A review study of Edge Computing Architecture For Predicting and Monitoring Water Quality in IOT System

[1.] **Mr. Dewarka Nath and Mrs. Jyoti Sharma**
[1]**Assistant Professor and Assistant Professor**
[1]Department. of Computer Science, Govt. Degree College Rajouri, J&K
[1]Department of Computer Science, Govt. Degree College Udhampur, J&K
dnsharam.po@gmail.com and jyoti.vashisth@gmail.com

Abstract - Clean water is crucial not only for drinking but also for cooking, washing, and personal hygiene. It helps prevent infections and promotes overall health. Manual tools and techniques used for monitoring quality of water are not sufficient to measure the quality of water in real-time. With the development of computing technology the monitoring of water quality indicators accurately are possible but not in real-time. The development of advance ICT devices and artificial intelligence, it's been possible to build potential and predictive models for monitoring and controlling pollution based on the real-time data. An intelligent water quality monitoring model is built on the real-time data available can predict the values of environment pollutants in real time. IOT based water sensors are embedded in water to measure the water quality indicators. By using the smart monitoring devices and machine learning techniques, the current level of pollution at the location is sent to users for information in real time. Machine learning model is used for the prediction of pollution level in the environment based on the previous data and the current data obtained by the sensors. In this review study, our aim to focus on the combination of machine learning and the edge computing paradigm in IOT environment. The presented research commences with the topic of edge computing, its benefits, such as reduced data transmission, improved scalability, and reduced latency, as well as the challenges associated with this computing paradigm, like energy consumption, constrained devices, security, and device fleet management strikes shortcomings of existing framework. It explores the areas of improvement and move towards a better and efficient model which allows data from environment sensors to be stored and processed locally at the edge and warns the users real-time. It implementation near the monitoring station saves network traffic and computing resources while ensuring the accuracy, privacy, security of data, low latency and lowering reliance on centralized cloud.

Keywords: Sensors, IoT, Edge Computing, Machine Learning, Environment Pollution.

Introduction:

The use of chemicals and pesticides in agriculture, population growth, and the expansion of industry have all contributed to the global problem of environmental pollution, particularly water contamination. Industries release harmful gases, hazardous waste (liquids, solids, and sludge), and toxic chemicals into the soil, water, and air. Agricultural pesticides, herbicides, chemical fertilizers, etc. contaminate soil and water. Air and water pollution has a major direct and indirect impact on human existence and the environment in the modern world. Sewage from septic tanks and drainage systems, as well as excess hazardous nutrients from the use of pesticides and fertilizers in agriculture, are the most common sources of pollution in rivers, lakes, and groundwater. The most widespread problems with water quality worldwide are sewage from septic tanks and drainage systems, as well as pollution of rivers, lakes, and ground water, particularly from the surplus hazardous nutrients that arise from the use of pesticides and fertilizers in agriculture. The biodiversity of the ecosystem is negatively impacted by this. Approximately 80% [50] of the wastewater produced by municipalities and industry worldwide is released into the environment untreated, damaging drinking water supplies, land, and the air. As a result, water is unfit for human consumption.

The environmental problem worsens with each passing day. Adequate supervision and innovative approaches are needed to ensure a tidy and clean environment and preserve a good society. For the monitoring and management of environmental toxins in real time, conventional methods were inadequate and slow.

Therefore, it is crucial to look for an intelligent and smart control system and a solution for monitoring environmental pollutants. Improved methods for real-time monitoring of environmental quality metrics must be developed. To reduce its detrimental impacts on people and the environment, it must be tackled with smart and sophisticated technology.

With the rapid advancements in integrated circuits (ICs), network and communication technologies, the Internet of Things (IoT), sensors, and actuators, the modern world may build a smart environmental monitoring system in real time. Smart environmental monitoring employs IoTs to routinely sample and analyze sensor data in order to better understand the natural environment, which includes the air, water, biota, and land. Environmental monitoring has developed into a Smart Environment Monitoring (SEM) system in recent years. The Internet of Things (IoT) may monitor and control environmental indicators that occur on land, in water, and in the air, send real-time data to a web server in a smart environment, and evaluate changes that might cause issues for people, animals, or plants.

The use of IoT devices has increased dramatically over the past 10 years, and there has been a growing interest in wireless communication network technology. As shown in 74, it is estimated that 39.6 billion IoT devices will be online by 2033 [1]. These devices produce a lot of data every second, but they are unable to handle raw data at IoT devices because of their short memory, low computing power, and low power backup. Furthermore, the low processing power of IoT devices prevents the adoption of machine learning (ML) techniques to make intelligent decisions close to the data source. Thus, their primary functions have been data generation, monitoring, and data transmission to centralized cloud systems for storage. and processing [2].

However, because it frequently results in high latency, requires a lot of bandwidth, causes network congestion, and raises data privacy issues because the data must be transferred and stored in the cloud indefinitely, this approach is not appropriate for real-time applications like drinking water pollution management, health indicators reporting, etc. Transporting all data to the cloud is becoming more difficult due to the rapid expansion of IoT devices as well as the quantity and frequency of data created by each IoT device (Figure 1). Additionally, even for little data, it necessitates a high level of user-cloud interaction. Numerous ideas have been put up to address the challenge mentioned above. One such method is edge computing, which stores and processes generated data close to its source and only sends necessary data to cloud servers for long-term processing and storage.

By integrating edge computing with cloud computing, complex data analytics activities can be carried out while reducing IoT network congestion, user response with remote servers, and associated latency[3–5]. Data dimensionality could be decreased by integrating machine learning (ML) capabilities into edge computing. In order to extract crucial data that is sent to the cloud for further analysis, this prepossessing would be carried out at the edge node.

