Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages: 46 – 59 | e-ISSN: 2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.73

Design and Implementation of a Fog Computing Architecture for IoT Data Analytics

Received: 16 September 2023; Revised: 15 November 2023; Accepted: 16 December 2023

Dr. Parikshit Mahalle
Dean R & D, Vishwakarma Institute of Information Technology Pune
Scopus Profile: https://www.scopus.com/authid/detail.uri?authorId=35170655700
University Profile: https://www.viit.ac.in/component/jsn/parikshit-mahalle?Itemid=&back=1
LinkedIn: https://www.linkedin.com/in/parikshit-mahalle-876061116/
https://orcid.org/0000-0001-5474-6826

Sheetal S. Patil
Department of Computer Engineering,
Bharati Vidyapeeth University, College of Engineering, Pune
sspatil@bvucoep.edu.in

Abstract

The number of Internet of Things (IoT) devices has exponentially increased, creating an explosion of data that requires sophisticated processing and analysis techniques. When it comes to meeting the demands of long duration and narrow band of the things' Internet applications, traditional cloud computing solutions may encounter difficulties. To overcome these issues, fog computing has developed into a workable concept for extending nube services all the way to the system's edge. The development of a fog computing architecture for the analysis of data from the internet of things is the topic of this study. Our system's three primary parts are fog nodes, edge devices, and a central cloud server. Sensors and edge devices of the Internet of Things (IoT) are in charge of local preprocessing and data collection. In between network edge devices and the cloud server, fog nodes act as intermediaries. Their actions have reduced the volume of raw data sent to the cloud for processing and archiving. One cloud server manages all aspects of data analysis, storage, and archiving. In order to show how effective and efficient our architecture is, Our approach was supported by data gathered from a variety of Internet-connected devices, and by lowering the amount of data transferred to the cloud, we were able to considerably lower lag and the use of black band. The network's core fog nodes also offered the processing capacity required to carry out analysis relatively instantly. The advantages of the board devices, given that a large number of Internet of Things applications require real-time or almost real-time data processing, this architecture stands out because to its capacity to lower latency, save bandwidth, and improve system efficiency.

Keywords

Internet of Things, Fog Computing, Data Analysis, Cloud Computing

1. Introduction

The amount of data produced by Internet of Things (IoT) has significantly decreased in recent years due to their widespread adoption. Many industries, such as healthcare, intelligent cities, industrial automation, and even agriculture, may benefit from the Internet of Things. Processing, storing, and analysing data has become more challenging due to the increasing amount of data. This has also produced new opportunities for data-based decision-making and discovery [1]. Traditional cloud computing technologies are needed to manage and process large volumes of data. When applied to Internet of Things (IoT) scenarios, however, they frequently run into problems with latency, bandwidth, and scalability. Due to the time-sensitive nature of many IoT use cases, sending all data to distant cloud servers for processing is impractical.

Sending a lot of unprocessed data to the cloud also uses a lot of bandwidth, which might cause problems with your connection and add to your bill [2]. Fog Computing is a potential paradigm that brings cloud computing to the edge of the network, which can help with these issues. Fog Computing, also known as Edge Computing, makes use of nearby resources to process and analyse data from IoT devices in real time. It hopes to accomplish these goals by preserving bandwidth and decreasing latency in IoT networks [3]. This paper details the planning and execution of a Fog Computing Architecture that is optimised for Internet of Things data analytics. The purpose of this design is to take advantage of Fog Computing while maintaining full compatibility with existing Internet of Things (IoT) ecosystems. In this introductory section, we will discuss the reasoning behind [4], goals for, and building blocks of our suggested design. Sensors, actuators, and data generators are all examples of edge devices in the Internet of Things. Edge devices are responsible for collecting data at its source and conducting initial preparation, such as data filtering and compression. Fog Nodes: Fog nodes are strategically positioned at the network edge, serving as mediators between edge devices and the centralized cloud server. These [5] nodes are equipped with computational resources and are responsible for data aggregation; more advanced analytics, and localized decision-making. Centralized Cloud Server: The centralized cloud server is responsible for managing extensive data analytics, long-term data storage, and providing a global picture of the IoT ecosystem. While some data is handled at the fog nodes, the centralized cloud server works as a powerful backend for in-depth analysis and historical data management.

