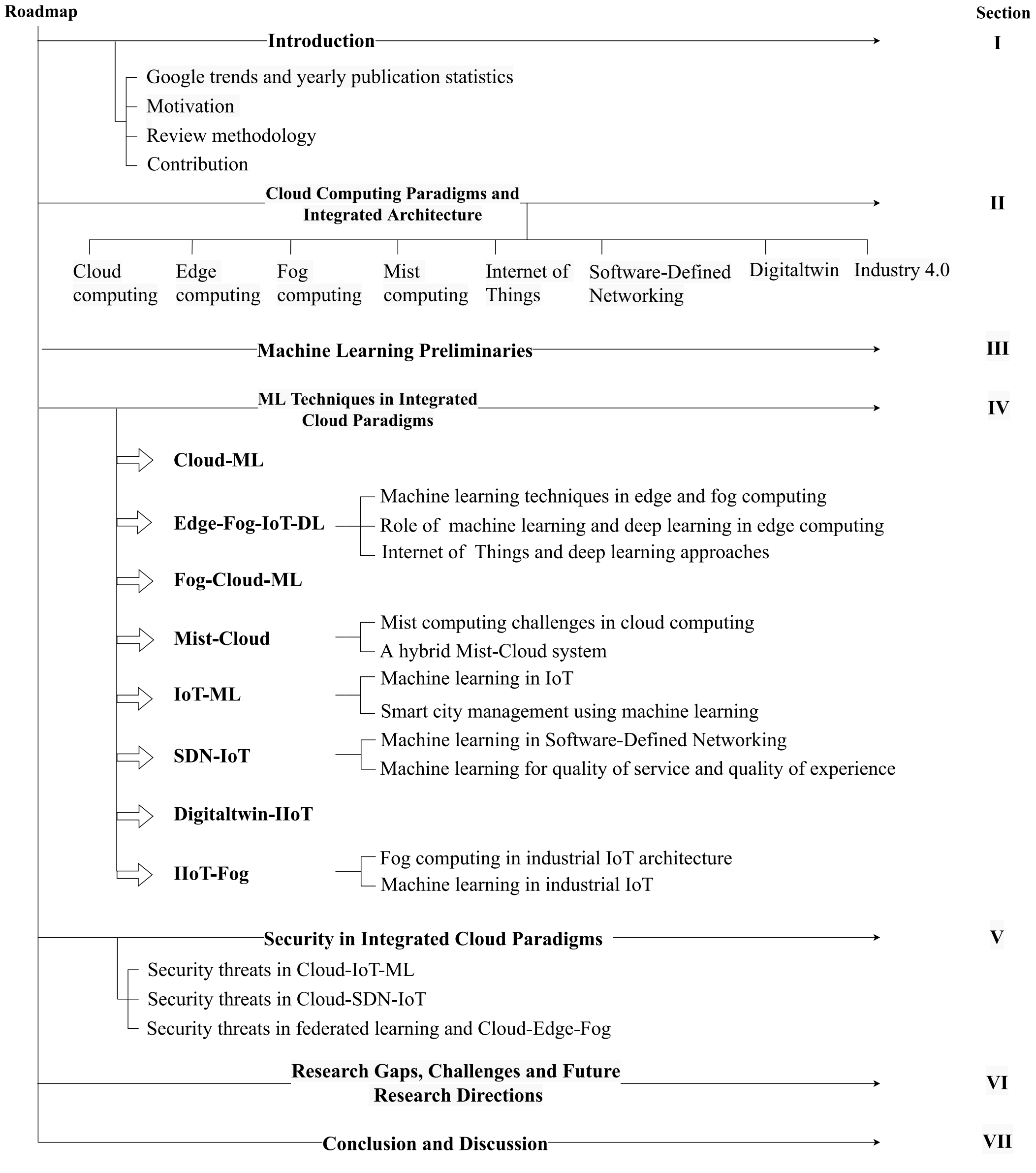

Roadmap
Section
Introduction
I
Google trends and yearly publication statistics
Motivation
Review methodology
Contribution
Cloud Computing Paradigms and Integrated Architecture
II
Cloud computing
Edge computing
Fog computing
Mist computing
Internet of Things
Software-Defined Networking
Digitaltwin
Industry 4.0
Machine Learning Preliminaries
III
ML Techniques in Integrated Cloud Paradigms
IV
Cloud-ML
Edge-Fog-IoT-DL
Machine learning techniques in edge and fog computing
Role of machine learning and deep learning in edge computing
Internet of Things and deep learning approaches
Fog-Cloud-ML
Mist-Cloud
Mist computing challenges in cloud computing
A hybrid Mist-Cloud system
IoT-ML
Machine learning in IoT
Smart city management using machine learning
SDN-IoT
Machine learning in Software-Defined Networking
Machine learning for quality of service and quality of experience
Digitaltwin-IIoT
IIoT-Fog
Fog computing in industrial IoT architecture
Machine learning in industrial IoT
Security in Integrated Cloud Paradigms
V
Security threats in Cloud-IoT-ML
Security threats in Cloud-SDN-IoT
Security threats in federated learning and Cloud-Edge-Fog
Research Gaps, Challenges and Future Research Directions
VI
Conclusion and Discussion
VII

Fig. 1. Road map of the survey.

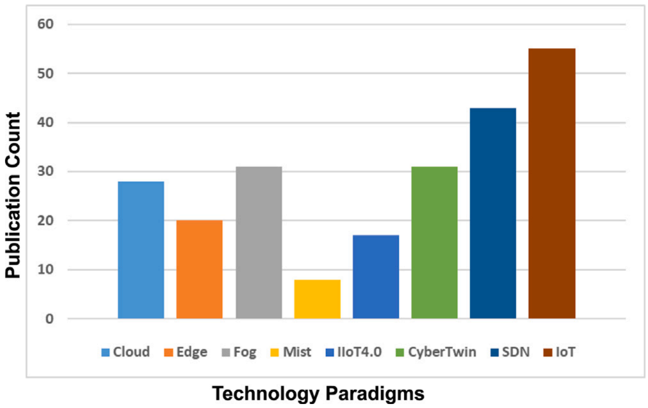


Fig. 2. Domain-wise studied articles.

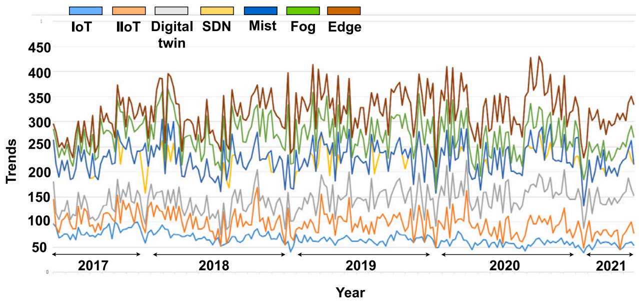


Fig. 3. Google trends during the last five years.

1.2. Motivation

As numerous state-of-art studies have been published in the context of cloud computing that discusses the application of ML and other related technologies (Huang et al., 2013). Based on the survey study (Buyya et al., 2010), it is observed that most of them have focused

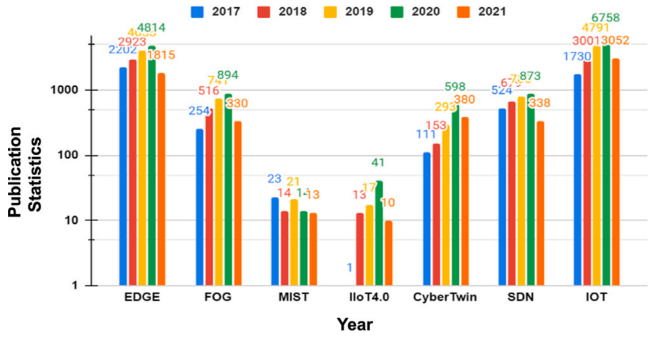


Fig. 4. Yearly publication statistics (on log scale).

on independent architectural study. On the other hand, Google trends data and state-of-the-art studies are evidence that future technological developments are highly dependent on the integration of emerging cloud computing paradigms along with ML appliances. With several ML applications/requirements of integration among various emerging cloud computing paradigms (edge computing Datta and Bonnet, 2017, fog computing Liu and Jin, 2013, mist computing Byers and Wetterwald, 2015, IoT Kurtz et al., 2016, SDN Bi et al., 2018, digitaltwin Bao et al., 2019, and industry 4.0 Čolaković and Hadžialić, 2018), it becomes very important to conduct a systematic survey to understand the broad overview of the possible integration, future trend, application demands, pros and cons, and research issues and challenges for the integration among several emerging cloud computing paradigms. Best of the author's knowledge, there is no systematic survey has been conducted, in the literature, for the same. Therefore, in this work, we focused on a survey paper to carryout a comprehensive study on integrated cloud computing paradigms, archetypes, and machine learning techniques in the integrated cloud computing paradigms.

1.3. Review methodology

The survey has been conducted using systematic literature review (SLR) process (Kitchenham and Charters, 2007; Ahmad and Alsmadi, 2021). The SLR has three phases, namely, planning the review, conducting the review, and reporting the review. During the planning phase, the study recognizes need for the review, defines the research areas, and designs a review protocol. This survey explained multi-paradigms integrated architectures and the applications of ML techniques to improve the services in integrated cloud computing paradigms. The survey discussed following research questions:

•$RQ1$: What are the major areas of integrated cloud computing paradigms?

•$RQ2$: Why there is a need to integrate different cloud computing paradigms?

•$RQ3$: What are the various machine learning techniques used in integrated cloud computing paradigms?

•$RQ4$: How machine learning techniques provide solutions to address the issues in these integrated cloud computing paradigms?

The review protocol has been defined based on the identified research questions and study objectives. In a research review, searching the most related studies connected to the research topic is important. To obtain the effective contents, ACM Digital Library, IEEE Xplore, Web of Science Core Collection, Scopus, ScienceDirect, Google Scholar, and electronic scientific databases sources were explored using following keywords cloud computing, internet of things, mist computing, edge computing, fog computing, industrial internet of things, cybertwin, software defined networking, internet of things, machine learning in cloud computing, machine learning and security in IoT/IIoT/cybertwin/edge/mist/fog/SDN etc. The survey defined inclusion and exclusion criteria to focus study on the most recent articles and future trends in machine learning techniques for integrated cloud computing paradigms. A filter was added to the search terms to look for peer-reviewed literature, conference paper, journal paper, blogs, and scientific book chapter well written in English language. The survey excluded research papers unrelated to above mentioned research questions and those articles that did not discuss the paradigms, posters, citations, non-English, preliminary studies, proof-of-concept, Powerpoint presentations, venue impact factor, etc.

Next, in the review phase, data extraction and quality assessment of the papers is performed. On the remaining articles, data extraction is done in order to obtain meaningful content. A reference management system (i.e., Mendeley) is used to extract the title, citations, publication channels, publication year, dataset, ML methodology and models, results, etc. The quality-assessment questions and scores are used to evaluate the articles that are relevant to the research-questions. The papers that have offered a thorough, comprehensive, understandable explanation of ML techniques, with the results of data analysis by mentioning IoT/IIoT/cybertwin/edge/mist/fog/SDN were given maximum score. The papers in which ML algorithms were simply mentioned with no further explanation and less technical information, and addressing only cloud computing not the other paradigms were given medium score. The papers without ML-based solutions and no real technical data-based content analysis were given minimum score. If the paper received more than 50% score, then it is included in the research review; else, it was rejected or excluded. The reporting the review phase represents a procedure for quickly and efficiently reviewing and examining each research studies included within the review. It integrates the results of the study and make conclusions from particular research outcomes. Finally, about 233 primary studies were recognized using data analysis, and information relevant to the research questions were retrieved and stored in database. The extraction of ML algorithms have done after reading entire-text of related primary studies. The primary studies show that the research for ML techniques in integrated cloud computing paradigms is less mature, and it requires further exploration.

1.4. Contribution

This survey presents, first time, a comprehensive study on integrated cloud computing architectures using ML techniques for emerging and integrated cloud computing paradigms. Our major contributions, in the survey, are itemized as follows:

- This work identifies the current research trends in the domain of emerging cloud computing paradigms, and explored the state of art thoroughly to investigate several integrated architectures, and the used ML techniques. In this work, a multidisciplinary study among several emerging cloud computing paradigms is carried out to understand the integration among them for meeting the application-specific QoS requirements.
- This survey classifies several integrated cloud computing paradigms based on the most dominant and trending technology i.e., ML. This work also investigates the frequently used ML techniques in the domain of integrated cloud computing paradigm.
- This survey majorly focuses on recently published articles, in the last five years (2017–21), which encourages the readers to quickly grasp the knowledge from scratch to the current advancements.
- This survey presents a brief tabulated summary on several QoS parameters, for almost every sub-domain, which presents a comparative analysis among similar state-of-the-art methods. This reduces a lot of readers effort to understand the domain/sub-domain-specific concepts, current research trends, advancements, and significant tools and techniques used.

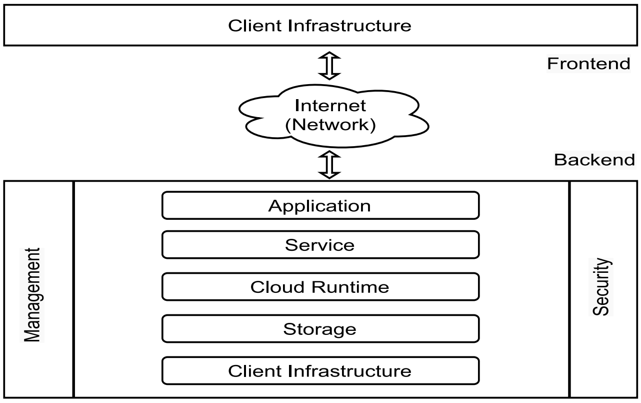


Fig. 5. Cloud computing architecture.

- Finally, this survey summarizes significant research gaps, challenges, and future research directions as an outcome of this thorough study. This will surely assist the researchers (including newborn) to carry forward the research in the domain of integrated cloud computing paradigms.

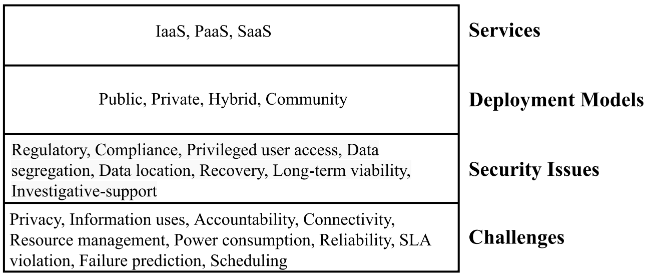
The rest of the paper is organized as follows. Section 2 defines the cloud computing paradigms and integrated architecture. Section 3 discusses machine learning preliminaries (supervised ML, unsupervised ML, reinforcement learning, deep neural network, and federated learning). Section 4 identifies ML techniques in the integrated cloud computing paradigms and presents a detailed comparative study. Section 5 discusses security in integrated cloud paradigms. Based on the study in Section 4, Section 6 presents the unidentified research gaps, challenges, and research future directions. Finally, Section 7 concludes the survey paper.

2. Cloud computing paradigms and integrated architecture

This section discusses various paradigms in the cloud computing domain i.e., edge computing, fog computing, mist computing, IoT, SDN, digitaltwin, and IIoT 4.0. In particular, this section briefs about the fundamental architectural study of the cloud computing paradigms as mentioned above.

2.1. Cloud computing

According to the National Institute of Standards and Technology (NIST), Mell et al. (2011) definition, *"Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction"*. A typical cloud computing architecture is shown in Fig. 5, which consists of front-end and back-end components that are connected using the internet (network). The front-end contains interfaces and applications to access cloud computing platforms that provide the user interface, client infrastructure, software, and network services. The back-end contains the resources that offer cloud computing services, traffic control, protocols, and security mechanism. It also provides application, storage, infrastructure, management software, security, cloud deployment, server, hypervisor-based services, etc. The NIST defined five key essential characteristics of cloud computing are broad network access, on-demand self-service, resource pooling, service metering, and rapid elasticity (Metri and Sarote, 2011). First, the on-demand self-service, provides the ability for users to provision computing capabilities as per their needs and with minimal third-party interactions. Second, broad network access, provides cloud resources access using standard and open protocols. This enables cloud resources access by various types of devices via the internet. Third, resource pooling and multi-tenancy, provides the ability to pool the cloud resources and serves multiple users utilizing a multi-tenant service model. Fourth, the rapid elasticity, provides the ability to scale-up and down the computing resources based on application requirements as most of the applications resource demand is unpredictable. Fifth, the service metering, measures the cloud services, and reports resource usage to optimize resource utilization using metering and measuring capability. The summary about basics of cloud computing system is shown in Fig. 6 (Buyya et al., 2010).



IaaS, PaaS, SaaS	**Services**
Public, Private, Hybrid, Community	**Deployment Models**
Regulatory, Compliance, Privileged user access, Data segregation, Data location, Recovery, Long-term viability, Investigative-support	**Security Issues**
Privacy, Information uses, Accountability, Connectivity, Resource management, Power consumption, Reliability, SLA violation, Failure prediction, Scheduling	**Challenges**

Fig. 6. Cloud computing system.

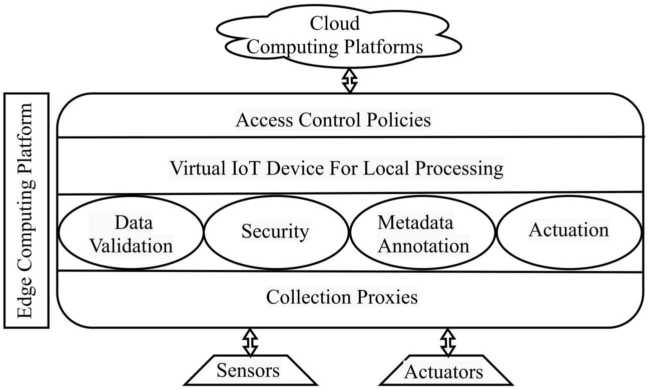


Fig. 7. Edge computing architecture.

2.2. Edge computing

NIST defines (Iorga et al., 2017) *"Edge computing as the network layer encompassing the end-devices and their users, to provide, for example, local computing capability on a sensor, metering or some other devices that are network-accessible"*. Edge computing is a distributed computing model where computation and data storage are brought close to the data source (edge) to minimize the response time and bandwidth usage. It reduces latency and potential failure points in an real-time IoT applications. It provides computation decentralization and data privacy. Fig. 7 depicts the architecture of edge computing along with its major components (Datta and Bonnet, 2017). The architecture has four layers, the first layer contains sensors, actuators, and collection proxies. The second layer contains data validation, security, metadata annotations, and actuation processes. The third layer contains virtual IoT devices for local data processing. Lastly, the fourth layer contains access control policies. The sensors and actuators are connected via collection proxies (Kumar et al., 2017). The collection proxies are the series of software drivers and scripts that are used to gather data. It allows actuators, sensors, and protocols (HTTP, CoAP) to communicate together. It also connects heterogeneous IoT devices to edge computing architecture. The next layer provides data validation, security, metadata annotations, and actuation processes (Wang et al.,

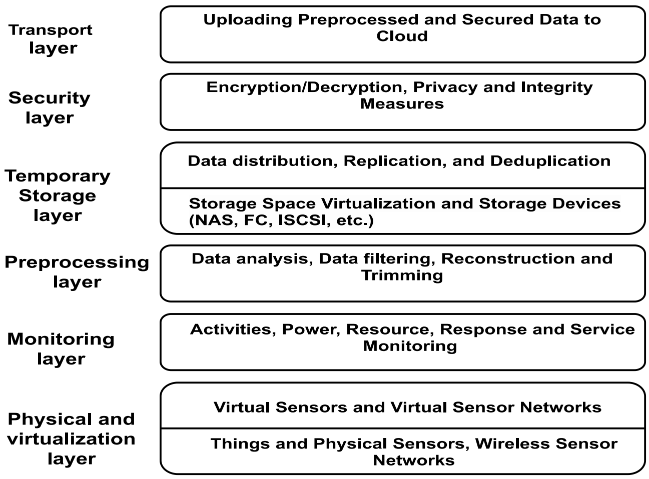

Transport layer
Uploading Preprocessed and Secured Data to Cloud
Security layer
Encryption/Decryption, Privacy and Integrity Measures
Temporary Storage layer
Data distribution, Replication, and Deduplication
Storage Space Virtualization and Storage Devices (NAS, FC, ISCSI, etc.)
Preprocessing layer
Data analysis, Data filtering, Reconstruction and Trimming
Monitoring layer
Activities, Power, Resource, Response and Service Monitoring
Physical and virtualization layer
Virtual Sensors and Virtual Sensor Networks
Things and Physical Sensors, Wireless Sensor Networks

Fig. 8. Fog computing architecture.

2018b). The data validation process is used to save bandwidth and reduce the load (Zhang et al., 2018b). It can run encryption–decryption techniques that use AES256 to encrypt the payload provides security before connecting with the cloud (Howard et al., 2017). The metadata annotations, which add more information to metadata, that enable virtual IoT devices to process data conveniently. The third layer also uses virtualization methods to virtualize the actuators (Tan and Le, 2019). It creates virtualized instances for actuators or sensors that are hosted on the cloud or at the edge (Ogden and Guo, 2018). By activating the actuators and satisfying specific conditions locally, it improves real-time operation needs (Mao et al., 2017).

2.3. Fog computing

Fog computing is defined by NIST (Iorga et al., 2018) as *"Fog computing is a layered model for enabling ubiquitous access to a shared continuum of scalable computing resources"*. According to NIST, fog nodes can be physical or virtual, and they were connected to smart end-devices and access networks (Ponz et al., 2015). Between the edge layer and centralized computing resources (cloud layer), fog nodes often provide some type of data management, computation and communication services. CISCO introduced fog computing in 2012 to provide cloud-based solutions (Susto et al., 2014). Edge devices are used for local computation, storage, and communication services (Monedero et al., 2012). Later, the fog architecture performs routing over the internet backbone. By pre-processing the sensor's data, it saves bandwidth and resource usages (Osaba et al., 2016). It improves both performance and security in real-time. Bypassing the cloud, fog nodes can connect directly with other fog nodes in a mesh network and can form federated fog architecture (Lieber et al., 2013). Fig. 8 depicts the fog computing architecture (Aazam et al., 2018). The fog computing architecture meets the real-time computational, storage, and communication requirements efficiently (Liu and Jin, 2013). Fig. 9 shows fog computing serving smart end devices in cloud-based ecosystem (Iorga et al., 2018). The architectures can vary as per use-cases to support the functionality of end-devices (Chakraborty et al., 2011).

Major characteristics of fog computing technology are contextual location awareness, low latency, geographical distribution, large-scale sensor network, distributed deployment, mobility support, real-time interactions, predominance of wireless IoT access, heterogeneity, interoperability, federation and real time streaming analytics (Samann et al., 2021). Additional features support in terms of hardware and software. Fog computing allows low-cost, flexible, and portable deployment. It also provides low latency, agility, and scalability for fog-node clusters (Osaba et al., 2016).

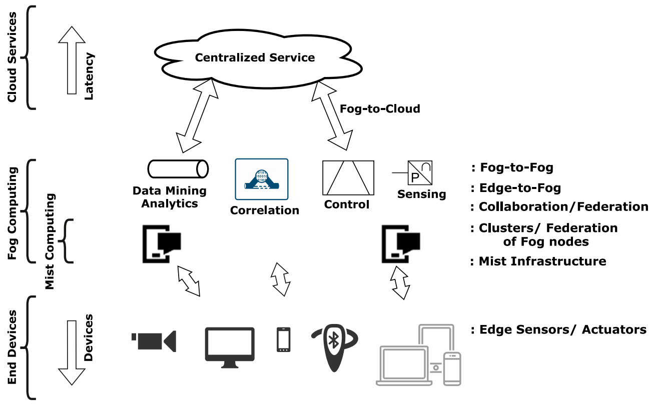


Fig. 9. A cloud-based ecosystem supported by fog computing.

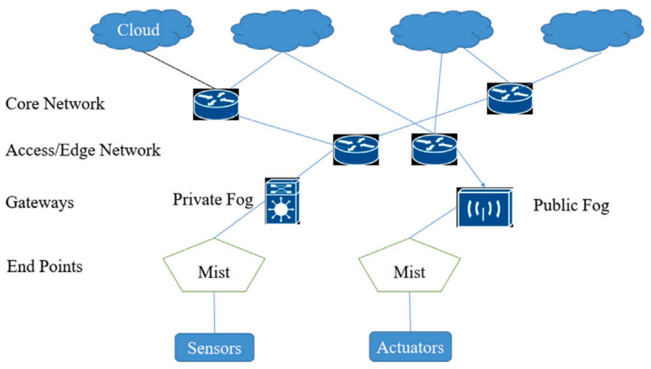


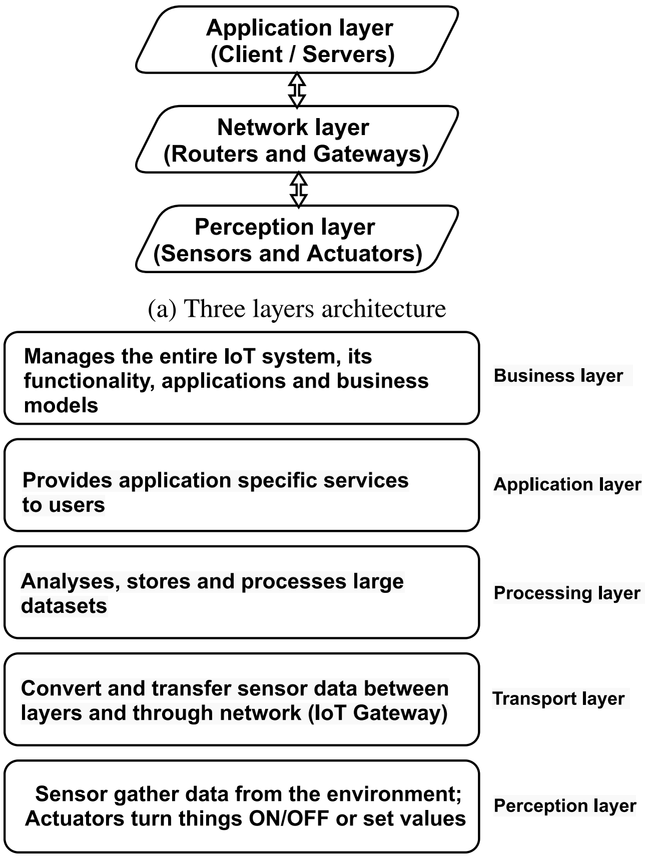
Fig. 10. Mist computing architecture.

2.4. Mist computing

According to NIST definition, Iorga et al. (2018), *"Mist computing is a lightweight and rudimentary form of fog computing that resides at the edge of the network fabric, bringing the fog computing layer closer to the smart end-devices"*. Micro-controllers and sensors are used in mist computing to function at the network's edge. They are used in mist computing to generate input data into fog computing nodes and possibly forward to centralized (cloud) computing services (Asif-Ur-Rahman et al., 2018). It was introduced by CISCO in 2014. Fig. 10 shows architecture of mist computing (Liyanage et al., 2018). The mist layer lies between the boundary and the fog domain that computes at the edge with a micro-controller. It is utilized to accomplish two objectives. First, resource harvesting is possible due to the sensor's inherent compute and communication capabilities. Second, it enables the deployment, management, and monitoring of any calculation on the sensor (Byers and Wetterwald, 2015). The computation is performed towards the edge of the IoT architecture in mist computing. The control is decentralized to endpoint nodes during the data transmission, resulting in minimal communication delay and maximum throughput (La et al., 2019). As a result, the load on fog nodes is reduced. Major characteristics of mist computing are as follows. First, mist provides a middle ground between cloud and edge/fog layers. Second, microcontrollers and microchips are embedded with mist devices. Third, mist computing works with cloud platforms and is used to make local decisions. Fourth, mist does not require a defined architecture. Finally, mist computing requires distributed servers over large geographical locations.

2.5. Internet of Things

Tarkoma and Katasonov (2011) defined the Internet of Things (IoT) as, *"A global network and service infrastructure of variable density and*


Application layer
(Client / Servers)
Network layer
(Routers and Gateways)
Perception layer
(Sensors and Actuators)
Manages the entire IoT system, its functionality, applications and business models
Business layer
Provides application specific services to users
Application layer
Analyses, stores and processes large datasets
Processing layer
Convert and transfer sensor data between layers and through network (IoT Gateway)
Transport layer
Sensor gather data from the environment; Actuators turn things ON/OFF or set values
Perception layer

(a) Three layers architecture

(b) Five layers architecture

Fig. 11. IoT architectures.

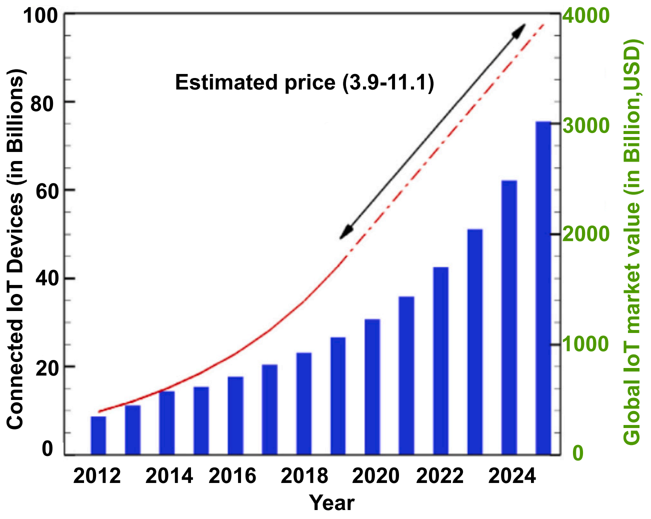


Fig. 12. IoT device growth per year (Manyika et al., 2015).

connectivity with self-configuring capabilities based on standard and interoperable protocols and formats [which] consists of heterogeneous things that have identities, physical and virtual attributes, and are seamlessly and securely integrated into the Internet". As indicated in Fig. 11, there are two types of IoT architectures i.e. three-layer and five-layer available in the literature (Kumar and Mallick, 2018). Fig. 11(a) shows three-layer IoT architecture that has application, network, and perception layers. The application layer delivers application-specific services and defines numerous IoT deployment services such as smart homes, smart cities, and smart health (Huynh et al., 2015). LTP, HTTP, DDS, AMQP, DNP, CoAP, SSH, IPfix, and NTP are the examples of application layer protocols. The network layer is responsible for routing, discovering, logical addressing, internetworking, and fragmentation throughout the network. It translates the logical addresses into physical addresses and passes the packets from the source node to the destination node. The IPv4/IPv6, SLIP, uIP, RLP, and TCP/UDP are examples of network layer protocols (Čolaković and Hadžialić, 2018). The perception layer is made up of actuators, sensors, and edge devices that interfaces with the outside world (Tarkoma and Katasonov, 2011). Sensors on the perception layer detect physical factors and recognize intelligent objects in the environment (Baldini et al., 2017). IEEE 802.11, 802.15, 802.16 series, PLC, WSN, GPS, and KNX are the some examples of physical layer protocols.