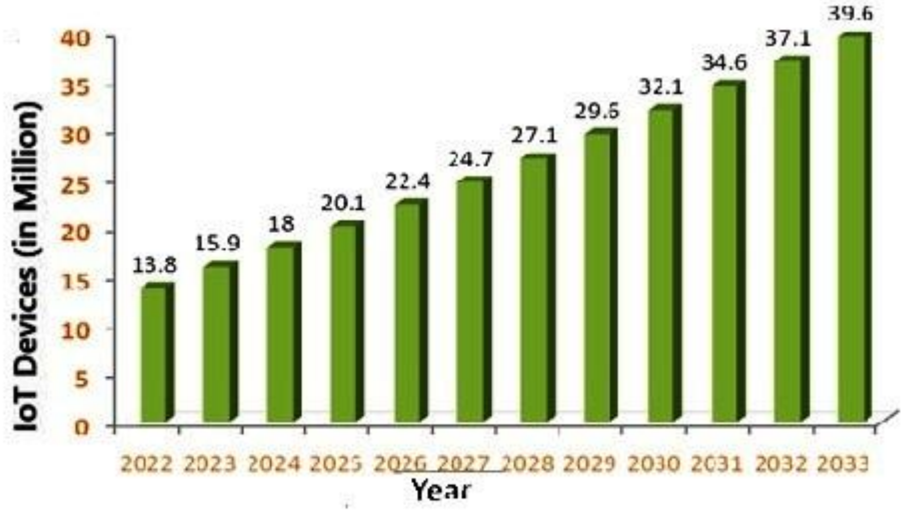

Figure 1: Growth of IoT Devices From 2022 to 2033

This paper is structured in the following manner: The general architecture of an IoT-based smart environment monitoring system is covered in Section 1, the principles of cloud-fog-edge computing in an IoT environment are reviewed in Section 2, Section 3 covers the integration of machine learning techniques at the edge is covered in.. Section 0 focuses on various potential IoT architectures, frameworks, and hardware to facilitate the deployment of ML models at the network edge in collaboration with cloud systems. In Section 5, various machine techniques used and their observation are covered. Section 6 describes the future directions of machine learning at the edge in an IoT environment, while Section 7 concludes the work and discusses the future direction

1. IoT Architecture

The centralized cloud system has stored data, carried out operations, and carried out data analytics at a considerable distance from the data sources and end users [6]. This centralized system involves large-scale traffic and congestion and lot of connected network devices. it is not practical and feasible to transmit all the IoT generated data to the cloud and get response after processing data when the applications requires instant reponse. To address these issues Fog computing and Edge computing are being developing by migrating storing and processing to the nearer of data source and edges of the network[7,8]. Edge computing and fog computing have some common featues[9] but the difference is that edge computing is focused on nodes that are nearer to data source while fog computing may encompass any data activities performed between the end device and the cloud [10]. Implementing ML algorithms directly on the IoT devices enable real-time decision-making through data analysis for end users controls congestion, reduces latency and secure the privacy of the user’s data. In order to achieve this, many IoT devices have been developed with inbuilt CPUs capable of implementing machine learning algorithms near the data source. Moreover, light weight frameworks and platforms that can support on-device machine learning are under development. The deployment and implementation of complex ML algorithms on the device itself is still limited due to the processing and memory constraints of IoT devices. These complex and time consuming ML based on applications distribute the data processing among IoT devices, edge, fog, and cloud computing for optimum efficiency.

While standardizing IoT architecture and real-time monitoring technology there is still need much advancement in framework and platform. Figure 2 precisely identify the major components of the IoT architecture commonly applied across a variety of applications. This IoT architecture includes :