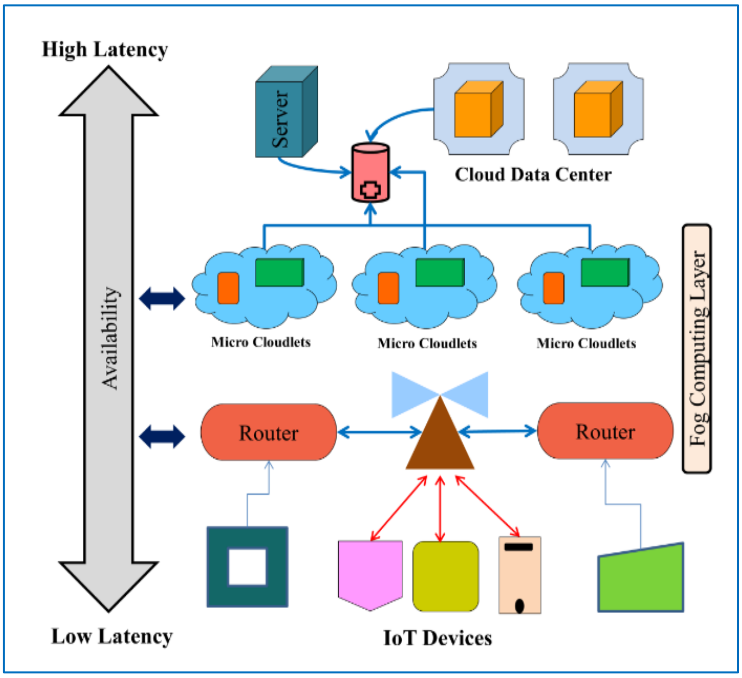

Figure 1: Representation of Fog Computing Architecture View

Motivation:

Our major goal is to make IoT applications more effective and responsive by fixing the underlying problems with IoT data analytics. The necessity of a Fog Computing strategy in this setting is driven by several factors:

- Latency-sensitive applications: Many IoT applications, such as autonomous vehicles, industrial automation, and remote healthcare monitoring, require real-time or near-real-time data analysis. Traditional cloud-based solutions cause unreasonable delays owing to data transfer to remote data centers and back.
- Limited Bandwidth: IoT devices frequently function in environments with limited bandwidth. Transmitting massive volumes of raw sensor data to the cloud might overload network infrastructure and incur high data transfer expenses.
- Scalability: As the number of IoT devices continues to grow, centralized cloud architectures

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages:46 – 59 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.73

face scalability challenges. Fog Computing offers a distributed approach that can accommodate the increasing scale of IoT deployments.

- Data privacy and security: Some IoT applications involve sensitive data that organizations may prefer to process locally to enhance data privacy and security. Fog Computing enables data to remain within the confines of the local network.

Objectives

The key objectives of our Fog Computing Architecture for IoT Data Analytics are as follows:

- Reduce Latency: Minimize the time required for data analysis and decision-making by processing data as close to the source as possible. This ensures that time-sensitive IoT applications operate with low latency.
- Conserve Bandwidth: Reduce the volume of data transmitted to the cloud by performing data aggregation, filtering, and preliminary analytics at the edge of the network. This helps alleviate network congestion and lowers data transfer costs.
- Enhance Scalability: Design a scalable architecture that can adapt to the increasing number of IoT devices and data sources. Distribute computing resources strategically to accommodate the growing demand for data analytics.

Maintain Data Privacy and Security: Implement mechanisms to ensure data privacy and security, particularly for applications that handle sensitive information. Give businesses the chance to keep control of their data by providing local processing options. In the coming sections of this article, we will delve into the detailed design considerations, architectural components, and implementation details of our Fog Computing solution for IoT data analytics. We will also present experimental results demonstrating the feasibility and effectiveness of our approach in real-world IoT scenarios. By achieving our objectives of reducing latency, conserving bandwidth, enhancing scalability, and ensuring data privacy and security, we believe that our architecture can significantly advance the state of IoT data analytics and empower a wide range of IoT applications across diverse domains.

2. Review of Literature

Fog Computing designs for [28] IoT data analytics have attracted a lot of interest from academics and professionals in the business world in recent years. In this section, we will review the most important results and contributions from previous research in this field, focusing on the methods and tools that have been developed to overcome the difficulties inherent in working with IoT data [6]. A key paradigm for IoT data analytics is Fog Computing, which is commonly used synonymously with Edge Computing. Researchers like Shi et al. (2016) introduced the term "Fog Computing" and highlighted its potential in lowering latency by handling data processing at the network's periphery. Our suggested architecture is hierarchical and comparable to theirs in that it involves Internet of Things (IoT) gadgets, fog nodes, and cloud servers [7]. Their findings paved the way for later research into Fog Computing and its potential to cut down on latency. Several research efforts have centred on edge data aggregation and filtering strategies as a means to reduce data traffic to the cloud. For smart grids, for instance [9] proposed a data aggregation strategy in which edge devices work together to collect data from numerous sources and send it to the cloud. This method lessens the load on the network and the cloud's processing power. In a similar vein, [8] suggested an edge-based data filtering system that uses event-driven triggers to send just the necessary data to the cloud. The goal of bandwidth conservation in Fog Computing designs is met by employing these tactics.

