ISSN: 0975-833X

Available online at http://www.journalcra.com

International Journal of Current Research
Vol. 15, Issue, 07, pp.25383-25387, July, 2023
DOI: https://doi.org/10.24941/ijcr.45624.07.2023

INTERNATIONAL JOURNAL OF CURRENT RESEARCH

RESEARCH ARTICLE

AN ANALYSIS OF MIST COMPUTING AND ITS APPLICATIONS IN MODERN TECHNOLOGY

***Karan Chawla**

Ashoka University, 131029, India

ARTICLE INFO

Article History:
Received 10th April, 2023
Received in revised form
08th May, 2023
Accepted 20th June, 2023
Published online 26th July, 2023

Key words:
Plastic, Compounding, Flexibility

**Corresponding Author:*
Dr. Mahak Gagain

ABSTRACT

Following the widespread adoption of cloud computing, researchers and businesses have been developing additional approaches that bring computing closer to the end users in order to realise ground-breaking applications, provide better QoS, exert more control over sensitive data, and maximise resource reuse. So, a general definition of mist computing is the downward push of cloudy computing characteristics to sensor networking at the very edge of the network. As is customary for Internet technology trends, Mist Computing has quietly become well-known, but it still needs a lot of work and time to evolve into a clear and understood technology that changes the industrial landscape. The conventional methods have drawbacks, such as a lack of generality and asignificantly slower rate of convergence. In order to address the challenge of sending enormous amounts of data to far-off cloud data centres, new computing paradigms like mist and fog computing have emerged as a result of the Internet of Things' (IoT) fast proliferation. The data is transmitted to the cloud server in the traditional setup of a GIS framework based on cloud computing, where it is processed and examined. On the Internet, this method used a lot of computing power and traffic. The Internet of Things (IoT), which consists of a network of connected sensors and devices, is continually growing. As a result, massive volumes of data of all kinds being created at speeds that were previously unheard of. Fog computing emerged as a solution to the problem of moving huge amounts of IoT data to distant cloud resources for processing by adding a layer between the IoT and cloud levels. This review paper addresses three of applications of mist computing, specifically Internet of Things, SmartHealth and Geospatial Data Analysis.

Citation: *Karan Chawla.* 2023. "An Analysis of Mist Computing and its applications in Modern Technology.*". International Journal of Current Research*, 15, (07), 25383-25387.

INTRODUCTION

Deep learning techniques are now being used more often in the analysis of large data in healthcare, including electronic health records, medical imaging, and sensor data from wearable devices. "Mist computing," a young subject, has showed considerable promise for processing data at the network edge, closer to where the data is created, to minimise latency and bandwidth needs and increase privacy and security. Deep learning's use in managing healthcare data is a new area of research, and there have been numerous papers in this field. It proposes a system that uses deep learning methods and a mist computing approach to manage large amounts of healthcare data. In order to analyse data at the network's edge in a resource-constrained setting close to where the data is created, mist computing is a distributed computing model that mixes cloud computing with edge computing. Mist Computing has quietly risen to prominence, as is typical for Internet technology trends, but it still requires a lot of work and time to develop into a mature and understandable technology that transforms the industrial environment. Despite this, there are several recent research lines that make use of this paradigm, either concentrating on aspecific issue or offering a comprehensive solution, as well as some Fog/Edge-oriented initiatives that accidentally touch on this issue.

