INTELLIGENT DECISION-MAKING FOR IOT DEVICES IN NETWORKS USING ARTIFICIAL INTELLIGENCE

By Aviral Jain, Student of Vellore Institute of Technology, Vellore (TN) - India

Registration number:21BCT0153

Abstract:

The Internet of Things (IoT) has become revolutionary technology, allowing the interconnection and flawless communication between a wide range of devices. However, the fast growth and scale of IoT networks pose important challenges in terms of resource optimization and decision-making. Artificial Intelligence (AI) methodologies have great potential in addressing these challenges by providing intelligent solutions. This research article concentrate on the problem of optimizing resource allocation and decision-making in IoT networks by integration of AI methodologies and develops competent and scalable algorithms that can semi/fully dynamically assign resources and make intelligent decisions based on real-time data from IoT devices.

The new AI technologies take intelligence to the edge and reduce the need for cloud analytics and associated costs. The combination of IoT and AI has the possibility to transform human life in to smart human living in smart city, including smart governance, education, economy, healthcare, mobility, communication, energy, security and agriculture. However, the limited capacity and poor performance of IoT devices cause serious challenges to the management and development of these resources. Intelligent decision-making algorithms using artificial intelligence can increase the performance and efficiency of IoT networks while providing efficient use.

As the future of IoT is promising and with advancement in edge computing, 5G connectivity, shared intelligence, blockchain security, enterprise-specific solutions, sustainability, data development and Intelligent Decision-Making for IoT Networks using Artificial Intelligence the best efficiency and effectiveness of IoT devices can be achieved.

Key Words: IoT network· Decision Making · AI algorithm· Edge computing · Cloud–fog system · IoT sensors · smart city · IGA-POP · 5G

Citation*: Aviral Jain, VIT Vellore, Intelligent Decision-Making for IoT Devices In Networks using Artificial Intelligence, (Jul 2023), Real-Time Task Scheduling Algorithm for IoT-Based Applications in the Cloud–Fog Environment,* https://doi.org/10.1007/s10922-022-09664-6

(submitted for course work under the guidance of Prof Shridhar Raj S)

Paper Organization:

The research paper is organised in the following sections:

Introduction:

In this section, we introduce the problems of resource allocation and decision accuracy improvement in IoT networks. The motivation behind the research is that while this decision space is already being made, there is still a lot of room to improve reality and the concept of smart living in smart city comes with IoT device network with the best and very efficient AI models.

Analysis and Review of Available Publications:

This section provides a comprehensive review of the existing literature on IoT network resource allocation with a focus on AI-based approaches. We divide the data according to the different allocation of resources and discuss various AI algorithms and methods proposed in this area. We analyze their strengths, limitations and applicability in various situations. Additionally, we identify research gaps and challenges that need to be addressed.

Recommended Framework and Algorithms for Intelligent Allocation of Devices in IoT network:

In this section, we present a framework for intelligent resource distribution in IoT networks. We explain the system design, data collection, algorithms and preliminary procedures. We discuss the selection and training of AI models, decision strategies, and their integration with IoT network infrastructure. We also decided to consider the timely allocation of resources to ensure that the framework is used effectively.

Proposal and Performance Analysis:

Proposed Methodology, Architecture, Experimental setup, descriptive data and performance evaluations are presented in this section. We conduct a comparative analysis of AI-based resource allocation by evaluating their performance, network performance and scalability. We discuss the results and analyze their effects.

Scope of Future Work and Conclusion:

In the conclusion, we summarize our findings and reiterate the importance of AI-based resource allocation in IoT networks. We discuss the implications of our proposed framework and consider their contribution. We also acknowledge the limitations of our study and recommend future research in this area.

Section 1: Introduction:

The Internet of Things is a framework for interconnecting devices, machines and digital systems, objects and living things with unique features and the ability to send information over the Internet without human-to-human or human-to-device interaction.

The combination of artificial intelligence and the Internet of Things (AIoT) creates "intelligent" devices that can learn from the data they generate and use those insights to make self-decisions.

New AI technologies take intelligence to the edge and reduce the need for cloud analytics and associated costs. The combination of IoT and AI has the potential to transform many things, including healthcare, transportation, energy, security and agriculture. However, the limited capacity and poor performance of IoT devices pose serious challenges to the management and development of these resources. Intelligent decision-making algorithms using artificial intelligence can increase the performance and efficiency of IoT networks while providing efficient use.

The future of IoT is quite promising with advances in edge computing, 5G connectivity, shared intelligence, blockchain security, enterprise-specific solutions, sustainability, data development identification, integration expansion and intelligent decision making algorithms using AI.

The Artificial Intelligence (AI) refers to the simulation of human intelligence in machines designed to think and learn like humans, combining computer science, mathematics and other disciplines. There are many methods for intelligence, Machine Learning focuses on training algorithms to learn from data and make predictions or decisions, given from the training system or Deep Learning uses multiple layers of neural networks to learn data representations. Supportive Learning involves training the agent to make decisions in a profit-maximizing environment. The agent learns by trial and error and receives feedback in the form of rewards. The Edge Computing is a distributed computing model that brings power and data storage closer to the edge of the network, where data is created or used, rather than relying gradually in the middle of the air. The Edge computing and IoT (Internet of Things) are closely related and have a significant impact on decision making in many areas:

(a) Reduced latency: Edge computing processes local data, bringing power consumption closer to the edge such as sensors and IoT devices. . It is sent to the cloud for analysis. This minimizes delays or delays in decision making, such as real-time processing of data in non-motorized vehicles.

(b) Edge computing allows for local data filtering, aggregation and analysis by allowing only important and applicable data to be sent to the cloud. Edge computing processes data at the edge, reducing the amount of data that must be sent to cloud IoT devices, which transmits more data and reduces bandwidth.

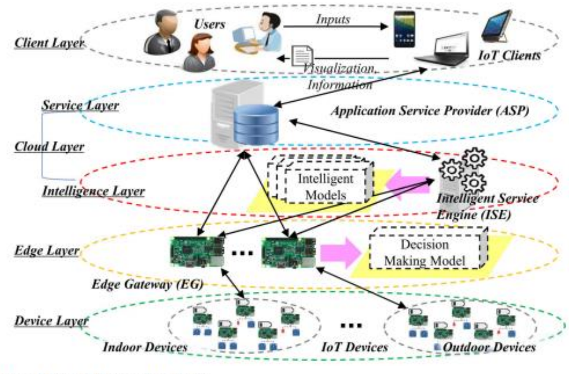
(c) Edge computing helps resolve privacy and security issues with IoT data.

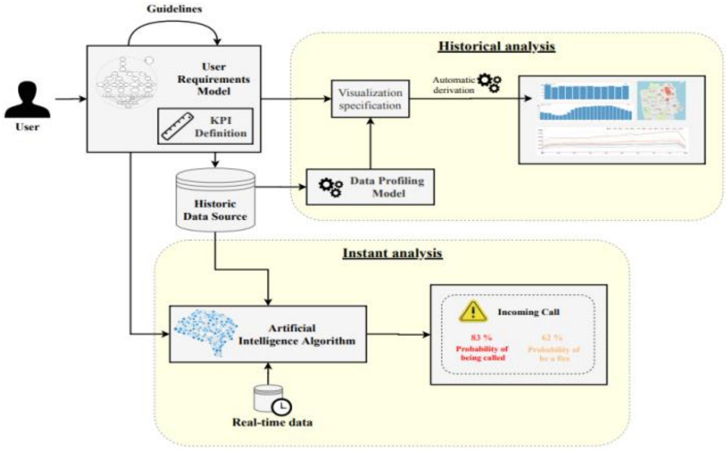
(d) By processing local data, sensitive or private data can be stored on edge devices or local networks,

(e) Real-time decision making: Edge computing enables real-time decision making by analyzing data at the edge and generating instant responses. This is particularly useful in applications such as predictive maintenance. Depending on the correctness of the problem representation and solution, the decision is called a model, no model or half model. Decision making is determined by solutions when decisions are not problematic due to a particular decision maker and there is little or no agreement on solving problems.

<u>Section 2: Analysis and Review of Available Publications</u>:

Research Paper/Pub	Analysis/Algorithm/Approach used	Limitations/improvement
[1]	Gateway centric iot system based on edge computing having IoT client, Application Service Provider (ASP)-provide multiple services to the users like mobiles, Intelligent Service Engine (ISE), Edge Gateway (EG), and IoT devices 4-tier architecture: Client-users with IoT clients to use the services in the cloud Cloud-servers including high performance computing units Edge-bridges iot devices to the cloud via EG (Edge Gateways) Device-includes sensors and actuators	The edge gateways need to have various elements to derive an optimized factor. More deep learning approaches need to be offloaded to the intelligent service environment to provide services from various perspective The energy consumption is predicted less than the original values at some moments

	 Proposed IoT system hierarchical architecture Intelligent edge computing and user environment update scenario	
[2]	A systematic review covering recently published material on artificial intelligence-based decision-making algorithms, Internet of Things sensing networks, cognitive automation, and deep learning-assisted smart process management in cyber-physical production systems by harnessing Preferred Reporting Items for Systematic Reviews and Meta-analysis (PRISMA) guidelines. Intelligent plant modules and smart factory automation have advanced CPPSs that are pivotal in collision identification, impedance monitoring, and assimilating machine learning-based tasks. Wireless sensor technology monitor manufacturing assets and networked production or logistics business operations in real time.	The scope of this systematic review does not approach complex connections between product decision-making information systems, real-time advanced analytics, cyber-physical smart manufacturing, and robotic wireless sensor networks in sustainable Industry 4.0. analyses should develop on real-time sensor networks so as to configure the importance of artificial intelligence-driven big data analytics
[3]	Artificial Intelligence methods based on resource allocation optimization without considering auction-based methods in various computing environments are provided such as cloud computing, Vehicular Fog Computing, wireless, IoT, vehicular networks, 5G networks, vehicular cloud architecture, machine-to-machine communication (M2M), Train-to-Train(T2T) communication network, Peer-to-Peer (P2P) network. Deep reinforcement learning, Q-learning technique, reinforcement learning, online learning, and also Classical learning methods such as Bayesian learning, Cummins clustering, Markov decision process	Many factors (such as the chance of resource failure and proper estimation of system load) affect the success of a resource management method in large computing systems, resource management is always known as a very complex issue
[4]	Users will be able to communicate their analytical needs and to obtain automatically the set of visualizations most suitable to achieve their goals. These visualizations will be grouped in a powerful dashboard that will help them to make strategic and tactical decisions analyze the historical data that will enable users to have an image of the current situation of the process an instant analysis, where the incoming data will be processed at the moment it arrives with the aim of anticipating to the events	Without the holistic view about the process and its outputs provided it would be hard for the users to understand the state in which their processes are, and therefore there would

	 **Figure 2.** Proposed Process.	not be enough evidence for making the correct decision.
[5]	A wide-ranging overview of the current literature related to the Internet of Things based on intelligent techniques in workable computing.	