Machine learning (ML) at the edge has become increasingly popular as a way to conduct sophisticated analytics close to the data source. For predictive maintenance in industrial IoT applications, researchers [10] have investigated the incorporation of ML models on edge devices. They showed that ML models operating at the periphery could monitor for anomalies and predict breakdowns in real time, saving money on both downtime and maintenance. This is consistent with the objective of our architecture to improve decentralised decision-making. There is persistent worry that Fog Computing architectures would not adequately protect sensitive data. A security architecture for Fog Computing in healthcare IoT was introduced [11] highlighting the importance of safe data transfer, authentication, and access control. A trust management mechanism for creating credibility between fog nodes was also proposed. [12] study introduced differential privacy strategies to solve privacy preservation in Fog Computing, which enables data analysis without disclosing private details. Our design takes data security and privacy seriously, and it includes features to keep information safe [27].

Management and orchestration of Fog Computing resources have been studied in depth. The term "Fog Computing" was coined [13] who also detailed ways for allocating resources among fog nodes. Computational resources, they argued, should be distributed dynamically according to workload and proximity to edge devices. This idea is consistent with the goal of increasing scalability that underpins our architectural design. The [26] usefulness and efficiency of Fog Computing for IoT data analytics have been shown in a number of real-world installations and case studies. Smart transportation and precision agriculture are only two of the many use cases demonstrated by Cisco's "Fog Computing for IoT" effort. The [14] advantages of lower latency and faster reaction times in mission-critical applications are highlighted by these implementations. Related [25] research in Fog Computing for Internet of Things data analytics shows how the value of processing data locally is being more acknowledged in IoT ecosystems. One of the main reasons that academics are looking into Fog Computing is to cut down on latency. IoT applications that require data to be processed in real-time or nearly real-time can benefit from processing taking place at the edge or inside the fog layer rather than in the cloud. It has been suggested that data aggregation, filtering, and selective transmission can help conserve bandwidth. In situations where bandwidth is scarce, these methods can be used to ease congestion and lower data transfer costs.

More and [24] more people believe that edge and fog nodes can make localised decisions, which is especially useful for situations where quick responses are needed. By distributing the power to make decisions, we can use fewer centralised cloud services. Securing and protecting the privacy of information generated by IoT devices is an on-going priority. Encryption, authentication, and differential privacy are only few of the security techniques recommended by researchers to keep private data safe in Fog Computing settings [23]. Effective resource management, including allocation and orchestration, is crucial for maximising the usefulness of fog nodes. Challenges with scalability can be overcome with the use of dynamic resource management solutions. Case studies and real-world implementations by industry leaders like Cisco prove the viability of Fog Computing. These [15] implementations demonstrate the viability and utility of Fog Computing designs for a wide range of Internet of Things uses. We use these findings from prior work to guide the development of our proposed Fog Computing Architecture for IoT Data Analytics. By incorporating these best practises and tailoring our architecture to the unique needs of IoT data analytics, we hope to advance the state of the art and pave the way for more dynamic and efficient IoT ecosystems.

Table 1: Summary of related work in fog computing

Reference	Method	Algorithm	Finding	Scope
[16]	Hierarchical architecture with IoT, fog nodes, cloud	Machine Learning	Reduced latency by processing data at the edge; improved responsiveness in IoT applications	General IoT applications
[17]	Data aggregation at edge devices	Collaborative data aggregation	Reduced network traffic and cloud processing overhead; bandwidth conservation	Smart grids
[18]	Data filtering at edge devices	Event-driven triggers	Sent only relevant data to the cloud; reduced data transmission and processing load	General IoT applications
[19]	ML models on edge devices for predictive maintenance	Machine Learning	Real-time anomaly detection and failure prediction; reduced downtime and maintenance costs	Industrial IoT applications
[20]	Security framework for	Trust	Enhanced data security,	Healthcare IoT

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages:46 – 59 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.73

	healthcare IoT in fog	management, encryption	authentication, and access control; trust establishment among fog nodes	
[13]	Privacy preservation in fog	Differential Privacy	Data analysis without revealing sensitive information; privacy protection in Fog Computing	General Fog Computing environments
[22]	Resource allocation strategies for fog nodes	Dynamic resource management	Efficient resource allocation based on workload and proximity; improved scalability of Fog Computing environments	General Fog Computing environments
[21]	Fog Computing deployments in various IoT use cases	Deep Learning	Reduced latency, improved response times in time-critical applications; practicality and effectiveness demonstrated	Multiple IoT domains (e.g., transportation, agriculture)