Smart Health: Heart disease continues to be a leading cause of death worldwide, and it can be difficult for medical professionals to identify the early signs of cardiac patients. To this purpose, a number of traditional machine learning models have become popular for accurately predicting cardiac illnesses while accounting for the patients' underlying health issues. These techniques have limitations including a lack of generalisation and a substantially slower rate of convergence. In order to enable low-latency and energy-efficient processing of the healthcare data, a CloudFog computing-based paradigm is required as the healthcare data connected with these systems grow up and cause healthcare big data difficulties. The development of cutting-edge communication and storage technologies, as well as advances in the medical, agricultural, and other industries, have all greatly profited from the widespread use of smart gadgets. The Internet of Things (IoT) has significantly impacted our daily lives as a result of the rapid advancements in digital technology over the past few years. The Internet of Things (IoT) is frequently envisioned as a network of linked computing devices that exchange data to accomplish predetermined objectives. These systems frequently have CPUs, actuators, and sensors. Sensing technologies are utilised by IoT-enabled devices to generate large amounts of data, which are then transported through fog computing or cloud computing to areas where deep learning algorithms may be applied to make appropriate judgements. The cornerstone of a sophisticated economy that depends on the Internet to offer client services is fog computing, which incorporates cloud computing standards. The huge slowdown in response time caused by cloud computing, however, makes it unsuitable forapplications that need frequent input. Fog, cloud, and big data management in the context of IoT have all been popular owing to their usability and capacity to offer reaction features dependant on the tracked target applications. These new technologies enable storage, computation, and communication amongst edge devices.

This enhances latency and network bandwidth optimisation, mobility concerns, security, and privacy, and makes fog computing suitable for real-time or latency-sensitive applications. It's crucial to create a complete community where sensors in a body-area network may use this IoT network to connect data to cloud storage rather than just creating wearable gadgets. The body area sensor networks, Internet-connected access points, and edge as well as cloud computing components are the three main components of the design. Numerous apps provide services to various system stakeholders using this system. The data produced by sensors connected to users is accessible to carers, family members, andother authorised parties, enabling them to keep track of the subject's vital signs whenever and wherever they like. To encourage mistfog processing of obtained data, provide a deep learning ensemble approach that integrates with a large framework architecture. Over the mist computing layer, a framework for lightweight computing of cardiac patient diagnostics was constructed using the deep learning ensemble approach DeepMist. A system that blends the IoT, mist computing, fog computing environment, and the cloud is used to simulate relevant data processing. Several performance metrics, including precision, recall, f-measure, accuracy, energy efficiency, and transmission latency, are used to quantify the effectiveness of the DeepMist deployment. The main goal is to distribute vital healthcare information quickly by reducing the computational burden on IoT devices with limited resources and offloading computationally expensive jobs to cloud computing devices. By utilising the DQN algorithm to provide distributed cardiac disease prediction via the mist computing devices, the DeepMist framework adds intelligence. The suggested paradigm consists of four layers: a physical layer for the Internet of Things, a communication layer, a layer for fog computing, and a layer for cloud computing.