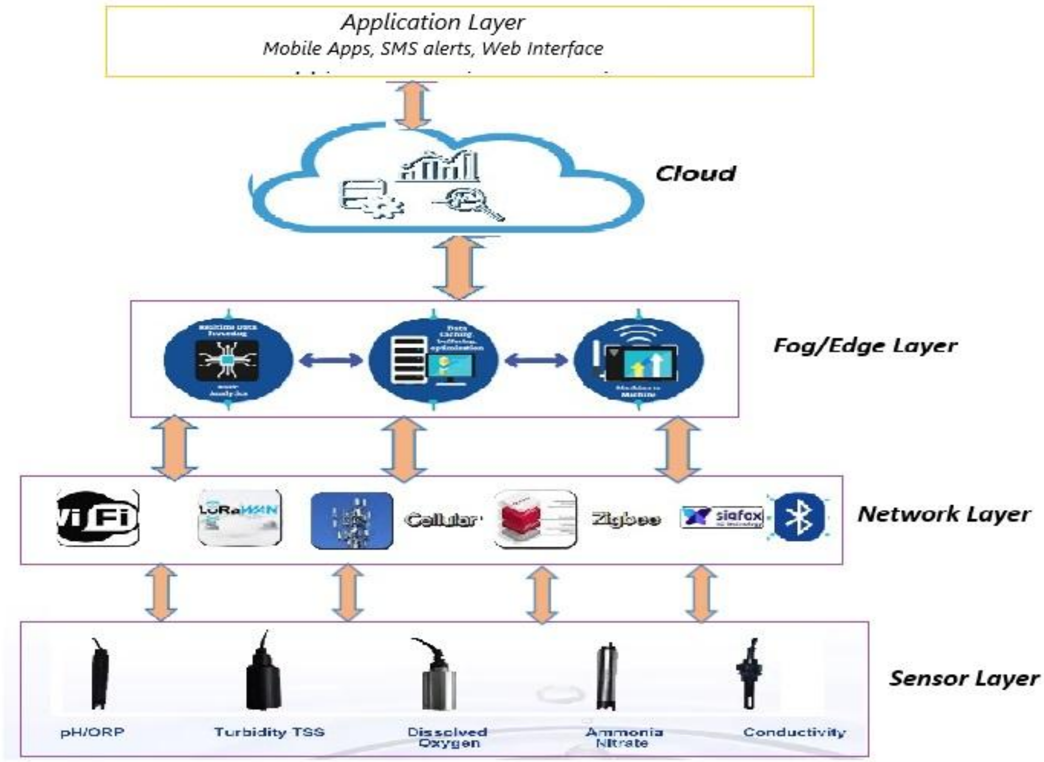

Figure 2 IOT Architecture of Smart environment monitoring System

Sensing Layer: This layer is responsible for object identification and data collection from environment. Additionally, it uses network technologies to send the created data to an edge or cloud server. Depending on the application, it comprises physical elements like sensors, actuators, and Internet of Things devices [11]. Light weight ML algorithms can be implemented in this layer, which we call as embedded intelligence.

Network Layer: Using wired or wireless transmission media, the network or communication layer routes and sends the data gathered from the physical devices to the higher layers(Edge of Fog layer). Wi-Fi, Bluetooth, cellular, LTE-Advanced, RFID, NFC, ZigWee, and other communication protocols are frequently used in IoT design.

Edge/Fog layer: This layer is referred to as the processing layer or middleware. It gathers and analyzes significant volumes of data sent from the network layer. It provides a variety of services to the foundational layers, encompassing databases, modules for big data processing, machine learning algorithms, and interaction with the cloud server. The computational workload might be distributed among the fog/edge and cloud servers

Cloud layer: Cloud Computing provides the most advanced and powerful computing paradigm and capable of handling thousands of requests at time[12]. In the architecture described, all the intricate computations take place on the cloud server. While Cloud Computing offers advantages such as cost reduction, efficiency, scalability, and reliability, it also presents challenges like low latency, the need for high internet bandwidth, and security and privacy concerns.

Application Layer: The aforementioned architecture's application layer offers customers services tailored to particular application. Tt communicates with end users/consumers through a range of IoT devices, including PCs, smart phones**,** SMS, and web interfaces.

2. Review of Literature:

Cloud, fog, and edge computing are current computing and storage concepts in the architecture of the internet of things. This computer system's storage layer manages massive amounts of data. When these paradigms are integrated with the Internet of Things, a reliable framework for collecting, storing, processing, and evaluating data is produced.

Smart cities are among the most popular applications of edge computing. In [13], The author explained how using the cloud and fog computing to manage services for smart cities may be made easier with the help of the service-oriented middleware (SOM) approach. For the full coordination and utilization of fog and cloud computing, they suggested SOM. This enhances the quality, flexibility, and integration of services required for a smart city applications. The processing of smart agricultural data from sensors that requires real-time operation was distributed in several nodes (Edge, Fog, Cloud) [14]. Thus, the real-time processing of sensors data was transferred to Edge and Fog nodes to reduce the burden on the Cloud for attaining the optimum objectives.

In [15], the proposed multi-layer fog architecture for smart city applications featuring an extensive data analysis service. The created fog-cloud computing framework, featuring separate functional modules, can mitigate the possible problems of a specialized computing platform and latency in cloud computing. The outcomes of experiments indicate that the analytical services are effective across multitier fogs and enhance QoS within the systems Other applications have used the edge-cloud framework for data analysis In [16], the author suggested a updated fog and cloud-based architecture for immediate extensive data evaluation in the Internet of Vehicles (IoV) setting. The architecture depicted concentrates on three aspects: smart computing, immediate large-scale data analysis, and intelligent computing

The main requirements of Edge-Fog-Cloud architecture, its reference architectures and applications were presented in [17]. it also presents interaction between Edge, Fog, and Cloud computing, its design and deployment considerations, such as: offloading, performance assessment, resource allocation and security issues.

In other study, water conservation edge computing framework by employing easy computing methods was used[18]. Considering various factors like the population size, household income levels, occupations of individuals, and historical water usage, the suggested model utilizes edge computing at the residential nodes within the network to assess the real-time water consumption of an individual household. The findings indicate that the suggested method performs more effectively on a self-created dataset compared to other comparable networks. An edge-IoT framework and a social computing architecture for energy-efficient system inside a public building scenario was developed [19]. The system is noticed, and the findings demonstrate how edge computing significantly enhances the framework and energy-saving conditions. The outcome of this work suggests that by splitting computation burdens from the cloud into the edge, systems with minimal computational capacity, consumption, and hardware costs may be deployed.

Another study [20] proposed a fog framework for mobile users to employ fog computing as a content delivery system. The fog nodes are situated in close proximity to end users. Users make requests to both the cloud and fog servers, and they respond to each request from either the cloud or the fog server, depending on whether the request has recently been initiated. According to the study, fog computing provides faster response times than cloud computing

3. Integration of ML techniques at the edge:

The system is made intelligent and smart through the application of machine learning techniques. The effectiveness of the ML algorithms employed determines how well the smart system performs. Supporting hardware, frameworks, data storage, model selection, etc., are necessary for integrating and implementing machine learning algorithms at the edge. The efficacy and efficiency of edge-based smart applications are significantly influenced by these aspects. In addition to meeting computing power and energy efficiency requirements, hardware choices must consider the type of application and particular circumstances. Additionally, when applying machine learning models at the edge network, model selection and compression are crucial.