Fig. 11(b) shows five-layer IoT architecture that has business, application, processing, transport, and perception layers. The business layer manages IoT systems with industry applications. The application layer is used to supply application-specific services, define IoT applications, provide data processing and storage. The processing layer includes modules for cloud, databases and big data processing (Patel et al., 2014). The transport layer employs network technology (3G, LAN, Bluetooth, RFID, etc.) to deliver sensor data between the physical and processing layers. Finally, the perception layer senses data using sensors and gathers data from the environment. Fig. 12 shows the expected annual growth of connected IoT devices and the global market of IoT from 2012 to 2025 (Manyika et al., 2015). By 2025, it is expected that the IoT will capture the market worth between 3.9–11.1 trillion dollars (USD). IoT has its applications in smart cities, entertainment, smart environment, energy conservation, social life, home automation, agriculture, fitness, health, logistics in supply chain and other related areas (Tarkoma and Katasonov, 2011).

2.6. *Software-defined networking*

The issues faced by the supporting network infrastructure are evolving in conjunction with the development of cloud computing. To overcome such issues, a more programmable, secure, and dynamic networking infrastructure is required. Software-Defined Networking (SDN) is one such type of network architecture approaches (Bi et al., 2018). It allows you to control the networking architecture in the cloud. SDN offers numerous advantages to cloud computing, like dynamic workloads support, programmable control planes, software-defined centralized control, network flexibility, higher performance, easy deployment, efficient configuration, network virtualization, enhancements in network security, Data Center Network (DCN) global view, and the integration between the hypervisor and cloud computing manager (Jain et al., 2016).

Fig. 13 depicts a SDN architecture. The SDN architecture is made up of three layers such as application layer, control layer, and infrastructure layer. Using the northbound API, the application layer establishes rules and provides various network services such as intrusion detection systems, load balancing, firewalls, access control, and quality of service, routing, proxy service, and monitoring. The control layer serves as the SDN's brain with SDN controller (Bi et al., 2018). This layer is an abstraction of network topology that exists on the server-side, and it is used to control network policies and traffic flow. Its capabilities include creating flow tables, data handling, network information via the southbound API and network complexity. Physical switches in the network are found at the infrastructure layer. The controller uses API (OpenFlow) to communicate with switches (Bi et al., 2018). The flow table of the OpenFlow switch is made up of flow rules. Match fields, counters, and a sequence of instructions make the flow rules. The flow rules are matched with the incoming packets at the switch, and the switches perform predetermined actions.

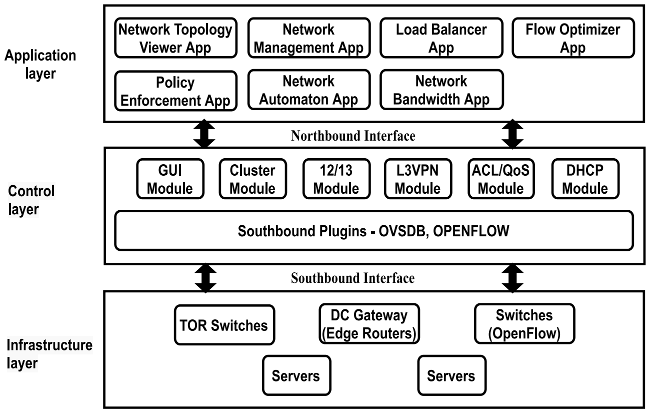


Fig. 13. A software-defined networking architecture.

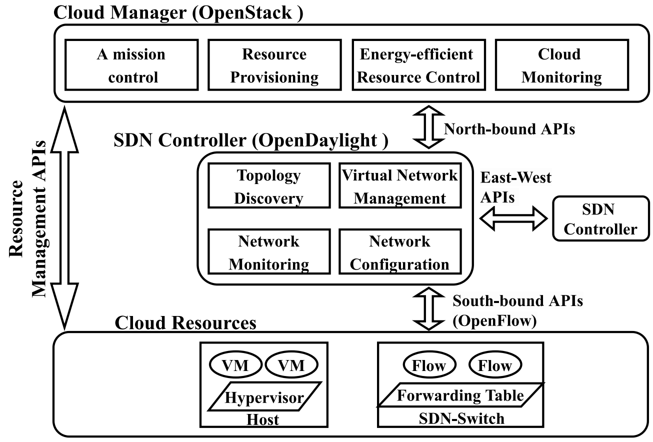


Fig. 14. A software-defined networking enabled cloud computing architecture.

Fig. 14 shows the architecture of cloud computing with an appropriate integration of SDN. The tenants and resources present in cloud are managed by the cloud manager. It manages incoming tenant requests, such as VM creation, and allocates cloud resources to deliver the cloud services. The cloud manager does energy-efficient resource management and resource monitoring (Mishra et al., 2022). OpenStack is an open-source cloud manager that is commonly used to create private clouds (Kim et al., 2017). The SDN controller, which is connected to the cloud manager via north-bound APIs, controls network-related services. Computing and networking resources are included in cloud resources. The cloud manager provisions compute resources (hosts) to run VMs on the hypervisor, whereas, the SDN controller manages networking resources (switches) by managing forwarding tables in switches via south-bound APIs.

2.7. Digitaltwin (Cybertwin)

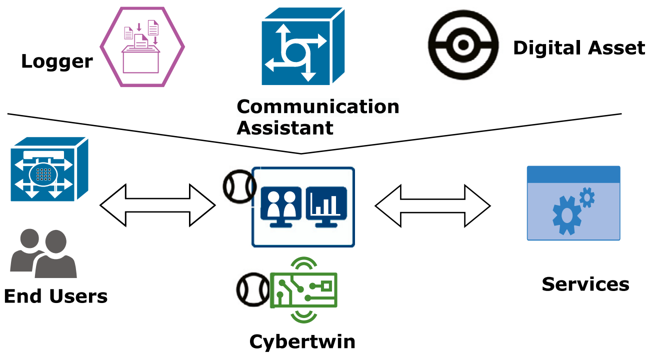


Fig. 15. Cybertwin based communication framework.

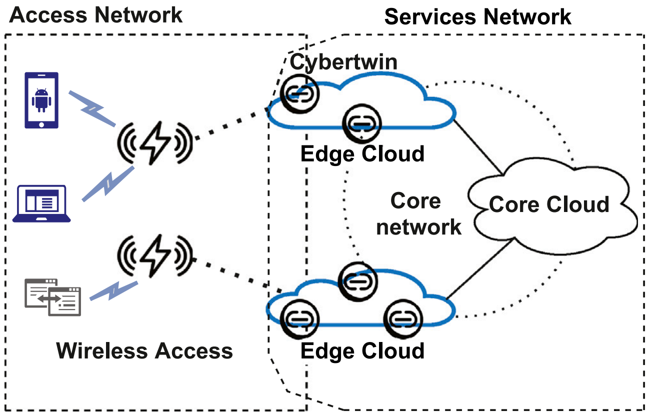


Fig. 16. A cloud-network architecture using cybertwin.

According to Madni et al. (2019), *“A digitaltwin is a virtual instance of a physical system (twin) that is continuously updated with the latter’s performance, maintenance, and health status data throughout the physical system’s life cycle”.* Michael Grieves of Florida Institute of Technology was the first to offer this concept in 2002. It is a virtual cyberspace representation of humans or things those are located at the edge (Bao et al., 2019). It replaces the end-to-end communication model with a new cybertwin-based communication model. Digitaltwin is a combination of two words “Digital + Twin”. Here, “Digital” means two-way real-time communication, simulation of scenarios, and decision-support functionalities while “Twin” means a replica of physical assets, availability in CAD models, and high fidelity (Liu et al., 2019c). It includes both physical and virtual products, as well as the links between them. Data flow between physical and digital items, and any changes in physical objects can be automatically mirrored in digital objects. Fig. 15 represents cybertwin-based communication framework (Yu et al., 2019). Cybertwin works as a communication assistant, network data logger, and digital asset owner, among the things. These three functions are provided by the cybertwin-based communication framework. First, as a *Communication Assistant*, end device (human or things) connects with its cybertwin. Next, cybertwin gets the response from the server and delivers them to the end devices. Second, as a *Network Behavior Logger*, cybertwin serves as a digital replica of the end devices. On the behalf of end devices, it collects and logs all data. Finally, as a *Digital Asset Function*, cybertwin removes sensitive information from user data. It transforms user data into digital assets (Zhuang et al., 2018). The cloud-network architecture using cybertwin (Yu et al., 2019) is shown in Fig. 16. It has four components, namely: core cloud, edge cloud, cybertwin, ends, etc. The core cloud supports caching, computing, and communication resources for the ends. The edge cloud lies in between the core cloud and the ends to give a quick response to the ends. It provides high-quality services to the network. The cybertwin is located in the edge cloud and provides digital representation for things in virtual space. The ends represent humans and things. The services are delivered straight from the network to the ends via cybertwin. These components are supported by different networks like access network, core network, and service network. The access network uses both wireless and wired methods to connect ends and edge clouds. The core network connects edge clouds and different core clouds. The service network is a type of logical network on the top of the cloud that offers services to ends.

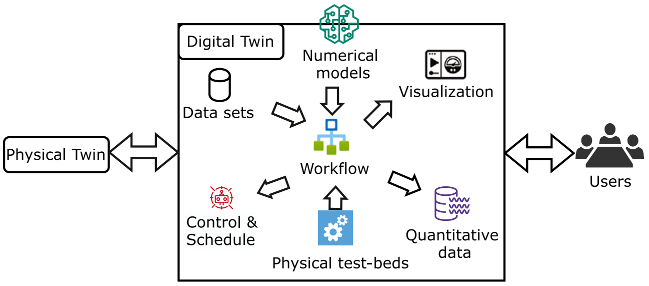


Fig. 17. Digitaltwin functions.

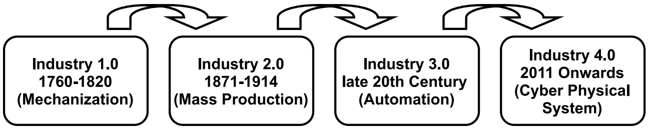


Fig. 18. Industrial revolution shift.

Digitaltwin performs a variety of functions, including numerical models, data sets, controlling and scheduling, physical test-beds, visualization, numerical models, processes, and quantitative data analysis as shown in Fig. 17.

2.8. Industry 4.0 (IIoT or fourth industrial revolution)

Digital revolution may be found all around us, but it appears more ubiquitous in cloud computing. Developing digital assets and operating life cycles are being increasingly important for businesses. The cloud is experiencing the next level of business-critical computing (Wichmann et al., 2019). Cloud computing is evolving beyond the commercial and consumer markets to include manufacturing and industrial automation. Boyes et al. (2018) defined Industrial Internet of Things (IIoT) as *“A system comprising networked smart objects, cyber–physical assets, associated generic information technologies and optional cloud or edge computing platforms, which enable real-time, intelligent, and autonomous access, collection, analysis, communications, and exchange of process, product and/or service information, within the industrial environment, to optimize overall production value”.* Fig. 18 shows the industrial revolution shift. The major goal of the IIoT is to increase productivity in industrial processes.

Fig. 19 describes the enabling technologies and domains in the IIoT (Čolaković and Hadžialić, 2018). First, the application domain provides visualization, application development, systems and devices monitoring, control and management services. It has three layers, IoT applications layer (smart homes, smart cities, and smart farms), architectural layer (software architectures, SOA and REST), and software and APIs layer (FreeRTOS, WebGL) that connect the application and middleware domains to keep the OS and applications up to date (Chalapathi et al., 2019).

The second, the middleware domain, provides data storage, data aggregation and processing, big data analysis, decision support, and ML services. It has three sub-layers, the cloud platform layer (OpenIoT, Sensorcloud, SmartThings, and Google cloud) provides on-demand computing resources utilizing the cloud, the data processing layer (BigQuery, Apache, and Storm) provides data mining and services, and data storage layer (MongoDB, HBase, and Hadoop) provides storage infrastructure/architectures services (Zezulka et al., 2018). The third, the networking domain provides seamless connectivity and data transfer services. It also has three sub-layers, the communication protocol layer consists of a system’s application (CoAp, MQTT), transport (TCP, UDP), and network protocols (IPv4, IPv6), which allows smooth communication. The network interface layer (RFID, NFC) provides technical standards for enabling seamless IoT integration. The adoption mechanism layer provides connectivity interfaces (RJ45, RS-232), gateways (ADLINK, BRC), and adaption methods (6LoWPAN, 6TiSCH) for the development of an IIoT system. The adoption layer includes standards, namely, 6TiSCH and IEEE 1095 that provide a more robust wireless connection, and the connectivity interface. The management layer provides all critical enabling technology standards for the development of an IIoT/IoT system. Finally, the object domain provides identification, sensing, actuating, data pre-processing, computation, and power supply services (Čolaković and Hadžialić, 2018). This domain contains three sub-layers, namely: hardware platforms, embedded objects, and mechanical–electrical parts. The hardware-platform layer contains hardware components like Arduino, Raspberry Pi, etc. The embedded-object layer integrates embedded sensors, displays, and firmware devices. The mechanical–electrical part layer offers batteries, micro-controllers, and digital signal processing services.

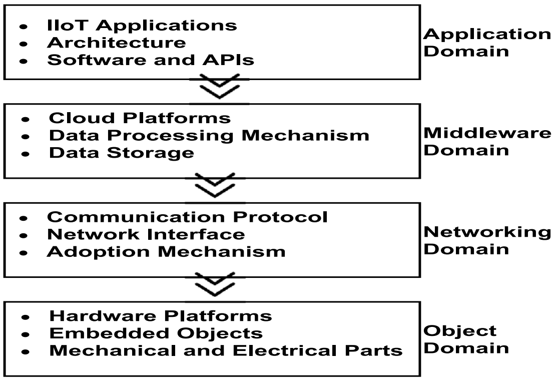


Fig. 19. Industrial IoT technology domains.

3. Machine learning preliminaries

Artificial Intelligence (AI) is the study of “intelligent agents”, or technologies that perceive their environment and then perform actions to optimize the chance of success at a certain goal. Machine Learning (ML) is a branch of AI that allows computers to learn without even being explicitly programmed. With the development of more powerful computer chips and microprocessors, researchers have discovered statistical models described as Artificial Neural Networks (ANNs). The ANN contains many layers (more than 3 layers) to handle increasingly complex information, resulting in the term “deep” learning. Different types of Deep Neural Networks (DNNs) are used in Deep Learning (DL) models (Xie et al., 2018). Fig. 20 presents the classification of major ML techniques.

Supervised learning, Unsupervised learning, Reinforcement learning, (Deep) neural network, and Federated learning are the major classification of ML techniques which are frequently used in Cloud integrated computing paradigms.

3.1. Supervised learning

According to NIST definition, Tabassi et al. (2019), *“Supervised learning is ML techniques in which training data are provided in the form of inputs labeled with corresponding outputs, and the model learns a mapping between inputs and outputs”.* It is a labeling learning technique that presents input data as well as proper output data to the machine learning model (Xie et al., 2018). The major supervised learning algorithms are K-Nearest Neighbor (K-NN), Decision Tree (DT), Random Forests (RF), Artificial Neural Networks (ANNs), Support Vector Machine (SVM), Naive Bayes (NB), Hidden Markov Model (HMM), Linear Regression (LR), logistic regression, etc. In the K-NN, data sample’s

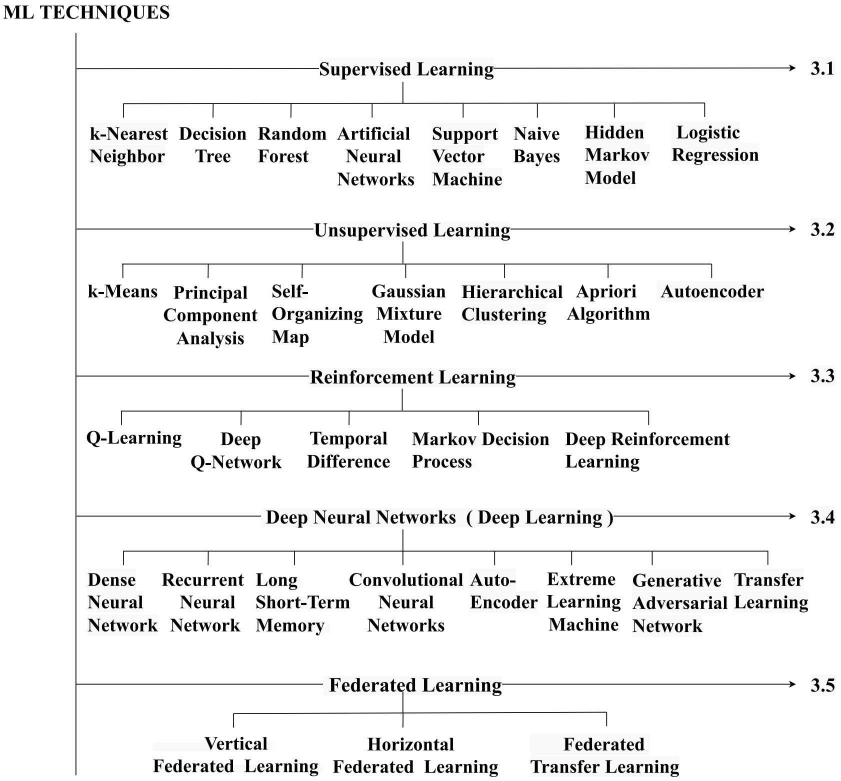


Fig. 20. Common machine learning techniques.

classification is defined by the k closest neighbors of the unclassified sample (Xie et al., 2018). DT is a classification technique that uses a learning tree to perform classification (Xie et al., 2018). By comparing the unlabeled sample's feature values to the decision tree nodes, the unlabeled sample can be classified using DT. To avoid decision tree overfitting and increase accuracy, the RF technique constructs each decision tree by randomly selecting a fraction of the feature space. With this, most voted class will be used to classify the sample (Xie et al., 2018). The ANN can analyze input information in the same manner as the human brain does (Xie et al., 2018). Neuron nodes are used to accomplish nonlinear and parallel complex calculations by using activation functions. In SVM, input vectors are mapped into a higher dimensional space (Xie et al., 2018). The SVM is used to discover a hyperplane that separates the classes inside the feature space by maximizing the margin between classes. The conditional probability is used in Naive Bayes (NB) theory to determine the probability of an event occurring that has prior knowledge of the factors concerning to the event (Xie et al., 2018). The samples are classified into the target classes having maximum posterior probability. The HMM is a probabilistic model that determines the joint probability of a set of random variables using observations and states (Xie et al., 2018). A system is modeled as a Markov process with unknown variables in HMM. Regression analysis refers to set of approaches for modeling the relation among one or more independent variables and a dependent variable (Xie et al., 2018). Linear Regression (LR) is used for the prediction of the value of the dependent variable by using the value of another independent variable (Yang et al., 2018). It predicts values in a continuous range by using two linearly correlated variables. The sigmoid function is used to transform a LR into logistic regression (Xie et al., 2018). The logistic regression is effective approach when the dependent variables is binary, but the independent variable are continuous in nature.

3.2. Unsupervised learning

According to NIST definition, Tabassi et al. (2019), *"Unsupervised learning is ML techniques in which training data are unlabeled inputs, and the model learns an underlying structure of the data".* It works with unlabeled data, and then act on these data without supervision (Xie et al., 2018). Fundamental unsupervised learning algorithms are k-Means, Principal Component Analysis (PCA), Self-Organizing Map (SOM), Gaussian Mixture Model (GMM), hierarchical clustering, apriori algorithm, AutoEncoder (AE) etc. The K-means clustering algorithm divides unlabeled data into different clusters and each data point related to the nearest mean cluster (Xie et al., 2018). PCA is a dimension-reduction approach that reduces a large set of attributes to a small set (Xie et al., 2018). It transforms the data using orthogonal linear transformations to a different coordinate system. A SOM is a group of neurons arranged in a 1-D or 2-D array, with weight vectors corresponding to points in an n-D feature space for each neuron (Xie et al., 2018). SOM is used for clustering and dimensionality reduction. The GMM is a probability distribution model for a finite mixture of probabilities. It represents a linear superposition of the combination of Gaussian distributions used for soft clustering purpose. Hierarchical clustering is a cluster analysis technique to create a hierarchy of clusters, by recursively splitting the data in a topdown or bottom-up manner (Xie et al., 2018). Apriori algorithm uses iterative approach to find frequently occurring common items, associations and correlations between the item sets (Xie et al., 2018). It applies "join" and "prune" methods to reduce the search space. The AE is a type of ANN that is designed to handle data encoding (mapping input data to a reduced feature space) as well as data decoding (reconstructing input data).

3.3. Reinforcement Learning

According to NIST definition, Tabassi et al. (2019), *"Reinforcement Learning (RL) is a reward-based policy for acting in an environment is learned from training data represented as sequences of actions, observations, and rewards".* It is a type of trial-and-error process in which the agent interacts with the environment in order to improve the long-term reward (Xie et al., 2018). Major RL algorithms are Q-Learning, Deep Q-network (DQN), Temporal Difference (TD), Markov decision process

(MDP), Deep Reinforcement Learning (DRL), etc. The Q-Learning finds the optimum Q-value function iteratively by optimizing action-selection policy using Bellman equation (Xie et al., 2018). It maximizes the reward by determining the next best action based on the current state. The DQN is a type of ANN that combines Q-Learning with DNN at scale to learn quality functions (Xie et al., 2018). The Q-function represents the expected reward by taking a specific action in the context of a particular state (Xie et al., 2018). MDP simulates a sequence of decisions taken by a learning agent as it interacts with its environment over a period of time to discovers a policy that maximizes cumulative rewards. Future states in MDP are entirely determined by the current state, rather than previous states. DRL improves sample efficiency during the learning by maximizing various pseudo-reward measures at the same time, and it predicts immediate reward by estimating the activation of a hidden unit (Xie et al., 2018). Furthermore, DRL uses DNN for value function and policy estimation purpose (Xie et al., 2018).

3.4. Deep Neural Network (Deep Learning)

According to Wang et al. (2020), *"Deep Neural Network (DNN) is a powerful class of ML that stack several layers of neural network models together"*. It consists of many layers, which connect neurons to solve a complex mathematical function. The input is processed by one layer, and it generates outputs for the next layer. The DNN are dense neural network, Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Convolutional Neural Networks (CNN), Auto-encoder, Extreme Learning Machine (ELM), Generative Adversarial Network (GAN), Transfer Learning (TL) etc. Dense neural network contains an input neuron with weights and an output neuron with an activation function (Wang et al., 2020). The input features are processed layer by layer across the network, with no direct link between non-consecutive layers in dense neural network. In RNN, the output at a given time step is determined by both the current and previous inputs (Wang et al., 2020). It contains a "memory" element that stores all necessary data about the operations. In the LSTM, a variable-length sequence is analyzed by progressively placing the new content into a single memory location (Wang et al., 2020). The gates (activation function layers) control the amount of information that forgot the irrelevant old information. LSTM also solves the long-term dependency issue present in RNN (Wang et al., 2020). CNN is a feed-forward-NN that consists of different convolutional layers stacked on top of one another, each capable of recognizing specific patterns in images (Wang et al., 2020). Auto-Encoder (AE) is a DNN that includes both an encoder and a decoder stage (Yang et al., 2018). The AE generates a latent representation, which is used as an input parameter for the subsequent auto-encoder layer. ELM is a generalized single-hidden-layer feedforward-NN with randomly generated hidden node parameters (input weights) and analytically obtained output weights (Wang et al., 2020). Generative models are used to model implicitly or explicitly underlying inputs and outputs distributions (Wang et al., 2020). The Generative Adversarial Network (GAN) is a implicit generative model that determines the probable distribution of real-world data samples and generates new samples from it (Wang et al., 2020). Transfer learning is an approach that aims to use information from one particular domain for assisting learning in a different domain. Some other key deep learning algorithms are Restricted Boltzmann Machine (RBM), Deep Belief Network (DBN) (Ahmed et al., 2021), etc. The Restricted Boltzmann Machine(RBM) is a generative-stochastic RNN capable of learning a probability-distribution from input information that consists of several visible and binary hidden units layers. The Deep Belief Network (DBN) contains stacked modules of RBM as middle layers and a classifier at the output layer.

3.5. Federated Learning

According to Lim et al. (2020), *"Federated Learning (FL) is a distributed ML technique that allows models to be trained on a large amount of decentralised data"*. FL is based on two main concepts: local computation and model transmission, which mitigates some of the privacy risks and costs associated with standard centralized ML methods. The customer's original data is saved locally and cannot be transferred or migrated. The device uses local data for local training purpose, after that it uploads the model for aggregation to the server. With this, server delivers the model update to the agents in order to meet the learning goal. FL protects user privacy by transmitting encrypted processed variables, thus, attackers are unable to access data source (Nguyen et al., 2021). It uses privacy protection techniques such as homomorphic encryption, secure aggregation, and noise addition of differential privacy to the model attributes. FL key applications are Google-keyboard search-suggestions, keyboard prediction and spotting, rank history-suggestions in browser, visual object-detection in computer vision (Fed-Vision), patient-clustering using electronic medical records, drug discovery, autism spectrum disorders using functional MRI, segmentation of brain tumor, named-entity recognition in medical (FedNER), recommendation of news in industries (FedRec), etc. (Nguyen et al., 2021).

Based on the data distribution characteristics of training set across the samples and feature spaces, FL is categorized in three major types, namely, vertical FL (feature-based), horizontal FL (sample-based) and Federated Transfer Learning (FTL). In horizontal FL, multiple learner agents work together for training the global FL models by having the same data feature-set and different sample-space on the local dataset (Nguyen et al., 2021). To determine the local update, the client trains their ML model locally that improve the security. It is used for wake-word-detection (voice assistant), Google keyboard, and drug detection in IoT application (Nguyen et al., 2021). In vertical FL, data sets may have identical sample-spaces and different data-features for learning the shareable ML model to improve the privacy. It improves privacy using encryption methods. It is used in shared ML model among entities in smart-city, smart-healthcare, smart-transportation, etc. The FTL transfers the attributes from various feature spaces into a single representation for training the data that is aggregated from different agents (Nguyen et al., 2021). The random-masks-encryption algorithm is applied to encrypt gradients-changes in the model update phase. This way it ensures security and data privacy during the model learning. Commonly, FTL used in disease diagnosis purpose in IoT that improves diagnosis performance (Nguyen et al., 2021).

4. ML techniques in integrated cloud computing paradigms

This section elaborates the integration of various emerging cloud computing paradigms and explores the ML techniques applied in integrated architectures such as Cloud-ML, Edge-Fog-IoT-DL, Fog-Cloud-ML, Mist-Cloud, IoT-ML, SDN-IoT, Digitaltwin-IIoT, and IIoT-Fog. The integration of Cloud-ML focuses on the applications of ML in Cloud computing. The parameters of the study are properties, services, deployment models, QoS parameters, security issues, and challenges related to cloud computing. For machine learning, the parameters are types of ML techniques, challenges, metrics/parameters, and different objectives. For the Edge-Fog-IoT-DL integration, the study parameters are the type of algorithms, the dataset used, issues, accuracy, the domain of computing, and devices used with the target objectives. In particular, for the use of ML and DL in edge/fog computing, the study of parameters are packages, libraries, models, frameworks, devices, and types of systems. Further, this study also includes the ML algorithms, data-sets/models, QoS parameters, problems addressed, and the type of domains. For the integration of Fog-Cloud-ML, this study focuses on the Cyber–Physical System (CPS) applications. The parameters are the type of ML algorithms, CPS domain, applications, method/tools, data-set, and the attribute used in the related research study. For Mist-Cloud integration, this study majorly focuses on the parameters such as resource constraints, scalability, privacy, security, interoperability, intelligence levels, and offline capability.

For the integration of IoT-ML, this study focuses on IoT devices identification and smart city applications. The study parameters include the types of IoT devices, ML techniques, feature extraction, and accuracy. Further, the smart city applications with IoT-ML and cloud are also explored, for the same, the parameters include smart city application domains, approaches, QoS parameters, data-sets, and the related issues. The SDN-IoT integration briefly summarizes the application of IoT in SDN backed with cloud, and QoS/QoE prediction. The study parameters are the ML method types, data-set, focused research areas, related applications, and software/hardware used for the experiments. For QoS/QoE prediction, the literature study includes methods and tools, data-sets, objectives, and future research directions. The Digitaltwin-IIoT integration is the new integration concepts where digitaltwin is backed by the cloud, and the IIoT is the integration of IoT and AI/ML. For this domain, the parameters for the literature study are algorithms, architectures, computational processing tools, data exchange protocols, data formats, modeling, and simulation tools. The IIoT-Fog integration is the newly emerging domain of research in which fog computing is backed by cloud technology. In this domain, the study explores the current project descriptions and typical integration of fog computing in the IIoT domain. The parameters of the study are architecture and application domain with attributes such as resource allocation, latency, energy, power, and bandwidth.