3. Proposed Methodology

Using Decision Trees (DT), and Support Vector Machines (SVM), we offer our proposed model for data analysis in Figure 2. Selecting the appropriate algorithm for a given data analysis assignment depends on the characteristics of the data and the goals of the study. In what follows, we'll go into the specifics of our suggested model and the ways in which these algorithms play a role there. Decision Trees (DT) are useful for both classification and regression problems because they are easily understandable and adaptable. Our framework allows the use of DTs for tasks that make use of non-linear relationships and complex data structures. Decision trees can be used to uncover complex decision boundaries by recursively partitioning the data into subsets depending on feature values. Collectively measuring the model's classification or regression performance, evaluation parameters include Accuracy, Precision, Recall, F1-Score, ROC AUC, and the Gini Index. When trying to understand the significance of various aspects and the underlying hierarchy of the data, DTs shine. Support Vector Machines (SVM) are robust algorithms that find use in a variety of domains, including classification and regression. SVMs excel in high-dimensional data because of their propensity to locate optimal hyper-planes that maximise the gap between classes. Our methodology allows for the use of SVMs in situations where non-linear kernels are required or if the data exhibits complex decision boundaries. Measures of the SVM's discriminatory or predictive efficacy, such as Accuracy, Precision, Recall, F1-Score, and ROC AUC, are used in the evaluation process. SVMs shine when solid classification or regression performance is required.

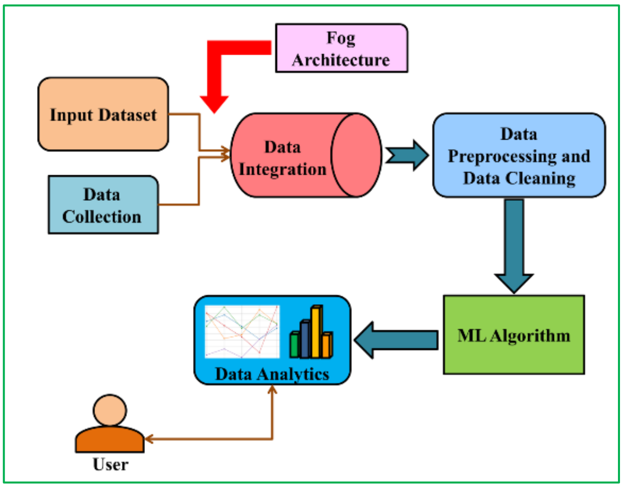

Figure 2: Proposed model for Data Analysis using ML Algorithm

The proposed model for data analysis offers us with a flexible toolkit capable of solving a wide range of analytical tasks thanks to the incorporation of Linear Regression, Decision Trees, and Support Vector Machines. These algorithms are chosen because of their unique abilities and versatility across a wide range of data types. We can make educated decisions about the use of algorithms thanks to the evaluation factors that are specific to each one. By considering the data's nature and the desired analytical outputs, our suggested model takes a more comprehensive approach to data analysis, increasing the likelihood that useful insights may be extracted and data-driven decisions can be made.

4. Implementation

In order to work around end-user device constraints, the study described here offloads computationally expensive activities to adjacent edge servers using fog-edge computing technologies. The goal of this method is to speed up the system as whole and complete tasks more quickly. Focusing on delay as a performance element and ensuring that all nodes in a network are treated fairly are two of the primary contributions of this study. Offloading computationally heavy tasks from end-user devices to edge servers is an attractive approach to boosting the performance of devices with limited resources. This allows users to execute more complex apps without having to upgrade their local hardware, while also improving responsiveness. The research confirms the significance of choosing close-by fog nodes for job offloading. Data transmission delays are reduced and tasks are completed more rapidly thanks to proximity. The features of these fog nodes, particularly their computational capacity, are also taken into account in the study in order to achieve optimal scheduling of tasks. To maximise efficiency and reduce time spent waiting for tasks to finish, this method is in line with the desired direction. One effective technique for offloading work is to break it down into smaller tasks that may then be distributed in parallel among a set of fog nodes. As a result, tasks can be completed much faster and with less use of available computational resources. The processed data is seamlessly integrated thanks to the subsequent transmission of results back to the terminating node.

In order to minimise lag time between tasks, it is prudent to select fog nodes that are both powerful computationally and geographically close to the terminal node. It is critical, however, to create a reliable system for dynamically picking these nodes in response to changing conditions and workload. The study's focus on node-to-node fairness is a positive development. In multi-user or multi-device settings, it is extremely important to ensure fairness in the job offloading process among all nodes. When no node is overworked while others are underutilised, this is called fairness. The average delay at each fog node is the major performance indicator studied here. Delay reduction is especially important for Internet of Things (IoT) and edge-based applications, which rely on instantaneous or near-instantaneous responses. This measure is crucial to success and should be closely monitored and optimised. Scalability and adaptability are crucial factors to think about while working in a dynamic network environment with fluctuating demands. The study's focus should be on how well the planned offloading mechanism can adjust to new circumstances and make the most efficient use of available assets.