4. IoT Hardware in Edge Intelligence:

IoT have been developed to interconnect physical devices with embedded intelligent devices using communication network. In [21] the study review the IoT is connecting our devices (fans, cell phones, vehicles, lamps, etc.) via the internet to facilitate our day to day work . The connected devices are equipped with a range of sensors, such as pH, DO, temperature, speed, etc., to gather and track environmental data, depending on the application. After that, the created data is transferred to the edge for storage and analysis. Additionally, IoT devices can use short- or long-range communication technology to communicate with other IoT architectures. Communication technologies can be either wired or wireless. While 4G, 5G, GSM, GPRS, LoRa, Sigfox, and other technologies are used for long-range communication, WiFi, ZigBee, Bluetooth, and other technologies are appropriate for short-range communication. IoT platforms like Thingspeak, DeviceHive, and others use a variety of protocols, such as HTTP, MQTT, AMQP, and XMPP, to manage M2M communication. However, with the exponential development of smart IoT devices that generate a massive amount of data every day, hence the possibility to congest the network.

4.1 IoT Design and Selection:

Implementing on-device machine learning inference or training is one of the current issues facing IoT devices because of their restricted resource capabilities both storage and computation power. Applications will require quick CPUs with low latency to process large amounts of data in order to support machine learning at the network edge. The most popular gadgets for implementing ML models at the edge will now be discussed. In [22], the study discusses the most commonly used single-board computer "Raspberry Pi", which has been used to run machine learning inference without the need for additional hardware. The Model 4 of Raspberry Pi board has a Quad Cortex A72 @ 1.5 GHz processor and supports up to 8 GB of SDRAM. Another AI enabled development board "Arduino Nano 33 BLE Sense" [23] is one of the cheapest boards available in the market. The Nano 33 BlE Sense is a straightforward and user-friendly device that allows you to build your machine learning model and upload it using the Arduino IDE. Because it can only run one program at a time, it is excellent for handheld technology. Arduino Nano Board 33 is used various applications such as Smart Health [24]. STM32 is a family of integrated circuits of 32-bit microcontroller produced by STMicroelectronics [25]. The internal components of each microcontroller include a processor core, flash memory, static RAM, a debugging interface, and a variety of peripherals. Specifically for its devices, such as the STM32Cube, it has created specialized libraries for implementing ML at the edge. It generates models in the industry-standard ONNX format as well as STM32-compatible C code-supplied NN models from Tensorflow and Keras.

4.2 Edge Frameworks and Libraries

Edge computing has been gaining its momentum as a result of the quick advancements in ML-based applications and the impossibility of processing solely in the cloud, which has led to the majority of ML-based applications being deployed at the edge. Consequently, innovative methods are needed to facilitate DL models at the edge. This increasing need has led to the development of specialized ML frameworks for edge computing architecture. These frameworks are essential for enabling the creation and deployment of lightweight models that can function in resource-constrained environments.. The popular AI frameworks and libraries utilized in most of the reviewed research papers are Pytorch, TensorFlow, Apache MXNet etc.

Primarily iOS and Android devices can run machine learning models with PyTorch. [26]. It has open-source library for building ML models uses dynamic computation and Python language [27]. The new PyTorch 1. 0 framework brings together the user experience of the PyTorch frontend with the scaling, deployment, and embedding features of the Caffe2 backend. PyTorch is favored by data scientists and deep learning researchers because of its simplicity and user-friendliness. TensorFlow is the most widely used open source framework for machine learning based applications. It is used for creating, training, evaluating, and deploying machine learning models [28,29] developed by the Google team in 2011 and made open-source by 2015. TensorFlow supports a wide range of applications, runs on Mac, Windows, Linux and even Android. TensorFlow light is designed for on-device inference and makes ML run everywhere with low latency.

The framework Apache MXNet [30] is a deep-learning library that is both powerful and versatile. It was designed to carry out training and inference activities by operating applications on a variety of platforms (cloud or edge). Additionally, it supports the Windows, Linux, and OSX operating systems as well as the Ubuntu Arch64 and Raspbian ARM-based operating systems. Because of its light weight, MXNet may be used in a distributed environment and integrated with a variety of host languages.

5. Machine Learning Techniques

Machine learning (ML) algorithms are employed to enhance the cognitive capabilities and intelligence of systems. This is a subset of artificial intelligence (AI), primarily concentrating on the enhancement of performance through autonomous learning or experiential acquisition. The efficacy of ML is contingent upon the specific ML algorithm implemented, the training conducted with extensive datasets, and the subsequent evaluation. ML is explicitly articulated by Mitchell, wherein ML encompasses the process of acquiring knowledge from experience E to augment the performance metric P on designated tasks. [5]. Mainly ML algorithms can be classified into four major categories: supervised, unsupervised, reinforcement and deep learning demonstrates a detailed taxonomy of ML algorithms.