[6]	The Comparative Analysis Intelligent Decision Model (CAIDM)uses artificial intelligence techniques like fuzzy logic, machine learning, neural networks to identify and analyze cost-effective solutions for resource utilization. The IDMA algorithm is based on machine learning, artificial intelligence, and data-driven insights.
	
	Deep learning, reinforcement learning and dynamic programming
	In: Network input attributes;
	Out: Resource allocation decision rate (DRA);
	For all time slots (*t*), then
	find the earlier arrival of resource request;
	Calculate the maximum resource allocation;
	Compute the order of resourceaccess;
	Repeat
	Initiate $t = t+1$;
	For $t = 1$ to A do
	$A = At+1$
	Again with the resource allocation;
	Resolve the resource shortage and maximization problem;
	. Compute the evaluation of queue size;
	If (queue size > 0)
	Then allot the resource for all users;
	else
		It still needs a large amount of spectrum to work on a 5g network
		
		Highly reliant on low latency connections which can be challenging in certain areas
		
		Need more power for 5g networks which will lead to additional costs

[7]	An edge intelligence-based IoT network in which a set of edge servers learn a shared model using federated learning (FL) based on the datasets uploaded from a multi-technology-supported IoT network. The data uploading performance of IoT network and the computational capacity of edge servers are entangled with each other in influencing the FL model training process. We propose a novel framework, called federated edge intelligence (FEI), that allows edge servers to evaluate the required number of data samples according to the energy cost of the IoT network as well as their local data processing capacity and only request the amount of data that is sufficient for training a satisfactory model.	The optimal uploading scheme is more suitable for cost-sensitive IoT network that is not very sensitive to the accuracy or convergence of the model.

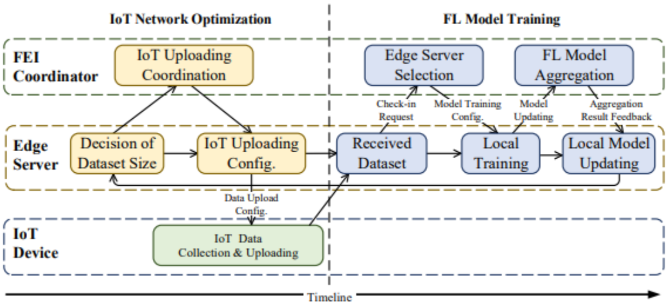
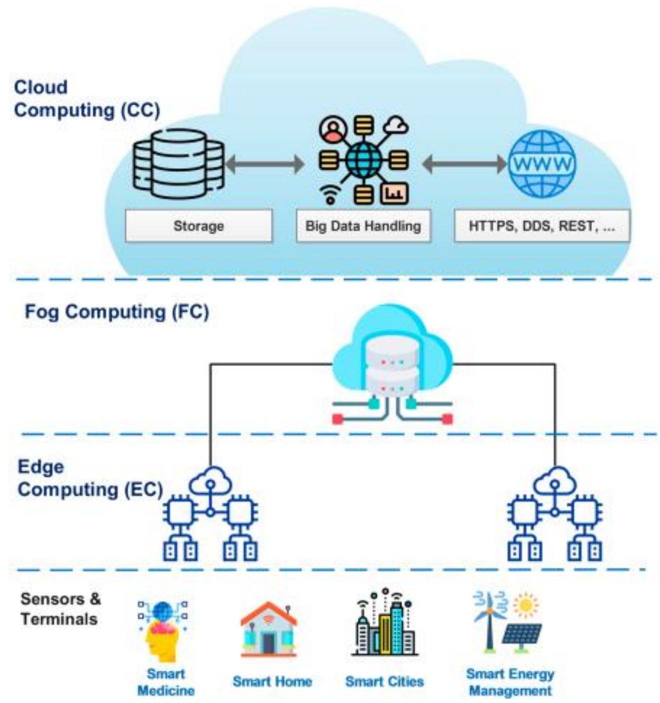
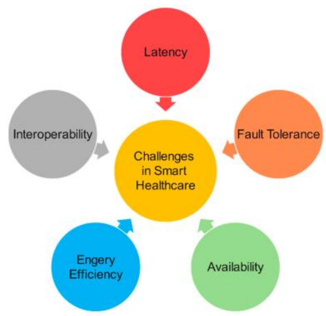
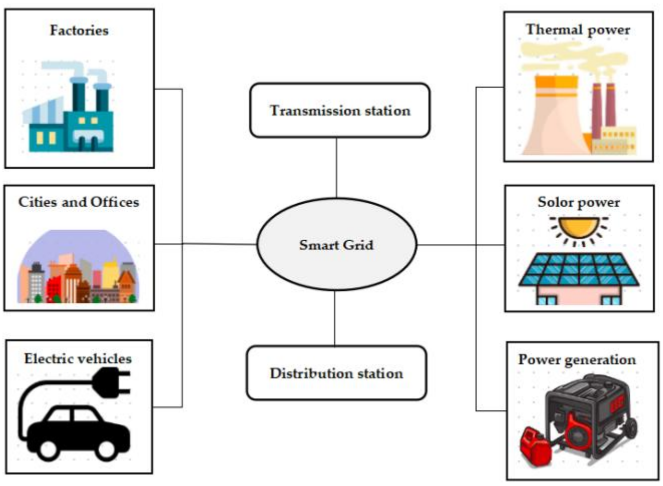
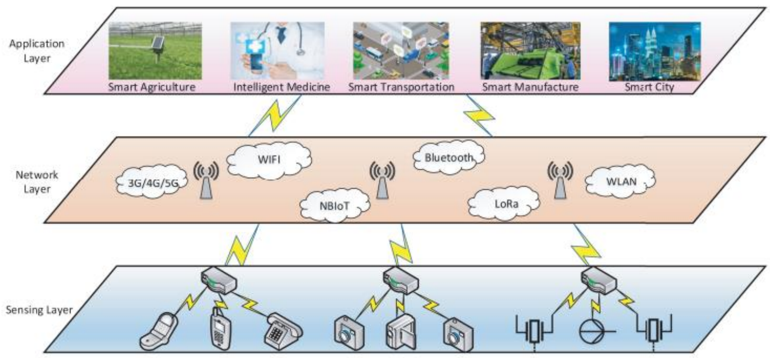
<table>
<tr><td></td><td>

Fig. 2. Proposed framework and procedure of the FEI.</td><td></td></tr>
<tr><td>[8]</td><td>Whale optimization algorithm in our research to determine the ideal set of parameters for DNN method training as well as hyperparameter selection. We establish three things to specify our problem. The system distributes resources instantly in fast time changing fading channels, and consumers. This supports examining vector ergodic average x = E·f·p(h)

Algorithm 1 The whale optimization algorithm
1. Initial population Y_i where $(i = 1,2,3,\ldots\ldots n)$
2. Fitness evaluation for every solution
3. $Y^* =$ best search agent
4. For each solution
5. While $(u < \text{Max\_iteration})$
6. Updated b, B, C, M and p
7. If 1 $(p < 0.5)$
8. If 2$(/B/ < 1)$
9. Random search agent selection (Y_{rand})
10. Else if $(/B/ > 1)$
11. Current search agent location changes.
12. End if
13. Else if 1 $(p \geq 0.5)$
14. End While
15. Change Y^* if better solution is accessible $u = u + 1$
16. Return Y^*
17. Output: $F_i^S \leftarrow$ Best Feature Vector</td><td></td></tr>
<tr><td>[9]</td><td>It presents trust based IoT recommendation techniques so far. Detailed classifications based on extracted parameters as well as investigation existing techniques in three different IoT layers put forth. Moreover, the advantages, disadvantages and open issues of each approach are introduced that can expand more frontier in obtaining accurate IoT recommendation in the future.</td><td></td></tr>
<tr><td>[10]</td><td>ICT infrastructure that facilitates information exchange among the various stakeholders within the urban environment, regardless of the specific application or service being used communication patterns:
(i)Utilising Cellular Mobile Networks for communication, these networks are primarily designed to enable communication between human-to-human and human-to-machine interactions, such as telephony, texting services, multimedia downloading, and streaming</td><td>Most critical issues are improving the coverage area and defining light weight data handling procedures for M2M devices to avoid data overloading difficulties [90]. It is imperative to take action to tackle traffic congestion applications may</td></tr>
</table>