4.1. Cloud-ML

The integration of cloud computing and ML benefits in terms of increasing throughput, minimizing delay, improving Quality-of-Service (QoS), and minimizing response time for cloud applications.

This sub-section details the ML techniques, challenges, parameters, and objectives from various research articles in the domain of cloud computing. The summary for the same is presented in Table 2. The key ML techniques which are widely used in cloud computing to handle several issues are explored as nearest neighbor, Naive Bayes (NB), Decision Tree (DT) (Liao et al., 2009), federated learning (Lim et al., 2020), Deep Learning (DL) (Wang et al., 2020), Support Vector Machine (SVM) (Bankole and Ajila, 2013), Artificial Neural Network (ANN) (Kundu et al., 2012), statistical ML (Bodík et al., 2009), modified Prediction of Query Run-time2 (PQR2) (Gupta et al., 2008), Linear Regression (LR), time-series analysis (Jiang et al., 2013), reinforcement learning (Xu et al., 2012), parallel Q-learning (Barrett et al., 2013), and Genetic Algorithm (GA) (Huang et al., 2013).

As the mobile data traffic is growing in cloud computing with exponential growth due to enormous growth in IoT devices. Caching techniques are used in Fog-Radio Access Networks (F-RANs), cache-enabled Cloud-Radio Access Networks (C-RANs), and other related cellular networks (Tandon and Simeone, 2016). It also handles mobile data traffic and reduces back-haul traffic. It stores the data near the edge utilizing the storage devices and improves the spectrum efficiency, energy efficiency, transmission delay, and also removes duplicate data transmission (Tandon and Simeone, 2016). Singular Value Decomposition (SVD) and Collaborative Filtering (CF) are also experienced in wireless networks to develop edge caching strategies to improve the wireless connectivity in cloud paradigm (Li et al., 2018b). In particular, ML predicts content popularity and rates by using social networks, locating systems, and assessing user velocity data.

In Mobile Edge Computing (MEC), the data sources are located outside the cloud, and the computing and storage functions are performed near to the data source. Federated Learning (FL) techniques are used in MEC for learning tasks. The FL is a decentralized ML architecture suggested by Google (Lim et al., 2020). The end devices employ local data for training ML models and send models updates for aggregation to the server. It applies collaborative training for ML models and uses mobile edge network optimization for DL (Lim et al., 2020). There are several issues present in FL, namely, resource allocation, privacy, security, and communication cost. The poor resource allocation happens due to the data quality control of heterogeneous devices, processing capacity, and engagement. It lacks in privacy and security due to the existence of malicious users or servers in the system and can use shared attributes to predict the characteristics of other participants (Wang et al., 2020). It increases the communication cost because of the high dimensionality of model updation and the low communication bandwidth of the devices.

Deep learning (DL) is a multi-layer ML architectural model used to extract accurate data from IoT-enabled devices that are placed in a complex-networking environment (Wang et al., 2020). DL techniques i.e., Deep Neural Networks (DNN), Convolutional Neural Network (CNN), Deep Q-Learning (DQL), Deep Reinforcement Learning (DRL) are applied in edge computing to improve the prediction accuracy. Through an iterative process, Q-learning is used to reduce this cost, and the simulation results show that the approach reduces backhaul traffic in macrocells and enhances user QoE in wireless network. The techniques DNN and Reinforcement Learning (RL) applied to improve the routing performance by predicting the whole path (Sun et al., 2019b). It is used to increase performance on edge devices by adaptive edge maintenance and management policies. DL improves efficiency by reducing latency and provides reliable intelligent services. Machine learning techniques such as naive bayes (Liao et al., 2009), decision tree (Jiang et al., 2013), SVM (Bankole and Ajila, 2013), linear regression (Xiong et al., 2011), time-series analysis (Kundu et al., 2012), etc. are widely used for configuration for memory prefetchers, prediction of the number of requests, resource demand prediction, adjust allocations, profit maximization, performance and resources allocation in the wireless network. SVM is the best-suited algorithm for unbalanced data sets and cloud intrusion detection. Many sensors are used for sensing the environment and generating a bunch of the data in an IoT environment (Kundu et al., 2012). For the proper functioning of the network, such sensor devices should be identified. ML algorithms like SVM, GMM, KNN, RF, etc. are applied to identify devices, which can extract device features (Tuama et al., 2016). In cloud computing, the ANN allows extensive operating capabilities and provides a quick and easy method of computing (Chen et al., 2019). The ANN is an ML technique that contains interconnected and multi-level neurons. The ANN training discovers several patterns which can be used to detect intrusion for cloud security. Cloud computing uses statistical ML with computer algorithms to provide optimal control for data centers (Bodík et al., 2009). The combination of statistical techniques and machine learning gives the power to analyze various kinds of data. Statistical models help to find the relationship among variables. It makes inference and significance of relationships, and then use them for the prediction.

Cloud computing uses databases to store different types of data/information. To retrieve the data, the query is processed in the database. This query takes some time to run/execute. The execution time of a query becomes the key factor for workload management. Typically, this workload management is done by efficient control policies, scheduling, resource allocation, and load balancing policies. Predicting Query Run-time (PQR) tree modeling uses ML techniques to predict query execution time (Gupta et al., 2008). PQR tree model can be improved by learning data skew, varying workloads, and applying time of day patterns in the future. To meet service demand in cloud computing, there is a need for real-time provisioning, service availability, and performance data. For this purpose real-time auto-configuration of VMs and appliances is necessary. To address this issue Unified Reinforcement Learning (URL) is proposed (Xu et al., 2012). URL uses App-Agent and VM-Agent-based approaches for tuning application parameter settings and adjusting VM configuration. However, this approach can be further improved by using multiple physical machines for VM clusters configuration and applying distributed RL

Table 2
ML technique in cloud network.

Ref.	ML technique	Challenges	Metrics/parameters	Objective
Li et al. (2018b)	ML (collaborative filtering, SVD)	Reducing costs & raising revenue for operators, big data processing, caching in mmWave communications, coded caching, network slicing, mobility-aware caching	Throughput, backhaul cost, power consumption, network delay	Edge caching
Lim et al. (2020)	Federated learning	Communication costs, resource allocation, privacy, security, dropped participants, unlabeled data, interference, asynchronous FL	–	Edge network
Wang et al. (2020)	DNN (CNN, DQL, DRL)	Non-IID training data, limited communication, unbalanced contribution, privacy, security	Communication rounds, scalability, accuracy, latency, communication cost, computation, convergence rate, communication load, fairness, latency, resource overhead, inference scalability, energy efficiency	Edge computing
Sun et al. (2019b)	ML (Q-Learning, DNN, RL)	Heterogeneous backhaul or fronthaul Management, infrastructure update, network slicing, standard datasets, transfer learning	Transmission power, data rate, interference, throughput, spectrum utilization, spectral efficiency, energy consumption, backhaul parameter, latency, cost	Wireless network
Chen et al. (2019)	ANN	Mobile edge caching & computing, wireless virtual reality, internet of things, multi-RAT, unmanned aerial vehicles, diversity of IoT devices, large scale & dynamics of IoT system	Content correlation, caching efficiency, hit ratio, resources management, path, channel, handover, content, computation, demand, LoS link	Wireless network
Liao et al. (2009)	ML (nearest neighbor, naive bayes, decision tree, SVM)	Parameter value optimization, data Prefetching, system performance	Temporal/spatial correlation, sampling, frequency of instructions, performance tuning, prefetcher configuration, hardware events	Configuration for memory prefetchers
Bodík et al. (2009)	Statistical ML (linear & LOESS regression)	Optimizing control parameters, model management, complex real-life workload or performance, internet datacenters	Usage patterns, hardware failures, application changes sharing resources, queuing models, energy efficiency, G/G/1 queues, concurrency levels and workload, requests, SLA, threshold, workload, number of servers, performance, latency, gain controller	Optimal control for data centers
Gupta et al. (2008)	Supervised ML (modified PQR2)	Sudden changes in workloads management, discover time ranges, time of day patterns, data skew	Query execution time, query plan, system load, span, query cost estimation, predict execution times of tasks and resource consumption	Workload management
Jiang et al. (2013)	ML (linear regression) & time series analysis	Resource Autoscaling, true elasticity, cost-effectiveness in the pay-per-use, migration of web applications, queuing models, multi-tenant VMs	Web requests, resource demand	Predict number of requests
Islam et al. (2012)	Error correction NN and linear regression	On-demand allocation, usage prediction, dynamic provisioning, trade-off b/w SLAs and constraints (VM setup overhead, cost effectiveness)	Root-mean-squared-error, PRED, number of samples, R2 Prediction, mean-absolute-percentage-error, benchmark duration, sampling interval, sampled resource, max & min EC2 instances	Predict resource usage patterns
Gong et al. (2010)	Statistical ML (PRESS)	Reduce resource waste and SLO violations, large-scale cloud computing infrastructures	Fine-grained dynamic patterns, resource prediction accuracy, cyclic and non-cyclic workloads, load traces, over-under estimation errors, SLO violations rate, penalty functions	Predict resource demands
Bankole and Ajila (2013)	ML (SVM, NN, LR)	VM provisioning, public cloud infrastructure, database server	CPU utilization, response time, throughput, mean absolute percentage error, root mean square error, percentage of observation	Resource demand prediction
Xiong et al. (2011)	ML (regression tree & boosting)	Virtual resource management for database systems, optimal resource allocation	Service latency, response time, CPU share, memory size, workload, replica number, SLA penalty cost, arrival rates, infrastructure cost	Adjust allocations and maximize profits
Xu et al. (2012)	Reinforcement learning	Configurations of VMs, appliances in clouds, distributed RL	CPU time, memory size, network bandwidth	Auto-configuration and optimal configurations for VMs

(*continued on next page*)

Table 2 (*continued*).

Ref.	ML technique	Challenges	Metrics/parameters	Objective
Barrett et al. (2013)	Parallel Q-learning	Optimizing resource allocation, live virtualized test-bed	User requests, number of VMs, UTC time, response time, arrival rates	Improving convergence times, auto-scale resources
Kundu et al. (2012)	ANN & SVM	Allocation and resizing of resources, storage I/O latency	MIMO queue model, CPU-memory limit, I/O latency	Performance and resources allocation
Huang et al. (2013)	GA & SVR	Internet, auxiliary memory, resource optimization module	Response time, CPU, RAM	Reduce service response times

algorithms. IaaS, in the clouds, scales the resources on-demand. The dynamic scaling gives unpredictable and different results which are executing on IaaS clouds.

To determine optimal scaling policies, a temporal difference RL (Q-learning) technique is used. Parallel Q-learning (PQR) mitigates the problem of the curse of dimensionality in RL (Barrett et al., 2013). PQR maintains Q-matrix (for the convergence), rewards, and strategies throughout the training process. Q-Matrix offers a strategy for state transition. In cloud computing, PQR supports the scalability of applications by making optimal resource allocation decisions. For future work, this approach can be integrated with the live virtualized test-bed environment. Resource Management is the key issue in cloud Computing. When the resources are limited, there is a need for an optimal resource allocation strategy. To address this, Support Vector Regression (SVR) prediction method was proposed (Huang et al., 2013). SVRs reduce service response time by measuring response time in the next period. After observing the current status of VMs, the resources are redistributed. For the reallocation of resources, a resource dispatch mechanism is applied using GA (Huang et al., 2013). By utilizing other types of resources, such as the internet and auxiliary memory, will make it a real-world scenario. The convergence speed of GA is also enhanced by predicting and reducing the processing time.

This discussion concludes that the application of various ML techniques in cloud-integrated emerging technologies such as edge caching, optimal control for data-centers, workload management, resource demand prediction, auto-configuration of VMs, and resources management are the key domains of the study. Further, with this study, it is also inferred that the researchers are majorly utilizing ML, DL, and GA techniques in cloud computing and other integrated emerging technologies.

4.2. Edge-Fog-IoT-DL

This sub-section presents an integrated study of edge computing, fog computing, IoT, and deep learning as the emerging cloud computing paradigm. It also elaborates edge computing using ML and DL with various systems, packages, libraries, models, frameworks, devices used in fog computing, IoT, and edge computing.

4.2.1. Machine learning techniques in edge and fog computing:

This sub-section briefs about the machine-learning techniques used in the edge/fog computing domain. The relevant summary of the state-of-the-art algorithms is tabulated as shown in Table 3. The major algorithms available in the literature include Support Vector Machine (SVM), KNN, hidden Markov model, linear SVM, collaborative filtering, stacked auto-encoder, linear regression, bayesian networks, and k-means clustering for various fog/edge computing-based applications.

For clustering analysis, a set of certain patterns must be separated into different clusters, and the cluster contains similar members. Some of the patterns are also very different from the clusters, and these patterns are identified in the data-set as the outliers or noises that are ignored during the analysis. In one cluster, the patterns are identical and different from those in another. It is applied in fog computing to identify Parkinson's disease for smart telehealth (Borthakur et al., 2017). Markov's model is a probabilistic model which uses the distribution of probabilities for all possible outcomes. It also maps observation and hidden attributes. The decision-making process in Markov makes sequential decisions with immediate and rapid rewards. It is applied for the detection of anomalies in edge computing for image recognition purposes (Drolia et al., 2017). The SVM is a sort of maximum margin classifier that uses a hyper-plane to distinguish positive and negative label data. It is used to detect anomalies and provides cloud security (Azimi et al., 2017). Linear SVM is a low-cost supervised machine learning binary classifier that separates binary hypotheses. It is applied in edge/fog computing, to improve healthcare and security (Zissis, 2017). Collaborative filtering has been used to make predictions from user rating (Wang et al., 2019b). It is an unsupervised ML algorithm that learns an unknown feature set. It gathers people's feedback on various items and then makes personalized recommendations based on past preferences or history profiles. In this way, it finds similar people and recommends what they like. It is applied in MEC, for service recommendation and solving problems with overload (Zissis, 2017).

Linear Regression (LR) takes the sensor's data and uses the information provided by neighbors to assist the sensor node in locating itself (Yang et al., 2018). For large data sets, LR is computationally costly, thus it is best to outsource the processing to the cloud. It is used to preserve the privacy of sensors as well as data that has been tampered with on a cloud server. It is applied in fog computing to provide privacy protection. Bayesian network is a probabilistic graphical model containing random variables and conditional dependencies between them. It is used to predict latency with more accuracy by maximizing conditional probability. It has been applied in MEC to calculate path latency traces with high accuracy and better performance (Hogan and Esposito, 2017). Path latency measurements are discretized using K-means clustering, which helps to eliminate small dependencies by removing variation from the same cluster points in MEC (Hogan and Esposito, 2017).

From this discussion, it is observed that in edge/fog computing, there is a need to improve the accuracy of ML algorithms. The issues related to latency, bandwidth consumption, limited computational capacity, and cognitive aspects are required to be addressed.

4.2.2. Role of machine learning and deep learning in edge computing

This sub-section discusses systems, packages, libraries, models, frameworks, and devices used for edge computing as shown in Table 4. In edge computing, there are several actuators, sensors, and IoT devices that work together. Without edge computing, the IoT would be reliant on the cloud for connectivity and computing services. Sending data back and forth between an IoT device and the cloud might cause slower response time and reduced operational efficiency. When a network connection is lost, the system must be available to work offline. The main edge computing systems are Azure IoT edge, AWS IoT Greengrass, and cloud IoT edge. In 2017, Amazon announced Amazon Web Services (AWS) Greengrass, which allows devices to perform local actions on the data generated by AWS. Azure IoT edge enables cloud solutions to be migrated to IoT devices and offers hybrid Cloud-Edge analytics services. Azure IoT edge offers Azure stream analytics, Azure functionalities, and Azure machine learning. Using Google's Artificial Intelligence (AI) products like TensorFlow lite, edge Tensor Processing Unit (TPU), and cloud IoT edge enhances the processing capacity of the Google cloud and extends ML power near edge devices.

Table 3
Edge computing using ML.

Ref.	Objective	Algorithms	Dataset	Issues	Accy	Domain	Devices
Borthakur et al. (2017)	Identify Parkinson’s disease	Clustering	Speech data	Bandwidth, resource constrained devices, stringent low latency	–	Smart telehealth (fog computing)	Smart watches
Drolia et al. (2017)	Image recognition	Markov model	Images	Bandwidth consumption, latency edge server usage, accuracy	90%	Edge computing	Smartphones, drones, cars, edge servers
Azimi et al. (2017)	Detection of arrhythmia	Linear SVM	ECG	Limited computational capacity, accuracy, adaptability, sensors’ battery life	93.6%	Healthcare IoT (fog computing)	Single-channel ECG, microcontroller unit, sensor node, jetson-TK1, HP Compaq 8200 Elite
Wang et al. (2019b)	Service recommendation	Collaborative filtering	User mobility, QoS data	Accuracy, large deviation, cognitive aspects	64.4%	Mobile edge computing	Edge servers
Zissis (2017)	Anomalies detection	SVM	Sensor data	Abnormal vessel movements	90%	Security (edge computing)	Rasberry Pi 2 900 MHz quad-core ARM Cortex-A7 CPU, AIS base receiver
Abeshu and Chilamkurti (2018)	Distributed attack detection	Deep learning (stacked autoencoder)	Traffic data (NSL-KDD)	Accuracy, scalability, false alarm rate	99.2%	Cyber security (fog computing)	IoT devices
Yang et al. (2018)	Privacy protection	Linear regression	Sensor data (REDD, MHEALTH)	Data aggregation, communication efficiency, computation overhead, single aggregation, privacy, utility	90%	Fog computing	Sensors, fog nodes, fog center, cloud server
Hogan and Esposito (2017)	Traffic engineering	Bayesian Networks, k-means clustering	Path latency traces	Latency, bandwidth, queue size estimation	(80%–90%)	Mobile edge computing	–

Table 4
ML/DL systems, packages, libraries, models, framework, devices on edge computing.

Systems (Liu et al., 2019a)	Packages (Liu et al., 2019a)	Libraries (Murshed et al., 2019)	Models (Murshed et al., 2019)	Frameworks (Murshed et al., 2019)	Devices (Murshed et al., 2019)
AWS IoT Greengrass, azure IoT edge, cloud IoT edge	TensorRT, CoreML, PyTorch, MXNet	TensorFlow, DL4J, SNPE, MACE, FANN, Paddle-Mobile	YOLO, DetectNet, MobileNet, GoogLeNet, DeepSense, AlexNet, LSTM, VGG, Deepface, SENNA, Faster R-CNN, VGG16, OpenALPR	TensorFlow Lite, Caffe2, Caffe2Go, Core ML2, ML Kit, AI2GO, DeepThings, DeepIoT, DeepCham, SparseSep, DeepX, edgent, daBNN, CONDENSA	NVIDIA Jetson TX1, TX2, NANO, google coral dev board, intel Movidius NCS, ARM ML, RISC-V GAP8, OpenMV Cam, BeagleBone AI, SparkFun edge, raspberry Pi

The packages used in edge computing are TensorRT, CoreML, PyTorch, and MXNet. NVIDIA introduced TensorRT, a high-performance deep learning inference SDK that enables minimal latency and maximizes throughput in deep learning applications. Apple launched CoreML, which is used to incorporate a trained machine learning model into Apple products. Deep learning models, tree ensembles, generalized linear models, and SVMs are all supported by it. It also supports an iOS-based framework that is used to run ML on edge devices. Facebook offered PyTorch as an open-source machine learning platform for research purposes. It has GPU acceleration and DNN for tensor processing. The Carnegie Mellon University (CMU) and the University of Washington developed MXNet, which supports Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN) networks (Chen et al., 2015). It is suitable for heterogeneous distributed systems because it is a memory and computation efficient system. This deep learning framework was adopted by Amazon. Apache MXNet is an open-source framework to train DNN and to deploy them on edge devices that are resource-constrained. It provides public cloud interaction and distributed ecosystems.

The libraries available to support edge computing are TensorFlow, DL4J, SNPE, Paddle-Mobile MACE, FANN, etc. Tensorflow is a deep learning open-source framework developed by Google and used in cloud data centers. Deep Learning for Java (DL4J) is a java-based deep learning library. Qualcomm’s SNPE (Snapdragon Neural Processing Engine) SDK is a useful platform for Snapdragon edge devices. SNPE also supports TensorFlow models. Mobile AI Compute Engine (MACE) is a mobile heterogeneous computing-optimized deep learning inference framework (Deng, 2019). MACE improves the speed of the application, memory utilization, library footprint, and UI responsiveness. FANN (Fast Artificial Neural Network) is an open-source cross-platform programming library for building multilayer feed-forward ANN. Paddle-Mobile is part of the PaddlePaddle deep learning project, which focuses on embedded systems.

The models used in edge computing are YOLO, DetectNet, MobileNet, GoogLeNet, DeepSense, AlexNet, LSTM, VGG, Deepface, SENNA, Faster Region-based Convolutional Neural Networks (R-CNNs), VGG16, and OpenALPR. MobileNets is a tiny, efficient, low-computation, low power, and low-latency CNN model that minimizes the number of parameters (Howard et al., 2017). It is a streamlined framework for creating lightweight DNNs that leverages depthwise separable convolution. SqueezeNet is a tiny CNN architecture with 50 times fewer attributes than AlexNet (Iandola et al., 2016). The frameworks used in edge computing are TensorFlow Lite, Caffe2, Caffe2Go, Core ML2, ML Kit, AI2GO, DeepThings, DeepIoT, DeepCham, SparseSep, DeepX, edgent, daBNN, and CONDENSA. TensorFlow Lite is a lightweight solution for on-device inference for mobile and edge computing devices. It can run on multiple CPUs and GPUs, it is ideal for distributed machine learning methods. Facebook introduced the Caffe2 deep learning framework. Caffe2Go is a deep learning model development framework built on top of Caffe2. Caffe2Go decreases the model size by using fewer convolutional layers. Google introduced the ML Kit framework as a mobile SDK framework. It is used for image labeling, text recognition, bar-code scanning, and smart response. Core ML is a machine learning framework that can be used across Apple products to perform fast real-time prediction. The Xnor proposed AI2GO, which can execute deep learning models on low-resource devices and is utilized for on-device interfacing. DeepThings can execute CNN-based inference on edge devices and has a small memory footprint because it uses Fused Tile Partitioning (FTP), this partitions the CNN model and distributes the partition parameters and weight to edge devices (Murshed et al., 2019). DeepIoT is utilized to compress the DNN model without affecting performance while reducing execution time and energy consumption. DeepChamp allows users to build deep learning models on edge computing devices and to recognize objects on android devices (Murshed et al., 2019). SparseSep is a large-scale DL model optimization tool for wearable devices that run on ARM Cortex, NVidia Tegra K1, and Qualcomm Snapdragon, processors. DeepX is a software accelerator that reduces resource overhead by creating unit-blocks from the deep model network, which are then processed on mobile devices using heterogeneous processors (Murshed et al., 2019). To improve DNN inference, edgent is utilized to partition DNN computations amongst small mobile and edge devices and apply early exit at middle DNN layers. On ARM devices, daBNN is used to create a binary neural network. To reduce convolution cost and speed up the inference, it uses a bit-packing method and binary direct convolution techniques. The CONDENSA is a programmable system for compressing DL models, which uses bayesian optimization to derive hyperparameters and can reduce memory footprint and execution time (Murshed et al., 2019).

The following are the major devices employed at the edge: RISC-V GAP8, SparkFun edge, NVIDIA Jetson TX1, TX2, NANO, Google Coral Dev Board, ARM ML, OpenMV Cam, Intel Movidius NCS, Raspberry Pi, and BeagleBone AI. Google Coral Dev Board was created to perform machine learning on the edge TPU (Co-processor) (Deng, 2019). It has a 40-pin GPIO header on the baseboard for IoT devices, and the system-on-module runs Linux on a Cortex-A53 processor. Google, SparkFun, and Ambiq created SparkFun edge to deliver real-time audio analysis, voice, and gesture recognition at the edge (Deng, 2019). As a visual processing unit, Intel Movidius is used in drones and intelligent security cameras. To construct ML, computer vision, image classification, and semantic segmentation, BeagleBone AI includes the Texas Instrument deep learning ML API (Deng, 2019). NVIDIA Jetson TX1, TX2, NANO, etc. are NVIDIA Jetson-powered embedded computing boards and AI computing devices that are designed to process complicated real-time data and intelligent systems on edge devices. The OpenMV Cam is a compact camera board with a low-power ARM Cortex-M7 processor for running machine learning-based computer vision algorithms (Deng, 2019).

This discussion presents the development of recently used tools in the edge computing domain. A variety of packages and libraries are used, however, there is a lack of models and frameworks that support integrated multi-platforms.

4.2.3. Internet of things and deep learning approaches

This subsection discusses ML approaches, data-set/model, algorithms, solution parameters, issues/problems, domain, working area, and Deep Learning (DL) with cloud integration as summarized in Table 5. Deep learning is applied for following IoT areas such as massive IoT, critical IoT, enhancing IoT security and privacy, AI modeling using IoT data pre-possessing, smart home, smart agriculture, and interactive art applications (Saleem and Chishti, 2019). It is also used to extract temporal relationships for IoT data. For image recognition and time-series analysis in IoT applications, ANN, CNN, Recurrent Neural Network (RNN), LSTM are popular DL techniques (Saleem and Chishti, 2019). In DL, several optimization strategies have been developed for fast solution finding such as Stochastic Gradient Descent (SGD), AdaBoost (Adam), and Nesterov-accelerated Adaptive Moment Estimation (Nadam). Deep learning techniques Restricted Boltzmann Machine (RBM) are useful in recognizing human activity in smart home applications and power demand forecasting in smart grid (Saleem and Chishti, 2019). DBNs are applied in electricity utilization of AC in smart homes and optimal load balancing, meteorological time series analysis in the smart city. CNN has been used in voice pathology detection and real-time ECG monitoring (Saleem and Chishti, 2019). RNNs used in early prediction of crop yield and traffic flow forecasting (Wang et al., 2018b). LSTMs are used in the prediction of urban traffic congestion and remaining life prediction of machines (Saleem and Chishti, 2019). Auto-Encoders are used in locomotive activity recognition and wind power prediction. Generative Adversarial Networks (GANs) are used in path planning and to generate multi-label discrete patient records (Saleem and Chishti, 2019).

As shown in Table 5, for resource-constrained devices in IoT ProtoNN and Bonsai ML approaches are used (Gupta et al., 2017). ProtoNN is a compressed and accurate model that uses kNN to provide accurate prediction and classification for resource-constrained devices (Gupta et al., 2017). It learns fewer prototypes for the entire training set, generates a sparsely low-dimensional projection of given data, and performs jointly discriminative learning of the projection. Bonsai is a tree-based learner for resource-constrained IoT devices that improves prediction accuracy while reducing the model size and prediction costs (Kumar et al., 2017). For better bandwidth utilization in IoT, Gradient Descent (GD) based techniques are used (Wang et al., 2018b). A distributed learning approach based on GD evaluates the convergence rate of distributed GD to determine the local update and global parameter aggregation for minimizing the loss function. To reduce the model size, optimize memory utilization and reduce computation cost ShuffleNet, MobileNets, and EfficientNet are used. ShuffleNet is a computationally efficient CNN architecture for mobile devices that provides high accuracy while reducing computation costs (Zhang et al., 2018b). It uses depth-wise separable convolutions, group convolution, and channel shuffling. EfficientNet has been utilized to increase transfer learning performance, reduce memory need, and make the quick inference. It increases the network bandwidth width, depth, and input resolution of the CNN model (Tan and Le, 2019).

The distributed DNN architectures such as mobile deep inference, Mapped DNNs, and MoDNN are used to improve response time, energy-saving, and reduce communication cost. In the Edge-Cloud-based architecture, mobile deep inference delivers vertical distributed DNN inference is utilized to make decisions about where and which model to execute (Ogden and Guo, 2018). To deliver fast inference, mapped DNNs provide distributed deep learning inference over distributed computing hierarchy (Teerapittayanon et al., 2017). The distributed mobile computing system MoDNN is utilized to deploy DNN (Mao et al., 2017). For unstable latency problems in IoT, DNN computation partitioning, dynamic distribution of DNN, Faster DNN inference, etc. are utilized (Hadidi et al., 2019). DNN computation partitioning is used to reduce latency and deploy DNN computation across IoT devices (Hadidi et al., 2019). Dynamic distribution of DNN used to harvest computing power. It is used in distributed robotic systems with collaboration and allows dynamic, real-time recognition in low-power robot system (Hadidi et al., 2018). Faster DNN inference uses a crowd-sourcing approach that takes advantage of robot's idle computational power and combines it with latency prediction and optimal partition selection algorithm.

For optimal bandwidth utilization, a layer fusion scheme is proposed (Stahl et al., 2019). In DNN inference, the layer fusion approach reduces communication time and bandwidth. The edgeNet (Pradeep

Table 5
ML and DL algorithms in edge computing.