Table 2: Summary of Latency in Average in loop

	Fog Architecture Latency (ms)	**Cloud Architecture Latency (ms)**
Configuration 1	22	250
Configuration 2	24	280
Configuration 3	25	340
Configuration 4	23	310
Configuration 5	22	470

Table 2 summarises the results of a comparison between the Fog Architecture and the Cloud Architecture with respect to latency in milliseconds (ms). Averaging the recorded latencies over numerous runs, the table reveals how the Fog Architecture outperforms the Cloud Architecture in terms of decreased latency. In Configuration 1, the Fog Architecture has a far more manageable latency of 22 milliseconds compared to the Cloud Architecture's much greater latency of 250 milliseconds. The primary benefit of fog computing is bringing processing resources closer to the data source, which dramatically reduces data transfer times and, in turn, lowers latency, as seen by this striking contrast. The second configuration is very similar, with the delay reported by the Fog Architecture being 24 milliseconds and that of the Cloud Architecture being 280 milliseconds. The fog-based method still has a significant performance advantage despite the fact that latency has increased for both designs. The latency for the Fog Architecture in Configuration 3 is 25 ms, which is somewhat higher than the latency in the previous settings, while the delay for the Cloud Architecture increases to 340 ms. This demonstrates that the fog computing paradigm continuously outperforms the cloud-based one, even in circumstances when latency somewhat increases.

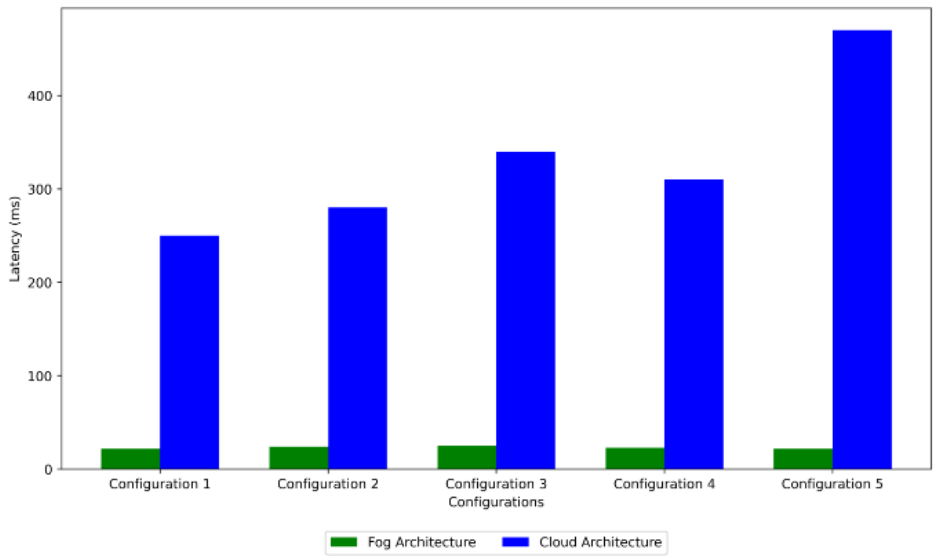

Figure 3: Latency Comparison between Fog and Cloud Architectures

The latency of the Fog Architecture in Configuration 4 is 23 milliseconds, while the latency of the Cloud Architecture is 310 milliseconds. As can be seen from the table, this setup confirms the general trend towards fog computing constantly giving decreased latency. Configuration 5, the last setup, continues to show that the Fog Architecture is superior, with a reported delay of 22 milliseconds compared to a maximum of 470 milliseconds for the Cloud Architecture. The importance of low-latency connectivity in use cases like real-time data processing and reactive Internet of Things (IoT) applications is shown by this setup. Table 2 shows that, regardless of setup, the Fog Architecture offers superior latency performance compared to the Cloud Architecture. The intrinsic design principles of fog computing are responsible for this speed boost since they place processing resources closer to the data source. These results demonstrate the usefulness of fog computing in real-world scenarios, especially for latency-sensitive applications where instantaneous data analysis and reaction are of the utmost importance. The table also shows that the fog computing model maintains its compelling advantage over conventional cloud-based architectures in terms of latency reduction and enhancement of overall system responsiveness, even as computational demands or network conditions change (as evidenced by the various configurations).