The real-time sensor data or non-real-time data provided by the end users via end devices like smartphones, laptops, or PCs is computed locally by the mist computing devices. In order to collect large amounts of sensory data from a dispersed network of IoT devices or end devices, the DeepMist framework is integrated at the mist computing layer. The mist computing layer makes it easier to handle data that has been offloaded from IoT devices that are resource- and power-constrained while also minimising the quantity of data sent across the communication network to higher levels like fog and cloud computing layers. The aforementioned IoT application layer devices may both pre-process and execute the IoT produced tasks locally with reduced latency and energy consumption by utilising the mist computing paradigm. The pre-processing and intelligent prediction modules are integrated into the IoT physical layer, which makes it easier to process sensor data. The IoT and mist computing devices are connected to the upper levels, such as the fog and cloud computing layers, by means of the communication layer, which is the second layer of the DeepMist architecture. The IoT devices need to be connected to its higher levels since they have limited computational power, bandwidth, and storage. The network and computing resources are optimised because IoT devices may effectively use the communication layer to transfer their computational tasks to neighbouring devices or fog nodes when the computational load on those devices increases. The fog computing layer, which is the third layer in the DeepMist architecture, is made up of fog nodes, gateways, and access networks. The operating system and processing unit of this layer, which may include hardware, microcontrollers, and signal processing unit, are included in the gateways. It offers the infrastructure and resources required by the fog nodes to carry out activities produced by the IoT physical layer. The access network is also in charge of defining the kind of communication that mist devices use to deliver information to the fog layer. Wide Area Network (WAN), Ethernet, WiFi, and Metropolitan Area Network are a few examples of access network concepts. The reinforcement learning paradigm, which may be thought of as an interactive framework that enables a learning agent to take some random action and modify its stateaccording on the reward for the action in a majority of predictable environment, is where the DQN algorithm gets its start. Consequently, the Markov Decision Process (MDP) may be used to represent reinforcement learning. To determine the goodness of a certain action to be in a given state, we need to estimate the value function. Based on the quantity of anticipated future benefits and the accuracy of anticipated number of returns, the learning agent determines the goodness of a given action. A specific manner of acting as a result of the learning process based on the policies may be defined as the value function for actions. In layer 1, there are sensor nodes. The mist computing nodes at layer 2 receive the data from the sensor nodes, which collect information about each object's state in the surrounding area. Sensors can be homogeneous, heterogeneous, single-dimensional, or multi-dimensional. The mist layer continually gathers data from the sensors and analyses it in an energy-efficient way with little expense. The information obtained from the mist layer is sent to the layer 3 fog nodes. The fog organiser is in charge of resource virtualization and grouping various mist nodes. It acts as a dispersed connection between the mist layer and the cloud layer. After receiving a request from an application manager or user, it allocates the resources and is also in responsible of monitoring and allocating the available resources. Data from sensors and the healthcare industry are processed by the fog layer. Four fog nodes coupled to the gateway node at layer 4 make up our experimental configuration. The simulation environment includes a variety of sensor nodes that are heterogeneously networked with the mist nodes to gather medical data. Furthermore, the heterogeneous setup is portrayed by the mist and fog computer nodes. It is situated in layer 5. In a many-to-one mapping, several layer 1 sensor nodes are combined into a single layer 2 mist node.

Geospatial Data Analysis: Geospatial data belonging to different stakeholders may now be shared and exchanged thanks to a GIS framework built on cloud computing. It has developed a setting that enables a wide range of users to securely access, retrieve, and distribute geographic data as well as related information. Land use and urban planning, environmental monitoring, natural resource management, marine, coastal, healthcare, and watershed management have all benefited from this cloud computing-based GIS framework. The GIS framework built on cloud computing has a wide range of new uses. To generate and visualise different planning scenarios, it has the capacity to combine and analyse various theme layers together with the data pertaining to their attributes. It combines distinctive visualisation features with typical geographic database activities including query creation, statistical calculations, and overlay analysis. These characteristics set the Cloud computing-based framework apart from existing systems for supporting spatial decisions.It is a technique that is frequently used in both the public and corporate sectors to describe events, forecast results, and develop plans.

Geospatial data includes useful time data as well as geographical distributions. In the conventional configuration of a GIS framework based on cloud computing, we transmit the data to the cloud server where they are processed and analysed. This technique consumed a lot of processing time and bandwidth on the Internet. This issue is solved by fog computing, which places local processing close to the edge of the clients. By lowering latency while increasing throughput, mist computing improves cloud and fog computing for processing geographical data. Additionally, it decreases geographic big data's cloud storage. Additionally, since we now transmit the analysis findings to the cloud rather than the data itself, the amount of transmission power needed to deliver the data to the cloud is decreased. The result is an increase in overall effectiveness. Clients like mobile phones and wearable sensing devices can communicate with one another using fog devices. An enormous amount of geographic big data was produced as a result of the growing usage of smart devices.These data are used by cloud, fog, and mist services to aid in various analyses. It implies that low-resource machine learning is used on fog devices that are near smart devices. At the network's edge, data are created and processed gradually. Similar work has been done in micro data centres and cloudlets in the past since cloud computing isn't always powerful enough to handle data when the data are created in large quantities at the network's edge. More edge devices or sensors may interact with the cloud on a greater scale thanks to edge computing. Because the infrastructures for cloud computing are not designed for the volume, velocity, and diversity of data. To better make usage of cloud services accessible, it is necessary to react to changes in the systems that exist before the cloud.