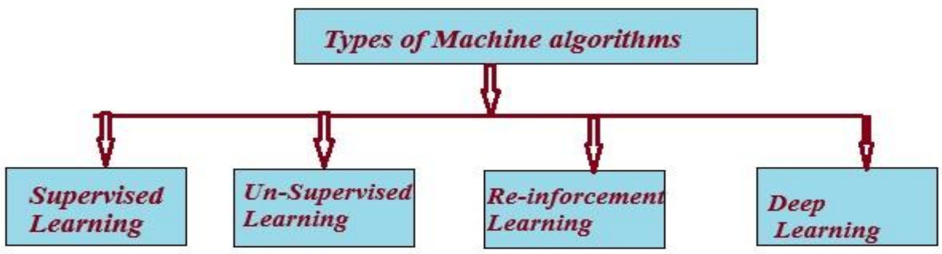

Figure 3 : **Types of Machine algorithms**

5.1 Supervised Learning: Supervised Learning, a category of machine learning algorithms, employs labeled datasets to refine its models and is predominantly applied in scenarios where both the input and output variables are predetermined. Within the realm of Supervised Learning, one can find methodologies such as linear regression, logistic regression, decision trees, and support vector machines. The objective is to cultivate the data in order to establish a predictive model that can proficiently forecast outcomes when confronted with unseen data instances[31,32]. Supervised machine learning implements classification and regression techniques. Some of the most popular supervised machine learning algorithms used in environment monitoring are support vector machine (SVM) and random forest (RF),

5.2 Unsupervised Learning: This particular category of machine learning algorithms employs unlabeled datasets to unveil patterns or characteristics inherent within the data. This encompasses techniques such as clustering, dimensionality reduction, and association rule mining. The primary aim is to discern patterns and structures within unlabeled datasets and endeavors to organize the data into coherent clusters accordingly. [33,34]. Clustering is among the most widely used unsupervised learning methods. Data with similar features are placed in the same groups, while dissimilar features are placed in other groups.

5.3 Reinforcement Learning: This type of machine learning algorithm adopt different approach and uses rewards and punishments to learn a task [35]. Examples include Q-learning and evolutionary algorithms. Reinforcement learning does not rely on data but rather on past experiences. It is commonly applied in autonomous systems like robotic , and AI gaming forming a reward-based learning applications.

5.4 **Deep Learning:** This category of machine learning algorithms employs neural networks to acquire proficiency in intricate tasks. It utilizes multilayered neural networks to emulate the sophisticated decision-making capabilities inherent to the human brain. Deep learning models undergo training utilizing annotated data. Upon completion of the training process, these models possess the capacity to autonomously categorize new data and recognize various data types independently. The following table

table compare the performance with limitation of each ML algorithms used in smart environment monitoring using IoT as per the study of researcher shown.

Table 1: **Comparison of ML algorithms used with their accuracy**

Research Topic	Purpose of study	Method/Device Used	Findings and Challenges	Limitation
Urban air quality monitoring[36]	Urban air pollution in terms of O3, NO2, SO2 concentrations	Forecasting models	Efficient and real time monitoring	Implement few air quality parameter
Smart air quality system [37]	Air quality	Temperature, humidity, dust and carbon dioxide sensor; LoRaWAN	Wide coverage	Minimum end user interaction and security issue
Intelligent air quality system [38]	Air quality for detection of CO2, NOx, temperature and humidity	UV light, AI and sensors	Use AI for quality monitoring	Minimum end user interaction
IoT based SEM [39]	Environmental monitoring	Heterogeneous sensors	W3C standard for interoperability;	interoperability issues
Smart Environment Monitoring [40];	Dust and humidity monitoring	IoT and sensors	Wide coverage and efficiency low cost and small size IoT	Security issues
Water quality monitoring [41]	Water Contamination analysis	Neural network for classification: drinkable or non-drinkable water	ANN and SVM ML used: variation in results	No notification to user in case sudden increase in pollution level
Water pollutant security [42]	Water contamination surveillance	SVM for classification as polluted or clean water	SVM with 93.8% accuracy and security	Issue in Real time monitoring
Water Quality pollutants parameters [43]	Water contamination analysis	Neural network for prediction for alkalinity, chloride, sulphate values	Liebenberg–Marquardt algorithm with 87.23% accuracy	Low accuracy and security issue
Big data and SVM [44]	Water contamination analysis	Machine learning based classification	SVM with 91.38% accuracy	No notification to user in case of sudden increase in pollution level

Water quality monitoring [45]	Drinking water analysis	Machine learning DT, KNN, SVM w	DT, KNN, SVM with 97% accuracy	Data Security issue
Pollution Monitoring [46]	Air pollution monitoring System	Sensors with MQ3 Model, Raspberry Pi and IoT	Real time monitoring;	Accuracy issues
Water quality forecasting[47]	TN, TP prediction	River flow, temperature, flow travel time, rainfall, DO, TN, and TP	ANN and SVM SVM>ANN	Accuracy issues

In the context of machine learning applications, the effectiveness and predictive precision of machine learning methodologies are typically contingent upon two factors, namely, the selection of the model and the integrity of the training dataset. Additionally, the quantity and nature of parameters employed in machine learning can also significantly influence the predictive accuracy of the model. Artificial Neural Networks (ANN) and Support Vector Machines (SVM) have demonstrated remarkable efficacy in the forecasting of water quality parameters. [47,48]. In some cases, SVM may produce better prediction accuracy and show higher generalization ability than ANN. Moreover, SVM uses an upper bound on the generalization error rather than reducing the training error [49]; therefore, it is more effective than ANN in minimizing this error.

6. Future Research Directions

The usage of edge computing in low latency applications, such as pollution and health monitoring, is growing rapidly due to the development of IoT and network technologies. To fully exploit the potential of edge computing, particularly in the context of the Internet of Things, a number of obstacles and unresolved concerns in the current research must be addressed. The section discusses these unresolved problems from the aforementioned study and suggests further research avenues.