	(ii) Using IoT-Dedicated Cellular Networks for communication, tries to design low-cost, low-energy, and low-traffic congestion IoT devices with reduced traffic requirements. IoT-dedicated cellular networks provide many advantages, including reduced power consumption and Total Cost of Ownership (iii) Employing Multi-Tier Networks for communication. Rely on a layered approach to collect and process sensing data and establish a network infrastructure in a multi-hop manner. The sensor data are typically transmitted to central gateways, which then transmit the data to the internet	not be feasible with traditional GPRS/GSM networks due to their higher costs, subscription fees, and power consumption
[11]	The healthcare focuses on IoT technology and real-time computing solutions through integration of IoT technology and fog computing (FC) solutions with edge computing (EC) has been proposed. The EC and FC technologies aim to bridge the gap between databases and end-users or bring cloud capabilities closer to users. As a result, FC and EC reduce service response times, energy consumption, and computational costs, and improve reliability. 	Nowadays, smart healthcare systems are growing explosively, in both number and scale, due to increasing human needs. Despite the positively achieved results, smart health care systems face several challenges. In this section, we discuss some important challenges, such as issues with fault tolerance, latency, power efficiency, interoperability, and availability. Still the Challenges are as follows: 
[12]	DAER: A resource preallocation algorithm of edge computing server by using blockchain in intelligent driving Crowd-MECS: A novel crowdsourcing framework for mobile edge caching and sharing Federated deep reinforcement learning for Internet of Things with decentralized cooperative edge caching and federated DRL-based cooperative edge caching (FADE) framework	
[13]	Checking packets that are passing through the border router for Communication between physical and network devices. Their methodology is based on the botnet attacks by checking the packet length	No information is presented about the technique employed to create a normal behaviour profile

	Semi-supervised Fuzzy learning-based distributed attack detection framework for IoT on the NSL-KDD dataset Threat analysis of IoT using ANN to detect DDoS/DoS attacks. A multi-level perceptron, a type of supervised ANN, is trained using internet packet traces and then the model is assessed on its ability to thwart (DDoS/DoS) attack	The detection effect of this method is highly dependent on the design of the traffic features used in training. IoT devices have small memory space, which is a challenge to keep track of as the system runs continuously with the high-speed network connection, huge packets have emerged as a challenge because the traffic might critically go over the limited processing
[14]	It presents a systematic literature review (SLR) of the existing literature, organizes the evidences in a systematic way, and then analyzes it for future research to explore, analyse and propose new solution for solving particular problem AI-enabled sensing and decision-making for the Internet of Things system. The uprising of IoT has been recognized around the world for making life easier with the use of intelligent devices such as smart sensors, actuators, and many other devices.	
[15]	Integration of IoT with Machine Learning to Create Smart Grid Artificial artifacts may use reinforcement learning to learn from their activities and make accurate predictions. The trial-and-error approach teaches this. Existing methods struggle with realtime building energy optimization in huge areas Decision Tree Classification Algorithm It helps to choose the machine learning algorithm best suited for the dataset being used and the issue being addressed since many machine learning algorithms can be used tobuild a model. 	Big data analytics: Vast amounts of varied, high-resolution data are being produced by the IoT, with Availability of services and networks intelligent buildings, coordinating the operations of several linked building systems is a big task Cyber security: Increased demands on building operations need to incorporate IP cams into building automation systems. These changes put individuals in harmful way and opens up new attack methods
[16]	Selective sensing It means to focus on information appealing to the observers most, and in sensing layer, various AI algorithms are	the security and privacy issues are the most

	defined for improving accuracy and efficiency of selective sensing neural gas network for automatic selection of observable regions with relevant measurements. This architecture starts from a sparse scanning model and is continuously enhanced via data training. It finally achieves significant benefits in time cost when doing 3D sampling and sensing. AI-enabled IoT networks, where deep learning and neural networks were adopted for different network architectures.Selecting the most appropriate routing path is important in improving the communication efficiency of the network layer a friend recommendation system on the basis of the Big-Five traits model, which can be regarded as with machine learning, semantic technologies, as well as large-scale data from a personality social network. 	fundamental challenges during the convergence. Due to the fact that data serves as the core part of both IoT and AI, leaks of sensitive information will suffer from greater risks and threats. Traditional IoT only focuses on establishing interconnections between fragmented things, while with the convergence of AI, IoT needs to pay much more attention to intelligence. All need to formulate strategies in order to gain advantages
[17]	“Flower End-to-End Detection Based on YOLOv4 Using a Mobile Device” by Z. Cheng and F. Zhang proposes a flower detection approach for smart garden applications. “Quasiconformal Mapping Kernel Machine Learning-Based Intelligent Hyperspectral Data Classification for Internet Information Retrieval” by J. Liu and Y. Qiao focuses on the proposed intelligent data classification algorithm based on machine learning. The approach of quasiconformal kernel mapping learning utilized for Internet-based hyperspectral data retrieval, which is important for many AIoT applications such as image and video-based ones “Fuzzy Obstacle Avoidance for the Mobile System of Service Robots” by S.-P. Tseng et al. Designs and implements a service robot featured by obstacle avoidance and target user tracking	
[18]	Artificial Intelligence and the Internet of Things. Wireless sensor network is used to connect all sensors and IoT devices. IoT devices establish a connection with centralized cloud server, where all heart sound files are accumulated PASCAL data set is used for training and testingfor deep convolutional neural network. PASCAL data set is used to train and test the deep convolutional neural network. 300 images are used to train the deep CNN	It is very challenging for reassembly of a whole audio signal from a corrupted observation. Separating HSS from the ambient noise is one of the

<table>
<tr><td></td><td>model, and 149 images are used to test the CNN model. Wearable sensors->wireless network->iot device->cloud storage->separating heart sound and noise->training and testing using cnn->problem detection and recommendation</td><td>obstacles that must be overcome.

Number of image in database must be in large number which means high memory usage</td></tr>
<tr><td>[19]</td><td>The potential of federated machine learning in addressing the critical challenges of intelligent IoT, so as to ensure data security, reduce network congestion, and make full use of distributed computation resources is presented. Also identified two emerging technologies, i.e. AirComp and RIS, that can tackle the challenge of limited communication bandwidth during the model aggregation phase of federated machine learning.</td><td>AirComp approach with the novel DC algorithm significantly improves the performance of model aggregation for federated machine learning, it may still suffer from unfavourable signal propagation conditions of wireless links, such as deep fading.</td></tr>
<tr><td>[20]</td><td>Three core architecture of IoT, i.e. the perception, transmission, and application layers is presented.

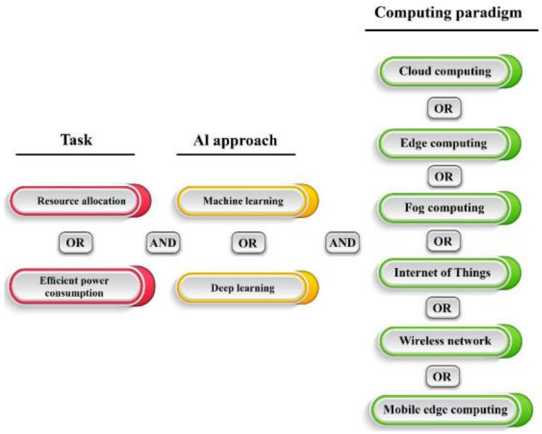
The perception layer is the lowest layer of the architecture of IoT. This layer involves high-tech used for identification, identification, actuation, and communication with a minimum level of human communication. like GPS

The transmission layer is interrupted between the application and the perception layer. This is considered the integration for different heterogeneous legacy networks, protocols, and technologies.

The Application layer is aimed at regulating and providing the application internationally based on the collected info that is controlled by the unit of information processing. It gives access to personalized facilities to IoT users

<table>
<tr><th>Layers</th><th>Sub-layers</th><th>Key Features</th><th>Key Technologies</th></tr>
<tr><td rowspan="2">Application layer</td><td>IoT applications</td><td rowspan="2">Handheld devices, Terminals and User Interface</td><td rowspan="2">Cloud computing, Middleware, M2M, Service support platform</td></tr>
<tr><td>Application support layer</td></tr>
<tr><td rowspan="3">Transmission layer</td><td>Local & Wide Area Network</td><td rowspan="3">Connectivity establishment and Information transmission</td><td rowspan="3">Internet, GPRS, Wi-Fi, Ad hoc Network</td></tr>
<tr><td>Core network</td></tr>
<tr><td>Access network</td></tr>
<tr><td rowspan="2">Perception layer</td><td>Perception network</td><td rowspan="2">Sensing, Identification, Actuation and Communication technologies</td><td rowspan="2">RFID, WSN, GPS, Bluetooth</td></tr>
<tr><td>Perception nodes</td></tr>
<tr><td>Network management</td><td colspan="2">Physical and information security management</td><td>Trust management</td></tr>
</table></td><td>The basis of IoT is laid down on the current WSN, and it architecturally receives a similar level of security and privacy issues as WSN possesses.