Edge computing may transfer request and delivery service from the cloud computing layer to client tier layer, conduct data storage, computing, offloading, caching, and processing. The variety of tasks in the edge network necessitates a well-designed edge node to effectively satisfy the demands for security, dependability, and privacy protection. Edge computing calls for placing the computer close to the data sources. These advantages above the conventional cloud computing paradigm have proved substantial. A paradigm of computation known as "cloud computing" makes use of the internet to deliver shared resources and on-demand services. It has the necessary computing tools and storage space for data visualisation and analysis. Moving from desktop to cloud servers was made possible by cloud computing. Different web processing architectures have been developed in a shared, open environment. Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS), and Database as a Service (DaaS) are four different types of service models that helped make it possible. It implements a one-of-a-kind multi-tenant design that enables several clients to contribute assets without interfering with one another. A paradigm of computation known as "cloud computing" makes use of the internet to deliver shared resources and on-demand services. It has the necessary computing tools and storage space for data visualisation and analysis. Moving from desktop to cloud servers was made possible by cloud computing. Different web processing architectures have been developed in a shared, open environment. Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS), and Database as a Service (DaaS) are four different types of service models that helped make it possible. It implements a one-of-a-kind multi-tenant design that enables several clients to contribute assets without interfering with one another.

By moving some of the processing to the network's edge, actuators, and sensors that make up the complete network for cloud data centres, fog computing has developed the notions of edge and fog computing further. The calculation has been carried out at the network's edge in the embedded nodes' microcontrollers with the aid of mist computing. The Mist computing paradigm boosts a solution's autonomy while reducing latency. When it comes to application activities, cloud, fog, and mist computing are complimentary to one another. The more computationally demanding work can be carried out in the fog layer's gateway while the less computationally heavy jobs may be carried out in edge devices. For the user's accessibility, the processing and data collection are still kept in the cloud data centre. Mist computing's most significant use is a group of various services that have been dispersed across the computer nodes. The terms "fog computing" and "mist computing," which fall in between the terms "fog computing" and "edge computing," were both developed by Cisco. They both expand thetraditional client-server architecture to a more peer-to-peer oriented approach, comparable to or equivalent to edge. A framework called MistGIS has been suggested for processing geospatial data analysis by taking into account mist computing. An example of an unsupervised machine learning method is K-means clustering. Here, we must collect objects of a similar nature. The K-means method is now a part of the open source R tools. Quantum GIS programme translated the shape file database into a.csv file format. K-means clustering algorithms are used on the resulting.csv files.

Internet of Things: Through thorough simulation studies with various security level probability for the first IoT input data of the entry tasks of the workflow jobs, the performance of both heuristicsis assessed. According to the simulation findings, the suggested technique not only offers a higher Quality of Service (QoS) than the baseline plan, but also saves money and energy. The Internet of Things (IoT), which includes an exponentially increasing number of networked sensors and gadgets, is still gaining pace. As a result, enormous amounts ofdata of various sorts are produced at previously unheard-of rates. By creating a layer between the IoT and cloud layers, fog computing arose as a solution to the issue of transporting massive volumes of IoT data to distant cloud resources for processing. By moving it closer to the IoT sensors and devices where the data are produced, it extends the cloud to the network edge. IoT data often has to be processed in real-time. As a result, the accuracy of the computations depends not only on their logical conclusions, but also on the speed at which they are reached. Examples of such applications include traffic management, weather forecasting, monitoring vital infrastructure, and healthcare monitoring. A simplified version of fog computing is mist computing. In an effort to reduce data transmission latency even further, it creates a layer between the fog and IoT layers, putting fog computing even closer to the IoT sensors and devices. As a paradigm, fog computing—and subsequently mist computing—shares many traits with cloud computing, albeit on a smaller scale, such as virtualization and resource sharing. In light of this, fog and mist resources, also known as fog nodes and mist nodes, respectively, can be virtual machines. When compared to fog resources, which are in turn scarce when compared to cloud resources, mist resources have a smaller capacity.