- ✓ The lack of standardized IoT devices and architecture [21-27] that presents significant challenges in achieving interoperability, scalability, privacy and security in diverse IoT systems. There is a need of defining standard and common communication protocols, data formats, and security mechanisms to make compatible integration and communication between diverse IoT devices, architecture and platforms.
- ✓ Deployment of efficient and appropriate ML techniques in edge computing paradigms will play a crucial role. This ML techniques with error methods used playing the important role in determining the performance of applications. Not every ML algorithm with training sets and error methods (44,45,46,47) perform with same efficiency in each application. The selection of ML algorithms, type and number of parameters used and training dataset need to be considered based on the application
- ✓ Computing frameworks and architectures that optimize the distribution of computational tasks across edge devices, fog nodes, and cloud servers also crucial role. Advanced edge computing paradigms federated learning [51], distributed learning [52] are being investigated to enhance

efficiency and effectiveness in edge computing environments keeping in view the latency, privacy and bandwidth consumption.

- ✓ The development of high speed wireless communication and 5G/6G networks [53] will enhance the capability of edge computing by providing faster and more reliable data transmission, thus supporting the real-time processing needs of critical applications.
- ✓ The advancement of edge-native services and applications tailored to particular industries and use cases is expected to pick up speed. For example, real-time monitoring and diagnostics in healthcare can be made possible via edge native applications. The popularity of edge computing will be driven by the ability to customize applications to the specific needs of various environments, opening up fresh opportunities for innovative thinking.

7. Conclusions

This paper provides a comprehensive review of the edge-fog-cloud framework in IoT architectures, machine learning algorithms, and criteria for solutions that implement ML on IoT devices at various processing layers, with the primary goal of defining the current state of the art and anticipating emerging needs. Because of the rapid growth of IoT devices and the frequency of generated data, the traditional cloud computing method is unable to fulfill the changing demands of real-time monitoring applications particularly in time constraint applications. Deploying machine learning near the data source reduces network congestion, latency, and power consumption while maintaining the security and privacy of user data. This paper provided a comprehensive overview of all the key ML techniques frameworks, and even hardware selection for the efficient and successful execution of machine learning models. Moreover, it is observed in recent works that the most of the research on smart environment monitoring applications is mainly focused on Cloud-based and sensors. Few researches have applied combinations of Cloud, Fog, and Edge in the environment monitoring. It also observed that traditional applications relied heavily on Cloud, and new applications started to apply the combinations of computing paradigms such as Cloud-Edge and Cloud-Fog. This review study also highlighted the main research directions for effective machine learning deployment at the IoT edge. It highlights a number of problems and difficulties with edge-based IoT frameworks for real-time application monitoring.

Acknowledgment

I would like to express my sincere gratitude to all those who supported me throughout the course of this research. First and foremost, I am deeply indebted to my academic Supervisor, Dr. Bawinder Kaur, Associate Professor, LPU, for their invaluable guidance and technical support. I am also grateful to Govt. PG College Rajouri for providing the resources and facilities necessary for conducting this research. My heartfelt appreciation goes to my peers and colleagues, whose insightful discussions and suggestions have greatly enriched my research work. This research represents the collective efforts and inspiration of many, and I am profoundly grateful for the contributions of each individual and group