Bandwidth and power consumption is more as IoT generation increase such as 5G range from no unified cloud services to other standards like a shortage of homogeneous machine to machine diversities since iot system is heterogeneous.</td></tr>
<tr><td>[21]</td><td>A comprehensive literature review on the application of artificial intelligence (AI) methods such as deep learning (DL) and machine learning (ML) for resource allocation optimization in</td><td>The limitations are Cost, Quality of data, hyperparameter tunning,</td></tr>
</table>

	computational paradigms. The ML-based approaches are categorized as supervised and reinforcement learning (RL). Moreover, DL-based approaches and their combination with RL are surveyed.  Hassannataj Joloudari, Javad & Alizadehsani, Roohallah & Nodehi, Issa & Mojrian, Sanaz & Fazl, Fatemeh & Khanjani Shirkharkolaie, Sahar & Kabir, H M Dipu & Tan, Ru San & Acharya, U Rajendra. (2022). Resource allocation optimization using artificial intelligence methods in various computing paradigms: A Review. 10.13140/RG.2.2.32857.39522.	ability to make model contextual and ability to withstand with resource shortcomings.
[22]	Weights & Biases web site is for ML Ops platform. It is to build tools for machine learning which are being used by cutting-edge machine learning teams including OpenAI, NVIDIA, and Co:here. It provides services to streamline ML workflow from end to end experiments, experiment tracking Reports, Collaborative dashboards, Artifacts, Dataset and models.	
[23]	This website is Comet's machine learning platform which integrates existing infrastructure of the client and tools so he can manage, visualize, and optimize models—from training runs to production monitoring.	
[24]	A semi-dynamic real-time task scheduling algorithm is proposed for bag-of-tasks applications in the cloud–fog environment that achieves a good balance between the makespan and the total execution cost and minimizes the task failure rate compared to the other algorithms.	More elapse runtime and requires further optimisation for fully dynamic real-time task scheduling
[25]	Fog computing is an emerging paradigm that extends computation, communication, and storage facilities toward the edge of a network. Compared to traditional cloud computing, fog computing can support delay-sensitive service requests from end-users (EUs) with reduced energy consumption and low traffic congestion. Basically, fog networks are viewed as offloading to core computation and storage. Fog nodes in fog computing decide to either process the services using its available resource or send to the cloud server. Thus, fog computing helps to achieve efficient resource utilization and higher performance regarding the delay,	

	bandwidth, and energy consumption. Summary of Resource allocation approaches to address several critical issues such as latency, and bandwidth, and energy consumption in fog computing with overview of state-of-the-art network applications and major research aspects to design these networks.	
[26]	The scheduling of service requests to VMs is presented as a bi-objective minimization problem, where a tradeoff is maintained between the energy consumption and makespan. It proposes a metaheuristic-based service allocation framework using three metaheuristic techniques, such as particle swarm optimization (PSO), binary PSO, and bat algorithm. These proposed techniques are evaluated rigorously and facilitate to deal with the heterogeneity of resources in the fog computing environment.	

Section 3: Recommended Framework and Algorithms for Intelligent Allocation of Device in IoT Network:

AI algorithm for decision making for allocation of IoT devices proposed in recent years include deep learning, reinforcement learning, genetic algorithms, fuzzy logic, and swarm intelligence the effectiveness of any algorithm in a specific situation may depend on various factors such as the complexity of the task, the amount and frequency of data collected from IoT devices, the computing resources available, and the specific needs and constraints of the system being managed. Therefore, finding the most suitable algorithm for a given situation may require careful evaluation and experimentation.

Predictive Analysis Approaches. The IoT sensors take information in machine and via internet it displays the data in computation devices through the cloud it will give real time information that can be set in a dynamic dataset. Further, to make predictive analysis of dynamic datasets using machine learning and artificial intelligence, there are several approaches:

(a) Supervised Learning: This approach involves supplying the machine learning model with a 12abelled dataset, where the input data and corresponding output data are specified. The model then creates a prediction function based on the 12abelled dataset, which can be used to generate predictions for new data.

(b) Unsupervised Learning: This approach involves using an unlabeled dataset, where the model must find patterns or structure on its own without any specific guidance. This can be useful for identifying hidden patterns or unsupervised anomalies.

(c) Deep Learning: This approach uses artificial neural networks to train models to process large datasets, such as images or speech. Deep Learning is particularly suited for datasets with high levels of complexity, such as those generated by IoT sensors.

(d) Time Series Analysis: This approach uses statistical models to analyze time-series data, typically by looking at patterns in the dataset, such as trends, seasonality, and cyclicality, and creating a forecasting model that predicts future trends based on those patterns.

(e) Reinforcement Learning: This is a type of machine learning that allows the model to learn from experience rather than being explicitly programmed. Reinforcement learning can be useful in scenarios where the kmodel must learn over time, such as in automated control or decision-making systems.

Predictive Analysis Algorithms. The important machine learning algorithms that can be used for predictive analysis of dynamic datasets are:

(a) Linear Regression: This algorithm is used for predicting continuous numerical values based on a set of input features. It's a popular choice for time series analysis.

(b) Decision Trees: This algorithm is used to construct a tree-like model of decisions and their possible outcomes. Decision trees are often used in classification problems, where the goal is to predict a categorical label based on a set of input features.

(c) Random Forest: This algorithm is an ensemble method that uses multiple decision trees to improve prediction accuracy. It is often used in industrial IoT applications.

(d) K-Nearest Neighbors (KNN): This algorithm is used for classification and regression problems. It works by finding the k closest data points in the feature space and using their labels or values to make a prediction.

(e) Support Vector Machines (SVM): This algorithm is used for classification and regression problems. It works by finding the hyperplane that separates the data points in the feature space and using it to make a prediction.

(f) Neural Networks: This is a family of algorithms that are based on the structure and function of the human brain. Neural networks are often used for complex data analysis tasks, such as image recognition or natural language processing.

(g) Gradient Boosting: This is another ensemble method that combines the power of multiple weak learners to improve prediction accuracy. It's often used in conjunction with decision trees.

The specific use of algorithm for predictive analysis of dynamic data set will depend on the nature of the data, the available computing resources, and the specific purpose.

<u>Decision making Algorithms for Allocation of IoT Devices</u>. The important algorithms for decision making and allocation of IoT devices are :

(a) AHP (Analytical Hierarchical Process): This approach allows decision makers to break down a complex decision into a series of sub-decisions, evaluated on multiple criteria, and prioritize the criteria to make an informed decision. Python code is as follows:

```
import numpy as np

# Define the problem
criteria = ['power', 'response_time', 'reliability']
devices = ['dev1', 'dev2', 'dev3', 'dev4', 'dev5']
task_matrix = np.array([[10, 5, 8],
                        [7, 3, 6],
                        [8, 4, 9],
                        [6, 3, 7],
                        [9, 3, 8]])

# Define the AHP functions
def pairwise_comparison_matrix(matrix):
    n = matrix.shape[0]
    pcm = np.zeros((n, n))
    for i in range(n):
        for j in range(n):
            pcm[i, j] = matrix[i, j] / matrix[j, i]
    return pcm
```

```
def weight_vector(matrix):
   n = matrix.shape[0]
   w = np.zeros(n)
   for i in range(n):
      w[i] = np.prod(matrix[i, :]**(1.0/n))
return w / np.sum(w)

def device_selection(matrix, criteria):
   # Pairwise comparison of criteria
   pcm_criteria = pairwise_comparison_matrix(matrix.T)
   # Compute weight vector for criteria
   w_criteria = weight_vector(pcm_criteria)
   # Pairwise comparison of devices for each criteria
   pcm_devices = np.zeros((len(criteria), len(devices), len(devices)))
   for i, criterion in enumerate(criteria):
      pcm_devices[i, :, :] = pairwise_comparison_matrix(matrix[:, i:i+1])
   # Compute weight vector for devices for each criterion
   w_devices = np.zeros((len(criteria), len(devices)))
   for i in range(len(criteria)):
      w_devices[i, :] = weight_vector(pcm_devices[i, :, :])
   # Compute overall weight of each device
   w_overall = np.multiply(w_criteria.reshape(3,1),
w_devices).sum(axis=0)
   # Return the index of the device with the highest weight
   return np.argmax(w_overall)

# Use AHP to select device for each task
allocation = {}
for i, task in enumerate(tasks):
   device_index = device_selection(task_matrix, criteria)
allocation[task] = devices
```

(b) TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution): This algorithm helps decision makers evaluate different alternatives based on multiple criteria by comparing them to an ideal solution and choosing the one that is closest. The python code is as follows:

```
import numpy as np

# Define the problem
criteria = ['power', 'response_time', 'reliability']
devices = ['dev1', 'dev2', 'dev3', 'dev4', 'dev5']
task_matrix = np.array([[10, 5, 8],
            [7, 3, 6],
            [8, 4, 9],
            [6, 3, 7],
            [9, 3, 8]])

# Normalize the task matrix
col_max = np.max(task_matrix, axis=0)
task_matrix_normalized = task_matrix / col_max
```

```
# Define the TOPSIS functions
def weighted_matrix(matrix, weights):
    return np.multiply(matrix, weights)

def ideal_solutions(matrix):
    ideal_best = np.max(matrix, axis=0)
    ideal_worst = np.min(matrix, axis=0)
return ideal_best, ideal_worst

def distance_to_ideal(matrix, ideal_best, ideal_worst):
    d_pos = np.sqrt(np.sum(np.square(matrix - ideal_best), axis=1))
    d_neg = np.sqrt(np.sum(np.square(matrix - ideal_worst), axis=1))
    return d_pos, d_neg

def relative_closeness(d_pos, d_neg):
    return d_neg / (d_pos + d_neg)

# Determine the relative closeness of each device for each task
closeness_matrix = np.zeros((len(devices), len(tasks)))
for i, task in enumerate(tasks):
    weighted_matrix_normalized =
weighted_matrix(task_matrix_normalized, criteria_weights[i])
    ideal_best, ideal_worst =
ideal_solutions(weighted_matrix_normalized)
    d_pos, d_neg = distance_to_ideal(weighted_matrix_normalized,
ideal_best, ideal_worst)
    closeness_matrix[:, i] = relative_closeness(d_pos, d_neg)

# Determine the best device for each task based on the relative closeness
allocation = {}
for i, task in enumerate(tasks):
    best_device_index = np.argmax(closeness_matrix[:, i])
allocation[task] = devices[best_device_index]
```