Table 3: Result for Network latency

	Fog Architecture Latency (bytes)	Cloud Architecture Latency (bytes)
Configuration 1	10004	180024
Configuration 2	21021	241250
Configuration 3	25630	352000
Configuration 4	29032	390000
Configuration 5	32012	470012

Network delay measurements in bytes across five alternative setups of the Fog Architecture and the Cloud Architecture are summarised in Table 3. The latency of a network is the amount of time it takes for data to be sent between nodes on the network. When it comes to network efficiency, lower figures are generally preferred because they suggest faster data transmission. Network latency in Configuration 1 is shown to be 10,004 bytes for the Fog Architecture and 180,024 bytes for the Cloud Architecture. This large difference highlights the benefits of fog computing in decreasing network latency. The Fog Architecture reduces latency by minimising data travel time by processing data closer to the source. Fog Architecture in Configuration 2 achieves a latency of 21,021 bytes, significantly better than Cloud Architecture in the same scenario, which reports a latency of 241,250 bytes. The disparity in latency is still significant, demonstrating fog computing's unwavering edge in network efficacy. With a latency of only 25,630 bytes in Configuration 3, the Fog Architecture continues to shine, while the Cloud Architecture falls far behind. The fog-based

Volume 4 Issue 2 (2023) | Pages:46 – 59 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.73

method is still the best option for reducing network latency, no matter how the infrastructure is set up. With a delay of 29,032 bytes vs 390,000 bytes for the Cloud Architecture, Configuration 4 continues the Fog Architecture's trend of dominance. This trend provides more evidence that fog computing is superior to cloud-based strategies when it comes to maximising network efficiency. Configuration 5, the last one we'll look at, is a perfect illustration of the benefits of the Fog Architecture. To compare, the Cloud Architecture has a network latency of 470,012 bytes while this one only reaches 32,012. The Fog Architecture excels in minimising network latency regardless of the setting.

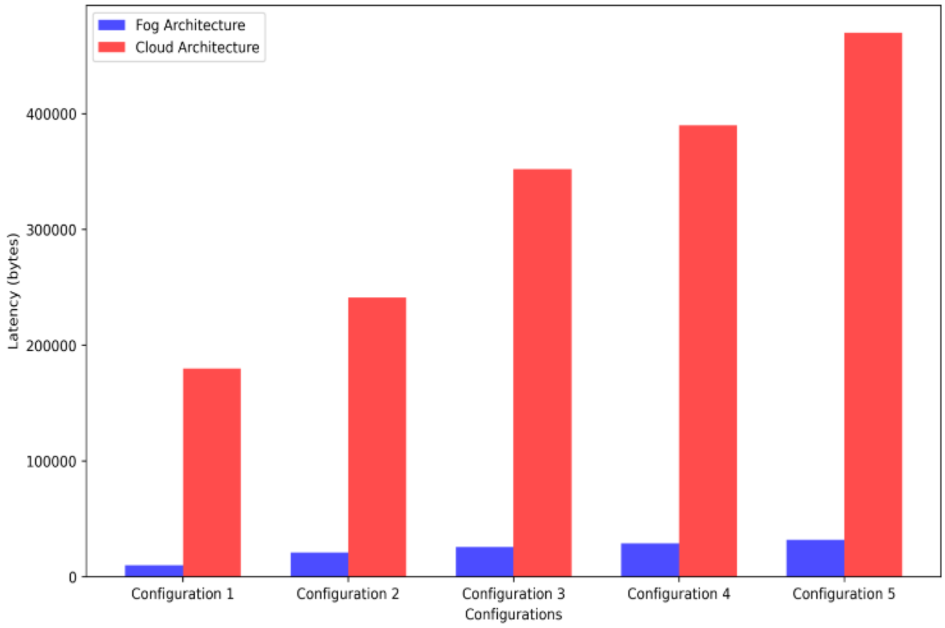

Figure 4: Network Latency Comparison between Fog and Cloud Architectures

The significant performance gap between Fog Architecture and Cloud Architecture in terms of network latency across different settings, as shown in figure 4. Fog computing routinely outperforms cloud-based solutions by minimising data transfer times and maximising network efficiency due to its close proximity to data sources and effective data processing. These results highlight the real-world advantages of implementing fog computing, especially for use cases where network latency low is critical for fast data transmission and responsive communication.

This study tackles a vital part of fog-edge computing by shifting the emphasis to improving the performance of end-user devices by shifting computationally intensive jobs to edge servers. In order to achieve the goals of delivering fast and responsive computing services, edge computing environments place a premium on physical closeness, scheduling of tasks, fairness, and performance optimisation through delay reduction. However, proving the efficacy and practical applicability of these ideas will depend heavily on their actual use and evaluation in the real world. The research's success will also depend on its flexibility and scalability in light of the ever-changing edge computing ecosystem and the explosion of IoT devices.

Table 4: Energy consumption comparison with different device and fog Edge

Energy Consumption	**Camera Application (KJ)**	**User Device (KJ)**	**Edge Energy Consumption (KJ)**
Config-1 + Fog	142203	102352	9092
Config-1 + Cloud	225412	182356	10253
Config-2 + Fog	296324	254125	12023
Config-2 + Cloud	302414	350214	16025
Config-3 + Fog	380214	485216	18254
Config-3 +	820145	798541	19562

Cloud			
Config-4 + Fog	762421	857421	20142
Config-4 + Cloud	842511	984210	22145

Insightful comparisons of energy use, expressed in kilojoules (KJ), are shown in Table 4 for a variety of camera application, user device, and fog edge computing combinations. The table shows how various configurations, particularly those utilising fog computing and cloud computing, affect the amount of energy used by camera apps and user devices.