References

1. Number of Internet of Things (IoT) connected Devices Worldwide from 2022 to 2033. Available online:https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/ (Published on 11Sep 2024).
2. Shah, P.; Jain, A.K.; Mishra, T.; Mathur, G. IoT-Based Big Data Storage Systems in Cloud Computing. In Proceedings of theSecond International Conference on Smart Energy and Communication, Bilaspur, India, 19–22 December 2021; pp. 323–333.
3. Shu, Y.; Zhu, F. An edge computing offloading mechanism for mobile peer sensing and network load weak balancing in 5Gnetwork. J. Ambient. Intell. Humaniz. Comput. **2020**, 11, 503–510.
4. El-Sayed, H.; Sankar, S.; Prasad, M.; Puthal, D.; Gupta, A.; Mohanty, M.; Lin, C.T. Edge of things: The big picture on theintegration of edge, IoT and the cloud in a distributed computing environment. IEEE Access **2017**, 6, 1706–1717.
5. Rajavel, R.; Ravichandran, S.K.; Harimoorthy, K.; Nagappan, P.; Gobichettipalayam, K.R. IoT-based smart healthcare videosurveillance system using edge computing. J. Ambient. Intell. Humaniz. Comput. **2022**, 13, 3195–3207.
6. Gorelik, E. Cloud Computing Models. Ph.D. Thesis, Massachusetts Institute of Technology, Boston, MA, USA, 2013.
7. Mahbub, M.; Gazi, M.S.A.; Provar, S.A.A.; Islam, M.S. Multi-Access Edge Computing-Aware Internet of Things: MEC-IoT. InProceedings of the 2020 Emerging Technology in Computing, Communication and Electronics (ETCCE), Dhaka, Bangladesh,21–22 December 2020; pp. 1–6.
8. Pan, J.; McElhannon, J. Future edge cloud and edge computing for internet of things applications. IEEE Internet Things J. **2017**,5, 439–449.https://dx.doi.org/10.1109/JIOT.2017.2767608
9. Liu, Y.; Fieldsend, J.E.; Min, G. A framework of fog computing: Architecture, challenges, and optimization. IEEE Access **2017**,5, 25445–25454.https://dx.doi.org/10.1109/ACCESS.2017.2766923
10. Cha, H.J.; Yang, H.K.; Song, Y.J. A study on the design of fog computing architecture using sensor networks. Sensors **2018**,18, 3633.https://dx.doi.org/10.3390/s18113633.
11. Burhan, M.; Rehman, R.A.; Khan, B.; Kim, B.S. IoT elements, layered architectures and security issues: A comprehensive survey.Sensors **2018**, 18, 2796. https://doi.org/0.3390/s18092796.
12. De Donno, M.; Tange, K.; Dragoni, N. Foundations and Evolution of Modern Computing Paradigms: Cloud, IoT, Edge, and Fog.IEEE Access **2019**, 7, 150936–150948. https://dx.doi.org/10.1109/ACCESS.2019.2947652.
13. Mohamed, N.; Al-Jaroodi, J.; Jawhar, I.; Lazarova-Molnar, S.; Mahmoud, S. SmartCityWare: A service-oriented middleware forcloud and fog enabled smart city services. IEEE Access **2017**, 5, 17576–17588. https://dx.doi.org/10.1109/ACCESS.2017.2731382.
14. Alharbi HA, Aldossary M (2021) Energy-efficient Edge-Fog-Cloud architecture for ioT-based smart agriculture environment.IEEE Access. https:// doi. org/ 10. 1109/ ACCESS. 2021. 31013 97.
15. He, J.; Wei, J.; Chen, K.; Tang, Z.; Zhou, Y.; Zhang, Y. Multitier fog computing with large-scale iot data analytics for smart cities.IEEE Internet Things J. **2017**, 5, 677–686dx.doi.org/10.1109/JIOT.2017.2724845.
16. Darwish, T.S.; Bakar, K.A. Fog based intelligent transportation big data analytics in the internet of vehicles environmentMotivations, architecture, challenges, and critical issues. IEEE Access **2018**, 6, 15679–15701. 10.1109/ACCESS.2018.2815989.
17. Firouzi F, Farahani B, Marinšek A (2021) The convergence andinterplay of edge, fog, and cloud in the AI-driven Internet ofThings (IoT). Inf Syst. https:// doi. org/ 10. 1016/j. is. 2021. 101840.
18. Thakur, T.; Mehra, A.; Hassija, V.; Chamola, V.; Srinivas, R.; Gupta, K.K.; Singh, A.P. Smart water conservation through a machinelearning and blockchain-enabled decentralized edge computing network. Appl. Soft Comput. **2021**, 106, 107274. dx.doi.org/10.1016/j.asoc.2021.107274.
19. Sitton-Candanedo, I.; Alonso, R.S.; Garcia, O.; Munoz, L.; Rodriguez-Gonzalez, S. Edge computing, iot and social computing insmart energy scenarios. Sensors **2019**, 19, 3353. dx.doi.org/10.3390/s19153353.
20. Sheltami, T.R.; Shahra, E.Q.; Shakshuki, E.M. Fog computing: Data streaming services for mobile end-users. Procedia Comput. Sci.**2018**, 134, 289–296.https://doi.org/10.1016/j.procs.2018.07.173.
21. Burhan, M.; Rehman, R.A.; Khan, B.; Kim, B.S. IoT elements, layered architectures and security issues: A comprehensive survey. Sensors **2018**, 18, 2796 https://dx.doi.org/10.1016/j.procs.2018.07.173.
22. Raspberry Pi. Raspberry pi Products. 2021. Available online: https://www.raspberrypi.com/products/ (accessed on 10May 2021).
23. Store, A. Arduino Nano 33 BLE Sense. 2021. Available online: https://store-usa.arduino.cc/products/arduino-nano-33-ble-sense(accessed on 10 January 2022).
24. Daskalos, A.C.; Theodoropoulos, P.; Spandonidis, C.; Vordos, N. Wearable Device for Observation of Physical Activity with the Purpose of Patient Monitoring Due to COVID-19. Signals **2022**, 3, 11–28. https://dx.doi.org/10.3390/signals3010002.
25. ST. STM32. 2022. Available online: https://www.st.com/en/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus/ (accessed on 10 February 2022).
26. Gondi, S.; Pratap, V. Performance Evaluation of Offline Speech Recognition on Edge Devices. Electronics **2021**, 10, 2697. https://dx.doi.org/10.3390/electronics10212697.
27. Stevens, E.; Antiga, L.; Viehmann, T. Deep Learning with PyTorch; Manning Publications: Shelter Island, NY, USA, 2020.
28. Warden, P.; Situnayake, D. Tinyml: Machine Learning with Tensorflow Lite on Arduino and Ultra-Low-Power Microcontrollers; O'Reilly: ebastopol, CA, USA, 2020.
29. Zaccone, G.; Karim, M.R.; Menshawy, A. Deep learning with TensorFlow; Packt Publishing Ltd.: Birmingham, UK, 2017.
30. Chen, T.; Li, M.; Li, Y.; Lin, M.; Wang, N.; Wang, M.; Xiao, T.; Xu, B.; Zhang, C.; Zhang, Z. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. arXiv **2015**, arXiv:1512.01274. https://doi.org/10.48550/arXiv.1512.01274.
31. Devi, I.; Karpagam, G.; Kumar, B.V. A survey of machine learning techniques. Int. J. Comput. Syst. Eng. **2017**, 3, 203–212. https://dx.doi.org/10.1504/IJCSYSE.2017.089191.

32. Alsharif, M.H.; Kelechi, A.H.; Yahya, K.; Chaudhry, S.A. Machine learning algorithms for smart data analysis in internet of things environment: Taxonomies and research trends. Symmetry **2020**, 12, 88. dx.doi.org/10.3390/sym12010088.