(c) DEA (Data Envelopment Analysis): This is a mathematical programming-based approach that evaluates different alternatives based on multiple inputs and outputs, and assigns efficiency scores to help make the best decisions. The python code is as follows:

```
mport numpy as np

# Define the problem
criteria = ['power', 'response_time', 'reliability']
devices = ['dev1', 'dev2', 'dev3', 'dev4', 'dev5']
task_matrix = np.array([[10, 5, 8],
                        [7, 3, 6],
                        [8, 4, 9],
                        [6, 3, 7],
                        [9, 3, 8]])

# Define the DEA functions
```

```
def efficiency_score(task_matrix, device):
    numerator = np.dot(task_matrix, device)
    denominator = np.dot(np.power(device, 2), np.ones(len(criteria)))
    return numerator / denominator

def dea_algorithm(task_matrix, devices):
    efficiency_scores = np.zeros(len(devices))
    for i in range(len(devices)):
        device_efficiency_scores = []
        for j in range(len(devices)):
            if i != j:
                device_efficiency_scores.append(efficiency_score(task_matrix,
devices[j]))
        if np.all(np.greater_equal(np.array(device_efficiency_scores),
efficiency_score(task_matrix, devices[i]))):
            efficiency_scores[i] = efficiency_score(task_matrix, devices[i])
    return efficiency_scores

# Define the device class with the criteria as attributes
class Device:
    def _init_(self, power, response_time, reliability):
        self.power = power
        self.response_time = response_time
        self.reliability = reliability

# Create devices
dev1 = Device(5, 10, 8)
dev2 = Device(7, 4, 6)
dev3 = Device(9, 6, 9)
dev4 = Device(8, 3, 7)
dev5 = Device(6, 5, 8)

# Get the DEA efficiency scores for each device
devices_matrix = np.array([[getattr(dev, criteria[0]), getattr(dev,
criteria[1]), getattr(dev, criteria[2])] for dev in [dev1, dev2, dev3, dev4,
dev5]])
efficiency_scores = dea_algorithm(task_matrix, devices_matrix)

# Allocate a device for each task
allocation = {}
for i, task in enumerate(range(len(task_matrix))):
 allocation[task] = devices[np.argmax(task_matrix[i, :] / efficiency_scores)]
```

(d) OR (Operations Research): This approach uses a variety of modeling techniques and mathematical algorithms like linear programming, dynamic programming, and queuing theory to help optimize decision making for IoT device allocation. The python code id as follows:

```
import numpy as np

def operation_research_algorithm(decision_matrix, criteria_weights):
    # Convert decision matrix and weights to numpy arrays
    decision_matrix = np.array(decision_matrix)
```

```
        criteria_weights = np.array(criteria_weights)

        # Normalize decision matrix
        normalized_matrix = decision_matrix / decision_matrix.sum(axis=0)

        # Apply weights to criteria
        weighted_matrix = normalized_matrix * criteria_weights

        # Calculate overall score for each option
        scores = weighted_matrix.sum(axis=1)

        # Return index of option with highest score
    return np.argmax(scores)
```

(e) Genetic Algorithm: This approach simulates a natural selection process used for finding the best possible outcome of a function. For IoT device allocation, the algorithm can evaluate multiple criteria and make decisions based on optimizing the best possible fitness function. The python code for this algorithm for allocation of IoT device is as follows:

```
import random

# Define the problem
tasks = ['task1', 'task2', 'task3', 'task4', 'task5']
devices = ['dev1', 'dev2', 'dev3', 'dev4', 'dev5']
criteria = {'power': [0.4, 0.2, 0.15, 0.15, 0.1],
      'response_time': [0.25, 0.15, 0.2, 0.3, 0.1],
      'reliability': [0.1, 0.2, 0.2, 0.15, 0.35]}

# Define the genetic algorithm parameters
population_size = 20
generations = 100
mutation_rate = 0.1

# Define the fitness function
def fitness(chromosome):
  # Calculate the fitness score of a chromosome based on the criteria
  score = 0.0
  for i, task in enumerate(tasks):
    device_index = chromosome[i]
    score += criteria['power'][device_index] *
criteria['response_time'][device_index] *
criteria['reliability'][device_index]
  return score

# Define the chromosome encoding functions
def random_chromosome():
  # Generate a random chromosome where each gene represents the
index of an IoT device
  chromosome = []
  for i in range(len(tasks)):
```

```
            chromosome.append(random.randint(0, len(devices) - 1))
        return chromosome

    def decode_chromosome(chromosome):
        # Translate the chromosome to an actual allocation of devices to tasks
        allocation = {}
        for i, task in enumerate(tasks):
            device_name = devices[chromosome[i]]
            allocation[task] = device_name
        return allocation

    # Define the genetic algorithm functions
    def select_parents(population):
        # Select two parents using tournament selection
        tournament_size = 5
        tournament_population = random.sample(population,
    tournament_size)
        tournament_population.sort(key=lambda c: fitness(c), reverse=True)
        return tournament_population[0], tournament_population[1]

    def crossover(parent1, parent2):
        # Perform single-point crossover to create two offspring
        crossover_point = random.randint(1, len(parent1) - 1)
offspring1 = parent1[:crossover_point] + parent2[crossover_point:]
```

(f) Reinforcement Learning: This approach helps IoT systems learn to allocate their devices based on user satisfaction, which is measured by the Q-value. The Q-value describes the expected long-term reward of a particular action taken in a state. The Federated deep reinforcement learning-based task offloading for IoT devices is one of the latest algorithms. The code in Python is as follows:

```
import random
import numpy as np
import gym
from gym import spaces
from keras.models import Sequential
from keras.layers import Dense
from keras.optimizers import Adam
from keras.utils import to_categorical
from collections import deque

class DQNAgent:
    def _init_(self, state_size, action_size):
        self.state_size = state_size
        self.action_size = action_size
        self.memory = deque(maxlen=2000)
        self.gamma = 0.95
        self.epsilon = 1.0
        self.epsilon_min = 0.01
        self.epsilon_decay = 0.995
        self.learning_rate = 0.001
```

```
            self.model = self._build_model()

      def remember(self, state, action, reward, next_state, done):
         self.memory.append((state, action, reward, next_state, done))

      def act(self, state):
         if np.random.rand() <= self.epsilon:
            return random.randrange(self.action_size)
         act_values = self.model.predict(state)
         return np.argmax(act_values[0])

      def replay(self, batch_size):
         minibatch = random.sample(self.memory, batch_size)
         for state, action, reward, next_state, done in minibatch:
            target = reward
            if not done:
               target = (reward + self.gamma *
   np.amax(self.model.predict(next_state)[0]))
            target_f = self.model.predict(state)
            target_f[0][action] = target
            self.model.fit(state, target_f, epochs=1, verbose=0)
         if self.epsilon > self.epsilon_min:
            self.epsilon *= self.epsilon_decay

   class OffloadingEnv(gym.Env):
      def _init_(self, num_users):
         self.num_users = num_users
         self.action_space = spaces.Discrete(num_users + 1)
         self.observation_space = spaces.Box(low=0, high=1,
   shape=(num_users+1,), dtype=np.float16)
         self
```

(g) An architecture of IoT service delegation and resource allocation decision rules of linearised decision tree algorithm based on predictive analysis. Pytrhon code is as follows:

```
from sklearn.tree import DecisionTreeClassifier
import pandas as pd
import numpy as np

# Load the data
data = pd.read_csv('iot_data.csv')

# Split the data into training and testing sets
train_data = data.iloc[:80, :]
test_data = data.iloc[80:, :]

# Define the features and target variables
features = ['sensor1', 'sensor2', 'sensor3', 'sensor4']
target = 'resource_allocated'
```

```
        # Train the decision tree classifier
        clf = DecisionTreeClassifier()
        clf.fit(train_data[features], train_data[target])

        # Make predictions on the testing set
        test_predictions = clf.predict(test_data[features])

        # Evaluate the accuracy of the classifier
        accuracy = np.mean(test_predictions == test_data[target])
print("Accuracy:", accuracy)
```

The choice of algorithm for decision making and allocation of IoT devices will depend on the specific requirements of your IoT system and the nature of the data involved.

Section 4: Proposed Methodology and Archetechture

Methodology: The proposed stepwise methodology for implementing AI algorithms for decision making by IoT networks is as follows:

(a) Define the Problem: Identify the specific area of decision making that needs to be improved in the IoT network, such as resource allocation, fault detection, security, or predictive maintenance. Define the objectives, constraints, and metrics that will guide the development and evaluation of the AI algorithm.

(b) Gather and Preprocess Data: Collect relevant data from sensor nodes, network traffic, or other sources that can inform the decision-making process. Preprocess the data to clean, transform, and extract features that are relevant to the problem domain. Use statistical methods and visualization tools to explore and understand the data.

(c) Design and Train the AI algorithm: Choose an AI algorithm that is suitable for the problem domain and data characteristics, such as deep learning, reinforcement learning, or fuzzy logic. Define the model architecture, hyperparameters, and optimization criteria. Train the model on a representative subset of the data and validate its performance using appropriate metrics.

(d) Integrate the AI algorithm into the IoT network: Deploy the trained AI model on the edge or cloud nodes of the IoT network, depending on the computational and latency requirements. Use application programming interfaces (APIs) or software development kits (SDKs) to integrate the AI model with the network infrastructure and other software components.

(e) Evaluate and refine the AI algorithm: Monitor the performance of the AI algorithm in real-world conditions and collect feedback from users and stakeholders. Analyze the results and identify areas for improvement or fine-tuning. Retrain the AI model periodically on updated data to improve its accuracy and scalability.