Sometimes, edge computing—another significant computing paradigm—is confused with fog and mist computing. But there are significant distinctions between them. Edge computing is specifically made available at the IoT layer by embedded local computing resources in IoT sensors and devices (i.e., edge computing is end-device dependant). As aresult, edge computing devices are a frequent name for the sensors and devices at the IoTlayer. Fog/mist computing, on the other hand, is end-device agnostic and offers a scalable, hierarchical architecture that is positioned between the IoT and cloud layers. In contrast to fog/mist computing, edge computing often does not allow resource pooling and virtualization technologies. Consequently, the IoT/edge layer, mist layer, fog layer, and cloud layer make form the bottom-up hierarchy of a multi-tier system that encompasses all of these paradigms. In this study, we simply refer to the IoT/edge layer as IoT layer and to the IoT sensors and devices as IoT sensors and devices for the purpose of simplicity. Workflow tasks are often used to process IoT data. A workflow job is made up of a number of component tasks that are linked together by data dependencies and precedence restrictions, with one task's output data feeding into another task's input. Usually, a workflow job's tasks all have deadlines that they must meet. An exit task is a task without any children, whereas an entry task is a task without any parents. An entrance job requires initial IoT input data that must be sent from the IoT layer in order to begin execution. Without finishing all of their parent tasks, non-entry tasks cannot begin execution. Additionally, in order to begin processing, all of their necessary input data must be sent (if it is not already accessible) to the resource that has been given to them. It may be essential to use more public cloud resources when demand is at its highest since mist nodes work together with fog nodes to handle IoT data. However, there are financial costs associated with using public cloud services, which should be considered while allocating resources. Additionally, the energy usage of the resources in the mist, fog, and cloud levels needs to be taken into account. Additionally, the IoT data that has to be handled may be sensitive, requiring that certain security standards be followed. This is crucial, especially in light of current rules like the General Data Protection Regulation (GDPR), which is among the strictest privacy and security regulations in the world. Generally speaking, it is more secure to handle highly secure data near to its source. Asdata are relocated from the controlled environment where they were created to a remote data centre of a public cloud provider, security threats, such as unauthorised access anddata interception, tend to rise.

The resources in the mist layer are the next physically nearest (and hence safest) resources to process the data since the sensors and devices in the IoT layer that generated the data often have relatively low processing capabilities. The mist or fog tiers can be used to process data with less stringent security standards. On the other hand, non-sensitive data can be handled on any tier, mist, fog, or cloud, depending on the security needs. As a result, it is essential to use a security-aware heuristic when scheduling workflow jobs that handle IoT data in a multi-tier context. The scheduling algorithm specifies the sequence in which the tasks will be handled on their allocated resources and is in charge of assigning the workflow jobs' tasks to the (security-wise) appropriate resources. Additionally, the scheduling strategy must adhere to the deadlines for the workload, minimise the financial expense of using additional public cloud resources, and maximise energy savings by effectively utilising the computing resources in the mist, fog, and cloud levels. IoT task scheduling in multi-tier contexts has been the subject of a significant amount of study. For the scheduling of stochastic operations in a fog-cloud environment, a trust model was put forth. Direct trust and reputation-based trust levels were the foundations of the suggested trust paradigm. It ensured that user processes would be executed in accordance with the degree of security specified in the Service degree Agreement (SLA) agreed with the cloud provider. The simulation trials proved that the suggested strategy works. But no mist tier and no workload deadlines were taken into account. Additionally, the suggested scheduling method did not include costs or energy. On the other hand, a scheduling technique that was cost- and security-conscious and took into consideration the workload's real-time limits in a fog-cloud context was provided. Jobs submitted by users were classified as private, semi-private, or public in the framework under examination. Additionally, fog and cloud resources were divided into categories of semi-trusted and untrustworthy. The network latency and the security tags of the jobs were taken into consideration while choosing the best fog and cloud resources for the job scheduling. Private jobs were assigned by the suggested technique to nearby fog resources or previously trusted cloud resources. On the other hand, it gave distant semi-trusted and untrusted fog and cloud resources semi-private and public tasks, respectively.