S.No.	Learning approach	Dataset/model	Algorithms	Solution parameters	Issues/problems	Domain	Working area
1	ProtoNN (Gupta et al., 2017)	Character recognition, Eye, MNIST, ALOI	KNN, gradient descent	Lower storage, minimal working memory	Poor accuracy, large model size, Prediction in real time	Deep learning (IoT)	Resource constrained devices
	Bonsai (Kumar et al., 2017)	Chars4K, WARD, RTWhale, CUReT	Tree-based	Minimizing model size, prediction costs, fit in KB of memory, lower battery consumption	Latency, bandwidth, privacy, energy	Deep learning (IoT)	
2	Gradient-descent (Wang et al., 2018b)	MNIST	SVM, CNN, linear regression, k-means	Optimum ML models, minimize loss function	Resource limitation of raspberry Pi	Deep learning (IoT)	Bandwidth
3	ShuffleNet (Zhang et al., 2018b)	ImageNet	CNN	Reduce computation cost, speedup	Small networks	Deep neural networks	Reduce model size, less memory, computation
	MobileNets (Howard et al., 2017)	ImageNet	Depthwise separable convolutions	Small & low latency models	Do not use side heads, label smoothing	Mobile and embedded vision	
	EfficientNet (Tan and Le, 2019)	ImageNet, CIFAR-100, Birdsnap, stanford cars, flowers	CNN (compound scaling method)	Better accuracy and efficiency	Balance dimensions of network width or depth or resolution	DNN	
4	Mobile deep inference (Ogden and Guo, 2018),	Flower images	Deep Neural Network	Accuracy and time, improving response time and saving energy	Storage constraints, hardware heterogeneity	Edge computing	Distributed DNN architectures
	Mapped DNNs (Teerapittayanon et al., 2017),	Multiview multicamera	Deep Neural Network	Fast & localized inference, enhance sensor fusion, system fault tolerance, data privacy, Communication cost	Latency issues and privacy, Reduced system accuracy	Edge computing (cloud computing)	
	MoDNN (Mao et al., 2017)	VGG-16	Deep Neural Network	Minimize non-parallel data delivery time, wakeup time and transmission time	Only for mobile platforms	Distributed mobile computing system	
5	Partitioning of DNN computations (Hadidi et al., 2019)	AlexNet	Coded distributed computing (DNN)	Robustness and close-to-zero recovery latency, tolerate failures, straggler problem	Unstable latencies, intermittent failures	IoT	Latency
	Dynamic distribution of DNN (Hadidi et al., 2018),	AlexNet, VGG16, HMDB	Deep Neural Network	Performance, profiling	Computationally intensive and resource hungry, heavy real-time computations	IoT	
	Faster DNN inference (Hadidi et al., 2018)	AlexNet, MNIST, CNDS, VGG	Deep Neural Network (Crowdsourcing)	Computing power of robots, latency speedup, model partition	Huge computational complexity and memory consumption, unstable or unacceptable network latency	Robot intelligence (cloud computing)	
6	Layer fusion scheme (Stahl et al., 2019)	YOLOv2	Integer linear programming, deep learning	Layer fusion, optimize memory, computation, demands	DNN cannot store the large weight data on a single device	Edge computing (IoT)	Bandwidth

(continued on next page)

et al., 2018) is a set of embedded FPGA and SqueezeNet CNN models that boost performance on edgeAI by reducing resource constraints. Reduced memory footprint (Gupta et al., 2015) uses a 16-bit fixed point for the weights to train the neural network with stochastic rounding for an energy-efficient hardware accelerator. To reduce communication cost in distributed training Federated-learning-with-Client-Selection (FedCS) protocol, edge aggregation, federated learning, edge Stochastic Gradient Descent (eSGD) (Lin et al., 2017), and deep gradient compression (Tao and Li, 2018) have been proposed. FedCS maintains clients regardless of resource restrictions and proposes a client

Table 5 (*continued*).

S.No.	Learning approach	Dataset/model	Algorithms	Solution parameters	Issues/problems	Domain	Working area
7	EdgeNet (Pradeep et al., 2018),	Image-Net	CNN on embedded FPGA	Mitigate resource constraints, accelerate SqueezeNet, improve accuracy	Limited battery power, computation resources	Embedded vision, edge AI	Weight quantization and model compression
	Reduced memory footprint (Gupta et al., 2015)	MNIST, CIFAR10	Stochastic rounding	Computational performance and energy efficiency, improved data-level parallelism	Limited precision arithmetic, noise-tolerance DNN, overhead	Numerical precision	
	SqueezeNet (Iandola et al., 2016)	ImageNet	Model compression, design-space exploration	More efficient distributed training, less overhead, deployment	Small CNN architecture	Deep CNN	
8	FedCS (Nishio and Yonetani, 2019),	CIFAR-I0, Fashion-MNIST	Federated learning client selection	Client selection, accelerate performance, training process efficient	Heterogeneous client, practical bandwidth limitation, simple DNN	Mobile edge computing	Communication cost and distributed training
	Edge aggregation	MNIST	CNN, hierarchical federated averaging, logistic regression	Reduce communication overhead, performance	Intolerable communication latency, burdens on the backbone network	Mobile edge computing	
	Federated learning (Luping et al., 2019)	MNIST CNN, next-word-prediction LSTM	CMFL	Mitigating communication overhead, reduction of network footprint	Learning convergence, update relevance	Machine learning	
	eSGD (Tao and Li, 2018)	MNIST	Momentum residual accumulation	Reduce the communication cost	More smooth, exploding gradient	Deep learning (IoT)	
	Deep gradient compression (Lin et al., 2017)	Cifar10, ImageNet, Penn Treebank, Librispeech Corpus	Compression to image classification, speech recognition, language modeling	Reducing communication bandwidth, scalability and speed up, distributed training, convergence	Lower throughput, intermittent poor connections	Large-scale distributed training	

selection strategy in a mobile edge computing framework with a small training time (Nishio and Yonetani, 2019). Federated learning (FL) is a framework for training the DNN model that eliminates the requirement to acquire private data from users. In federated learning, edge aggregation is utilized to reduce communication costs. Communication-Mitigated Federated Learning (CMFL) enhanced FL by minimizing communication overhead (Luping et al., 2019). CMFL provides feedback about model change global tendency. The eSGD reduces the size of the gradient in the CNN model by only transferring essential gradients and allows for effective communication in Distributed Deep learning (DDL) (Tao and Li, 2018). Deep gradient compression reduced the gradient 600 times while maintaining accuracy, using momentum correction, warm-up training, momentum factor masking, and local gradient clipping (Lin et al., 2017). Federated Learning (FL) supports many IoT services such as IoT data-sharing, optimizing IoT data-offloading and data-caching operations, along with attack detection, localization, mobile-crowdsensing, privacy, security, etc. (Nishio and Yonetani, 2019). FL also integrates blockchain to address security threats in fog-based IoT networks (Nishio and Yonetani, 2019). Further, FL has been applied in many IoT applications including smart city (city data management, smart energy and energy load prediction), smart transportation (vehicular traffic planning, vehicular resource management, unmanned aerial vehicles communication and its network/energy/security management), smart healthcare (electronic health records management, healthcare cooperation), smart industry (federated robotics arms, industrial intelligence, industrial edge-based IoT, network resource allocation, IIoT security), etc.(Nguyen et al., 2021). The above discussion indicates that the currently existing DL approaches in IoT face challenges related to collecting and managing data regularly, models training, limited resources in IoT devices (hardware constraints), and rapid expansion in IoT data. The lack of efficiency and performance of these DL techniques need to be addressed.

4.3. *Fog-Cloud-ML*

This sub-section explores the state of the art with the integration of fog computing, cloud computing, and machine learning. It focuses on how machine learning methods are applied for Cyber–Physical System (CPS) data stream utilizing cloud computing and fog computing-based architectures.

The concepts of CPS and fog computing are closely related. CPS is mainly associated with real-time applications. As per NIST (Yang et al., 2018), CPS is defined as *"System that integrates the cyber world with the physical world are often referred to as cyber–physical systems. The computational and physical components of such systems are tightly interconnected and coordinated to work effectively together, sometimes with humans in the loop"*. When integrating fog and cloud computing networks, administrators consider most time-sensitive data. The most critical time-sensitive data should be evaluated within established control loops. Processing latency is reduced by bringing storage and computing systems as close as possible to the applications, components, and devices that require them. In this way, fog computing is enhancing cloud computing performance by reducing latency, improving response time, increasing compliance for business, high security and data privacy, saving bandwidth, increasing overall speed and efficiency, reducing frequently use of WAN services, higher uptime of critical systems, and with improved remote location services.

Zhou et al. (2018) proposed a fog computing-based Cyber–Physical Machine Tool System (FC-CPMTS). This system minimizes the cloud platform's network traffic and calculation workload. The results reveal that the CNC machine tool's autonomy, interconnection, interoperability, and intelligence have improved. Moon et al. (2016) proposed CPS-healthcare system provides real-time and secure data collection services via fog nodes. This system reduces network latency as compared to cloud computing. Some other fog computing-based CPS examples are IFCIoT, Cloud-to-Things continuum, trust management system, exogenous coordination model, PsCPS, CPSWare, and ICPS.

Table 6 shows problem, ML algorithms, field, topic, methods/tools used, dataset, attribute/parameters related to cloud and fog computing. This shows the effectiveness of ML approaches for CPS data stream (Sideratos et al., 2015). The widely applied ML techniques in fog integrated cloud architecture are Generalized Linear Model (GLM), Gaussian Processes (GP), Multilayer Perceptrons (MLP), Random Forests (RF) (Sideratos et al., 2015), ANN, SVM, KNN, DT, BN (Ferreira et al., 2017), particle swarm clustering (Yuwono et al., 2016), PSO (Chakraborty et al., 2011), and Hidden Markov Model (HMM) (Tai et al., 2009) in CPS applications. The applications are mainly divided into Smart Grid (SG), Intelligent Transportation Systems (ITS) (Kumar et al., 2021) and Smart Manufacturing (SM) fields (Sideratos et al., 2015). The SG is a new paradigm for energy supply and electric networks that use advanced monitoring, control, and communication technologies to offer a stable and secure energy supply. The ITS refer to the distribution of traffic-related information to drivers in real-time with high-reliability (Ferreira et al., 2017). It offers safety-critical features such as blind-spot warnings during changing lanes, as well as enhancements to the driving experience. It provides traffic congestion notification and rerouting advice to overcome traffic congestion. The SM applies intelligent data-driven manufacturing for the entire product development life cycle. It uses advanced sensor, control, modeling, and platform technology for the rapid development of new items. It offers real-time optimization for manufacturing production systems, supply chain networks, and gives dynamic responses to product demand (Gupta et al., 2014).

The SG faces issues like electrical power prediction, short-term load forecasting, blackout warning, energy demand, economic load dispatch, and decision-making in smart home-usage. The issues found in intelligent transportation systems include behavior and event recognition in driver monitoring and analysis, obstacles classification, and traffic congestion level. In smart manufacturing key issues are energy consumption, machinery prognostics, machine maintenance, quality control, fault detection, process optimization, and fault diagnosis of bearings (Tai et al., 2009).

To address above mentioned issues, machine learning techniques are investigated. The ANN techniques forecast trends by performing regression on a time-series data stream. The ANN predicts energy consumption to help clients to regulate energy demand (Lahouar and Slama, 2015). The SVM is utilized in fault detection, forecasting, clustering, feature design, and in application domains such as smart grid and transportation since it offers a shorter response time (Ponz et al., 2015; Susto et al., 2014; Monedero et al., 2012). The RF is used to detect faults in manufacturing machines or products as well as spurious electricity records from sensors (Monedero et al., 2012; Osaba et al., 2016; Lieber et al., 2013). In the power system, a Decision Tree (DT) classification technique is utilized for fault detection, energy demand prediction, and bus travel time (Chakraborty et al., 2011; Yuwono et al., 2016; Li and Jayaweera, 2014; Tai et al., 2009). The HMM is used for process optimization, customer real-time decision-making, and detection of machine failures (Li and Jayaweera, 2014). Customers can learn to act optimally in the context of Markov dynamics via Q-learning. It allows users to experience the implications of their actions without needing to make models beforehand. Further, the PSO is applied for Economic Load Dispatch (ELD) in the smart grid. The ELD allocates generators' powers to meet specific demands while minimizing generated costs. It is a non-linear constrained optimization problem. The hybrid quantum-inspired PSO lowers the fuel cost of generator (Chakraborty et al., 2011).

This sub-section described the use of ML techniques in fog and cloud computing. It elaborates on the problems related to smart grid, transport, and manufacturing fields. It also addresses issues related to these fields and applies ML techniques to solve them. The key ML techniques mostly applied in fog-integrated cloud computing are ANN, SVM, RF, DT, HMM, and PSO.

4.4. *Mist-Cloud*

This sub-section focuses on the integration of mist computing and cloud computing. Mist computing extends computing, storage, and networking capabilities over the fog. In the IoT-Fog-Cloud continuation, it is termed as the first computing location and provides dispersed computing at the extreme edge. It maintains users' data privacy through local processing. Virtualized instances are deployed on single-board PCs by utilizing mist computing capabilities. It is mainly utilized in time-centric applications to improve overall processing with low latency and high throughput. It achieves optimal utilization of the resources. For video streaming applications, mist computing reduces the load on standard WiFi infrastructures. It mainly focuses on overlay analysis that is used to explain the utilization of mist-assisted cloud architecture in MistGIS architecture (Barik et al., 2019). In mist computing domain, key challenges are scalability (Byers and Wetterwald, 2015), resource constraints (Asif-Ur-Rahman et al., 2018), security (La et al., 2019), offline capability, privacy, interoperability (Battistoni et al., 2019), and intelligence levels (Battistoni et al., 2019; Liyanage et al., 2018).

4.4.1. *Mist computing challenges in cloud computing*

By processing data near the edge, mist computing minimizes latency and enhances throughput. It collects services that are distributed across multiple computing nodes. As indicated in Table 7, mist computing addressed and implemented these challenges. El-Hasnony et al. (2021) proposed Mist-Cloud computing-based framework on the Internet of Healthcare Things (IoHT), for monitoring and treatment purposes. It contains five layers, namely, the mist, the perception, the cloud, the fog, and the service provider layers. The mist layer is used to pre-process the sensor data by rule-based methods and performs time-critical data analysis. The Cloud-Mist-based IoHT framework is shown in Fig. 21. The author also applied four ML techniques for analysis of IoT and big data, namely, naïve bayes, MultiLayer Perceptron (MLP), Sequential Minimal Optimization (SMO), and Reduced-Error Pruning tree (REP-tree) to calculate CPU processing time and classification accuracy. It has used the dataset, namely, "home activity monitoring utilizing gas sensors" provided by the University of California, Irvine (UCI) ML repository.

4.4.2. *A hybrid Mist-Cloud system*

MistGIS is a model based on mist computing, used for geospatial big data analytics in the mining area. Geospatial data was processed at the edge and then misted using fog devices before being saved in the cloud layer. It contains four layers as edge layer, fog layer, mist layer, and cloud layer. Mist layer improves computational performance near the edge computing devices. The Spark Machine Learning Library (MLlib) library is used for data analysis purposes (Asif-Ur-Rahman et al., 2018). MLlib supports ML algorithms for classification, regression, collaborative filtering, and clustering purpose. Barik et al. (2019) applied machine learning techniques for cluster analysis and considered nearby universities in the same cluster. Three clusters were formed by using the location of the different universities (latitude and longitude). The MistGIS proposed framework is shown in Fig. 22.

Table 6
ML methods for CPS data stream in the cloud and fog architecture.

Ref.	Problem	ML Algo.	Field	Topic	Methods/Tools	Dataset	Attr./parameters
Sideratos et al. (2015)	Electrical power prediction	GLM, GP, MLP, RF	Smart grid	Load Forecast	Prediction, combination	Load time series	–
Ferreira et al. (2017)	Behaviour and event recognition	ANN, SVM, RF, BN	Transport	Driver monitoring and analysis	Smartphones sensor	Driving data	Aggressive breaking, Turn, acceleration, lane change
Shin et al. (2014)	Energy Consumption	ANN	Manufacturing	Metal cutting industry	Big data analytics	STEP-NC, MTConnect	Cutting depth, feedrate, spindle speed, cutting diameter, wattage
Lahouar and Slama (2015)	Short term load forecasting	ANN, SVM, RF	Smart grid	Electrical power	–	Tunisian Power	Load demand
Wu et al. (2016)	Tooling wear/Errors detection	Parallel RF	Manufacturing	Machinery prognostics	Amazon EC2	PHM	Spindle speed, feed rate, Y-Z depth of cut, sampling rate, material
Gupta et al. (2014)	Blackout warning	SVM	Smart grid	Grid resilience	Probabilistic	Normal-cascade failure states, IEEE 30-bus	Line contingency, mean, variance, skewness, kurtosis
Ponz et al. (2015)	Obstacles classification	SVM	Transport	Traffic accidents	Information fusion	Labeled images, point clouds	Euclidean, HOG, concentration, planicity, sphericity, cubicity, triangularity
Susto et al. (2014)	Machine maintenance	SVM, KNN	Manufacturing	Semiconductor manufacturing	Multiple Classifier	R2F historical data, time series	Maximum, minimum, average, variance, skewness, kurtosis
Monedero et al. (2012)	Energy demand	DT, bayesian network	Smart grid	Detection of non-technical losses	IBM SPSS modeler 14	Endesa Company NTLs	MinConsumption, StatisStreak, DiffReadingBills
Osaba et al. (2016)	Traffic congestion level	C4.5	Transport	Predict traffic congestion and pollution level	Open street maps, classification	Road data	Station, lane, time, total vehicles
Lieber et al. (2013)	Quality control	SOM, NB, DT	Manufacturing	Inline quality prediction	RapidMiner	Sensor data	Time, frequency, value statistics, deviation, rolling force, speed, temp
Liu and Jin (2013)	Fault detection	Bayesian	Manufacturing	Source diagnosis	BN software Netica®	Sensor data	Inline measurement gages
Chakraborty et al. (2011)	Economic load dispatch	PSO	Smart grid	Hybrid quantum mechanics	C++, PHP5	–	Production cost, fuel cost function, system power balance, operating power, ramp rate
Yuwono et al. (2016)	Process optimization	Particle swarm clustering, HMM	Manufacturing	Bearing fault diagnosis	HP motor, torque transducer/encoder, dynamometer, control electronic	Vibration test data	Defect frequency, ball pass freq, shaft freq, cepstral liftering
Li and Jayaweera (2014)	Smart home usage	HMM, Q-learning	Smart grid	Decision-making	–	–	Action, state, belief mode, accumulated reward
Tai et al. (2009)	Fault diagnosis of bearings	HMM	Manufacturing	Detect machine failure	–	Historical production data	Sequences of deterioration

Table 7
Implemented mist computing challenges.

Challenges	Papers
Resources constraints	Asif-Ur-Rahman et al. (2018), Byers and Wetterwald (2015), La et al. (2019), Battistoni et al. (2019) and Liyanage et al. (2018)
Scalability	Battistoni et al. (2019) and Liyanage et al. (2018)
Security	Asif-Ur-Rahman et al. (2018), Byers and Wetterwald (2015) and La et al. (2019)
Privacy	Battistoni et al. (2019)
Offline capability	Battistoni et al. (2019)
Interoperability	Battistoni et al. (2019)
Intelligence Levels	Low Asif-Ur-Rahman et al. (2018), Byers and Wetterwald (2015), La et al. (2019), Battistoni et al. (2019) and Liyanage et al. (2018)

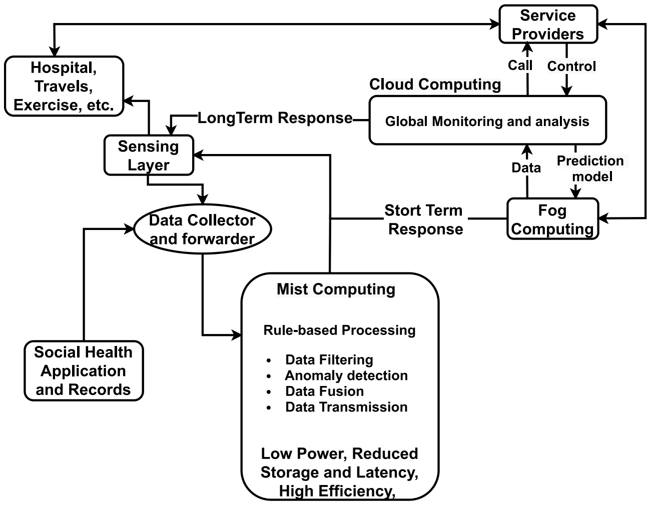


Fig. 21. Cloud-Mist based IoHT framework.

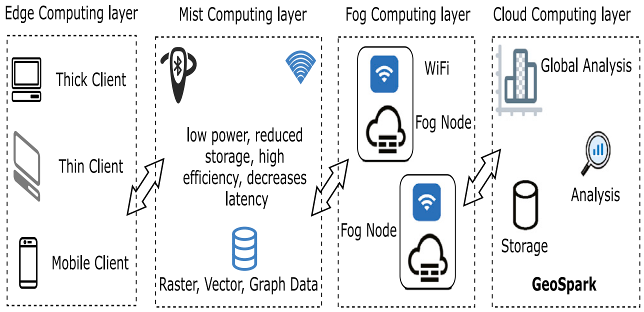


Fig. 22. Proposed framework of MistGIS.

From the above study, it is observed that there is a lack of mist computing implementations and standards. There is a need to efficiently configure them using the best strategies to meet the requirements of heterogeneous nodes in massive-scale IoT networks. Due to the dynamic nature of mist computing networks, it is challenging to allocate a node to execute a long-running job. There is a need to use hybridization for boosting system exploration such as swarm intelligence and bio-inspired computation which can be explored in the future (Barik et al., 2019).

4.5. IoT-ML

This sub-section presents the integration of the Internet of Things and machine learning. It briefs about challenges in IoT, IoT device identification techniques, and applications of ML algorithms for smart city IoT systems.

4.5.1. Machine learning in IoT

There are several issues in IoT like power-saving, network management, heterogeneity, interoperability, overloading, network congestion, QoS, long-range network, network management, security and privacy, scalability, coverage, and network mobility. The use of ML techniques offers benefits to resolve the above-mentioned issues. ML maintains privacy by preventing unauthorized identification and tracking. It also performs power analysis and detects intrusion in the network systems (Huynh et al., 2015). The design of a privacy-setting interface for IoT device employs a data-driven approach using ML to predict the user behaviors (Baldini et al., 2017). Clustering analysis is applied for smart default policies or profiles to improve the results. To provide interoperability and heterogeneity, gradients are calculated during the learning phase and the "Adadelta" optimizer which gives the best results. ML applied to handle network congestion and overload issues. In cognitive radio, ML is used to solve two types of issues: classification and decision making. It learns environmental facts using a ML approach and makes optimal decisions.

ML techniques also assist in handling network mobility and coverage issues. Smart data mobility provides data sharing by using communication within IoT entities in a self-evolving manner. The ANN is used to provide learning assistance for the system. It aims to bridge the gap between knowledge reasoning and learning. Nahrstedt et al. (2016) used the naive bayesian technique used to detect people's future movements, using traces and mobility patterns. SVM is used to check whether the contact exists within two devices. Chafii et al. (2018) applied reinforcement learning algorithms used to enhance Narrow Band-Internet of Thing (NB-IoT) coverage using dynamic spectrum. Additionally, ML techniques also assist to enhance the performance of the system by improving the QoS parameters. To predict the missing QoS values, the kernel machine learning algorithm and Kernel Least Mean Square (KLMS) algorithm are used to investigate the hidden connections between all known QoS data and the relevant QoS data with the best similarity. Wu et al. (2017) proposed context-aware QoS prediction using crowd intelligence. This utilized the Gradient Descent (GD) algorithm in a deviation-based neighborhood model. The problem of data sparsity can be efficiently addressed by optimizing QoS prediction models through ML. There are various devices working in the IoT domain such as Camera (Tuama et al., 2016), iOS-based (Kurtz et al., 2016), android based (Miettinen et al., 2017), PC (Patel et al., 2014; Meidan et al., 2017) and smartphones (Tuama et al., 2016). These devices need to be located and mapped for proper functioning. For this purpose, support vector machine, random forest, Gaussian mixture modeling, K-nearest neighbor, and boosting algorithm are used (Meidan et al., 2017). Table 8 represents the brief details about IoT device identification using ML techniques.

Based on this study, it is inferred that the key algorithms applied to solve the several issues in IoT are gradient descent, reinforcement learning, least mean square, ANN, SVM, KNN, and naive bayes for classification and decision making. For IoT device identification GMM and KNN algorithms offer the highest accuracy. The SVM, KNN, and RF are mostly used algorithms for the identification purpose.

4.5.2. Smart city management using machine learning

This sub-section presents the study for ML algorithms used in IoT based smart city applications such as smart traffic, smart health, smart environment, smart weather prediction, smart citizen, smart agriculture, smart home, smart air control, smart public place monitoring, and smart human activity control in IoT integrated cloud computing paradigms. Kafi et al. (2013) explained intelligent transportation systems based on wireless sensor networks for smart traffic monitoring and utilized classification learning algorithms such as K-Nearest Neighbor (KNN) and SVM. Author presented details about the cloud-based architectures used in a variety of smart traffic applications. Qin et al. (2016) discussed clustering algorithms such as Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and K-Means for future IoT and RFID enabled information systems used in smart health for RFID readers to scan and monitor medical devices. Furthermore, Jakkula and Cook (2010) explored how to discover anomalous power consumption, and the outlier detection algorithms. Authors used statistical approaches in power datasets, KNN, and discrete-time warping for distance measures in a smart environment. Ni et al. (2014) also presented a hybrid technique for sensor data prediction in the IoT. For the same, Empirical Mode Decomposition (EMD) was utilized for data pre-processing, it had the problem of frequent emergence of mode mixing. To address this problem, Ensemble Empirical Mode Decomposition (EEMD) is employed for non-linear and non-stationary signal sequences. It also applied Support Vector Regression (SVR) after mapping the data to a high-dimensional space using a kernel function.

Table 8
IoT device identification using ML.

Ref.	Device	ML Technique	Features	Accy
Tuama et al. (2016)	Camera	SVM	Image quality metrics, wavelet, block covariance, cross-correlation of CFA, local binary patterns, conditional probability	98.75%
Kurtz et al. (2016)	iOS	Threshold based classifier, SVM	Device name, URL, protected resources, icon cache, top 50 songs, public resources, WiFi and device name	97%
Miettinen et al. (2017)	Android, iOS	Random forest	Device configuration, sensor specific, inter-arrival time, rate of packets	–
Huynh et al. (2015)	Android, iOS	GMM, KNN	Time & frequency domain, MFCC, RMS	98%–100% (detection rate)
Baldini et al. (2017)	Android, iOS	SVM, KNN, naive bayes, decision tree	Statistical, log energy entropy, Shannon entropy, standard deviation, variance, skewness, kurtosis	85.08%
Patel et al. (2014)	ZigBee device	Random forest, AdaBoost, MDA-ML, GRLVQI	Instantaneous amplitude, phase, frequency	75%–90%
Meidan et al. (2017)	PC, smartphone, printer, socket, refrigerator, thermostat	GBM, XGBoost, random forest, binary classifier	Network traffic, src IP, dest IP, port number	99.281%

The parameters of SVR are optimized using Particle Swarm Optimization (PSO). Ma et al. (2013) proposed a model for transit riders' travel patterns that uses smart card transaction data and employed the DBSCAN algorithm to discover historical travel patterns as well as KMeans with rough set techniques for clustering and classification. Due to the massive amount of data presented on the Internet of Agricultural Things (Han et al., 2014), the bayesian network is used to determine the final product's trust value. Costa and Santos (2015) utilized data mining techniques for massive data processing in smart homes to predict future energy consumption. They also used K-means algorithm to detect intra-cluster similarities. Shukla et al. (2015) proposed outlier detection techniques for stream data mining. They demonstrated that a density-based algorithm produces more purity than a distance-based approach. The density-based technique gives the best results in terms of memory consumption and arbitrary shape. Monekosso and Remagnino (2013) explored the data-driven approach for validating sensor-generated historical data utilizing Curvilinear Component Analysis (CCA) and Principle Component Analysis (PCA) algorithm to detect intermittent faults and failure masking. Shilton et al. (2015) proposed a one-class support vector machine to model data patterns in IoT applications. For the same, the details are shown in Table 9.

This study infers that the key algorithms applied in smart city management are KNN, PSO, DBSCAN, K-means, naive bayes, PCA, CCA, ensemble learning, and DT for classification and clustering. As the IoT sector grows, new and hybrid ML approaches are required to tackle complex problems.

4.6. SDN-IoT

The Software-Defined Networking (SDN) provides intelligence to Internet of Things (IoT) networks (Kim et al., 2017). This sub-section discusses the integration of SDN and IoT based architectures. It elaborates on machine learning techniques used in the IoT and SDN integrated architecture for several QoS/QoE parameters predictions.

4.6.1. Machine learning in software-defined networking

This sub-section discusses the objectives, learning methods, dataset, accuracy, attributes, domain study, and software/hardware in software-defined networking integrated architectures.

Table 10 presents the summary of the ML techniques applied to the SDN domain. This study focuses on several key SDN objectives as service context identification, SQL injection attack detection, DNS query attack detection, dynamic attack detection, IoT solution & recommendation engine, arrhythmia detection, stress detection, smart meter working, and parking space detection. To address these objectives, various ML techniques are used and briefly described as follows. Kim et al. (2017) presented hierarchical clustering techniques used to obtain service context in network traffic identification. The SQL injection, DNS query attacks, and dynamic attacks were investigated for SDN security (Ahmed et al., 2017). Uwagbole et al. (2017), also explored the Structured Query Language (SQL) injection attack using traffic data. It is observed that SVM and logistic regression are used to enable scalable big data mining. Ahmed et al. (2017), presented bayesian algorithm that can track and learn the behavior of IoT devices over time at SDN controllers. According to Bhunia and Gurusamy (2017), SVM offered better accuracy (98%) for traffic classification. SDN manages security risks dynamically and adaptively without having any load on IoT devices. Using electronic health records and demographic data, machine learning models such as logistic regression, LibSVM, One Rule (OneR), and DT were applied to health monitoring to predict the occurrence of diseases (Asthana et al., 2017). The IoT healthcare applications are arrhythmia detection, IoT health, and stress detection. For arrhythmia classification, KNN and bagged-DT predict and identify heart disease (Walinjkar and Woods, 2017). To manage a large amount of data in IoT healthcare, both supervised and unsupervised ML were used to learn from data and improve through experience (Nguyen et al., 2017). Pandey (2017) presented that logistic regression and SVM algorithms which offered good accuracy between (66%–68%) for stress detection. Further, Siryani et al. (2017) discussed bayesian network and naïve bayes algorithm to provide a decision support system for electric smart meter operation. Ling et al. (2017) explored mean shift clustering and haar-feature classifier for parking space identification and occupancy detection in smart cities. The haar-feature classifier trains a cascade function and applies unique for feature extraction to identify the object using a sequence of annotated figures.