The methodology for implementing AI algorithms for decision-making by IoT networks should be iterative, adaptive, and data-driven, and should involve multidisciplinary collaboration between domain experts, data scientists, and IT professionals.

Architecture: The architecture for implementing AI algorithms for decision-making with IoT devices can vary depending on the specific use case. It involves the use of sensors or other IoT devices to gather data, which is then processed by AI algorithms to generate insights and make decisions. The AI algorithms can be implemented on the edge, in cloud-based systems, or a combination of both. The type of AI algorithm used and how it is integrated into the IoT device architecture will also depend on the specific use case and desired outcomes. Here the architecture for smart city is proposed which involves the following:

(a) IoT devices: which include sensors and other devices that collect data from the environment, such as traffic flow, weather, and air quality.

(b) Cloud-based systems: which provide the processing power and storage needed to analyze large amounts of data generated by the IoT devices.

(c) Communication infrastructure: which includes network protocols and security measures needed to ensure that data is transmitted securely between devices and the cloud-based processing systems.

(d) Artificial Intelligence algorithms: These AI algorithms can be implemented using methods like deep learning, reinforcement learning, and machine learning to generate insights from the data collected by the IoT devices.

(e) Decision-making systems: These systems use the insights generated by the AI algorithms to make decisions and take actions. This can include things like traffic flow optimization, air quality management, and energy efficiency measures.

Permutation-Based Optimization Problems (IGA-POP) algorithm: The architecture for implementing AI algorithms for decision-making for IoT device networks in smart cities is a complex and integrated system that requires careful planning, implementation, and management to be effective. The Permutation-Based Optimization Problems (IGA-POP) is proposed as a real time task scheduling algorithm for IoT network devices in the Cloud–Fog environment of a smart city as it is one of the most effective model [24]. The goal of the IGA-POP is to suggest different permutations for the real-time tasks submitted by IoT end users. Then, the tasks are assigned, in the suggested order, to a VM that provides the minimum expected finish time. The proposed scheduling algorithm is periodically executed every fixed time to schedule the set of tasks arriving since the last scheduling round. Additionally, the proposed scheduling algorithm fulfills the requirements of various users by achieving a good trade-off between the makespan and total execution cost, where some users prefer to give higher priority to the execution time and others prefer to execute their tasks within a tight budget.

The Cloud–fog computing can support delay-sensitive and computing-intensive IoT tasks by jointly exploiting cloud and fog resources. The three-tier architecture shown in the following figure-1 is the widely adopted architecture of cloud–fog computing. IoT devices tier consists of GPS-equipped IoT devices (e.g., sensors, smart phones, wearable devices, and smart vehicles) that enable end users to submit their tasks. The fog computing tier comprises edge devices such as routers, switches, and gateways. It provides computing and storage facilities close to data generation. Cloud computing tier consists of a large pool of resources in the form of datacenters [25, 26].

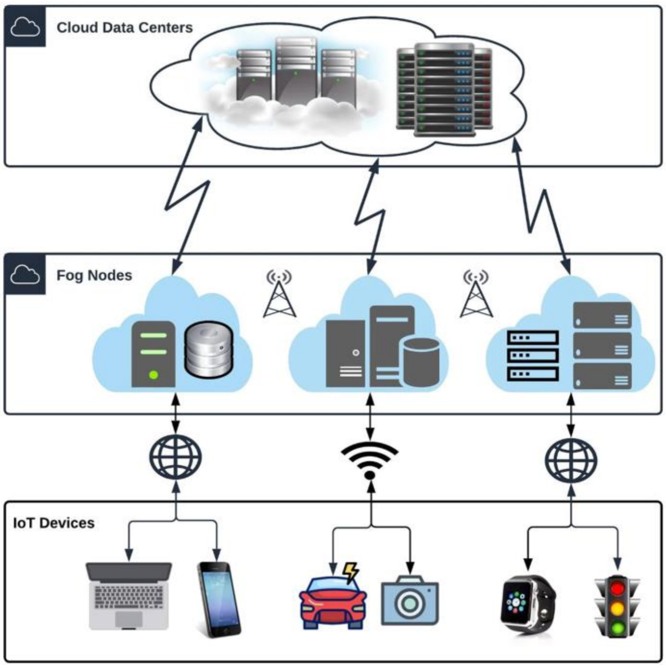


Fig-1

The real-time task scheduling in virtualized cloud–fog computing aims to find the optimal assignment of real-time tasks submitted by IoT end users to available VMs. As shown in Figure- 2, the cloud–fog system consists of *f* virtualized fog nodes, *c* virtualized cloud nodes, a fog broker, and a cloud broker. Each broker contains three main components: a task scheduler, a resource monitor, and a task monitor. The task scheduler applies the task scheduling algorithm that achieves the adopted scheduling policy. The resource monitor tracks the available capacity of the different VMs. The task monitor tracks the progress of task execution and returns the execution results to IoT users. The following assumptions are made about the cloud–fog model: The tasks submitted by IoT end users have different arrival times and deadline times.

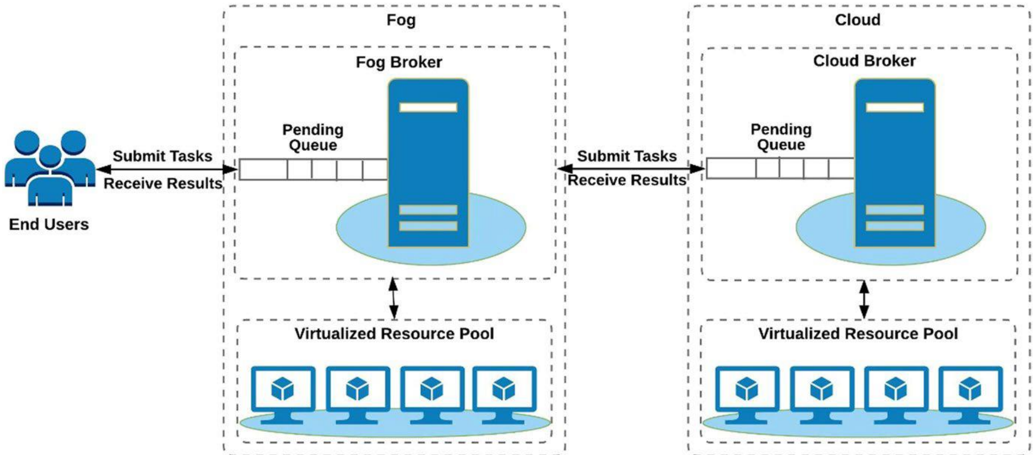


Figure-2

The following Figure 3 shows the sequence diagram of the task scheduling operation in a cloud–fog system. First, all requests are nodes, collects execution results, and sends the results back to the fog broker, who in turn sends the results back to the IoT devices.execution results, and sends the results back to the IoT devices. However, if the fog nodes are saturated, the fog broker immediately forwards the tasks to the cloud broker. The cloud broker assigns the arrived real-time tasks to the cloud.

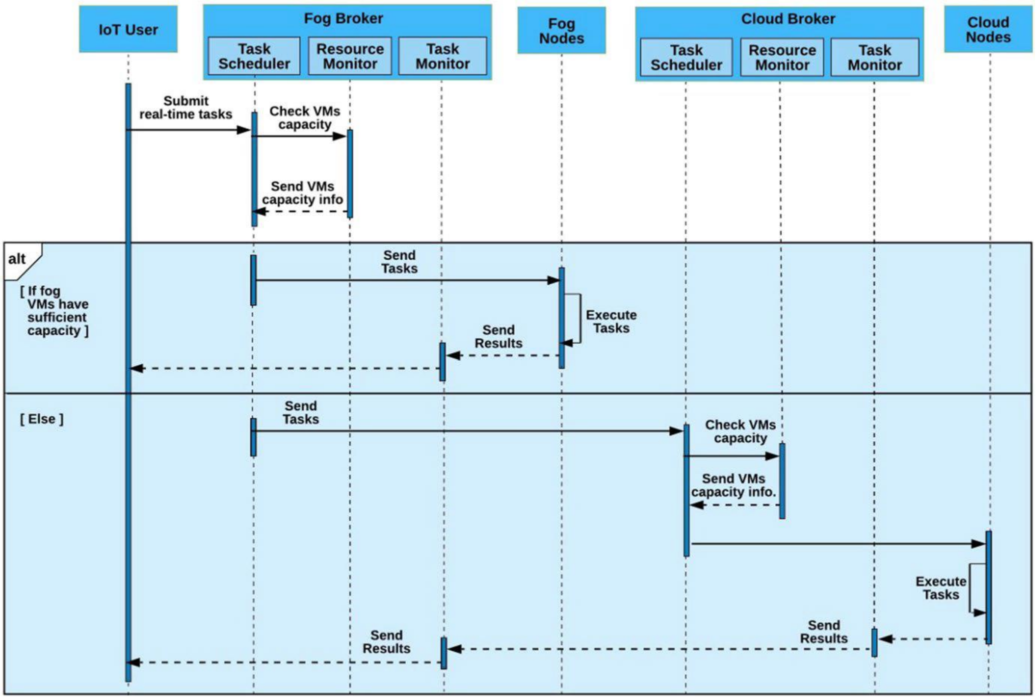


Fig-3

The IGA-POP algorithm approaches task scheduling as a permutation-based optimization problem. Hence, it employs the IGA-POP to provide the best permutation for the submitted real-time tasks. Then, the real-time tasks are assigned to the VMs that achieve the minimum expected finish time.