A heuristic that considers energy use, cost, and security when scheduling processes in a fog-cloud context. Authentication, integrity, and secrecy are the three factors that were taken into account in order to satisfy the security criteria. These characteristics' additional costs were taken into consideration. With the aim of reducing resource usage (and therefore energy consumption), network usage, and security overhead, a data mining approach was used. The outcomes demonstrated that the suggested method performed better than the other algorithms compared. Due to the disregard for deadlines, the suggested technique proved unsuitable for real-time workloads. Also, no mist layer was taken into account. Alternatively, a scheduling method that optimises energy and time for real-time processes in a fog-cloud environment. The tradeoff between competing objectives while distributing fog and cloud resources served as the foundation for the scheduling heuristic. While computationally intensive jobs were planned on cloud resources, latency-sensitive tasks were scheduled on fog resources. Additionally, the scheduling strategy made use of the dynamic voltage and frequency scaling (DVFS) method to save energy. However, the financial cost of the cloud resources was not included in the suggested strategy. Additionally, neither security specifications nor a mist layer were taken into account.

CONCLUSION

With varying numbers of mist nodes, the heart disease dataset was used to comparetraining accuracy for the QRL, DRL, and DQN approaches. It was found that the training accuracy consistently increases with the addition of each subsequent node as the number of mist computing nodes rises. This rise occurs because, as the number of mist nodes in the network increases, each node learns based on the experiences of its predecessors as well as the conceptual models that correlate to the heart disease data it gets for training. Because the models have a tendency to over-fit the sample training set, this causes the training accuracy of the models to increase for different iterations of the training set. It is also significant to note that, for all computing instances of the mist layer, the suggested DQN strategy beats the QRL and DRL approaches. It is important to highlight at this point that as the number of nodes rose, the accuracy of the test decreased since each node receives a smaller portion of the training dataset and cannot approximate to the DQN model. Additionally, when compared to QRL and DRL, the suggested DQN approach's adaptation consistently yields better performance. The learning algorithm used affects performance as well. However, when it comes to reducing the amount of energy used to create the model across a mist computing environment, the suggested DQN algorithm performs better than all the other methods. By surpassing the QRL and DRL techniques, which resulted in delays of 4.1211 ms and 3.8721 ms, respectively, the suggested DQN algorithm accounts for the lowest delay of 2.8002 ms in creating themodel.

The suggested framework was developed on top of the low latency, energy-efficient Mist computing platform, and it made use of the DQN algorithm to aid prediction. In order to establish how frequently the Mist computing layer offloads tasks to the Fog layer, we would place greater attention on calculating the best CPU utilisation of the Mist computing layer. Additionally, we would concentrate on creating task offloading techniques for safe offloading of medical data in the MistFog computing infrastructure. For IOT, The baseline policy, SAH, which was cost and energy conscious but not cost and security conscious, was contrasted with the suggested scheduling approach, SCEAH. Through simulation, the effectiveness of both policies was assessed under various security level probability values for the first IoT input data of the entry tasks of the workflow jobs. According to the simulation results, SCEAH was more successful than SAH overall, outperforming it in each of the areas that were tested. We want to research our suggested scheduling heuristic in conjunction with a dynamic scaling method for the cloud VMs in order to use the additional cloud resources more effectively, achieve larger energy savings, and incur fewer financial costs. We also intend to look into other techniques because in this study the security needs of the non-entry tasks of the workflow jobs were estimated using a cautious methodology. For instance, certain workflow processes may conduct partial or complete anonymization on the data they output as output. By using data masking or pseudonymization, for instance, sensitive information may be removedfrom a dataset and made anonymous. As a result, depending on how fully, partially, or not the job that processed the data anonymized it, the security level of the output data of a task may differ. Mist nodes result in efficient transmission at better speed and latency while also reducing storage needs. Scalable cloud computing adds a support layer thanks to edge computing on fog nodes. Huge amounts of data are being produced as wearable and internet-connected sensors are used more often. The cloud could be set aside for in-depth investigation. In contrast to cloud computing, mist computingplaces an emphasis on close proximity to end users as well as local resource pooling, a decrease in latency, greater service quality, and improved user experiences. Future plans include for the addition of more sophisticated processing techniques. With the availability of huge data, fog computing lessened the burden of dependency on Cloud services. Mist architecture is likely to have a significant role in determining how large data handling and processing are done in the near future. It would be fascinating to investigate the application of MistGIS in the fields of health, education, coastal management, natural resource management, watershed management, and energy.