This study shows that the widely used ML techniques in SDN are SVM, DT, bayesian network, bagged tree, naive bayes, logistic regression, and KNN. In these algorithms, bagged tree and KNN algorithms are mostly used algorithms in the SDN domain.

4.6.2. Machine learning for quality of service and quality of experience

This sub-section discusses objectives, methods, tools, input and output datasets, and related works for QoS/QoE parameters prediction in IoT based on SDN. Table 11 presents the machine learning

Table 9
ML algorithm for smart city IoT systems.

Ref.	Smart city cases	Approach	Parameters	Objective	Dataset	Applications	Issues
Kafi et al. (2013)	Smart traffic (WSN based ITS)	kNN,SVM	Ad-hoc on-road, Ad-hoc hybrid, Ad-hoc on-vehicles, on-vehicles with BS, on-road with BS, hybrid ad-hoc, hybrid with BS	Traffic monitor	Sensor data	Pollution control, traffic light, parking management, safety, traffic optimization, smart cities, reliability, real time	–
Qin et al. (2016)	Smart health	DBSCAN and K-Means	Data quality, data generation, data interoperability	Manage IoT data	Linked stream	Cities, homes, environment monitoring, health, energy, business	Data quality and uncertainty, co-space data, transaction handling, mining, security, privacy
Jakkula and Cook (2010)	Smart environment	KNN, DTW	Electricity consumption, total population, window size, outliers	Smart home energy conservation	CASAS smart real-synthetic, power datasets	MavHome project gator tech smart house, iDorm, georgia tech aware home	Identify abnormal power, monitor energy consumption, privacy, personalization
Ni et al. (2014)	Smart weather prediction	Support vector regression, PSO, ensemble empirical mode decomposition	Wind speed, atmospheric pressure, temperature, relative humidity	Sensor data forecasting	Beijing district sensor data, periodicity, fluctuation, trend	Energy consumption, stream-flow of river, economic trend	Prediction of sensor data, sensor error data detection, sensor data noise reduction
Ma et al. (2013)	Smart citizen	DBSCAN, rough set-based, k-Means, C4.5	Route-number, remaining balance, driver ID, smart card ID, transaction time, transaction amount, remaining balance, alighting-stop, boarding-stop	Model transit riders' travel patterns	Transaction data from smart cards	Activity-based travel, transit market analysis, transit OD estimation	Travel pattern recognition, travel regularity mining
Han et al. (2014)	Smart agriculture	Naive bayes	Time, location, signature, chain, coverage, reputation	Trust worthiness of food safety	Pork production historical data, electronic pedigrees , reference database	Supply chain, internet of agricultural things, food production, transportation	Subjectivity and time-delay of the trust evaluation, data veracity
Costa and Santos (2015)	Smart home	K-Means	Cluster number, week, electricity and gas consumption, energy, heating/cooling, lights	Homes energy consumption, future electricity consumption	Electricity and gas consumption	Reinventing the energy bill, energy monitoring service	Energy control and distribution
Shukla et al. (2015)	Smart air controlling	K-Means, micro-cluster based Continuous Detection	Memory usage, total process time, range query, arrival rate, concept drift, uncertainty, cross-correlation,	Outlier detection	KDDCUP99	Fraud detection, plagiarism, communication network management	Stream data mining, transient, notion of time, memory and CPU utilization
Monekosso and Remagnino (2013)	Smart public place monitoring	PCA , CCA	Data rates, computation time/speed, sample rate	Sensor failure and recovery	Sensor data	Smart home, smart office building, shopping mall	Detecting the fault, extrapolating, faulty sensor
Shilton et al. (2015)	Smart human activity control	One-class SVM	Gradients, kernel cache, memory	Automatic analysis of the sensor data	Banana dataset, UCI "adult"	IoT deployment	Detecting emerging anomalies, computational complexity

techniques applied for QoS/QoE matrices prediction in IoT for SDN architectures. The main objectives in SDN are delay prediction, QoS prediction, service-level QoS prediction, QoE prediction, response time, and transmission delay.

The QoE considers perception and satisfaction of user services such as mean opinion score (Laghari et al., 2018). QoE metrics are discrete in nature, therefore, the classification algorithms are used. The subjective methods are used to calculate the QoE matrices (Laghari et al., 2018). Further, the QoS measures are continuous data, therefore, regression based supervised learning algorithms are used. The QoS metrics include loss rate, jitter, latency, and network throughput. For delay prediction in IoT applications, Carner et al. (2017) applied Artificial Neural Networks (ANN) and Stochastic Gradient Descent (SGD) optimization algorithms on traffic load data-set. For QoS prediction, Jain et al. (2016) applied Decision Tree (DT) and Linear Regression (LR) using Key Performance Indicators (KPIs). It predicts traffic congestion through multi-dimensional analysis of KPIs. For service-level QoS prediction, Pasquini and Stadler (2017) utilized RF and regression tree based ML techniques. RF shows better estimations than regression tree analysis. For QoE prediction in video streaming services, Letaifa (2017) used Bayesian Networks (BNs), random neural networks, DTs, Hidden Markov Models (HMMs) techniques. Abar et al. (2017) used

Table 10
Machine learning and internet of things in SDN.

Ref.	Objective	Methods	Dataset	Accy.	Attributes	Area	Software/Hardware
Kim et al. (2017)	Service context identification	Hierarchical clustering	Traffic data	-	Standard deviation, mean packet-length, inter-packet arrival-time, mean inter-packet arrival-time, timestamp, src IP, dest IP, protocol, packet length, src port, dest port Number	Network traffic	–
Uwagbole et al. (2017)	Detect SQL injection attack	SVM, logistic regression	Traffic data	98%	SQL tracing	Security	Azure ML studio
Ahmed et al. (2017)	Attack detection (DNS query)	Bayesian	Traffic data	75%	Traffic patterns, total number of packets transmitted, ratio of source and destination bytes, connection duration time	Security	Dual core i7-4790 CPU (3.60 GHz), memory (8 GB)
Bhunia and Gurusamy (2017)	Attack detection (Dynamic)	SVM	Traffic data	98%	Source of requests, number of failed authentication attempts, device usage at different time periods, number of sent requests, bandwidth consumption	Security	Mininet emulator, IoT devices, SDN-enabled switch, cluster SDN controller, master SDN controller
Asthana et al. (2017)	IoT solution & Recommendation engine	Lib SVM, decision tree	Health history	–	Demographic features, electronic health records, IoT data	Recommend wearables	Home health monitoring, wristwatches, headphones, smartphones
Walinjkar and Woods (2017)	Detect arrhythmia	Bagged tree, KNN	ECG waveform (MIT-BIH, Physionet)	99.4%	Age, gender, signal strength, inter-beat interval, heart rate, impact factor, blood pressure	Healthcare	3-lead ECG kit, MATLAB, arduino microcontroller
Nguyen et al. (2017)	IoT health	Supervised & unsupervised	Sensor data	-	Blood pressure, respiration, SpO2, pulse rate, heart rate	Healthcare	Accelerometers, gyroscopes, bluetooth, zigBee, RFID, NFC, UWB, arduino, phidgets, intel galileo, Contiki, TinyOS, LiteOS
Pandey (2017)	Stress detection	SVM, logistic regression	Pulse waveform	66%–68%	Heart rate	Healthcare	Node MCU, Read–Eval–Print loop interpreter, pulse sensor
Siryani et al. (2017)	Smart meter working	Bayesian network, naive bayes	Meter data	96.69%	Signal strength, network type, network coverage, ESM CQ, travel expense average, FTSV, total operations costs savings	Decision-support	Point-to-point (mobile connection), power line communication, automated meter reading system, RapidMiner, data concentrator
Ling et al. (2017)	Detection of parking Space	Mean shift clustering	Camera data	91%	Vehicle pictures, images	Smart city	Raspberry Pi, OpenCV, eight mega-pixel camera board, amazon cloud

DCR (Degradation Category Rating) as a subjective method to develop the training model and validate it. Later, DTs, NNs, RFs, K-Nearest Neighbor (KNN), etc. are applied to determine the QoE objectively. The RF achieved the best results for predicting the perception of the user. This study briefly identifies that for QoS/QoE predictions in SDN architectures, majorly used ML techniques are NN, LR, RF, DT, HMM, and BN. Later, the SDN controller uses these prediction results to configure the devices. For the implementation purpose widely used tools are Weka, Mininet, and OpenFlow SDN emulator.

4.7. Digitaltwin-IIoT

This sub-section brings digitaltwin and industrial IoT integration together. Pires et al. (2019) analyzed that the growth of the digitaltwin gains significant importance in the development of industry 4.0 solutions. Therefore, it explores ML usage in digitaltwin and IIoT such as bearing with anomalies and crack sizes identification, digitaltwin framework for the petrochemical industry, and digitaltwin-driven industrial AI along with various parameters for the creation of digitaltwin. In recent years, developments in cloud computing, and AI have resulted in substantial advancements in the industrial digitization process. A digitaltwin-driven cutting tool can provide a modern solution to these ever-increasing digitization requirements such as a digitaltwin-based anomaly detection platform. It is used for a real-time industrial system for health monitoring and anomaly prediction. Cloud computing and industrial robots are combined in Industrial cloud Robotics (ICR). A digitaltwin-driven industrial cloud robotics framework efficiently synchronizes and integrates digital and physical industrial robots. Piltan and Kim (2021) presented a classification of bearing anomalies and identification of crack sizes using the SVM with intelligent digitaltwin. It achieved average accuracy 99.5% for bearing fault pattern recognition and 99.6% for crack size identification.

Further, Min et al. (2019) discussed a framework for digitaltwin based on industrial IoT in the petrochemical industry. For digitaltwin

Table 11
QoS/QoE prediction in IoT-SDN using machine learning.

Ref.	Objective	Methods	Tools	Dataset input	Dataset output	Future Works
Carner et al. (2017)	Delay prediction	SGD, ANN (M/M/1)	Omnet++ (SCIKIT)	Traffic load, routing policy	Network delay	Provide the behavior of the network
Jain et al. (2016)	QoS prediction (big data multi-dimensional analytics)	Linear regression, DT, (Java, R)	Mininet, Weka (D-ITG, IPerf3, POX)	24 network KPIs	QoS parameters	Discover new correlations, perform root cause analysis, predict traffic congestion
Pasquini and Stadler (2017)	Service-level QoS prediction	RF, regression tree, statistical learning	Server cluster, OpenFlow SDN emulator	Port granularity statistics, operating system granularity statistics, flow granularity statistics	Frame rate, response time	QoS estimation techniques to produce accurate real-time estimates with small overhead
Letaifa (2017)	QoE prediction (SDN context video streaming service)	DT, RNN, HMM, BN	Mininet	RTT, jitter, bandwidth, delay	Mean opinion score value	Use more factor, apply on video conference, real-time
Abar et al. (2017)	QoE prediction (degradation category rating), response time, transmission delay	DT, NN, kNN, RF (Pearson correlation coefficient, Root-MeanSquareError)	Weka, Mininet	Video quality parameters (SSIM: perceptual effects of distortions, VQM: structural similarity based on HVS model)	Mean opinion score value	Improve the dataset with QoS (response time, transmission delay), privacy, security of users

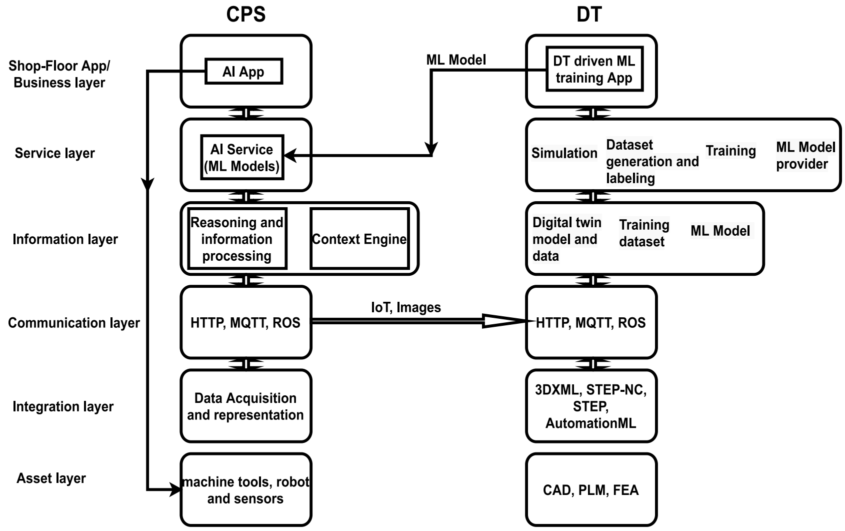

CPS
DT
Shop-Floor App/ Business layer
AI App
ML Model
DT driven ML training App
Service layer
AI Service (ML Models)
Simulation
Dataset generation and labeling
Training
ML Model provider
Information layer
Reasoning and information processing
Context Engine
Digital twin model and data
Training dataset
ML Model
Communication layer
HTTP, MQTT, ROS
IoT, Images
HTTP, MQTT, ROS
Integration layer
Data Acquisition and representation
3DXML, STEP-NC, STEP, AutomationML
Asset layer
machine tools, robot and sensors
CAD, PLM, FEA

Fig. 23. ML-based applications using digitaltwin for smart manufacturing.

model training, the basic regression techniques, such as Classification And Regression Tree (CART) and the integration algorithms such as Random Forest (RF), Gradient Boosted Decision Trees (GBDT), Adaptive Boosting (AdaBoost), Light Gradient Boosting Machine (LightGBM), eXtreme Gradient Boosting (XGBoost), and ANNs have been used. According to this study, Min et al. (2019), it is inferred that LightGBM offers maximum predicting accuracy and performance, and acts as a key training algorithm of the digitaltwin model. Alexopoulos et al. (2020) developed ML models using a digitaltwin-driven industrial artificial intelligence. It is used to train vision-based recognition for part-orientation to control the production process using Transfer Learning (TL). TL detects and applies prior knowledge and skills to new tasks. The ML-based system using digitaltwin for manufacturing is shown in Fig. 23.

Table 12 presents algorithms, architectures, computational processing tools, data exchange protocols, data formats, modeling, simulation tools, etc. for the creation of digitaltwin in IIoT 4.0.

The ANN is employed as a computing system, without the necessity for task-specific procedures to create digitaltwin (Park et al., 2019). Design Failure Mode and Effects Analysis (DFMEA) was used for the identification of design functions, severity effects, and failure modes (Madni et al., 2019). In a simulation-based controlled experimental testbed, a neural network is used to determine operator/user preferences and priorities. In virtual and real-world situations, clustering techniques are used to group objects and patterns (Madni et al., 2019). Further, reinforcement learning is applied for uncertain and partially observable operational environmental states. Ding et al. (2019) proposed a DigitalTwin-based Cyber–Physical Production System (DT-CPPS) for smart manufacturing. During operating flow, to find available solutions for future operations, the dynamic scheduling engine was used. It had utilized ANN and Ant Colony Optimization (ACO) algorithm for computation optimization. The authors in Rodič (2017) suggested a model that matches existing production processes and allows them to test optimization strategies using Discrete Event Simulation (DES). The DES is a technique for simulating system behavior over time as a sequence of events in industry 4.0. It is also used to create a model that represents existing manufacturing processes and experiments with different optimization strategies during the construction of an automated XML model.

Zheng et al. (2018) proposed Smart Product-Service Systems (Smart PSS) that achieves customer satisfaction with minimal environmental

Table 12
Digitaltwin creation.

Algorithms	Architecture	Computational processing tools	Data exchange protocols	Data formats	Modeling and simulation tools
Artificial neural network (Park et al., 2019)	J2EE SSH programming architecture (Leng et al., 2019)	AWS elastic MapReduce (Lu and Xu, 2019)	OPC UA (Ardanza et al., 2019)	Automation ML (Bao et al., 2019)	ANSYS simplorer (Popa et al., 2018)
Boundary element method (Zhuang et al., 2018)	Master–slave architecture (Arafsha et al., 2019)	Elastic stack, ELK stack (Damjanovic-Behrendt and Behrendt, 2019)	MQTT (Haag and Anderl, 2018)	SHDR (Liu et al., 2018a)	Autodesk revit (Lu and Brilakis, 2019)
DFMEA, clustering, reinforcement learning (Madni et al., 2019)	RESTFul (Park et al., 2019)	HBase (Lee et al., 2018)	MTConnect (Liu et al., 2018b)	STEP (Liu et al., 2019c)	Avatar software (Datta, 2016)
Deep neural network, ACO, ANN (Ding et al., 2019)	Service oriented architecture (Park et al., 2019)	IoT EL20 Edge computing (MacDonald et al., 2017)	CoAP (Leng et al., 2019)	–	Dymola (Schneider et al., 2019)
Discrete event simulation (Rodič, 2017)	Server–client architecture (Zheng et al., 2018)	MatLAB, Simulink (Denos et al., 2018)	SOAP (Park et al., 2019)	–	EasySim (Lovas et al., 2018)
Discrete Fourier transform (Zheng et al., 2018)	–	MS excel VBA (Tan et al., 2019)	AMQP (Damjanovic-Behrendt and Behrendt, 2019)	–	JMonkeyEngine 3.0 (Sierla et al., 2018)
Dynamic bayesian network (Li et al., 2017; Alam and El Saddik, 2017)	–	OMPL (Sierla et al., 2018)	NTP (Nikolakis et al., 2019)	–	Lantek expert Punch (Moreno et al., 2017)
FEA analysis (Zhuang et al., 2018; Li et al., 2017)	–	QFSM (Alam and El Saddik, 2017)	PTP (Kim et al., 2018)	–	MWorks (Luo et al., 2019a)
Gaussian filtering (Denos et al., 2018)	–	Reduce (Lee et al., 2018)	Profinet (Zhang et al., 2019)	–	PlantSimulation (Lee et al., 2018)
Hidden Markov model (Petković et al., 2019)	–	Tensorflow (Damjanovic-Behrendt and Behrendt, 2019)	Wireless-HART (Zhang et al., 2019)	–	Siemens' STAR-CCM (Ferguson et al., 2017)
Monte Carlo simulation (Söderberg et al., 2017)	–	–	TCP/IP, UDP (Ardanza et al., 2019)	–	SIMIO (Rodič, 2017)
NSGA-II algorithm (Liu et al., 2019b)	–	–	OpenFlow (Kim et al., 2018)	–	Unity3D engine (Omer et al., 2019)
Savitzky-golay filtering (Oyekan et al., 2019)	–	–	–	–	WITNESS horizon (Popa et al., 2018)
VV&A (Tao and Zhang, 2017)	–	–	–	–	–

impacts. It applied digitaltwin-enabled service innovation for data analytics. The discrete Fourier transform is executed on sampled data and calculated frequency bin. The Dynamic Bayesian Network (DBN) is used to connect variables during adjacent time steps. Li et al. (2017) designed digitaltwin-based aircraft wing health monitoring system. A DBN is utilized to make probabilistic predictions about future crack growth. Alam and El Saddik (2017) proposed cloud-based cyber–physical systems using a digitaltwin-architecture reference model. The combination of a fuzzy rule basis with the bayes network enhanced the system reconfiguration capabilities. A digitaltwin-based smart production framework was designed for product assembly shop floors. The digitaltwin was created in three stages as element-level, behavior-level, and rule-level. At the element level, the boundary element method is used to simulate the physical functions. The behavior level is made up of the behaviors of elements and the mechanisms that respond to them. The rule-level contains association rules, operation rules, and evolution rules utilizing ML and data mining algorithms (Zhuang et al., 2018).

Some others applications based on digitaltwin include aircraft-wing health monitoring system using DBN and that uses Finite Element Analysis (FEA) for monitoring crack growth (Li et al., 2017). The digitaltwin-based autonomously-controlled micro-punching system contains a detection system that applies gaussian filtering algorithm. This algorithm reduces noise data to increase data accuracy (Zhao et al., 2019). The digitaltwin-based autonomous manufacturing in smart shop floors system used HMM algorithm. The HMM uses historical data to generate a machine sequence that helps to choose the optimal machines. For personalized manufacturing in the production system, a digitaltwin-based real-time geometry assurance system is developed that optimizes quality. It uses FEA for sensitivity and variation analysis. Monte Carlo simulation was performed to determine the level of uncertainty and risk associated with visualization probable outcomes (Söderberg et al., 2017). The digitaltwin-based automated flow-shop manufacturing framework developed for rapid individualized design. In which fast sorting and multi-objective genetic algorithm were utilized to optimize machine performance (Liu et al., 2019b). Virtual reality-digitaltwin based physical layout system proposed strategies for human–robot collaboration in factories using Savitsky–Golay filtering techniques for noise reduction (Oyekan et al., 2019). Tao and Zhang (2017) proposed a DigitalTwin Shopfloor (DTS) based on digitaltwin paradigm. It describes domain knowledge using rules of associations, constraints, and deductions. The apriori, SVM, and K-means are also used to create rule models.

This study explored digitaltwin and IIoT 4.0 integration which helps in improving the performance of future industrial sectors by offering real-time monitoring and saving the cost. The major ML techniques that are widely applied for the same are SVM, K-Means, RF, LightGBM, ANNs, ACO, TL, DBN, and HMM.

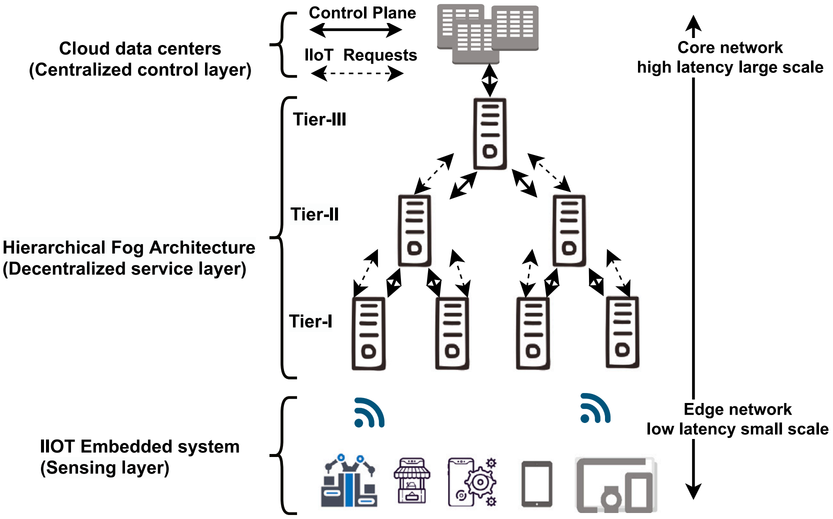


Fig. 24. Fog hierarchical architecture for IIoT.

Table 13
Fog computing and IIoT domains.

Ref.	Architecture	IIoT application domain	Resource allocation	Latency	Energy	Power	Handover	Bandwidth
Bi et al. (2018)	SDN	Mobility	0	0	0	0	1	0
Wang et al. (2019a)	HetNets	Big data analytics	1	1	0	0	0	0
Du et al. (2017)	Fog & Cloud	Smart IoT devices	1	1	1	0	0	0
Sun et al. (2019a)	RAN	Smart IoT devices	1	0	0	1	0	0
Zhang et al. (2018a)	NOMA & RAN	Big data analytics	1	0	0	1	0	0
Ning et al. (2019)	Cloud	VANETS	0	1	0	0	0	0
Khattak et al. (2019)	Cloud	Healthcare	1	1	0	0	0	1
Abkenar and Jamalipour (2019)	Fog & IoT	Smart IoT devices	0	1	1	1	0	0
Zeng et al. (2019)	NFV & RAN	5G network	0	1	1	1	0	0

4.8. IIoT-Fog

This sub-section presents a study on the integration of fog computing and industry 4.0 to deal with the issues related to latency, sensitivity, computing, minimize energy consumption, increase security, real-time data processing, and storage in IIoT. Samann et al. (2021) presented two methods for embedding intelligence in fog computing. Firstly, device-driven intelligence is achieved by adding smart data processing and networking services to ML-capable gateways. In which, ANN monitors wireless channel parameters can facilitate efficient coverage and connectivity. Secondly, human-driven intelligence is achieved by translating human domain information into network domain decision-making. In which, human behavioral patterns are used to train the network and make it smarter.

4.8.1. Fog computing in industrial IoT architecture

Chekired et al. (2018) proposed a novel hierarchical fog server architecture for the IIoT to minimize processing and communication delays in IIoT systems. Fig. 24 shows that fog servers are grouped into hierarchical architecture to serve industrial object requests and analyze data in real-time. This used Simulated Annealing (SA) approach to aggregate optimal solutions over different fog tiers.

Table 13 presents a comparative study on the architectures such as SDN, HetNets, Fog+ Cloud, Non-Orthogonal Multiple Access (NOMA)+ Radio Access Network (RAN), Fog+ IoT, and Network Function Virtualization (NFV)+ RAN, which are frequently used in IIoT and fog integration based applications. This study has been done on several QoS parameters as shown in Table 13.

Further, SDN-based fog computing architecture for mobility support reduces handover and applies route optimization algorithm in the mobile network (Bi et al., 2018). Heterogeneous Network (HetNets) driven by big data analytic reduces resource allocation and minimizes latency to provide routing recommendations (Wang et al., 2019a). It uses tensor decomposition methods that are used in deep learning and clustering applications. Fog-Cloud-based architectures proposed for smart IoT devices to reduce resource allocation, latency, and energy requirements (Du et al., 2017). It also optimizes computation offloading decisions for cost conservation problems based on min–max fairness (Wang et al., 2019a). Zhang et al. (2018a) presented NOMA-based Fog-RANs architecture for resource allocation which explains resource allocation and power attributes. Ning et al. (2019) introduced Vehicular Fog Computing (VFC) architecture for traffic management in real-time for smart cities. This fog-based-VANETS reduce latency in the network. The cloud based-healthcare model optimizes resource allocation, latency, and bandwidth (Khattak et al., 2019). In this model, foglets are used to handle the requests. Abkenar and Jamalipour (2019) also proposed the energy balancing algorithm for the Fog-IoT network. It reduces energy consumption, delay of the network and does efficient energy balancing for all fog nodes. Zeng et al. (2019) discussed that F-RAN can be used to reduce front-haul congestion and enhance End-To-End (E2E) latency. The NFV performed the transition from hardware-based functions to software-based functions. This NFV+RAN based framework improves energy efficiency, reduces latency and power consumption in 5G networks.

4.8.2. Machine learning in industrial IoT

Natesha and Guddeti (2021) proposed a monitoring system for machine malfunction in an industrial environment. Fog servers are used to analyze the sound of machines as normal or abnormal. In fog servers, ML models such as LR, RF, SVM, AdaBoost (Adam), and ANN-based MLP are used to identify faulty machines based on the functional sound. For this purpose, Mel-Frequency Cepstral Coefficients (MFCC) and Linear Prediction Coefficient (LPC) features are extracted from the machine sound. This minimizes service time and avoids failures in the

Table 14
Industry 4.0 projects.

Projects	Description
I4MS	Gateways are used to gain access to the most cutting-edge technologies for industries in Europe
ENTOC	Provide a standardized description of components for virtual commissioning
ARIZ	Generate human–robot cooperation safely and learning system for the employee for future
MetamoFAB	Integrate CPS with current modernization and development plans
ParsiFAl 4.0	Evaluate the condition of components to service plants, reduce maintenance cost and track transport route
SOPHIE	In real-time connects the digital factory with real production, use to develop and test concepts
OPAK	Make complexity control for the production process and open engineering platform
PLANSEE	Develop autonomous factory and secure data traffic
#1 Smart Factory	Produce smart, flexible, and future electronics in manufacturing industries
e-F@actory	For easy digital transformation and utilize robust technological improvements

Industrial IoT environment. Li et al. (2018a) utilized ML techniques to optimize machine maintenance for manufacture inspection systems in the IIoT using Convolutional Neural Network (CNN). O'donovan et al. (2018) proposed the ML models in factory operations for the equipment failures that reduce device failures and latency. Using detection engines, a predictive analytics mechanism for device discovery was proposed to predict machine failures. These engines use predetermined rules standards to compare the current status of machine data.

Additionally, Table 14 presents the current core projects running in the same domain with their brief descriptions.

The identified list of the projects is I4MS, ENTOC, ARIZ, MetamoFAB, ParsiFAl 4.0, SOPHIE, Open Engineering Platform for Autonomous Mechatronic Automation Components (OPAK), Power Semiconductor and Electronics Manufacturing 4.0 (PLANSEE), #1 Smart Factory, and E-F@actory. According to the study (Oztemel and Gursev, 2020), these projects are concentrating on large-scale digitization in various countries like Germany (ENTOC, ARIZ, MetamoFAB, ParsiFAl, SOPHIE, OPAK), Europe (I4MS), Austria (PLANSEE), and JAPAN (e-F@actory). As per the analysis, it is observed that widely used key ML techniques are LR, RF, SVM, AdaB, MLP, CNN, etc. to reduce service time, predict device failures, reduce latency, and improve performance in IIoT applications (Oztemel and Gursev, 2020).

This study explored various fog computing and IIoT 4.0 integration approaches along with related architectures such as SDN, HetNets, Fog-Cloud, NOMA-RAN, Fog-IoT, and NFV-RAN. These architectures optimized key performance metrics such as resource allocation, latency, energy, power, handover, and bandwidth. Out of these architectures, Fog+Cloud, Fog+IoT, and NFV+RAN based models are optimizing maximum QoS performance metrics.

5. Security in integrated cloud paradigms

Security is the most significant challenge to prevent the widespread adoption of cloud computing (Ahmad and Alsmadi, 2021; Alsharif and Rawat, 2021). A threat/attack occurs when an intruder gains access to a system and reveals private data without permission of the user. Attacks are posing a threat to the network ability to protect identification, authorization, accessibility, privacy, and integrity of the user. Thus, this section enlightens the security threats in emerging Cloud computing integrated paradigms.