Python Code of Proposed Algorithm IGA-POP for Real-Time Task Allocation for IoT Devices in Network

```
# Import libraries
import random
import math

# Fitness Function
def fitness_function(individual, fog_num, fog_capacity, processing_time):
    fitness = 0
    fog_load = []
    for i in range(fog_num):
        fog_load.append(0)
    for i in range(len(individual)):
        if (fog_load[individual[i]] + processing_time[i]) <= fog_capacity[individual[i]]:
            fog_load[individual[i]] += processing_time[i]
        else:
            fitness += fog_capacity[individual[i]] - fog_load[individual[i]]
            fog_load[individual[i]] = fog_capacity[individual[i]]
    fitness += sum(fog_load)/len(fog_load)
    return fitness

# IGA-POP Algorithm
def iga_pop(fog_num, fog_capacity, processing_time, pop_size, max_gen):
    pop = []
    pop_fitness = []
    for i in range(pop_size):
        ind = [random.randint(0, fog_num-1) for j in range(len(processing_time))]
        pop.append(ind)
        pop_fitness.append(fitness_function(ind, fog_num, fog_capacity, processing_time))
    for i in range(max_gen):
```

```
# Roulette wheel selection
selected_individuals = []
for j in range(pop_size):
    randnum = random.uniform(0, sum(pop_fitness))
    fit_sum = 0
    for k in range(pop_size):
        fit_sum += pop_fitness[k]
        if fit_sum > randnum:
            selected_individuals.append(list(pop[k]))
            break
# Crossover
new_pop = []
while len(new_pop) < pop_size:
    parent1 = random.choice(selected_individuals)
    parent2 = random.choice(selected_individuals)
    crossover_point = random.randint(1, len(processing_time)-1)
    child1 = parent1[:crossover_point] + parent2[crossover_point:]
    child2 = parent2[:crossover_point] + parent1[crossover_point:]
    new_pop.append(child1)
    new_pop.append(child2)
# Mutation
for i in range(len(new_pop)):
    if random.uniform(0, 1) < 0.1:
        rand_idx = random.randint(0, len
```

Intensive experiments have been conducted to evaluate the performance of the proposed scheduling algorithm on datasets of different scales under different scenarios against FF, BF, the traditional GA, and the BLA [24]. The obtained results reveal that the proposed algorithm can achieve a better balance between the makespan and the total execution cost than the other algorithms. Additionally, the conducted experiments show that the proposed scheduling algorithm has the minimum average delay time and failure rate. Additionally, it was noticed that the optimization algorithms (i.e., the proposed algorithm, the GA, and the BLA) have better performance compared to the state-of-art scheduling algorithms (i.e., FF and BF) with respect to the different performance metrics except the elapsed run time.

Section 5: Future Scope of Work:

Open research challenges and future work for Decision-Making for IoT Devices.

Resource allocation is usually formulated as an optimization problem. Accordingly, it is susceptible to intractable optimization problems, in which problem constraints would have to be relaxed at the cost of risking sub-optimal solutions. ML/DL methods have the potential to come up with acceptable near-optimal solutions in reasonable amounts of time for challenging optimization problems. To this end, researchers' interest in interdisciplinary approaches has increased, and various AI methods (ML/DL-based) are under investigation for resource allocation. In this work, we have comprehensively reviewed the findings of studies of AI approaches for decision making to the resource allocation problems in diverse computing paradigms and have discussed the successes and shortcomings. Future work is still needed to develop new methods capable of handling resource allocation with reasonable computational complexity and performance.

The performance of AI methods is dependent on the availability and quality of training data. There is a huge problem of noisy and unlabeled data in heterogeneous platforms such as IoT, mobile edge computing, etc. Many AI methods rely on supervised training, which requires labeled training data with good quality. Preparing such data may be difficult and time-consuming, which poses a limitation to applying AI methods. For example, one fast-growing use case is the Internet of Medical Things, which concerns remote healthcare services. Researchers' ultimate goal is to develop AI-based healthcare systems that dispense with the need for human intervention. However, there is extremely low tolerance for erroneous decisions in safety-critical domains such as healthcare. Therefore, high-quality labeled training samples are mandatory for training AI models. Such data are challenging to obtain since the labeling process is typically carried out fully manually by medical experts.

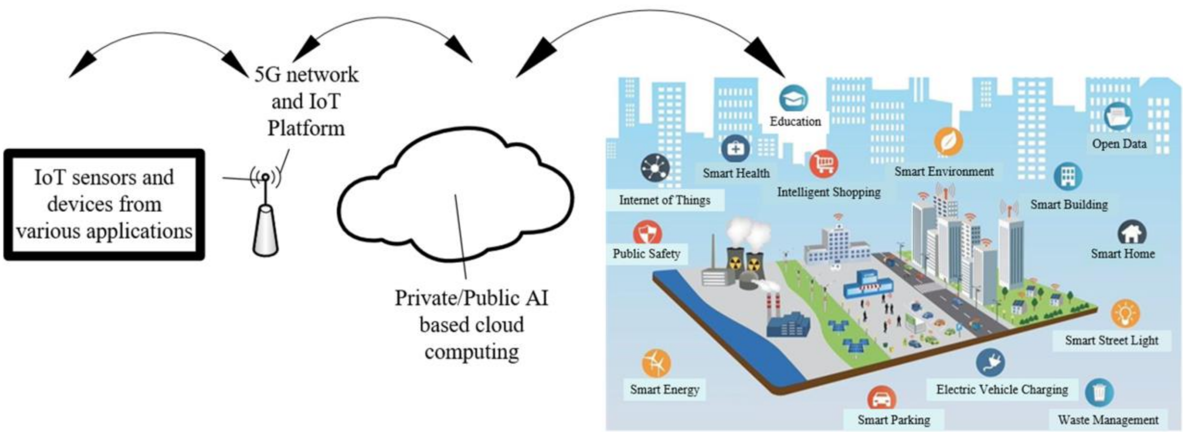
Another factor affecting the performance of complex AI models is hyperparameter tuning. For a DL model to be trained for a specific task, apart from model parameters that will have to be adjusted during the training process, model hyperparameters such as the number of hidden layers, learning rate, regularization coefficient, etc. are critical to successful training. Hyperparameter tuning requires searching a complex multidimensional space, which is onerous and can be confusing. Hyperparameter optimization tools like Wandb [22], Comet [23], etc., may facilitate the process by keeping track of the conducted optimization experiments.

Another important challenge regarding applying AI methods in different computing paradigms is the ability to make the models contextual. As in computing paradigms such as IoT, the tasks may dynamically change, and training DL models to adapt to new changes will impose high computation costs, which is impractical. Hence, the training effort is to be devoted to the generative adversarial network, and no further training will be required when running the application.

Effective resource allocation approaches must be able to withstand unforeseen resource shortcomings. For instance, should a cloud computing server temporarily loses some of its computation resources due to a cyber-attack, it will be necessary to change the resource allocation priorities. Tasks with higher priorities will be granted access to the available resources. Dynamically changing the resource allocation strategy is indirectly related to the contextual models as brought out in the previous paragraph.

Further, specific to the proposed algorithm IGA-POP, there is a scope of optimising it in fully dynamic data flow cloud-fog environment to utilise effectively in smart city.

Future Scope of Work for AI usage for Decision-making for IoT Devices and for more Practical and Effective Implementation .



Applications in Smart City

The future expansion and efficiency of the IoT networks and effective utilization in smart city is going to depend on the development of every sector, such as a smart environment, smart agriculture, smart economy, smart business, and smart governance. The ability to incorporate the latest 5G communication technology, IoT devices, and AI algorithms into a city's infrastructure is going to play a crucial role in the success of a smart city. Integrating these technologies can harmonize to produce a more efficient, sustainable, and livable city. Combining all the sectors can help build a smart city; after fulfilling the requirements brought out. The available data transfer standard developed for IoT networks is not compatible. Therefore, lots of work must be conducted to enable intercommunication between the sensor nodes using different communication protocols while operating under low power. Another area the researchers or stakeholders must focus on is developing efficient storage techniques and low-power hardware design to reduce operational costs. Heterogeneous networks for individual applications should be processed in one giant smart city network, and 5G can play a vital role in future smart city concepts. AI also has an enormous possibility for future work, including developing data fusion techniques for making heterogeneous data sources more accessible and intelligent data reduction for ensuring that surplus or 'uninteresting' data are not part of the AI development pipeline. In summary, the current communication technologies are inadequate in providing uninterrupted connectivity in smart cities as they were initially created to accommodate a restricted number of devices and possess limited communication capabilities. Thus, there is a pressing need to develop intelligent and standardized protocols, such as the Internet of Things enabled by 5G technology, to cater to the needs of future smart cities effectively. The current review paper has not thoroughly examined the ethical, social, and political ramifications of integrating AI with IoT in smart cities, including its potential impact on privacy, security, and social justice. Additionally, it lacks a comprehensive critical analysis of the advantages and drawbacks of this integration, leaving unanswered questions regarding the long-term sustainability, scalability, and efficacy of such approaches in smart cities.