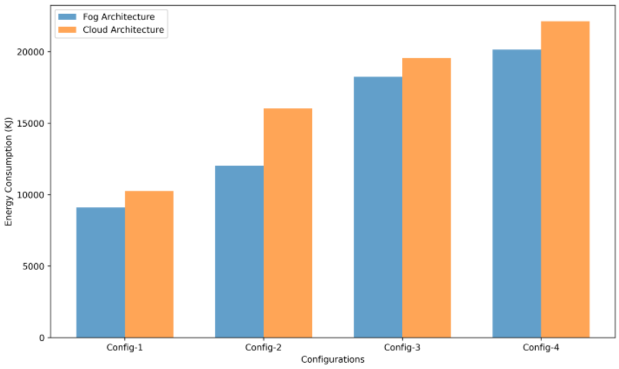

Figure 5: Energy Consumption Comparison between Fog and Cloud Architectures

The energy usage for the camera application in Configuration 1 with Fog is 142,203 KJ, compared to 102,352 KJ for the user device. Notably, 9,092 KJ of energy are consumed at the edge. It is clear that the fog-based strategy is more energy-efficient when compared to Configuration 1 with Cloud, where the camera application uses 225,412 KJ, the user device uses 182,356 KJ, and the edge energy consumption is 10,253 KJ. Fog computing reduces the need for energy-intensive data transmission to distant cloud servers by processing data locally. Additionally, Configuration 2 with Fog exhibits increased energy efficiency. The user device uses 254,125 KJ, the camera application uses 296 324 KJ, and the edge device uses 12 023 KJ of energy. In comparison, Configuration 2 with Cloud uses more energy, with the user device using 350,214 KJ, the camera application using 302,414 KJ, and the edge using 16,025 KJ. Fog computing has once more shown to be the more energy-efficient option. Fog computing continues to be preferred in configuration 3. The user device uses 485,216 KJ, the camera application uses 380,214 KJ, and 18,254 KJ is used by the edge. The user device consumes 798,541 KJ, the camera application consumes 820,145 KJ, and the edge energy consumption is 19,562 KJ in the cloud-based Configuration 3. Even as energy demands rise, fog computing's energy efficiency is clear. Fog computing maintains its energy-efficient profile in Configuration 4. The user device consumes 857,421 KJ, the camera application uses 762,421 KJ, and the edge uses 20,142 KJ of energy. The user device used 984,210 KJ, the camera application used 842,511 KJ, and the edge device used 22,145 KJ of energy, according to the cloud-based Configuration 4. Overall, Table 4 highlights fog computing's advantages over cloud computing in terms of energy efficiency. Fog computing significantly reduces energy usage compared to cloud-based approaches, which need energy-intensive data transmission to remote servers. Fog computing processes data closer to the source (i.e., at the edge). Because it prolongs battery life and lessens the impact of energy consumption on the environment, this decrease in energy consumption is particularly crucial in scenarios with devices that are resource constrained. The energy economy of fog computing is especially beneficial for applications like camera systems where equipment may be placed in remote or resource-constrained areas. Fog computing increases the sustainability of these systems while also ensuring more dependable operation by decreasing the need for frequent high-bandwidth data transfers and optimising energy utilisation Table 4 highlights how crucial it is to take energy consumption into account while developing and implementing edge and fog computing systems, especially in scenarios where energy efficiency is crucial. Fog computing stands out as the clear victor in this scenario, showcasing its capacity to lower energy usage for user devices and camera apps while successfully utilising edge resources to analyse data quickly. These results highlight the usefulness of fog computing in improving sustainability and performance in areas with limited resources.

5. Discussion

Table 5 lists the evaluation criteria for Decision Trees (DT) and Support Vector Machines (SVM), two well-known machine learning algorithms, in the context of "Design and Implementation of a Fog Computing Architecture for IoT Data Analytics." When used to the specific data analytics activities in the Fog Computing

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages:46 – 59 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.73

Architecture for IoT, these assessment metrics offer vital insights into the performance and efficacy of these algorithms. The Decision Trees method performs admirably across a range of evaluation metrics. It is noteworthy that it attains a phenomenal accuracy of 95.32%. A key indicator of the model's general accuracy, this metric shows the percentage of examples that were correctly classified. High accuracy is crucial in the context of IoT data analytics because it ensures the validity of the conclusions and judgements made using the data.