5.1. Security threats in Cloud-IoT-ML

Table 15 presents different types of attacks and threats in Cloud-IoT-ML integrated paradigm at different layers. From this study, it is inferred that KNN, RF, Q-learning and RL techniques are used to detect code attacks, privacy attacks and various malware at the application layer (Veena, 2018; Gurusamy et al., 2019). At the network layer, SVM, NN, KNN, ANN, RF, K-means, PCA, and LSTM are the widely used techniques to detect encryption attacks, Denial-of-Service (DoS) attacks, Distributed Denial-of-Service (DDoS) attacks, routing attacks, and middleware attacks (Luo et al., 2019b; Yeh et al., 2017). At the physical layer, Q-learning, RL, supervised ML, deep CNN, ANN are employed to detect side-channel, physical damage, node jamming, eavesdropping, rogue certificate, bluesnarfing, microprobing attacks, etc. (Volodymyr, 2020). To improve security in IoT, ML secures connected devices by identifying malicious code attacks (Thakkar and Lohiya, 2021). It will help the researchers to develop effective ML techniques for detecting several attacks. Applications of ML for Cloud-IoT security are majorly classified in three areas: intrusion detection, authentication, and privacy approaches. Majority of the articles (about 62%) are studied where ML is widely applied for intrusion detection. Major ML algorithms used for intrusion detection are SVM, KNN, Naïve Bayes, Random forest, k-Mean, Decision Tree, and ANN as per our findings in the literature. The SVM is mostly applied for intrusion detection and authentication (Andročec and Vrček, 2018). The intrusion detection and mitigation framework (IoT-IDM) provides network-level protection for smart home IoT devices. It allows users to utilize customized ML techniques for intrusion detection based on historical patterns of known attacks (Nobakht et al., 2016). Perez et al. (2017) discussed about hybrid ML techniques (supervised and unsupervised learning algorithms) that improve the intrusion detection rate. Authentication is the process of validating the authenticity of smart devices, and the trust building for key infrastructure is essential in the IoT. Yeh et al. (2016) proposed a method in which user bio-features are extracted as authentication tokens using ML, and performed real-time object verification in the background without the individual's consent. Gebrie and Abie (2017) developed a risk-based adaptive authentication scheme that continuously observes changes in channel parameters. It examines possible risks using a naive bayes algorithm, and optimizes the authentication solution for both user and the IoT devices in smart home eHealth. Privacy is another key parameter in Cloud-IoT security (Román-Castro et al., 2018). It includes approaches such as privacy preserving-identification and traffic-obfuscation, code obfuscation assessment, access control policies, biometric-encoding framework, etc.

Ahmad and Alsmadi (2021) presented IoT attack security threats taxonomy and explained various types of attacks such as large-scale attacks (LightAidra, Bashlite, Kaiten, Leet, Mirai, Hajime, Persirai, and Reaper), botnet attacks, packet flooding attacks, TCP SYN flooding attacks, ping of death attacks, slowloris attacks, jamming attacks, amplification attack, etc. Further, the author analyzed various ML and DL techniques i.e., Support Vector Machine (SVM), Random Forest (RF), Deep Learning (DL), Convolution Neural Network (CNN), Recurrent Neural Network (RNN), and Autoencoders (AE) for detecting malicious network traffic in IoT. Ioannou and Vassiliou (2019) used Radial Basis Function (RBF) kernel with a C-support optimizer (c-SVM) to discriminate between benign and malicious IoT traffic. Chaudhary and Gupta (2019) stated that random forest achieved maximum accuracy (99.17%) for DDoS attack detection. Auto-Encoder (AE) has been useful technique in fog-based IoT system to identify distributed attacks for cyber-security (Hogan and Esposito, 2017).

5.2. Security threats in Cloud-SDN-IoT

The Intrusion Detection System (IDS) is used to detect various types of attacks or intrusions present in the network (Ahmed et al., 2021). Intrusion Detection is the process of recognizing unauthorized network access by continuously monitoring and examining events for indications of possible risks. The IDS monitors network traffic, analyzes, and detects anomalous behavior or illegal access in the system. There are various security issues in SDN such as forwarding device attacks,

Table 15
Attack detection in CLOUD-IoT-ML using machine learning.

Layer	Attack categories	Description	ML-based solution
Application layer	Code Attacks (SQL Injection, Cross-Site Scripting, Malicious script, Session Hijacking, Logic Bomb)	Code that tries to access user data or executes for the purpose of non-validating the processes	K-NN and RF for malware detection (Veena, 2018), Adaptive deep-forest model and AdaBoost for SQL injection detection (Li et al., 2019)
	Privacy Attacks (SpearPhishing, Impersonation, Whaling, Bluejacking)	The potential of IoT are being used against users' privacy in numerous ways in these attacks	Q-learning and Dyna-Q improve detection latency. RL for Malware Detection (Gurusamy et al., 2019)
	Malware (Virus, Ransomware, Spyware, Trojan Horse)	The program generates a botnet and targets hardcoded sensitive-data in IoT devices	
Network layer	Encryption Attacks (Man in the middle, Caching, Spoofing, Packet manipulation, Crypt analysis, RFID Cloning)	When data from an IoT device is not encrypted, an intruder can sniff it and save it for later use	SVM, NN, and K-NN to detect the intrusion attack (Luo et al., 2019b; Wehbi et al., 2019), NN to detect DoS (Xiao et al., 2018), ANN to detect DDoS (Yavanoglu and Aydos, 2017)
	DOS or DDoS (Packet Flooding, Battery draining, Botnet, Sloloris, Jamming, Asymmetric)	Malicious software can take down victim machines and make them inaccessible to regular user	and anomalies (Mathonsi et al., 2019), SVM for attack Detection (Yassine and Malli, 2019), RF to avoid damage and unprotected device connection (Yeh et al., 2017), SVM and k-means for
	Routing Attacks (Sybil, Wormhole, Port, Forwarding, Decreased rank, Hello-flood, version number modification,sinkhole, black hole, wormhole)	Affect the IoT network performance by adversely affecting network resources, topology, and traffic data	abnormal behavior of IoT devices (Yusof et al., 2017), C-means clustering with PCA to improve detection rate, Association Rule for intrusion detection (Kleberger et al., 2011),
	Middleware Attacks (Brute force, Dictionary, Message Replay, Signature)	Attack on the entities, data, communication channels in IoT	LSTM for real-time malicious traffic detection (Hwang et al., 2019)
Physical layer	Side-Channel	A security vulnerability that uses indirect effects of the system or its hardware to acquire information from it or affects the execution flow of a program, rather than directly attacking the program or its code	Q-learning reduces the authentication error and RL for spoofing (Graves, 2019), supervised ML for spoofing detection, RL and deep CNN for the jamming attack (Gu et al., 2019), Supervised ML to detect and filter poisonous data (Volodymyr, 2020),
	Physical Damage	Causes physical damage to IoT devices	NN and ElGamal algorithm to avoid attacks (Vishwakarma and Jain, 2019),
	Node Jamming	Interrupts the network traffic by blocking a channel and draining the battery of resource-constrained small devices	RL for Eavesdropping (Swami et al., 2019)
	Eavesdropping	Information is theft while being transferred over a network	
	Rogue Certificate	Intercept traffic and add their own fake certificate. The integrity of a rogue certificate has been compromised, or it was issued to the wrong person	
	Bluesnarfing	Pairing with user's Bluetooth device and steals or compromises user personal information	
	Microprobing	To alter secure chip's tamper-resistance properties	

control plane threats, communication vulnerabilities, fake traffic flows, open programmable APIs attacks, etc. (Ahmed et al., 2021). The forwarding device attacks have used access points and switches to cause networking system failure or delay it for a long time period (Ahmed et al., 2021). The control plane threats have the potential to bring the complete network system down. The communication vulnerabilities occur when the Transport Layer Security (TLS) protocol for data control becomes disabled by the admin, making Man-in-the-middle attacks possible (Ahmed et al., 2021). The fake traffic flows (Using DoS and DDoS attacks) make the network resources overloaded and disrupt the services. The open programmable APIs have opened new doors for attacks. ML techniques such as SVM, naive bayes, decision tree, KNN, and random forest have been used to address these SDN security issues (Zaman and Lung, 2018). Fig. 25 shows the classification of intrusion detection system (Ahmed et al., 2021). The IDS techniques are classified into two parts, namely, detection based and data source based. The detection based methods are further divided into anomaly based and signature based methods. The data source based methods analyze the information received from various input sources and then classify the normal or abnormal attacks. In the signature-based detection techniques, each signature (the attack method) is saved in the database, and each new signature is compared to the previously recorded signatures for misuse detection (Ahmed et al., 2021). The pattern matching approaches work by considering the susceptible hosts which have behavioral patterns that are similar to normal hosts (Ahmed et al., 2021). The expert system is used to make decisions based on behavioral characteristics.

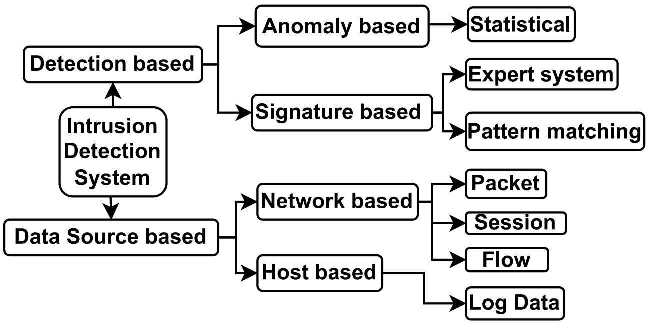


Fig. 25. Intrusion detection system classification.

The network-based detection methods are capable of recognizing specific types of protocol-based and network-based threats by analyzing network traffic data. The packet-based methods inspect the predefined rules or characteristics in the entire packet (payload and header) for application-layer information (Ahmed et al., 2021). The session-based approaches examine features of a session's header field and payloads to identify anomalous traffic data (Ahmed et al., 2021). The flow-based methods examine only the packet's header by using statistical approaches and do not check the internal payload (Ahmed et al., 2021). The host-based detection methods keep track of audit trails, log files, and system calls generated by running processes. It creates a historical list of events and activities which can be examined for abnormalities and intrusion. The log-based detection method takes the user's logs data after signing into a system to detect threats in a specific environment (Ahmed et al., 2021).

SDN faces different types of DDoS attacks due to communication between switches and controller at application plane (unauthorized applications attacks), data plane (flow table overflow, spoofing of the switch, buffer saturation attacks), control plane (packet-in-flooding, saturation of controller attacks), and communication links (congestion of southbound API attacks) (Alamri et al., 2021). Table 16 shows various intrusion detection techniques in Cloud-SDN-IoT environment using ML and AI techniques. It summaries major intrusion types, algorithms, datasets, tools and extracted features for intrusion detection using supervised ML, unsupervised ML, reinforcement learning, DNN, ensemble learning and hybrid models.

5.3. Security threats in federated learning and Cloud-Edge-Fog

There are various types of security threats for federated learning i.e., insider attacks (single, byzantine, sybil attacks), outsider attacks, free riding attacks, eavesdropping, backdoor attacks, malicious-server, communication bottleneck, semi-honest attacks (passive adversaries), malicious attacks (active adversaries), training-phase attacks, inference-phase attacks (evasion or exploratory), poisoning-attacks, and inference-attacks (infer-class representative, membership, properties, training data inputs and labels attacks, etc.) (Lyu et al., 2020). The insider attacks are performed by FL server and agents. The outsider attacks are done by different eavesdroppers on the communication medium. By generating several malicious players to train the global model in which the malicious player attempts to increase the efficiency of data poisoning during the sybil attacks (Lyu et al., 2020). In the free riding attacks, the malicious players use the global model without participating in the learning process (Lyu et al., 2020). The backdoor-attacks are the method of injecting a malicious job into an existing model while keeping the accuracy of the original job i.e., trojans threats (Lyu et al., 2020). The malicious-servers can grab confidential user information or alter the global ML model and integrate malicious jobs in the global ML model using shared computing capabilities (Lyu et al., 2020). The communication bottleneck attacks the communication bandwidth in FL environment. It has been solved by compressing data and enabling specific outputs to be transmitted back to the central system. There are two categories of poisoning-attacks, namely, data-poisoning and model-poisoning. In the data-poisoning attacks, low-quality data are used for training the ML model to tamper the model i.e., data injection, data modification, etc. The adversary has the ability to alter the updated global model parameters in the model-poisoning attacks (Lyu et al., 2020). The inference-attacks are key threat to privacy in FL, and they have been performed by a malicious participants or centralized server in the FL process (Lyu et al., 2020). The major FL defense or prevention techniques are sniper (poisoning), knowledge-distillation (eavesdropping inference-GAN), anomaly detection (poisoning attacks), moving target defense (eavesdropping), pruning (backdoor attacks), data sanitization (poisoning attacks), trusted execution environment (malicious server), Fools-Gold (sybil and backdoor poisoning attacks), and autoencoder-based anomaly-detection algorithms (free riding attacks) (Mothukuri et al., 2021).

There are various security threats at different layers of Edge-Fog paradigm i.e., application layer (HTTP flood, SQL injection, malwares), presentation layer (hypervisor, data leakage, VM-Based), transport layer (TCP flood, UDP flood, session-hijacking), network layer (DoS attack, man-in-the-middle attacks, spoofing attacks), physical layer (eavesdropping, tampering, replay attack), etc. (Ometov et al., 2022). In the edge computing, the application layer attacks have been addressed by different methods. The HTTP flood attack is tackled by filtering techniques and intrusion detection system. The SQL injection attack are tackled by applying random noise and delay to circuits in order to reduce data leaks (Ometov et al., 2022). It also uses behavior and signature based detection methods, encryption methods, light weight user authentication method, etc., to detect attacks at different layers (Ometov et al., 2022). In the fog computing, VM-based attacks have been addressed using ML techniques at presentation layer. It uses intrusion detection and prevention approaches to detect the anomalies. It detects DDoS attack by training the Self Organizing Map (SOM) network with traffic flow attributes. ML techniques (CNN, SVM, random forest, XGBoost, etc.) have been used to classify the attacks (Ometov et al., 2022). There are various security threats in fog computing such as attacks, cyber attack, vulnerable fog nodes, deep-ransomware attacks, and account-hijacking phishing-attacks (Das and Guria, 2022). To address these attacks various ML techniques have been proposed in the literature. The spam detection have been performed in outgoing SMS-messages using Naive Bayes (NB), Naive Bayes Multinomial (NBM), and SVM with 98.8% accuracy in Android devices (Das and Guria, 2022). In the Fog-based IoT network, different cyber attacks have been detected using Random Forest (RF) with 99.34% accuracy (Das and Guria, 2022). For the vulnerable fog nodes having limited resources used NN with back-propagation, SVM, Extreme Learning Machine (EML), and achieved 99.07% accuracy (Das and Guria, 2022). In the distributed IoT node, cyber attacks have been detected using deep learning and softmax ML classification techniques with 99.2% accuracy (Das and Guria, 2022). At the fog layer, CNN has achieved 99.6% accuracy for deep-ransomware threat-detection on the real dataset. Multilayer feedforward-ANN has achieved 98.36% accuracy on the phishing dataset for detection of account-hijacking phishing-attacks on the fog network (Das and Guria, 2022).

The IIoT-Digitaltwin paradigm faces different types of security threats. The reconnaissance attacks performed by scanning the network, zero-day vulnerability, enumerating services, etc. (Suhail et al., 2022). Another type of design phase attack performed by utilizing different twins' specification and ML based operation information (Suhail et al., 2022). In the digitaltwin, majorly, Triton and Stuxnet inspired malware approaches have been used for launching the attacks and break the isolation mechanisms (air gaps, virtualization, sandboxes, etc.) (Suhail et al., 2022). Gamification in cybersecurity attempts to offer security analysts with a safe virtual training mode. Gamification

Table 16
Intrusion detection techniques in Cloud-SDN-IoT using ML and DL.

Ref.	Intrusion types	Algorithms	Dataset and tools	Features	Ref.	Intrusion types	Algorithms	Dataset and tools	Features
Cheng et al. (2021)	Malicious payload identification	DT, RF, KNN, SVM, LR, Multinomial-NB (Supervised learning)	CTU-BOTNET (Mininet)	Term frequency, linguistic features	Maeda et al. (2019)	Detecting botnets	MLP (DNN)	CTU-13, ISOT	Statistical, IP, TCP, UDP and raw feature set
Boero et al. (2017)	Malware intrusions detection	SVM (Supervised learning)	Mininet	Number of packets and bytes, flow duration, byte rate, average length of the packet, packet rate, first packet length	Novaes et al. (2020)	Detect port scan and DDoS attack	LSTM (DNN)	CIC-DDoS2019 (Mininet)	Time-stamp features
Nanda et al. (2016)	Identify malicious hosts	DT, NB, decision table, BayesNet (Supervised learning)	Log data	Invader IP, compromised host, Number of efforts in an outbreak	Dey and Rahman (2019)	Flow-based anomaly detection	RF, GRU-LSTM (DNN)	NSL-KDD	Nominal, numeric, binary features
Satheesh et al. (2020)	Detect the intruder anomalies	RF, NB, BayesNet, Part (Supervised learning)	NSL-KDD (Mininet)	Nominal, numeric, binary features	Dawoud et al. (2019)	Recognize anomalies	AutoEncoder, K-Means (DNN)	KDD9́9	Feature set of KDD9́9
Ajaeiya et al. (2017)	Flow-based IDS	SVM, DT, RF, KNN and bagged trees (Supervised learning)	–	Duration, packet count, byte count, Src/Dst IP, protocol, Src and Dst port	Deepa et al. (2019)	DDoS attack detection	SVM, KNN, NB, SOM (Ensemble learning)	CAIDA 2016 (Mininet)	-
Bani-talebi Dehkordi et al. (2021)	Detect DDoS attack	J48, BayesNet, random tree, REPTree, NB, LR (Supervised learning)	UNB-ISCX (Mininet)	Statistical, IP, TCP, UDP, raw features	Alamri and Thayananthan (2020)	Detect DDoS attack	XGBoost (Ensemble learning)	CIC-DDoS-2019 (Mininet)	CICDDoS2019 features
Sebbar et al. (2020)	Man-in-the-middle attack recognition	RF (Supervised learning)	Self-collected from SDN traffic (Mininet)	–	Alzahrani and Alenazi (2021)	Attack detection	DT, RF, XGBoost (Ensemble learning)	NSL-KDD	Time-duration, protocol-type, source bytes, service count
Wang et al. (2018a)	Detecting low-rate DDoS attack	HMM, SOM, KNN, back-propagation (Unsupervised learning)	Synthetically created (Mininet)	Src and Dst IP	Alzahrani and Alenazi (2021)	DDoS attack detection	RF, KNN, bagging, SVD (Ensemble learning)	KDD CUP 1999	TCP connection basic, connective contents, time and host-based traffic features
Barki et al. (2016)	Identify the attacker hosts	K-Means, K-Medoids, NB, KNN (Unsupervised learning)	Self-collected (Mininet)	-	Phan and Park (2019)	DDoS attack defense	SVM, SOM (Ensemble learning)	CAIDA	Flow attributes
Jevtic et al. (2018)	DDoS attack recognition	SOM (Unsupervised learning)	Simulated data (NS-3)	-	Alamri and Thayananthan (2020)	Protecting against DDoS attacks	XGBoost (Ensemble learning)	CICDDoS2019, CAIDA	Network traffic, time, byte rate
Sampaio et al. (2018)	Anomaly Detection	RL	Self-generated (Mininet)	-	Nam et al. (2018)	Flooding attack detection	SOM, KNN (Hybrid models)	Caida DDoS attack 2007	Entropy of source IP, entropy of source port, entropy of destination port, entropy of packet protocol, total number of packets
Zolotukhin et al. (2020)	Attack detection and alleviation	DQN, PPO (Reinforcement learning)	Self-generated	Port, TCP flags, packet size, packet count, security logs and alerts, protocol, and SDN flows	Malik et al. (2020)	Reconnaissance and surveillance recognition	LSTM, CNN (Hybrid models)	CICIDS2017	Flow-based features

(*continued on next page*)

Table 16 (*continued*).

Ref.	Intrusion types	Algorithms	Dataset and tools	Features	Ref.	Intrusion types	Algorithms	Dataset and tools	Features
Akbari et al. (2020)	Threats detection	Neural-fitted Q-learning (Reinforcement learning)	Simulated traffic (Mininet)	–	Ding et al. (2020)	Abnormal flow detection	CNN, FCN (Hybrid models)	UNSW-NB15	Flow-based features
Phan et al. (2019)	DoS attack	Q-Learning (Reinforcement learning)	Simulated traffic (MaxiNet)	Average packets per flow, average packet size per flow, packet change ratio, flow change ratio, average duration per flow, percentage of pair-flows, growth of different ports, average flow inter-arrival time, fraction of TCP flows over total incoming flows and entropy of incoming flows	Garg et al. (2019)	Suspicious flow recognition	RBM, gradient-descent-based SVM (Hybrid models)	Real-time data traffic	Real-time, content-based, host-based, basic
Niyaz et al. (2016)	DDoS attack	SAE (DNN)	Real and private network data	TCP flow, UDP flows, ICMP flows	Khan and Akhunzada (2021)	Malware detection	CNN, LSTM (Hybrid models)	IoT-23	Flow-based features

uses Generative Adversarial Networks (GANs) deep learning methods for task execution (attacking and defending) (Suhail et al., 2022). Another type of scaling and ramp attacks manipulate the actual signal by injecting malicious data in IIoT system. The intrusion detection technique using digitaltwin utilizes support vector machine for classification of these type of attacks (Akbarian et al., 2020). On the digitaltwin-based CPS, there can be two types of attacks, namely, trivial attacks and non-trivial attacks (Wei et al., 2022). Typically, trivial attacks distort noncontinuous spikes present in the reported functional data values during digital twin procedure to affect the system (i.e., attack on the actuators). The non-trivial attacks usually distort minor changes present in the functional values within the threshold limit and, later over the time, make major disruption in the system (Wei et al., 2022). To detect these anomalies, deep learning method Natural-Gradient-Boosting (NGBoost) has been proposed that applies multi-attribute boosting using natural gradients for integration of the base learner (Wei et al., 2022). To secure the IIoT system against different cyber threats, various ML techniques have been utilized in the literature (Khan et al., 2022). There are various cyber threats in IIoT network have been studied such as botnet attacks, malicious domain, DoS attacks, login attacks, probe attacks, detection of Mirai attacks, Khan et al. (2022). To address these threats, the ML techniques have been widely applied such as LSTM, hybrid LSTM-RNN-CNN methods, RBM, and Gated Recurrent Unit (GRU) (Khan et al., 2022). Other type of security threats such as botnet-traffic analysis, botnet-attack identification, DDoS attack-detection, IIoT botnet-detection (Gafgyt and Mirai malwares) have been studied (Khan et al., 2022). To addressed these threats, CNN and auto-encoder, decision tree, random forest, hybrid-deep-learning methods (LSTM-DNN) have been applied in the state of the art (Khan et al., 2022).

6. Research gaps, challenges and future research directions

In this paper, we thoroughly investigated various emerging cloud computing integrated paradigms and explored various ML techniques usages in the same set of paradigms. Based on the above study, this section presents the identified research gaps, challenges, and future research directions for each integrated paradigm.

6.1. Cloud-ML

- Due to enormous growth in mobile data traffic caching techniques are used to store the data near the edge devices. The efficient techniques for edge caching can be developed utilizing ML techniques to reduce the mobile data traffic.
- For optimal control of data centers, cloud data-center utilize statistical techniques. The combination of ML with statistical techniques improves the prediction rate that gives another direction for investigation.
- For the workload management, ML and Prediction of Query Runtime (PQR) combination can be utilized to predict the query execution time considering parameters such as data skew and variation in workload for improving the prediction accuracy of query execution time.
- As in Mobile Edge Computing (MEC), data sources are outside the cloud, therefore, Federated Learning (FL) techniques can be utilized for learning. However, at the wider scale of FL, there are research opportunities to deal with challenges such as resource allocation, privacy, security, and communication cost considering the heterogeneous environment in FL.
- To fulfill the growing service demand in the cloud, Unified Reinforcement Learning (URL) adjusts the VM configuration and also tunes the application parameters for real-time auto-configuration of VMs. The URL technique lacks in performance for multiple physical machines for VM cluster configurations, therefore, the development of distributed RL algorithm need to be explored.
- A better resource allocation strategy to address the issues related to the scarcity of resources, resource contention, resource fragmentation, over and under-provisioning can be developed utilizing Parallel Q-learning techniques through the live virtualized test-bed environment.
- Additionally, some other domains such as deep learning for Non-IID (Independent and Identically Distributed) training data, heterogeneous backhaul or fronthaul management, mobile edge caching and computing, data pre-fetching, resource auto-scaling, usage prediction, and resource optimization using ML techniques can be explored to reduce the cost and increase the revenue.

6.2. Edge-Fog-IoT-DL

- MEC empowers Mobile Cloud Computing (MCC) by bringing cloud resources to the edge of the Radio Access Network (RAN) in 5G networks. The MEC requires an efficient mechanism for computing and network resource allocations. To address this, Deep Reinforcement Learning-based Resource Allocation (DRLRA) technique can be developed for resource allocation to reduce average service time and balanced resource usage in a changing MEC environment.
- As the security is a big threat in fog computing. Anomaly detection can be done efficiently by using ML/DL in the field of network security.
- There exists several challenges in federated learning such as data skewing, data imbalance, data heterogeneity, communication security, accountability, system heterogeneity and architectures, computation power, level of trust, model performance, network connectivity, trace-ability, dropped participants, privacy, unlabeled data handling,etc., which are required to be solved considering application requirements.
- In fog integrated architectures, Linear Regression (LR) technique enhances privacy protection. The stacked auto-encoder is mostly used to identify distributed attacks. In general, a stacked auto-encoder is made up of many layers of sparse auto-encoders, with each hidden layer output linked to the input of the next hidden layer. The investigation for packages, libraries, models, frameworks, and development of devices can be explored by considering the factors such as processing capacity, memory, and bandwidth requirement. It is necessarily required for the models and frameworks that support integrated multi-platforms.
- DL-based model can be designed considering IoT integrated edge computing environment to maximize the network speed and ensure user privacy during data uploading. Further, the reduction in the model size, optimized memory utilization, and communication costs are the key future research domain to develop the model which supports distributed learning architectures in real-time environments.

6.3. Fog-Cloud-ML

The integration of fog computing with CPS applications (smart grid, smart manufacturing, and transportation) reduces latency and improves response time. Further, ML techniques can be utilized for source diagnosis, machine failure detection, and quality prediction in the manufacturing domain; load forecasting, grid resilience, and electrical power in smart grids domain; non-technical losses detection, driver monitoring with traffic congestion, pollution levels, and traffic accidents in the transportation domain utilizing the integration of merging cloud computing paradigms to achieve the required QoS parameters.

6.4. Mist-Cloud

Mist computing integration with cloud/fog archetypes takes advantage of the processing and networking capacity of the devices at the edge of IoT networks and deals with the latency-sensitive requirements of the applications utilizing mist-based IoT framework, particularly in real-time applications. However, an integration of ML needs to be explored extensively that gives another future research dimension to deal with bandwidth prediction, load prediction, migration requirement prediction, risk of delay analysis, workload classification, real-time processing requirement prediction, resource metering, energy efficiency, and latency requirements analysis, reliability and service fault prediction.

6.5. IoT-ML

- In a typical IoT network, ML techniques are quite useful to effectively address security and privacy-related issues by identifying malicious code attacks, predicting user behaviors, making power analysis, and intrusion detection. Several ML models have been applied for addressing IoT devices' security and privacy concerns. However, due to the heavy computational requirements of the ML models, such models are not responsive in real-time. Therefore, the development of novel, dedicated, compressed, and light-weighted ML models are extremely required in this domain with acceptable accuracy requirements.
- Further existing ML models using IoT based can be redesigned to enhance the accuracy and multi-functional requirements by increasing multi-feature in the existing data-sets or using the large-scale database or utilizing tensor-flow learning techniques.
- For personalized computing, network congestion, overhead, and interoperability issues are the big concerns in heterogeneous IoT networks. To overcome them, several ML algorithms i.e., ANN, SVM, naive bayes, etc., can be utilized for clustering-based analysis with advanced deep learning and intelligent optimization techniques.
- ML algorithms in IoT require a high-quality training data-set to attain high accuracy. But still, no open-source data sets of IoT network data are available. That can be one possible way of contributing by creating such a database for several applications.
- IoT platforms are required to investigate ML-based performance analysis tools for testing IoT platforms to improve QoS, smart applications inter/intra-connectivity, and provisioning services.
- ML has been used to recover and compress data by reducing latency and sent-data size. CNN performs better for retrieving features and patterns from large datasets in the spatial domain, while RNN gives better results for making time relationships in the time domain. CNN and RNN techniques can be used to improve prediction accuracy.

6.6. SDN-IoT

SDN typically divides the IoT network into two planes: (i) data planes where nodes act as data forwarding nodes, and (ii) control planes where nodes act as network management nodes. Despite several advantages of SDN such as programmable interface, self-configuration, low-cost and efficient network management, and enabling network virtualization in the cloud, many gaps can be bridged via ML techniques to enhance the performance.

- In SDN based architectures for Cloud/Fog/IoT, ML-based techniques can be utilized for network traffic classification, intelligent routing, and its optimization, QoS/QoE prediction (to meet enhance user or service provider level satisfaction), virtual resource management, and security and privacy (network threats identification/prediction, intrusion-detection, etc.).
- Selection of the number of SDN controllers, and their appropriate position placement to optimize several QoS parameters such as energy, delay, routing hopes, and scalability in typical IoT networks, are the major challenges that can be tackled via ML/Soft computing based optimization techniques.
- Further, ML techniques can be utilized to configure the SDN controller to validate network management and ensure minimal reliability requirements to prevent any vulnerabilities and boost network availability.
- SDN controller may utilize bayesian algorithm and other similar ML algorithms found useful to track, monitor, and learn IoT device's behavior.