Section 6: Conclusion:

IoT technology plays a significant role in many aspects of modern life by providing a wide variety of applications, including home healthcare, smart manufacturing, smart agriculture, smart retailing, and smart transportation. Many of these applications require fast response times and low latency, which cannot be achieved by overwhelmed cloud resources. Fog resources are used together with cloud resources to achieve the requirements of real-time IoT applications. In this study, a semi-dynamic real-time task scheduling algorithm is proposed to schedule real-time tasks of IoT applications in the cloud–fog environment. The proposed scheduling algorithm approaches task scheduling as a permutation-based optimization problem. Hence, it employs the IGA-POP to provide the best permutation for the submitted real-time tasks. Then, the real-time tasks are assigned to the VMs that achieve the minimum expected finish time. Intensive experiments have been conducted to evaluate the performance of the proposed scheduling algorithm on datasets of different scales under different scenarios against FF, BF, the traditional GA, and the BLA. The obtained results reveal that the proposed algorithm can achieve a better balance between the makespan and the total execution cost than the other algorithms. Additionally, the conducted experiments show that the
proposed scheduling algorithm has the minimum average delay time and failure rate. Additionally, it was noticed that the optimization algorithms (i.e., the proposed algorithm, the GA, and the BLA) have better performance compared to the state-of-art scheduling algorithms (i.e., FF and BF) with respect to the different performance metrics except the elapsed run time. For future work, the proposed scheduling algorithm can be extended to be applicable in dynamic environments which use containers rather than VMs. The capability of containers environment is more suitable to the dynamic environment due to its flexibility and versatility to improve resource utilization. Moreover, the absence of virtualization layer in the container, incurs less performance overhead on the applications.

In addition, the proposed can be adapted to address the workflow scheduling problem. Moreover, the proposed scheduling algorithm can be integrated with load balancing techniques. Additionally, other optimization algorithms, such as the shark optimization algorithm and whale optimization algorithm, can be evaluated for task scheduling in a cloud–fog computing environment. Finally, deep learning techniques can be employed to tackle the dynamic scheduling problem.

Further, the necessary communication between different systems, including smart devices, storage, servers, communication networks, is an undeniable part of daily life. Optimizing and increasing the efficiency of this communication is an important consideration, and resource allocation is a critical bottleneck. Researchers are using innovative AI methods to optimize resource allocation according to the data flow during network operation to solve the challenge of resource allocation. These measures have moved the industry towards automated resource management on a large and complex scale. This article has reviewed various AI methods used for decision making for resource allocation problem in different computing environments and summarized the performance in terms of response time, energy efficiency, throughput, cost, service consuming delay, convergence time, latency, etc. New resource allocation methods are continually being developed, and the computing environments have shifted from cloud to fog and edge.

References:

[1]Jin, W., Lim, S., Woo, S. et al. Decision-making of IoT device operation based on intelligent-task offloading for improving environmental optimization. Complex Intell. Syst. 8, 3847–3866 (2022). https://doi.org/10.1007/s40747-022-00659-z

[2]Andronie M, Lăzăroiu G, Iatagan M, Uță C, Ștefănescu R, Cocoșatu M. Artificial Intelligence-Based Decision-Making Algorithms, Internet of Things Sensing Networks, and Deep Learning-Assisted Smart Process Management in Cyber-Physical Production Systems. Electronics. 2021; 10(20):2497. https://doi.org/10.3390/electronics10202497

[3]Joloudari, J. H., Alizadehsani, R., Nodehi, I., Mojrian, S., Fazl, F., Shirkharkolaie, S. K., ... & Acharya, U. R. (2022). Resource allocation optimization using artificial intelligence methods in various computing paradigms: A Review. arXiv preprint arXiv:2203.12315.

[4]Lavalle A, Teruel MA, Maté A, Trujillo J. Improving Sustainability of Smart Cities through Visualization Techniques for Big Data from IoT Devices. Sustainability. 2020; 12(14):5595. https://doi.org/10.3390/su12145595

[5]Jiayi Guo, Shah Nazir, "Internet of Things Based Intelligent Techniques in Workable Computing: An Overview", Scientific Programming, vol. 2021, Article ID 6805104, 15 pages, 2021. https://doi.org/10.1155/2021/6805104

[6]Logeshwaran, J., Kiruthiga, T., &Lloret, J. (2023). A novel architecture of intelligent decision model for efficient resource allocation in 5G broadband communication networks. ICTACT Journal On Soft Computing, 13(3), 2986-2994.

[7]Y. Xiao, Y. Li, G. Shi and H. V. Poor, "Optimizing Resource-Efficiency for Federated Edge Intelligence in IoT Networks," 2020 International Conference on Wireless Communications and Signal Processing (WCSP), Nanjing, China, 2020, pp. 86-92, doi: 10.1109/WCSP49889.2020.9299798.

[8]Ahmed QW, Garg S, Rai A, Ramachandran M, Jhanjhi NZ, Masud M, Baz M. AI-Based Resource Allocation Techniques in Wireless Sensor Internet of Things Networks in Energy Efficiency with Data Optimization. Electronics. 2022; 11(13):2071. https://doi.org/10.3390/electronics11132071

[9]Mohammadi, V., Rahmani, A.M., Darwesh, A.M. et al. Trust-based recommendation systems in Internet of Things: a systematic literature review. Hum. Cent. Comput. Inf. Sci. 9, 21 (2019). https://doi.org/10.1186/s13673-019-0183-8

[10]Alahi MEE, Sukkuea A, Tina FW, Nag A, Kurdthongmee W, Suwannarat K, Mukhopadhyay SC. Integration of IoT-Enabled Technologies and Artificial Intelligence (AI) for Smart City Scenario: Recent Advancements and Future Trends. Sensors. 2023; 23(11):5206. https://doi.org/10.3390/s23115206

[11]Dang, V.A.; Vu Khanh, Q.;Nguyen, V.-H.; Nguyen, T.; Nguyen,D.C. IntelligentHealthcare:Integration of Emerging Technologiesand Internet of Things for Humanity.Sensors 2023, 23, 4200. https://doi.org/10.3390/s23094200

[12]L. Yang, X. Chen, S. M. Perlaza and J. Zhang, "Special Issue on Artificial-Intelligence-Powered Edge Computing for Internet of Things," in IEEE Internet of Things Journal, vol. 7, no. 10, pp. 9224-9226, Oct. 2020, doi: 10.1109/JIOT.2020.3019948

[13]Khraisat, A., Alazab, A. A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. Cybersecur 4, 18 (2021). https://doi.org/10.1186/s42400-021-00077-7

[14]Hao Qinxia, Shah Nazir, Ma Li, Habib Ullah Khan, Wang Lianlian, Sultan Ahmad, "AI-Enabled Sensing and Decision-Making for IoT Systems", Complexity, vol. 2021, Article ID 6616279, 9 pages, 2021. https://doi.org/10.1155/2021/6616279

[15]Mazhar T, Irfan HM, Haq I, Ullah I, Ashraf M, Shloul TA, Ghadi YY, Imran, Elkamchouchi DH. Analysis of Challenges and Solutions of IoT in Smart Grids Using AI and Machine Learning Techniques: A Review. Electronics. 2023; 12(1):242. https://doi.org/10.3390/electronics12010242

[16]F. Shi et al., "Recent Progress on the Convergence of the Internet of Things and Artificial Intelligence," in IEEE Network, vol. 34, no. 5, pp. 8-15, September/October 2020, doi: 10.1109/MNET.011.2000009.

[17]Tien-Wen Sung, Pei-Wei Tsai, Tarek Gaber, Chao-Yang Lee, "Artificial Intelligence of Things (AIoT) Technologies and Applications", Wireless Communications and Mobile Computing, vol. 2021, Article ID 9781271, 2 pages, 2021. https://doi.org/10.1155/2021/9781271

[18]Muhammad Shafiq, Changqing Du, Nasir Jamal, Junaid Hussain Abro, Tahir Kamal, Salman Afsar, Md. Solaiman Mia, "Smart E-Health System for Heart Disease Detection Using Artificial Intelligence and Internet of Things Integrated Next-Generation Sensor Networks", Journal of Sensors, vol. 2023, Article ID 6383099, 7 pages, 2023. https://doi.org/10.1155/2023/6383099

[19]K. Yang, Y. Shi, Y. Zhou, Z. Yang, L. Fu and W. Chen, "Federated Machine Learning for Intelligent IoT via Reconfigurable Intelligent Surface," in IEEE Network, vol. 34, no. 5, pp. 16-22, September/October 2020, doi: 10.1109/MNET.011.2000045.

[20] Yao Jun, Alisa Craig, Wasswa Shafik, Lule Sharif, "Artificial Intelligence Application in Cybersecurity and Cyberdefense", Wireless Communications and Mobile Computing, vol. 2021, Article ID 3329581, 10 pages, 2021. https://doi.org/10.1155/2021/3329581

[21] Hassannataj Joloudari, Javad & Alizadehsani, Roohallah & Nodehi, Issa & Mojrian, Sanaz & Fazl, Fatemeh & Khanjani Shirkharkolaie, Sahar & Kabir, H M Dipu & Tan, Ru San & Acharya, U Rajendra. (2022). Resource allocation optimization using artificial intelligence methods in various computing paradigms: A Review. 10.13140/RG.2.2.32857.39522.

[22] "https://wandb.ai/site," 2022.

[23] "https://www.comet.ml/site/," 2022.

[24] Abohamama, A.S., El-Ghamry, A. & Hamouda, E. Real-Time Task Scheduling Algorithm for IoT-Based Applications in the Cloud–Fog Environment. *J Netw Syst Manage* **30**, 54 (2022). https://doi.org/10.1007/s10922-022-09664-6

[25] Mukherjee, M., Shu, L., Wang, D.: Survey of fog computing: Fundamental, network applications, and research challenges. IEEE Commun. Surv. Tutor **20**(3), 1826–1857 (2018). https:// doi. org/ 10. 1109/ COMST. 2018. 28145 71

[26] 10. Mishra, S.K., Puthal, D., Rodrigues, J.J., Sahoo, B., Dutkiewicz, E.: Sustainable service allocation using a metaheuristic technique in a fog server for industrial applications. IEEE Trans. Ind. Inform. **14**(10), 4497–4506 (2018). https:// doi. org/ 10. 1109/ TII. 2018. 27916 19