REFERENCES

Bonomi, F., Milito, R., Zhu, J., Addepalli, S.. Fog computing and its role in theinternet of things. In: ACM MCC. 2012, .

Dubey, H., Constant, N., Monteiro, A., Abtahi, M., Borthakur, D., Mahler, L., et al. Fog Computing in Medical Internet-of-Things: Architecture, Implementation, and Applications. In: Handbook of Large-Scale Distributed Computing in Smart Healthcare. Springer International Publishing AG; 2017, .

Yang, C., Huang, Q., Li, Z., Liu, K., Hu, F.. Big data and cloud computing: innovation opportunities and challenges. International Journal of Digital Earth2017;10(1):13–53.

Evangelidis, K., Ntouros, K., Makridis, S., Papatheodorou, C.. Geospatial services in the cloud. Computers & Geosciences 2014; 63:116–122.

Barik, R., Dubey, H., Lenka, R.K., Mankodiya, K., Pratik, T., Sharma, S.. MistGIS: Optimizing Geospatial Data Analysis Using Mist Computing. In: International Conference on Computing Analytics and Networking (ICCAN2017). Springer; 2017, .

Dubey, H., Kumaresan, R., Mankodiya, K.. Harmonic sum-based method for heartrate estimation using PPG signals affected with motion artifacts. Journal of Ambient Intelligence and Humanized Computing 2016;:1–14.

Andreu-Perez, J., Poon, C.C., Merrifield, R.D., Wong, S.T., Yang, G.Z.. Big data for health. IEEE journal of biomedical and health informatics 2015;19(4):1193–1208.

Dubey, H., Mehl, M.R., Mankodiya, K.. BigEar: Inferring the ambient and emotional correlates from smartphone-based acoustic big data. In: Connected Health: Applications, Systems and Engineering Technologies (CHASE), 2016IEEE First International Conference on. IEEE; 2016, p. 78–83

Aslanpour, M. S., Gill, S. S., & Toosi, A. N. (2020). Performance evaluation metrics for cloud, fog and edge computing: a review, taxonomy, benchmarks and standards for future research. *Internet of Things*, *12*, 100273. https://doi.org/10.1016/j.iot.2020.100273

Auluck, N., Rana, O., Nepal, S., Jones, A., & Singh, A. (2019). Scheduling real time security aware tasks in fog networks. *IEEE Transactions on Services Computing*, 1–14. https://doi.org/10.1109/TSC.2019.2914649

Buttazzo, G. C. (2011). *Hard real-time computing systems: Predictable scheduling algorithms and applications*, 3rd edn. New York: Springer. https://doi.org/10.1007/978-1-4614-0676-1

Chen, Y. (2020). *Service-oriented computing and system integration: Software, IoT, big data, and AI as services*, 7th edn. Dubuque: Kendall Hunt Publishing.