- Furthermore, in SDN, QoS management is a major concern, and key performance metrics including Round-Trip Time (RTT), jitter, delay, utilization, and packet drop can be explored using machine learning for root cause analysis, traffic congestion prediction, and correlation finding.
- For upgrading the data forwarding rules at the IoT devices or data plane, the analysis for mobility pattern of IoT devices is required using ML in an SDN-based IoT network. This will help in an efficient decision-making.

6.7. Digitaltwin-IIoT

The ML integration with digital-twin framework and industry 4.0 can significantly benefit the future industrial sectors. In this integration, there are issues related to multistage virtual model validation, design of efficient techniques, AI integration, and design of a unique structure for the industries in the IIoT setting. Further, other issues include information synthesis, real-time connection management, and synchronization in digitaltwin based IIoT framework. For this integration, researchers have good research opportunities to work for the digitaltwin implementation as data fusion and integration methods, rapid creation and management of digitaltwin models, ML-based model testing and model validation, ML-based Network congestion prediction, and ML-based protocol recommendation for accessing IoT devices and sensors via various protocols. Furthermore, the evaluation of predictive maintenance is possible by using ML techniques considering multidisciplinary settings in the design of digitaltwin, ML can be extensively utilized to handle large-dimensions dataset, latency lags, time-series data alignment, resource demand prediction, etc.

6.8. IIoT-Fog

An integration of IIoT-Fog and ML is very trending and full of research opportunity domains. In this environment, the utilization of fog servers enhanced response time for real-time applications, and ML can be used to perform prediction, analysis, and real-time learning capabilities. It makes the machines intelligent and responsive in real-time by minimizing the service time and machine failures. Further, in this integration, ML techniques can be effectively utilized to predict device failures in industries, optimize machine maintenance, increase automation with improved performance, analyze and decision making in real-time, recommend data routing, minimize energy consumption, and reduce network latency in the network.

7. Conclusion and discussion

In last decade, cloud computing has been emerged with several emerging paradigms such as edge, fog, mist, IoT, SDN, cybertwin, and industry 4.0 utilizing machine learning techniques. This survey, first, introduces briefly backgrounds of all cloud computing paradigms, and then identifies possible integration among them based on the recent state of the art study. To carry out this study, last five years (2017–21) research articles are explored thoroughly. Further, Google Trends are also investigated to measure the popularity of top Google search queries for each cloud computing paradigms. Based on the study, essential comparative study is prepared for the readers on several parameters to grasp the summary with minimal efforts. In particular, this study investigated the emerging cloud computing paradigms, their possible integration, and the scope of ML in the integration. This survey explored various integrated cloud computing paradigms with ML techniques used to solve various issues such as QoS requirements, dynamic scheduling, workload management, resource allocation decisions, real-time optimal resource provisioning, communication cost, energy, response time, anomaly detection, offloading strategies etc. in integrated cloud computing paradigms. Based on the comprehensive study, we identified the research gaps, challenges, and future directions for each identified integration of emerging cloud computing paradigms utilizing most dominant problem solving technology i.e., ML. This study would essentially assist the researchers to carry-forward their research in integrated cloud computing paradigms as understanding the fundamentals, archetypes, current integration and research trends, scope of other integration, role of ML to solve various problems, new challenges, and future research directions.

This study is limited to the research questions raised in the review methodology (Section 1.3). The integration among emerging Cloud computing paradigms is too vast; therefore, this survey lacks in all possible combinations among the emerging Cloud computing paradigms and their architectures. Further, there are a lot of Quality of Service (QoS) parameters considering service provider and client upcoming requirements; it could not cover all of those. Thus, an extensive study will be carried out in depth to understand the role of machine learning based techniques in effective utilization of the computing resources and offering QoS. Furthermore, hybrid machine learning models, in the same domain, are just about to be practiced, which also need to be explored in more depth. All of these limitations of this survey study give new research direction to future potential survey studies.

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.

References

Aazam, M., Zeadally, S., Harras, K.A., 2018. Fog computing architecture, evaluation, and future research directions. IEEE Commun. Mag. 56 (5), 46–52.

Abar, T., Letaifa, A.B., El Asmi, S., 2017. Machine learning based QoE prediction in SDN networks. In: 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC). IEEE, pp. 1395–1400.

Abeshu, A., Chilamkurti, N., 2018. Deep learning: The frontier for distributed attack detection in fog-to-things computing. IEEE Commun. Mag. 56 (2), 169–175.

Abkenar, F.S., Jamalipour, A., 2019. EBA: energy balancing algorithm for fog-IoT networks. IEEE Internet Things J. 6 (4), 6843–6849.

Ahmad, R., Alsmadi, I., 2021. Machine learning approaches to IoT security: A systematic literature review. Internet Things 14, 100365.

Ahmed, M.E., Kim, H., Park, M., 2017. Mitigating DNS query-based DDoS attacks with machine learning on software-defined networking. In: MILCOM 2017-2017 IEEE Military Communications Conference (MILCOM). IEEE, pp. 11–16.

Ahmed, M., Shatabda, S., Islam, A., Robin, M., Islam, T., et al., 2021. Intrusion detection system in software-defined networks using machine learning and deep learning techniques–a comprehensive survey.

Ajaeiya, G.A., Adalian, N., Elhajj, I.H., Kayssi, A., Chehab, A., 2017. Flow-based intrusion detection system for SDN. In: 2017 IEEE Symposium on Computers and Communications (ISCC). IEEE, pp. 787–793.

Akbari, I., Tahoun, E., Salahuddin, M.A., Limam, N., Boutaba, R., 2020. ATMoS: Autonomous threat mitigation in SDN using reinforcement learning. In: NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium. IEEE, pp. 1–9.

Akbarian, F., Fitzgerald, E., Kihl, M., 2020. Intrusion detection in digital twins for industrial control systems. In: 2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM). IEEE, pp. 1–6.

Alam, K.M., El Saddik, A., 2017. C2PS: A digital twin architecture reference model for the cloud-based cyber-physical systems. IEEE Access 5, 2050–2062.

Alamri, H.A., Thayananthan, V., 2020. Bandwidth control mechanism and extreme gradient boosting algorithm for protecting software-defined networks against DDoS attacks. IEEE Access 8, 194269–194288.

Alamri, H.A., Thayananthan, V., Yazdani, J., 2021. Machine learning for securing SDN based 5G network. Int. J. Comput. Appl. 174 (14), 9–16.

Alcaraz, C., Agudo, I., Nunez, D., Lopez, J., 2011. Managing incidents in smart grids a la cloud. In: 2011 IEEE Third International Conference on Cloud Computing Technology and Science. IEEE, pp. 527–531.

Alexopoulos, K., Nikolakis, N., Chryssolouris, G., 2020. Digital twin-driven supervised machine learning for the development of artificial intelligence applications in manufacturing. Int. J. Comput. Integr. Manuf. 33 (5), 429–439.

Alsharif, M., Rawat, D.B., 2021. Study of machine learning for cloud assisted iot security as a service. Sensors 21 (4), 1034.

Alzahrani, A.O., Alenazi, M.J., 2021. Designing a network intrusion detection system based on machine learning for software defined networks. Future Internet 13 (5), 111.

Andročec, D., Vrček, N., 2018. Machine learning for the internet of things security: a systematic. In: 13th International Conference on Software Technologies. http://dx.doi.org/10.5220/0006841205970604.

Arafsha, F., Laamarti, F., El Saddik, A., 2019. Cyber-physical system framework for measurement and analysis of physical activities. Electronics 8 (2), 248.

Ardanza, A., Moreno, A., Segura, A., de la Cruz, M., Aguinaga, D., 2019. Sustainable and flexible industrial human machine interfaces to support adaptable applications in the Industry 4.0 paradigm. Int. J. Prod. Res. 57 (12), 4045–4059.

Arockiam, L., Monikandan, S., Parthasarathy, G., 2011. Cloud computing: a survey. Int. J. Internet Comput. 1 (2), 26–33.

Asif-Ur-Rahman, M., Afsana, F., Mahmud, M., Kaiser, M.S., Ahmed, M.R., Kaiwartya, O., James-Taylor, A., 2018. Toward a heterogeneous mist, fog, and cloud-based framework for the internet of healthcare things. IEEE Internet Things J. 6 (3), 4049–4062.

Asthana, S., Megahed, A., Strong, R., 2017. A recommendation system for proactive health monitoring using IoT and wearable technologies. In: 2017 IEEE International Conference on AI & Mobile Services (AIMS). IEEE, pp. 14–21.

Azimi, I., Anzanpour, A., Rahmani, A.M., Pahikkala, T., Levorato, M., Liljeberg, P., Dutt, N., 2017. HiCH: Hierarchical fog-assisted computing architecture for healthcare IoT. ACM Trans. Embed. Comput. Syst. (TECS) 16 (5s), 1–20.

Baldini, G., Dimc, F., Kamnik, R., Steri, G., Giuliani, R., Gentile, C., 2017. Identification of mobile phones using the built-in magnetometers stimulated by motion patterns. Sensors 17 (4), 783.

Banitalebi Dehkordi, A., Soltanaghaei, M., Boroujeni, F.Z., 2021. The DDoS attacks detection through machine learning and statistical methods in SDN. J. Supercomput. 77 (3), 2383–2415.

Bankole, A.A., Ajila, S.A., 2013. Predicting cloud resource provisioning using machine learning techniques. In: 2013 26th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE). IEEE, pp. 1–4.

Bao, J., Guo, D., Li, J., Zhang, J., 2019. The modelling and operations for the digital twin in the context of manufacturing. Enterp. Inf. Syst. 13 (4), 534–556.

Barik, R.K., Misra, C., Lenka, R.K., Dubey, H., Mankodiya, K., 2019. Hybrid mist-cloud systems for large scale geospatial big data analytics and processing: opportunities and challenges. Arab. J. Geosci. 12 (2), 32.

Barki, L., Shidling, A., Meti, N., Narayan, D., Mulla, M.M., 2016. Detection of distributed denial of service attacks in software defined networks. In: 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE, pp. 2576–2581.

Barrett, E., Howley, E., Duggan, J., 2013. Applying reinforcement learning towards automating resource allocation and application scalability in the cloud. Concurr. Comput.: Pract. Exper. 25 (12), 1656–1674.

Battistoni, P., Sebillo, M., Vitiello, G., 2019. Computation offloading with MQTT protocol on a fog-mist computing framework. In: International Conference on Internet and Distributed Computing Systems. Springer, pp. 140–147.

Bhunia, S.S., Gurusamy, M., 2017. Dynamic attack detection and mitigation in IoT using SDN. In: 2017 27th International Telecommunication Networks and Applications Conference (ITNAC). IEEE, pp. 1–6.

Bi, Y., Han, G., Lin, C., Deng, Q., Guo, L., Li, F., 2018. Mobility support for fog computing: An SDN approach. IEEE Commun. Mag. 56 (5), 53–59.

Bodík, P., Griffith, R., Sutton, C.A., Fox, A., Jordan, M.I., Patterson, D.A., 2009. Statistical machine learning makes automatic control practical for internet datacenters. HotCloud 9, 12.

Boero, L., Marchese, M., Zappatore, S., 2017. Support vector machine meets software defined networking in ids domain. In: 2017 29th International Teletraffic Congress (ITC 29), Vol. 3. IEEE, pp. 25–30.

Borthakur, D., Dubey, H., Constant, N., Mahler, L., Mankodiya, K., 2017. Smart fog: Fog computing framework for unsupervised clustering analytics in wearable internet of things. In: 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP). IEEE, pp. 472–476.

Boyes, H., Hallaq, B., Cunningham, J., Watson, T., 2018. The industrial internet of things (IIoT): An analysis framework. Comput. Ind. 101, 1–12.

Buyya, R., Broberg, J., Goscinski, A.M., 2010. Cloud Computing: Principles and Paradigms. John Wiley & Sons.

Byers, C.C., Wetterwald, P., 2015. Fog computing distributing data and intelligence for resiliency and scale necessary for iot: The internet of things (ubiquity symposium). Ubiquity 2015 (November), 1–12.

Carner, J., Mestres, A., Alarcón, E., Cabellos, A., 2017. Machine learning-based network modeling: An artificial neural network model vs a theoretical inspired model. In: 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN). IEEE, pp. 522–524.

Chafii, M., Bader, F., Palicot, J., 2018. Enhancing coverage in narrow band-IoT using machine learning. In: 2018 IEEE Wireless Communications and Networking Conference (WCNC). IEEE, pp. 1–6.

Chakraborty, S., Senjyu, T., Yona, A., Saber, A., Funabashi, T., 2011. Solving economic load dispatch problem with valve-point effects using a hybrid quantum mechanics inspired particle swarm optimisation. IET Gener. Transm. Distrib. 5 (10), 1042–1052.

Chalapathi, G.S.S., Chamola, V., Vaish, A., Buyya, R., 2019. Industrial internet of things (IIOT) applications of edge and fog computing: A review and future directions. arXiv preprint arXiv:1912.00595.

Chaudhary, P., Gupta, B.B., 2019. Ddos detection framework in resource constrained internet of things domain. In: 2019 IEEE 8th Global Conference on Consumer Electronics (GCCE). IEEE, pp. 675–678.

Chekired, D.A., Khoukhi, L., Mouftah, H.T., 2018. Industrial IoT data scheduling based on hierarchical fog computing: A key for enabling smart factory. IEEE Trans. Ind. Inf. 14 (10), 4590–4602.

Chen, M., Challita, U., Saad, W., Yin, C., Debbah, M., 2019. Artificial neural networks-based machine learning for wireless networks: A tutorial. IEEE Commun. Surv. Tutor. 21 (4), 3039–3071.

Chen, T., Li, M., Li, Y., Lin, M., Wang, N., Wang, M., Xiao, T., Xu, B., Zhang, C., Zhang, Z., 2015. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. arXiv preprint arXiv:1512.01274.

Cheng, Q., Wu, C., Zhou, H., Kong, D., Zhang, D., Xing, J., Ruan, W., 2021. Machine learning based malicious payload identification in software-defined networking. J. Netw. Comput. Appl. 192, 103186.

Čolaković, A., Hadžialić, M., 2018. Internet of Things (IoT): A review of enabling technologies, challenges, and open research issues. Comput. Netw. 144, 17–39.

Costa, C., Santos, M.Y., 2015. Improving cities sustainability through the use of data mining in a context of big city data.

Damjanovic-Behrendt, V., Behrendt, W., 2019. An open source approach to the design and implementation of digital twins for smart manufacturing. Int. J. Comput. Integr. Manuf. 32 (4–5), 366–384.

Das, S., Guria, P., 2022. Adaptation of machine learning in fog computing: An analytical approach. In: 2022 International Conference for Advancement in Technology (ICONAT). IEEE, pp. 1–11.

Datta, S., 2016. Emergence of digital twins: Is this the march of reason?

Datta, S.K., Bonnet, C., 2017. An edge computing architecture integrating virtual IoT devices. In: 2017 IEEE 6th Global Conference on Consumer Electronics (GCCE). IEEE, pp. 1–3.

Dawoud, A., Shahristani, S., Raun, C., 2019. Unsupervised deep learning for software defined networks anomalies detection. In: Transactions on Computational Collective Intelligence XXXIII. Springer, pp. 167–178.

Deepa, V., Sudar, K.M., Deepalakshmi, P., 2019. Design of ensemble learning methods for ddos detection in sdn environment. In: 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN). IEEE, pp. 1–6.

Deng, Y., 2019. Deep learning on mobile devices: a review. In: Mobile Multimedia/Image Processing, Security, and Applications 2019, Vol. 10993. International Society for Optics and Photonics, p. 109930A.

Denos, B.R., Sommer, D.E., Favaloro, A.J., Pipes, R.B., Avery, W.B., 2018. Fiber orientation measurement from mesoscale CT scans of prepreg platelet molded composites. Composites A 114, 241–249.

Dey, S.K., Rahman, M.M., 2019. Effects of machine learning approach in flow-based anomaly detection on software-defined networking. Symmetry 12 (1), 7.

Ding, K., Chan, F.T., Zhang, X., Zhou, G., Zhang, F., 2019. Defining a digital twin-based cyber-physical production system for autonomous manufacturing in smart shop floors. Int. J. Prod. Res. 57 (20), 6315–6334.

Ding, P., Li, J., Wang, L., Wen, M., Guan, Y., 2020. HYBRID-CNN: An efficient scheme for abnormal flow detection in the SDN-based smart grid. Secur. Commun. Netw. 2020.

Drolia, U., Guo, K., Narasimhan, P., 2017. Precog: Prefetching for image recognition applications at the edge. In: Proceedings of the Second ACM/IEEE Symposium on Edge Computing. pp. 1–13.

Du, J., Zhao, L., Feng, J., Chu, X., 2017. Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee. IEEE Trans. Commun. 66 (4), 1594–1608.

El-Hasnony, I.M., Mostafa, R.R., Elhoseny, M., Barakat, S.I., 2021. Leveraging mist and fog for big data analytics in IoT environment. Trans. Emerg. Telecommun. Technol. 32 (7), e4057.

Ferguson, S., Bennett, E., Ivashchenko, A., 2017. Digital twin tackles design challenges. World Pumps 2017 (4), 26–28.

Ferreira, J., Carvalho, E., Ferreira, B.V., de Souza, C., Suhara, Y., Pentland, A., Pessin, G., 2017. Driver behavior profiling: An investigation with different smartphone sensors and machine learning. PLoS One 12 (4), e0174959.

Garg, S., Kaur, K., Kumar, N., Rodrigues, J.J., 2019. Hybrid deep-learning-based anomaly detection scheme for suspicious flow detection in SDN: A social multimedia perspective. IEEE Trans. Multimed. 21 (3), 566–578.

Gebrie, M.T., Abie, H., 2017. Risk-based adaptive authentication for internet of things in smart home eHealth. In: Proceedings of the 11th European Conference on Software Architecture: Companion Proceedings. pp. 102–108.

Gong, Z., Gu, X., Wilkes, J., 2010. Press: Predictive elastic resource scaling for cloud systems. In: 2010 International Conference on Network and Service Management. Ieee, pp. 9–16.

Graves, D., 2019. Reactive vs. Proactive cybersecurity: 5 reasons why traditional security no longer works.

Gu, Y., Li, K., Guo, Z., Wang, Y., 2019. Semi-supervised K-means DDoS detection method using hybrid feature selection algorithm. IEEE Access 7, 64351–64365.

Gupta, S., Agrawal, A., Gopalakrishnan, K., Narayanan, P., 2015. Deep learning with limited numerical precision. In: International Conference on Machine Learning. PMLR, pp. 1737–1746.

Gupta, S., Kambli, R., Wagh, S., Kazi, F., 2014. Support-vector-machine-based proactive cascade prediction in smart grid using probabilistic framework. IEEE Trans. Ind. Electron. 62 (4), 2478–2486.

Gupta, C., Mehta, A., Dayal, U., 2008. PQR: Predicting query execution times for autonomous workload management. In: 2008 International Conference on Autonomic Computing. IEEE, pp. 13–22.

Gupta, C., Suggala, A.S., Goyal, A., Simhadri, H.V., Paranjape, B., Kumar, A., Goyal, S., Udupa, R., Varma, M., Jain, P., 2017. Protonn: Compressed and accurate knn for resource-scarce devices. In: International Conference on Machine Learning. PMLR, pp. 1331–1340.

Gurusamy, D., Deva Priya, M., Yibgeta, B., Bekalu, A., 2019. DDoS risk in 5G enabled IoT and solutions. Int. J. Eng. Adv. Technol. 8 (5), 1574–1578.

Haag, S., Anderl, R., 2018. Digital twin–Proof of concept. Manuf. Lett. 15, 64–66.

Hadidi, R., Cao, J., Ryoo, M.S., Kim, H., 2019. Robustly executing DNNs in IoT systems using coded distributed computing. In: Proceedings of the 56th Annual Design Automation Conference 2019. pp. 1–2.

Hadidi, R., Cao, J., Woodward, M., Ryoo, M.S., Kim, H., 2018. Distributed perception by collaborative robots. IEEE Robot. Autom. Lett. 3 (4), 3709–3716.

Han, W., Gu, Y., Zhang, Y., Zheng, L., 2014. Data driven quantitative trust model for the internet of agricultural things. In: 2014 International Conference on the Internet of Things (IOT). IEEE, pp. 31–36.

Hogan, M., Esposito, F., 2017. Stochastic delay forecasts for edge traffic engineering via Bayesian networks. In: 2017 IEEE 16th International Symposium on Network Computing and Applications (NCA). IEEE, pp. 1–4.

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.

Huang, C.-J., Wang, Y.-W., Guan, C.-T., Chen, H.-M., Jian, J.-J., 2013. Applications of machine learning to resource management in cloud computing. Int. J. Model. Optim. 3 (2), 148.

Huynh, M., Nguyen, P., Gruteser, M., Vu, T., 2015. Poster: Mobile device identification by leveraging built-in capacitive signature. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. pp. 1635–1637.

Hwang, R.-H., Peng, M.-C., Nguyen, V.-L., Chang, Y.-L., 2019. An LSTM-based deep learning approach for classifying malicious traffic at the packet level. Appl. Sci. 9 (16), 3414.

Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., Keutzer, K., 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. arXiv preprint arXiv:1602.07360.

Ioannou, C., Vassiliou, V., 2019. Classifying security attacks in IoT networks using supervised learning. In: 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS). IEEE, pp. 652–658.

Iorga, M., Feldman, L., Barton, R., Martin, M., Goren, N., Mahmoudi, C., 2017. The NIST Definition of Fog Computing. Tech. rep., National Institute of Standards and Technology.

Iorga, M., Feldman, L., Barton, R., Martin, M.J., Goren, N.S., Mahmoudi, C., et al., 2018. Fog computing conceptual model.

Islam, S., Keung, J., Lee, K., Liu, A., 2012. Empirical prediction models for adaptive resource provisioning in the cloud. Future Gener. Comput. Syst. 28 (1), 155–162.

Jain, S., Khandelwal, M., Katkar, A., Nygate, J., 2016. Applying big data technologies to manage QoS in an SDN. In: 2016 12th International Conference on Network and Service Management (CNSM). IEEE, pp. 302–306.

Jakkula, V., Cook, D., 2010. Outlier detection in smart environment structured power datasets. In: 2010 Sixth International Conference on Intelligent Environments. IEEE, pp. 29–33.

Jevtic, S., Lotfalizadeh, H., Kim, D.S., 2018. Toward network-based ddos detection in software-defined networks. In: Proceedings of the 12th International Conference on Ubiquitous Information Management and Communication. pp. 1–8.

Jiang, J., Lu, J., Zhang, G., Long, G., 2013. Optimal cloud resource auto-scaling for web applications. In: 2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing. IEEE, pp. 58–65.

Kafi, M.A., Challal, Y., Djenouri, D., Doudou, M., Bouabdallah, A., Badache, N., 2013. A study of wireless sensor networks for urban traffic monitoring: applications and architectures. Procedia Comput. Sci. 19, 617–626.

Khan, S., Akhunzada, A., 2021. A hybrid DL-driven intelligent SDN-enabled malware detection framework for Internet of Medical Things (IoMT). Comput. Commun. 170, 209–216.

Khan, W.U., Malik, J., Hasan, T., Bibi, I., Al-Wesabi, F.N., Dev, K., Huang, G., 2022. Securing industrial internet of things against botnet attacks using hybrid deep learning approach.

Khattak, H.A., Arshad, H., ul Islam, S., Ahmed, G., Jabbar, S., Sharif, A.M., Khalid, S., 2019. Utilization and load balancing in fog servers for health applications. EURASIP J. Wireless Commun. Networking 2019 (1), 1–12.

Kim, H.J., Jung, M.Y., Chin, W.S., Jang, J.W., 2017. Identifying service contexts for qos support in iot service oriented software defined networks. In: International Conference on Mobile, Secure, and Programmable Networking. Springer, pp. 99–108.

Kim, H., Shin, H., Kim, H.-s., Kim, W.-T., 2018. VR-CPES: A novel cyber-physical education systems for interactive VR services based on a mobile platform. Mob. Inf. Syst. 2018.

Kitchenham, B., Charters, S., 2007. Guidelines for performing systematic literature reviews in software engineering.

Kleberger, P., Olovsson, T., Jonsson, E., 2011. Security aspects of the in-vehicle network in the connected car. In: 2011 IEEE Intelligent Vehicles Symposium (IV). IEEE, pp. 528–533.

Kumar, A., Goyal, S., Varma, M., 2017. Resource-efficient machine learning in 2 kb ram for the internet of things. In: International Conference on Machine Learning. PMLR, pp. 1935–1944.

Kumar, N.M., Mallick, P.K., 2018. The Internet of Things: Insights into the building blocks, component interactions, and architecture layers. Procedia Comput. Sci. 132, 109–117.

Kumar, N., Mittal, S., Garg, V., Kumar, N., 2021. Deep reinforcement learning-based traffic light scheduling framework for SDN-enabled smart transportation system. IEEE Trans. Intell. Transp. Syst..

Kundu, S., Rangaswami, R., Gulati, A., Zhao, M., Dutta, K., 2012. Modeling virtualized applications using machine learning techniques. In: Proceedings of the 8th ACM SIGPLAN/SIGOPS Conference on Virtual Execution Environments. pp. 3–14.

Kurtz, A., Gascon, H., Becker, T., Rieck, K., Freiling, F., 2016. Fingerprinting mobile devices using personalized configurations. Proc. Priv. Enhanc. Technol. 2016 (1), 4–19.

La, Q.D., Ngo, M.V., Dinh, T.Q., Quek, T.Q., Shin, H., 2019. Enabling intelligence in fog computing to achieve energy and latency reduction. Digit. Commun. Netw. 5 (1), 3–9.

Laghari, A.A., He, H., Khan, A., Kumar, N., Kharel, R., 2018. Quality of experience framework for cloud computing (QoC). IEEE Access 6, 64876–64890.

Lahouar, A., Slama, J.B.H., 2015. Random forests model for one day ahead load forecasting. In: IREC2015 the Sixth International Renewable Energy Congress. IEEE, pp. 1–6.

Lee, J., Noh, S.D., Kim, H.-J., Kang, Y.-S., 2018. Implementation of cyber-physical production systems for quality prediction and operation control in metal casting. Sensors 18 (5), 1428.

Leng, J., Zhang, H., Yan, D., Liu, Q., Chen, X., Zhang, D., 2019. Digital twin-driven manufacturing cyber-physical system for parallel controlling of smart workshop. J. Ambient Intell. Humaniz. Comput. 10 (3), 1155–1166.

Letaifa, A.B., 2017. Adaptive QoE monitoring architecture in SDN networks: Video streaming services case. In: 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC). IEEE, pp. 1383–1388.

Li, D., Jayaweera, S.K., 2014. Machine-learning aided optimal customer decisions for an interactive smart grid. IEEE Syst. J. 9 (4), 1529–1540.

Li, Q., Li, W., Wang, J., Cheng, M., 2019. A SQL injection detection method based on adaptive deep forest. IEEE Access 7, 145385–145394.

Li, C., Mahadevan, S., Ling, Y., Choze, S., Wang, L., 2017. Dynamic Bayesian network for aircraft wing health monitoring digital twin. Aiaa J. 55 (3), 930–941.

Li, L., Ota, K., Dong, M., 2018a. Deep learning for smart industry: Efficient manufacture inspection system with fog computing. IEEE Trans. Ind. Inf. 14 (10), 4665–4673.

Li, L., Zhao, G., Blum, R.S., 2018b. A survey of caching techniques in cellular networks: Research issues and challenges in content placement and delivery strategies. IEEE Commun. Surv. Tutor. 20 (3), 1710–1732.

Liao, S.-w., Hung, T.-H., Nguyen, D., Chou, C., Tu, C., Zhou, H., 2009. Machine learning-based prefetch optimization for data center applications. In: Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis. pp. 1–10.

Lieber, D., Stolpe, M., Konrad, B., Deuse, J., Morik, K., 2013. Quality prediction in interlinked manufacturing processes based on supervised & unsupervised machine learning. Procedia Cirp 7, 193–198.

Lim, W.Y.B., Luong, N.C., Hoang, D.T., Jiao, Y., Liang, Y.-C., Yang, Q., Niyato, D., Miao, C., 2020. Federated learning in mobile edge networks: A comprehensive survey. IEEE Commun. Surv. Tutor. 22 (3), 2031–2063.

Lin, Y., Han, S., Mao, H., Wang, Y., Dally, W.J., 2017. Deep gradient compression: Reducing the communication bandwidth for distributed training. arXiv preprint arXiv:1712.01887.

Ling, X., Sheng, J., Baiocchi, O., Liu, X., Tolentino, M.E., 2017. Identifying parking spaces & detecting occupancy using vision-based IoT devices. In: 2017 Global Internet of Things Summit (GIoTS). IEEE, pp. 1–6.

Liu, Y., Jin, S., 2013. Application of Bayesian networks for diagnostics in the assembly process by considering small measurement data sets. Int. J. Adv. Manuf. Technol. 65 (9–12), 1229–1237.

Liu, F., Tang, G., Li, Y., Cai, Z., Zhang, X., Zhou, T., 2019a. A survey on edge computing systems and tools. Proc. IEEE 107 (8), 1537–1562.

Liu, C., Vengayil, H., Zhong, R.Y., Xu, X., 2018a. A systematic development method for cyber-physical machine tools. J. Manuf. Syst. 48, 13–24.

Liu, C., Vengayil, H., Zhong, R.Y., Xu, X., 2018b. A systematic development method for cyber-physical machine tools. J. Manuf. Syst. 48, 13–24.

Liu, Q., Zhang, H., Leng, J., Chen, X., 2019b. Digital twin-driven rapid individualised designing of automated flow-shop manufacturing system. Int. J. Prod. Res. 57 (12), 3903–3919.