33. Ongsulee, P. Artificial intelligence, machine learning and deep learning. In Proceedings of the 2017 15th International Conference on ICT and Knowledge Engineering (ICT&KE), Bangkok, Thailand, 22–24 November 2017; pp. 1–6.

34. Ayodele, T.O. Machine learning overview. New Adv. Mach. Learn. **2010**, 2, 16.

35. Sharma, K.; Nandal, R. A literature study on machine learning fusion with IoT. In Proceedings of the 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 23–25 April 2019; pp. 1440–1445.

36. Shaban, K.B.; Kadri, A.; Rezk, E. Urban air pollution monitoring system with forecasting models. IEEE Sens. J. 2016, 16, 2598–2606.

37. Amado, T.M.; Cruz, J.C. Dela Development of Machine Learning-based Predictive Models for Air Quality Monitoring and Characterization. In Proceedings of the TENCON 2018, 2018 IEEE Reg, Jeju, Korea, 28–31 October 2018; pp. 668–672.

38. Rosero-Montalvo, P.D.; Caraguay-Procel, J.A.; Jaramillo, E.D.; Michilena-Calderon, J.M.; Umaquinga-Criollo, A.C.; Mediavilla-Valverde, M.; Ruiz, M.A.; Beltran, L.A.; Peluffo-Ordónez, D.H. Air quality monitoring intelligent system using machine learning techniques. In 3rd International Conference on Information, Systems and Computer Science (INCISCOS 2018), Quito, Ecuador, 14–16 November 2018; pp. 75–80.

39. Mihăi¸tă, A.S.; Dupont, L.; Chery, O.; Camargo, M.; Cai, C. Evaluating air quality by combining stationary, smart mobile pollution monitoring and data-driven modelling. J. Clean. Prod. 2019, 221, 398–418.

40. Beebi, F. Environmental Monitoring System Using IoT. India Res. Tech. Organiz. 2018, 5, 64–68.

41. Bouamar, M.; Ladjal, M. Evaluation of the performances of ANN and SVM techniques used in water quality classification. Proc. IEEE Int. Conf. Electron. Circuits Syst. 2007, 1047–1050; DOI:10.1109/ICECS.2007.4511173.

42. Pang, Z.; Jia, K.; Feng, J. A water environment security monitoring algorithm based on intelligent video surveillance. In Proceedings of the 2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2014), Kitakyushu, Japan, 27–29 August 2014; pp. 191–194.

43. Ragi, N.M.; Holla, R.; Manju, G. Predicting Water Quality Parameters Using Machine Learning. In Proceedings of the 4th IEEE International Conference on Recent Trends on Electronics, Information & Communication Technology (RTEICT-2019), Bengaluru, India, 17–18 May 2019; pp. 1109–1112.

44. Budiarti, R.P.N.; Sukaridhoto, S.; Hariadi, M.; Purnomo, M.H. Big Data Technologies using SVM (Case Study: Surface Water Classification on Regional Water Utility Company in Surabaya). In Proceedings of the 2019 International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE 2019), Jember, Indonesia, 16–17 October 2019; Volume 1, pp. 94–101

45. Jalal, D.; Ezzedine, T. Toward a smart real time monitoring system for drinking water based on machine learning. In Proceedings of the The 27 th International Conference on Software, Telecommunications and Computer Networks (SoftCOM 2019), Split, Croatia, 19–21 September 2019; pp. 1–5.

46. Dhas, Y.J.; Jeyanthi, P. Environmental Pollution Monitoring System Using Internet of Things (IoT). J. Chem. Pharm. Sci. 2017, 10, 1391–1395.

47. Sci. Pollut. Control Ser., Support vector machine-an alternative to artificial neuron network for water quality forecasting in an agricultural nonpoint source polluted river? 21 (2014), pp. 11036-11053, 10.1007/s11356-014-3046.

48. A. Parsaie, A.H. Nasrolahi, A.H. Haghiabi; Water quality prediction using machine learning methods Water Qual. Res. J., 53 (2018), pp. 3-13, 10.2166/wqrj.2018.025.

49. A novel machine learning-based approach for the risk assessment of nitrate groundwater contamination; Sci. Total Environ., 644 (2018), pp. 954-962, 10.1016/j.scitotenv.2018.07.054.

50. Bhanu Pratap and Saroj Kumar, et al; Wastewater generation and treatment by various eco-friendly technologies: Possible health hazards and further reuse for environmental safety; https://doi.org/10.1016/j.chemosphere.2022.13754; Volume 313, February 2023, 137547

51. 109. Hasan, M.K.; Jahan, N.; Nazri, M.Z.A.; Islam, S.; Khan, M.A.; Alzahrani, A.I.; Alalwan, N.; Nam, Y. Federated Learning for Computational Offloading and Resource Management of Vehicular Edge Computing in 6G-V2X Network. IEEE Trans. Consum. Electron. **2024**, 70, 3827–3847. https://dx.doi.org/10.1109/TCE.2024.3357530.

52. Yuan, Y.; Chen, S.; Yu, D.; Zhao, Z.; Zou, Y.; Cui, L.; Cheng, X. Distributed Learning for Large-scale Models at Edge with Privacy Protection. IEEE Trans. Comput. **2024**, 73, 1060–1070. https://dx.doi.org/10.1109/TC.2024.3352814.

53. Chandra, S.; Arya, R.; Singh, M.P. Intelligent resource management in 5G/6G network by adopting edge intelligence for higher. education systems. E-Prime Electr. Eng. Electron. Energy **2024**, 8, 100517. https://dx.doi.org/10.1016/j.prime.2024.100517