Dogo, E. M., Salami, A. F., Aigbavboa, C. O., & Nkonyana, T. (2019). Taking cloud computing to the extreme edge: a review of mist computing for smart cities and industry 4.0 in Africa. In F. Al-Turjman (Ed.) *Edge computing: from hype to reality* (1st ed., pp. 107–132). https://doi.org/10.1007/978-3-319-99061-3_7. Springer.

Felicetti, C., Furfaro, A., Saccà, D, Vatalaro, M., Lanuzza, M., & Crupi, F. (2019). Making IoT services accountable: a solution based on blockchain and physically unclonable functions. In *Proceedings of the 12th International Conference on Internet and Distributed Computing Systems (IDCS'19)*, (pp. 294–305). https://doi.org/10.1007/978-3-030-34914-1_28

Hensh, F., Gupta, M., & Nene, M. J. (2021). Mist-Edge-Cloud (MEC) Computing: an integrated computing architecture. In *Proceedings of the Second International Conference on Electronics and Sustainable Communication Systems (ICESC'21)*, (pp. 1035–1040). https://doi.org/10.1109/ICESC51422.2021.9532929

Hosen, A. S. M. S., Sharma, P. K., Ra, I. H., & Cho, G. H. (2021b). SPTM-EC: a security and privacy-preserving task management in edge computing for IIoT. *IEEE Transactions on Industrial Informatics*, 1–9. https://doi.org/10.1109/TII.2021.3123260

Ding Y, Fang W, Liu M, Wang M, Cheng Y and Xiong N. (2023). JMDC. *Journal of Parallel and Distributed Computing*. 173:C. (83-93). Online publication date: 1-Mar-2023. https://doi.org/10.1016/j.jpdc.2022.11.008

Naghib A, Jafari Navimipour N, Hosseinzadeh M and Sharifi A. (2023). A comprehensive and systematic literature review on the big data management techniques in the internet of things. *Wireless Networks*. 29:3. (1085-1144). Onlinepublication date: 1-Apr-2023.

Khan A, Haq I, Hussain T, Muhammad K, Hijji M, Sajjad M, De Albuquerque V and Baik S. (2022). PMAL: A Proxy Model Active Learning Approach for Vision Based Industrial Applications. *ACM Transactions on Multimedia Computing, Communications, and Applications*. 18:2s. (1-18). Online publication date: 30-Jun-2022.

Mazhar S, Sun G, Bilal A, Li Y, Farhan M, Awan H and Khattak H. (2022). Digital and Geographical Feature Detection by Machine Learning Techniques Using Google Earth Engine for CPEC Traffic Management. *Wireless Communications & Mobile Computing*. 2022. Online publication date: 1-Jan-2022.

López Escobar J, Díaz Redondo R and Gil-Castiñeira F. (2022). In-depth analysisand open challenges of Mist Computing. *Journal of Cloud Computing: Advances, Systems and Applications*. 11:1. Online publication date: 19-Nov-2022.

Fraga-Lamas P and Fernández-Caramés T. (2022). Tactical Edge IoT in Defense and National Security. IoT for Defense and National Security. 10.1002/9781119892199.ch20. (377-396). Online publication date: 28-Dec-2022.

Panigrahi S, Rani Jena J, Goswami V, Shekhar Patra S, Samaddar S and Barik R. (2022). Performance Evaluation of State Dependent Queueing based Geospatial Mist-assisted Cloud System for Crime Data Visualisation 2022 3rd International Conference on Computing, Analytics and Networks (ICAN). 10.1109/ICAN56228.2022.10007265. 978-1-6654-9944-6. (1-

Galán J, Carrasco R and LaTorre A. (2022). Military Applications of Machine Learning: A Bibliometric Perspective. *Mathematics*. 10.3390/math10091397. 10:9. (1397).

Heidari A, Jafari Navimipour N, Unal M and Zhang G. (2023). Machine Learning Applications in Internet-of-Drones: Systematic Review, Recent Deployments, and Open Issues. *ACM Computing Surveys*. 55:12. (1-45). Online publication date: 31-Dec-2024.

****