Liu, J., Zhou, H., Tian, G., Liu, X., Jing, X., 2019c. Digital twin-based process reuse and evaluation approach for smart process planning. Int. J. Adv. Manuf. Technol. 100 (5–8), 1619–1634.

Liyanage, M., Chang, C., Srirama, S.N., 2018. Adaptive mobile Web server framework for Mist computing in the Internet of Things. Int. J. Pervasive Comput. Commun..

Lovas, R., Farkas, A., Marosi, A.C., Ács, S., Kovács, J., Szalóki, A., Kádár, B., 2018. Orchestrated platform for cyber-physical systems. Complexity 2018.

Lu, R., Brilakis, I., 2019. Digital twinning of existing reinforced concrete bridges from labelled point clusters. Autom. Constr. 105, 102837.

Lu, Y., Xu, X., 2019. Cloud-based manufacturing equipment and big data analytics to enable on-demand manufacturing services. Robot. Comput.-Integr. Manuf. 57, 92–102.

Luo, W., Hu, T., Zhang, C., Wei, Y., 2019a. Digital twin for CNC machine tool: modeling and using strategy. J. Ambient Intell. Humaniz. Comput. 10 (3), 1129–1140.

Luo, X., Yan, Q., Wang, M., Huang, W., 2019b. Using MTD and SDN-based honeypots to defend DDoS attacks in IoT. In: 2019 Computing, Communications and IoT Applications (ComComAp). IEEE, pp. 392–395.

Luping, W., Wei, W., Bo, L., 2019. CMFL: Mitigating communication overhead for federated learning. In: 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS). IEEE, pp. 954–964.

Lyu, L., Yu, H., Yang, Q., 2020. Threats to federated learning: A survey. arXiv preprint arXiv:2003.02133.

Ma, X., Wu, Y.-J., Wang, Y., Chen, F., Liu, J., 2013. Mining smart card data for transit riders' travel patterns. Transp. Res. C 36, 1–12.

MacDonald, C., Dion, B., Davoudabadi, M., 2017. Creating a digital twin for a pump. ANSYS Adv. 1, 8–10.

Madni, C.C., Azad, M., Lucero, S.D., 2019. Leveraging digital twin technology in model-based systems engineering. Systems 7 (1), 7.

Maeda, S., Kanai, A., Tanimoto, S., Hatashima, T., Ohkubo, K., 2019. A botnet detection method on SDN using deep learning. In: 2019 IEEE International Conference on Consumer Electronics (ICCE). IEEE, pp. 1–6.

Malik, J., Akhunzada, A., Bibi, I., Imran, M., Musaddiq, A., Kim, S.W., 2020. Hybrid deep learning: an efficient reconnaissance and surveillance detection mechanism in SDN. IEEE Access 8, 134695–134706.

Manyika, J., Chui, M., Bisson, P., Woetzel, J., Dobbs, R., Bughin, J., Aharon, D., 2015. Unlocking the Potential of the Internet of Things, Vol. 1. McKinsey Global Institute.

Mao, J., Chen, X., Nixon, K.W., Krieger, C., Chen, Y., 2017. Modnn: Local distributed mobile computing system for deep neural network. In: Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017. IEEE, pp. 1396–1401.

Mathonsi, T., Tshilongamulenzhe, T., Buthelezi, B., 2019. Blockchain security model for internet of things. In: The Proceedings of Academics World 158th International Conference. pp. 52–56.

Meidan, Y., Bohadana, M., Shabtai, A., Guarnizo, J.D., Ochoa, M., Tippenhauer, N.O., Elovici, Y., 2017. ProfilIoT: a machine learning approach for IoT device identification based on network traffic analysis. In: Proceedings of the Symposium on Applied Computing. pp. 506–509.

Mell, P., Grance, T., et al., 2011. The NIST definition of cloud computing.

Metri, P., Sarote, G., 2011. Privacy issues and challenges in cloud computing. Int. J. Adv. Eng. Sci. Technol. 5 (1), 5–6.

Miettinen, M., Marchal, S., Hafeez, I., Asokan, N., Sadeghi, A.-R., Tarkoma, S., 2017. Iot sentinel: Automated device-type identification for security enforcement in iot. In: 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS). IEEE, pp. 2177–2184.

Min, Q., Lu, Y., Liu, Z., Su, C., Wang, B., 2019. Machine learning based digital twin framework for production optimization in petrochemical industry. Int. J. Inf. Manage. 49, 502–519.

Mishra, P., Kumar, N., Godfrey, W.W., 2022. An evolutionary computing-based energy-efficient solution for IoT-enabled software-defined sensor network architecture. Int. J. Commun. Syst. e5111.

Monedero, I., Biscarri, F., León, C., Guerrero, J.I., Biscarri, J., Millán, R., 2012. Detection of frauds and other non-technical losses in a power utility using Pearson coefficient, Bayesian networks and decision trees. Int. J. Electr. Power Energy Syst. 34 (1), 90–98.

Monekosso, D.N., Remagnino, P., 2013. Data reconciliation in a smart home sensor network. Expert Syst. Appl. 40 (8), 3248–3255.

Moon, J.-K., Song, Y.-J., Kim, J.-M., 2016. A delegation model of healthcare system based of AB-PRE in fog computing environment. Adv. Sci. Lett. 22 (11), 3432–3436.

Moreno, A., Velez, G., Ardanza, A., Barandiaran, I., de Infante, A.R., Chopitea, R., 2017. Virtualisation process of a sheet metal punching machine within the Industry 4.0 vision. Int. J. Interact. Des. Manuf. (IJIDeM) 11 (2), 365–373.

Mothukuri, V., Parizi, R.M., Pouriyeh, S., Huang, Y., Dehghantanha, A., Srivastava, G., 2021. A survey on security and privacy of federated learning. Future Gener. Comput. Syst. 115, 619–640.

Murshed, M., Murphy, C., Hou, D., Khan, N., Ananthanarayanan, G., Hussain, F., 2019. Machine learning at the network edge: A survey. arXiv preprint arXiv:1908.00080.

Nahrstedt, K., Li, H., Nguyen, P., Chang, S., Vu, L., 2016. Internet of mobile things: Mobility-driven challenges, designs and implementations. In: 2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI). IEEE, pp. 25–36.

Nam, T.M., Phong, P.H., Khoa, T.D., Huong, T.T., Nam, P.N., Thanh, N.H., Thang, L.X., Tuan, P.A., Loi, V.D., et al., 2018. Self-organizing map-based approaches in DDoS flooding detection using SDN. In: 2018 International Conference on Information Networking (ICOIN). IEEE, pp. 249–254.

Nanda, S., Zafari, F., DeCusatis, C., Wedaa, E., Yang, B., 2016. Predicting network attack patterns in SDN using machine learning approach. In: 2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN). IEEE, pp. 167–172.

Natesha, B., Guddeti, R.M.R., 2021. Fog-based intelligent machine malfunction monitoring system for industry 4.0. IEEE Trans. Ind. Inf..

Nguyen, D.C., Ding, M., Pathirana, P.N., Seneviratne, A., Li, J., Poor, H.V., 2021. Federated learning for internet of things: A comprehensive survey. IEEE Commun. Surv. Tutor..

Nguyen, H.H., Mirza, F., Naeem, M.A., Nguyen, M., 2017. A review on IoT healthcare monitoring applications and a vision for transforming sensor data into real-time clinical feedback. In: 2017 IEEE 21st International Conference on Computer Supported Cooperative Work in Design (CSCWD). IEEE, pp. 257–262.

Ni, P., Zhang, C., Ji, Y., 2014. A hybrid method for short-term sensor data forecasting in Internet of Things. In: 2014 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD). IEEE, pp. 369–373.

Nikolakis, N., Alexopoulos, K., Xanthakis, E., Chryssolouris, G., 2019. The digital twin implementation for linking the virtual representation of human-based production tasks to their physical counterpart in the factory-floor. Int. J. Comput. Integr. Manuf. 32 (1), 1–12.

Ning, Z., Huang, J., Wang, X., 2019. Vehicular fog computing: Enabling real-time traffic management for smart cities. IEEE Wirel. Commun. 26 (1), 87–93.

Nishio, T., Yonetani, R., 2019. Client selection for federated learning with heterogeneous resources in mobile edge. In: ICC 2019-2019 IEEE International Conference on Communications (ICC). IEEE, pp. 1–7.

Niyaz, Q., Sun, W., Javaid, A.Y., 2016. A deep learning based DDoS detection system in software-defined networking (SDN). arXiv preprint arXiv:1611.07400.

Nobakht, M., Sivaraman, V., Boreli, R., 2016. A host-based intrusion detection and mitigation framework for smart home IoT using OpenFlow. In: 2016 11th International Conference on Availability, Reliability and Security (ARES). IEEE, pp. 147–156.

Novaes, M.P., Carvalho, L.F., Lloret, J., Proenca, M.L., 2020. Long short-term memory and fuzzy logic for anomaly detection and mitigation in software-defined network environment. IEEE Access 8, 83765–83781.

O'donovan, P., Gallagher, C., Bruton, K., O'Sullivan, D.T., 2018. A fog computing industrial cyber-physical system for embedded low-latency machine learning Industry 4.0 applications. Manuf. Lett. 15, 139–142.

Ogden, S.S., Guo, T., 2018. {MODI}: MObile deep inference made efficient by edge computing. In: {USENIX} Workshop on Hot Topics in Edge Computing (HotEdge 18).

Omer, M., Margetts, L., Hadi Mosleh, M., Hewitt, S., Parwaiz, M., 2019. Use of gaming technology to bring bridge inspection to the office. Struct. Infrastruct. Eng. 15 (10), 1292–1307.

Ometov, A., Molua, O.L., Komarov, M., Nurmi, J., 2022. A survey of security in cloud, edge, and fog computing. Sensors 22 (3), 927.

Osaba, E., Onieva, E., Moreno, A., Lopez-Garcia, P., Perallos, A., Bringas, P.G., 2016. Decentralised intelligent transport system with distributed intelligence based on classification techniques. IET Intell. Transp. Syst. 10 (10), 674–682.

Oyekan, J.O., Hutabarat, W., Tiwari, A., Grech, R., Aung, M.H., Mariani, M.P., López-Dávalos, L., Ricaud, T., Singh, S., Dupuis, C., 2019. The effectiveness of virtual environments in developing collaborative strategies between industrial robots and humans. Robot. Comput.-Integr. Manuf. 55, 41–54.

Oztemel, E., Gursev, S., 2020. Literature review of Industry 4.0 and related technologies. J. Intell. Manuf. 31 (1), 127–182.

Pandey, P.S., 2017. Machine learning and IoT for prediction and detection of stress. In: 2017 17th International Conference on Computational Science and Its Applications (ICCSA). IEEE, pp. 1–5.

Park, K.T., Im, S.J., Kang, Y.-S., Noh, S.D., Kang, Y.T., Yang, S.G., 2019. Service-oriented platform for smart operation of dyeing and finishing industry. Int. J. Comput. Integr. Manuf. 32 (3), 307–326.

Pasquini, R., Stadler, R., 2017. Learning end-to-end application qos from openflow switch statistics. In: 2017 IEEE Conference on Network Softwarization (NetSoft). IEEE, pp. 1–9.

Patel, M., Chaudhary, S., Garg, S., 2016. Machine learning based statistical prediction model for improving performance of live virtual machine migration. J. Eng. 2016.

Patel, H.J., Temple, M.A., Baldwin, R.O., 2014. Improving ZigBee device network authentication using ensemble decision tree classifiers with radio frequency distinct native attribute fingerprinting. IEEE Trans. Reliab. 64 (1), 221–233.

Perez, D., Astor, M.A., Abreu, D.P., Scalise, E., 2017. Intrusion detection in computer networks using hybrid machine learning techniques. In: 2017 XLIII Latin American Computer Conference (CLEI). IEEE, pp. 1–10.

Petković, T., Puljiz, D., Marković, I., Hein, B., 2019. Human intention estimation based on hidden Markov model motion validation for safe flexible robotized warehouses. Robot. Comput.-Integr. Manuf. 57, 182–196.

Phan, T.V., Gias, T.R., Islam, S.T., Huong, T.T., Thanh, N.H., Bauschert, T., 2019. Q-MIND: Defeating stealthy DoS attacks in SDN with a machine-learning based defense framework. In: 2019 IEEE Global Communications Conference (GLOBECOM). IEEE, pp. 1–6.

Phan, T.V., Park, M., 2019. Efficient distributed denial-of-service attack defense in SDN-based cloud. IEEE Access 7, 18701–18714.

Piltan, F., Kim, J.-M., 2021. Bearing anomaly recognition using an intelligent digital twin integrated with machine learning. Appl. Sci. 11 (10), 4602.

Pires, F., Cachada, A., Barbosa, J., Moreira, A.P., Leitão, P., 2019. Digital twin in industry 4.0: Technologies, applications and challenges. In: 2019 IEEE 17th International Conference on Industrial Informatics (INDIN), Vol. 1. IEEE, pp. 721–726.

Ponz, A., Rodríguez-Garavito, C., García, F., Lenz, P., Stiller, C., Armingol, J.M., 2015. Laser scanner and camera fusion for automatic obstacle classification in ADAS application. In: Smart Cities, Green Technologies, and Intelligent Transport Systems. Springer, pp. 237–249.

Popa, C.L., Cotet, C.E., Popescu, D., Solea, M.F., Şaşcîm, S.G., Dobrescu, T., 2018. Material flow design and simulation for a glass panel recycling installation. Waste Manage. Res. 36 (7), 653–660.

Pradeep, K., Kamalavasan, K., Natheesan, R., Pasqual, A., 2018. Edgenet: Squeezenet like convolution neural network on embedded fpga. In: 2018 25th IEEE International Conference on Electronics, Circuits and Systems (ICECS). IEEE, pp. 81–84.

Qin, Y., Sheng, Q.Z., Falkner, N.J., Dustdar, S., Wang, H., Vasilakos, A.V., 2016. When things matter: A survey on data-centric internet of things. J. Netw. Comput. Appl. 64, 137–153.

Rodič, B., 2017. Industry 4.0 and the new simulation modelling paradigm. Organizacija 50 (3), 193–207.

Román-Castro, R., López, J., Gritzalis, S., 2018. Evolution and trends in IoT security. Computer 51 (7), 16–25.

Saleem, T.J., Chishti, M.A., 2019. Deep learning for Internet of Things data analytics. Procedia Comput. Sci. 163, 381–390.

Samann, F.E., Abdulazeez, A.M., Askar, S., 2021. Fog computing based on machine learning: A review. IJIM 15 (12), 21.

Sampaio, L.S., Faustini, P.H., Silva, A.S., Granville, L.Z., Schaeffer-Filho, A., 2018. Using NFV and reinforcement learning for anomalies detection and mitigation in SDN. In: 2018 IEEE Symposium on Computers and Communications (ISCC). IEEE, pp. 00432–00437.

Satheesh, N., Rathnamma, M., Rajeshkumar, G., Sagar, P.V., Dadheech, P., Dogiwal, S., Velayutham, P., Sengan, S., 2020. Flow-based anomaly intrusion detection using machine learning model with software defined networking for OpenFlow network. Microprocess. Microsyst. 79, 103285.

Schneider, G.F., Wicaksono, H., Ovtcharova, J., 2019. Virtual engineering of cyber-physical automation systems: The case of control logic. Adv. Eng. Inform. 39, 127–143.

Sebbar, A., Zkik, K., Baddi, Y., Boulmalf, M., Kettani, M.D.E.-C.E., 2020. MitM detection and defense mechanism CBNA-RF based on machine learning for large-scale SDN context. J. Ambient Intell. Humaniz. Comput. 11 (12), 5875–5894.

Shilton, A., Rajasegarar, S., Leckie, C., Palaniswami, M., 2015. DP1SVM: A dynamic planar one-class support vector machine for Internet of Things environment. In: 2015 International Conference on Recent Advances in Internet of Things (RIoT). IEEE, pp. 1–6.

Shin, S.-J., Woo, J., Rachuri, S., 2014. Predictive analytics model for power consumption in manufacturing. Procedia Cirp 15, 153–158.

Shukla, M., Kosta, Y., Chauhan, P., 2015. Analysis and evaluation of outlier detection algorithms in data streams. In: 2015 International Conference on Computer, Communication and Control (IC4). IEEE, pp. 1–8.

Sideratos, G., Ikonomopoulos, A., Hatziargyriou, N., 2015. A committee of machine learning techniques for load forecasting in a smart grid environment. Int. J. Energy Power 4, 98.

Sierla, S., Kyrki, V., Aarnio, P., Vyatkin, V., 2018. Automatic assembly planning based on digital product descriptions. Comput. Ind. 97, 34–46.

Siryani, J., Tanju, B., Eveleigh, T.J., 2017. A machine learning decision-support system improves the internet of things' smart meter operations. IEEE Internet Things J. 4 (4), 1056–1066.

Söderberg, R., Wärmefjord, K., Carlson, J.S., Lindkvist, L., 2017. Toward a digital twin for real-time geometry assurance in individualized production. CIRP Ann. 66 (1), 137–140.

Souri, A., Hussien, A., Hoseyninezhad, M., Norouzi, M., 2019. A systematic review of IoT communication strategies for an efficient smart environment. Trans. Emerg. Telecommun. Technol. e3736.

Stahl, R., Zhao, Z., Mueller-Gritschneder, D., Gerstlauer, A., Schlichtmann, U., 2019. Fully distributed deep learning inference on resource-constrained edge devices. In: International Conference on Embedded Computer Systems. Springer, pp. 77–90.

Suhail, S., Zeadally, S., Jurdak, R., Hussain, R., Matulevičius, R., Svetinovic, D., 2022. Security attacks and solutions for digital twins. arXiv preprint arXiv:2202.12501.

Sun, Y., Peng, M., Mao, S., Yan, S., 2019a. Hierarchical radio resource allocation for network slicing in fog radio access networks. IEEE Trans. Veh. Technol. 68 (4), 3866–3881.

Sun, Y., Peng, M., Zhou, Y., Huang, Y., Mao, S., 2019b. Application of machine learning in wireless networks: Key techniques and open issues. IEEE Commun. Surv. Tutor. 21 (4), 3072–3108.

Susto, G.A., Schirru, A., Pampuri, S., McLoone, S., Beghi, A., 2014. Machine learning for predictive maintenance: A multiple classifier approach. IEEE Trans. Ind. Inf. 11 (3), 812–820.

Swami, R., Dave, M., Ranga, V., 2019. Defending DDoS against software defined networks using entropy. In: 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU). IEEE, pp. 1–5.

Tabassi, E., Burns, K.J., Hadjimichael, M., Molina-Markham, A.D., Sexton, J.T., 2019. A taxonomy and terminology of adversarial machine learning. In: NIST IR. pp. 1–29.

Tai, A.H., Ching, W.-K., Chan, L.-Y., 2009. Detection of machine failure: Hidden Markov model approach. Comput. Ind. Eng. 57 (2), 608–619.

Tan, M., Le, Q., 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning. PMLR, pp. 6105–6114.

Tan, Y., Yang, W., Yoshida, K., Takakuwa, S., 2019. Application of IoT-aided simulation to manufacturing systems in cyber-physical system. Machines 7 (1), 2.

Tandon, R., Simeone, O., 2016. Cloud-aided wireless networks with edge caching: Fundamental latency trade-offs in fog radio access networks. In: 2016 IEEE International Symposium on Information Theory (ISIT). IEEE, pp. 2029–2033.

Tao, Z., Li, Q., 2018. Esgd: Communication efficient distributed deep learning on the edge. In: {USENIX} Workshop on Hot Topics in Edge Computing (HotEdge 18).

Tao, F., Zhang, M., 2017. Digital twin shop-floor: a new shop-floor paradigm towards smart manufacturing. Ieee Access 5, 20418–20427.

Tarkoma, S., Katasonov, A., 2011. Internet of things strategic research agenda (IoT–SRA). In: Finnish Strategic Centre for Science, Technology, and Innovation: For Information and Communications (ICT) Services, Businesses, and Technologies, Finland.

Teerapittayanon, S., McDanel, B., Kung, H.-T., 2017. Distributed deep neural networks over the cloud, the edge and end devices. In: 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS). IEEE, pp. 328–339.

Thakkar, A., Lohiya, R., 2021. A review on machine learning and deep learning perspectives of IDS for IoT: recent updates, security issues, and challenges. Arch. Comput. Methods Eng. 28 (4), 3211–3243.

Tuama, A., Comby, F., Chaumont, M., 2016. Camera model identification based machine learning approach with high order statistics features. In: 2016 24th European Signal Processing Conference (EUSIPCO). IEEE, pp. 1183–1187.

Uwagbole, S.O., Buchanan, W.J., Fan, L., 2017. An applied pattern-driven corpus to predictive analytics in mitigating SQL injection attack. In: 2017 Seventh International Conference on Emerging Security Technologies (EST). IEEE, pp. 12–17.

Veena, K., 2018. A survey on network intrusion detection. Int. J. Sci. Res. Sci. Eng. Technol. 4, 595–613.

Vishwakarma, R., Jain, A.K., 2019. A honeypot with machine learning based detection framework for defending IoT based botnet DDoS attacks. In: 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI). IEEE, pp. 1019–1024.

Volodymyr, B., 2020. Recurrent neural networks appications guide.

Walinjkar, A., Woods, J., 2017. ECG classification and prognostic approach towards personalized healthcare. In: 2017 International Conference on Social Media, Wearable and Web Analytics (Social Media). IEEE, pp. 1–8.

Wang, X., Han, Y., Leung, V.C., Niyato, D., Yan, X., Chen, X., 2020. Convergence of edge computing and deep learning: A comprehensive survey. IEEE Commun. Surv. Tutor. 22 (2), 869–904.

Wang, W., Ke, X., Wang, L., 2018a. A HMM-R approach to detect L-DDoS attack adaptively on SDN controller. Future Internet 10 (9), 83.

Wang, S., Tuor, T., Salonidis, T., Leung, K.K., Makaya, C., He, T., Chan, K., 2018b. When edge meets learning: Adaptive control for resource-constrained distributed machine learning. In: IEEE INFOCOM 2018-IEEE Conference on Computer Communications. IEEE, pp. 63–71.

Wang, X., Yang, L.T., Kuang, L., Liu, X., Zhang, Q., Deen, M.J., 2019a. A tensor-based big-data-driven routing recommendation approach for heterogeneous networks. IEEE Netw. 33 (1), 64–69.

Wang, S., Zhao, Y., Huang, L., Xu, J., Hsu, C.-H., 2019b. QoS prediction for service recommendations in mobile edge computing. J. Parallel Distrib. Comput. 127, 134–144.

Wehbi, K., Hong, L., Al-salah, T., Bhutta, A.A., 2019. A survey on machine learning based detection on DDoS attacks for IoT systems. In: 2019 SoutheastCon. IEEE, pp. 1–6.

Wei, Y., Law, A.W.-K., Yang, C., Tang, D., 2022. Combined anomaly detection framework for digital twins of water treatment facilities. Water 14 (7), 1001.

Wichmann, R.L., Eisenbart, B., Gericke, K., 2019. The direction of industry: a literature review on industry 4.0. In: Proceedings of the Design Society: International Conference on Engineering Design, Vol. 1. Cambridge University Press, pp. 2129–2138.

Wu, D., Jennings, C., Terpenny, J., Kumara, S., 2016. Cloud-based machine learning for predictive analytics: Tool wear prediction in milling. In: 2016 IEEE International Conference on Big Data (Big Data). IEEE, pp. 2062–2069.

Wu, H., Yue, K., Hsu, C.-H., Zhao, Y., Zhang, B., Zhang, G., 2017. Deviation-based neighborhood model for context-aware QoS prediction of cloud and IoT services. Future Gener. Comput. Syst. 76, 550–560.

Xiao, L., Wan, X., Lu, X., Zhang, Y., Wu, D., 2018. IoT security techniques based on machine learning: How do IoT devices use AI to enhance security? IEEE Signal Process. Mag. 35 (5), 41–49.

Xie, J., Yu, F.R., Huang, T., Xie, R., Liu, J., Wang, C., Liu, Y., 2018. A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges. IEEE Commun. Surv. Tutor. 21 (1), 393–430.

Xiong, P., Chi, Y., Zhu, S., Moon, H.J., Pu, C., Hacigümüş, H., 2011. Intelligent management of virtualized resources for database systems in cloud environment. In: 2011 IEEE 27th International Conference on Data Engineering. IEEE, pp. 87–98.

Xu, C.-Z., Rao, J., Bu, X., 2012. URL: A unified reinforcement learning approach for autonomic cloud management. J. Parallel Distrib. Comput. 72 (2), 95–105.

Yang, M., Zhu, T., Liu, B., Xiang, Y., Zhou, W., 2018. Machine learning differential privacy with multifunctional aggregation in a fog computing architecture. IEEE Access 6, 17119–17129.

Yassine, H., Malli, M., 2019. A lightweight IoT security solution. In: 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC). IEEE, pp. 567–572.

Yavanoglu, O., Aydos, M., 2017. A review on cyber security datasets for machine learning algorithms. In: 2017 IEEE International Conference on Big Data (Big Data). IEEE, pp. 2186–2193.

Yeh, T., Chiu, D., Lu, K., 2017. Persirai: New internet of things (IoT) botnet targets IP cameras. In: Blog, Trend-Labs, Vol. 9.

Yeh, K.-H., Su, C., Hsu, C.-L., Chiu, W., Hsueh, Y.-F., 2016. Transparent authentication scheme with adaptive biometrie features for IoT networks. In: 2016 IEEE 5th Global Conference on Consumer Electronics. IEEE, pp. 1–2.

Yu, Q., Ren, J., Fu, Y., Li, Y., Zhang, W., 2019. Cybertwin: An origin of next generation network architecture. IEEE Wirel. Commun. 26 (6), 111–117.

Yusof, M.A.M., Ali, F.H.M., Darus, M.Y., 2017. Detection and defense algorithms of different types of ddos attacks using machine learning. In: International Conference on Computational Science and Technology. Springer, pp. 370–379.

Yuwono, M., Qin, Y., Zhou, J., Guo, Y., Celler, B.G., Su, S.W., 2016. Automatic bearing fault diagnosis using particle swarm clustering and Hidden Markov Model. Eng. Appl. Artif. Intell. 47, 88–100.

Zaman, M., Lung, C.-H., 2018. Evaluation of machine learning techniques for network intrusion detection. In: NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium. IEEE, pp. 1–5.

Zeng, Y., Al-Quzweeni, A., El-Gorashi, T.E., Elmirghani, J.M., 2019. Energy efficient virtualization framework for 5G F-RAN. In: 2019 21st International Conference on Transparent Optical Networks (ICTON). IEEE, pp. 1–4.

Zezulka, F., Marcon, P., Bradac, Z., Arm, J., Benesl, T., Vesely, I., 2018. Communication systems for industry 4.0 and the iiot. IFAC-PapersOnLine 51 (6), 150–155.

Zhang, H., Qiu, Y., Long, K., Karagiannidis, G.K., Wang, X., Nallanathan, A., 2018a. Resource allocation in NOMA-based fog radio access networks. IEEE Wirel. Commun. 25 (3), 110–115.

Zhang, H., Zhang, G., Yan, Q., 2019. Digital twin-driven cyber-physical production system towards smart shop-floor. J. Ambient Intell. Humaniz. Comput. 10 (11), 4439–4453.

Zhang, X., Zhou, X., Lin, M., Sun, J., 2018b. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6848–6856.

Zhao, R., Yan, D., Liu, Q., Leng, J., Wan, J., Chen, X., Zhang, X., 2019. Digital twin-driven cyber-physical system for autonomously controlling of micro punching system. IEEE Access 7, 9459–9469.

Zheng, P., Lin, T.-J., Chen, C.-H., Xu, X., 2018. A systematic design approach for service innovation of smart product-service systems. J. Cleaner Prod. 201, 657–667.

Zhou, Z., Hu, J., Liu, Q., Lou, P., Yan, J., Li, W., 2018. Fog computing-based cyber-physical machine tool system. IEEE Access 6, 44580–44590.

Zhuang, C., Liu, J., Xiong, H., 2018. Digital twin-based smart production management and control framework for the complex product assembly shop-floor. Int. J. Adv. Manuf. Technol. 96 (1), 1149–1163.

Zissis, D., 2017. Intelligent security on the edge of the cloud. In: 2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC). IEEE, pp. 1066–1070.

Zolotukhin, M., Kumar, S., Hämäläinen, T., 2020. Reinforcement learning for attack mitigation in sdn-enabled networks. In: 2020 6th IEEE Conference on Network Softwarization (NetSoft). IEEE, pp. 282–286.

Dinesh Soni received his M.Tech. degree from the Indian Institute of Technology, New Delhi. Currently, he is pursuing Ph.D. in the Department of Computer Science and Engineering Indian Institute of Technology, Roorkee, India. His research interests include Cloud Computing, Distributed Computing, IoT, and Machine Learning.

Neetesh Kumar received his M.Tech and Ph.D. degrees from the School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi. Currently, he is working as a Faculty member at CSE department, IIT-Roorkee, India. He has published several research publications in world's top tier publishers like IEEE journals and transactions, Elsevier journals, Springer Journals etc. One of his articles published in IEEE sensors journal has been notified by IEEE council in the list of world's top 15 most downloaded articles in the month of Oct–Nov 2018. He is a regular reviewer of several reputed journals like, IEEE Transactions on (PDS, ITS, VT, Cybernetics, TNSM etc.), IEEE journal of Internet of things, FGCS etc. He is also acting as a lead PI for several sponsored research projects from IIT Roorkee/DST/CSIR agencies, Government of India. He has got several best-paper/special-mention awards from the reputed conferences/workshops. Broadly, His research interests include Algorithm Design, Cloud Computing, Parallel Computing, Software-defined Networking, Computational Intelligence, IoT and Intelligent Transportation System.