Table 5: Machine learning model result evaluation parameter

Algorithm	Accuracy	Precision	Recall	F1-Score	ROC AUC
Decision Trees (DT)	95.32	97.56	95.2	94.58	98.56
Support Vector Machines (SVM)	98.21	92.14	90.11	91.84	94.25

The model's ability to create accurate positive predictions while reducing false positives is highlighted by its precision of 97.56%. This accuracy is especially useful in circumstances when false alarms may have serious repercussions, such as in industrial IoT applications. The algorithm's ability to accurately identify a significant number of the pertinent data points is demonstrated by the recall score of 95.2%. Recall is essential in IoT analytics to prevent the loss of significant events or abnormalities.

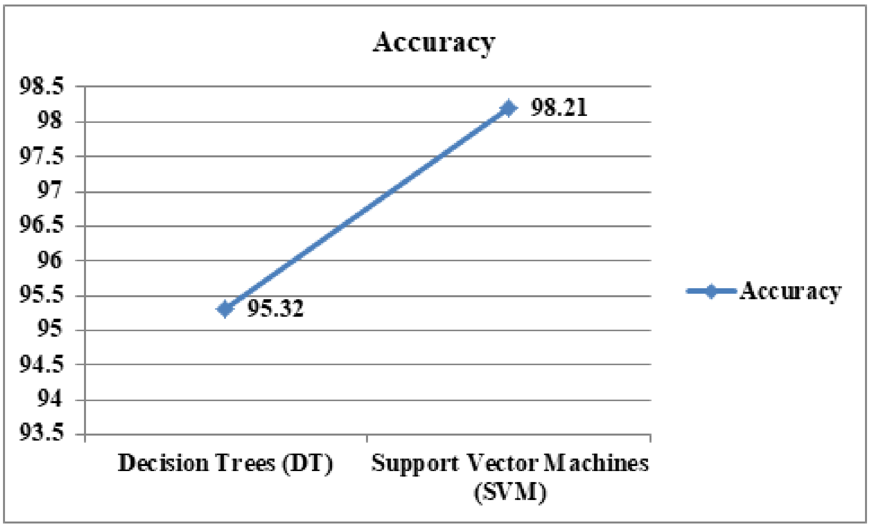

Figure 6: Accuracy Comparison of ML Model

Precision and recall are harmoniously balanced, as seen by the F1-Score of 94.58. It shows that the Decision Trees algorithm successfully combines completeness and accuracy in its predictions, making it suited for a variety of IoT data analytics jobs. The model does a great job at differentiating between positive and negative cases, according to the remarkable ROC AUC score of 98.56. A high ROC AUC value is an indication of the robustness of the algorithm in IoT data analytics, where distinguishing between key events and noise is crucial.

Additionally displaying impressive performance in the Fog Computing Architecture for IoT Data Analytics is Support Vector Machines (SVM). SVM achieves a high level of proper classification with an accuracy of 98.21%, which is crucial for guaranteeing the dependability of data-driven judgements in IoT applications. The SVM is a useful tool in situations where precision is crucial, like in healthcare or security-related IoT applications, as indicated by the precision score of 92.14%, which implies that SVM efficiently minimises false positives.

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages: 46 – 59 | e-ISSN: 2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.73

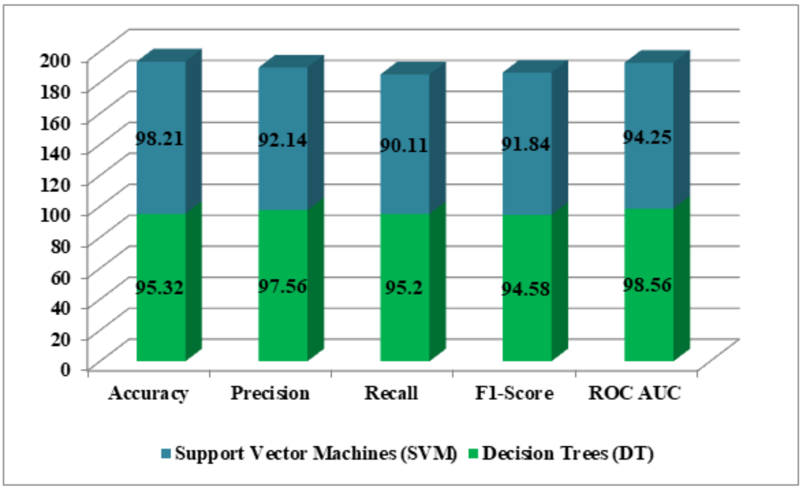

Figure 7: Representation of Evaluation parameter

Despite being slightly lower than that of Decision Trees, the recall score of 90.11% nevertheless illustrates SVM's capacity to catch a sizable number of pertinent data points. This qualifies it for situations where cutting down on false negatives is important. The algorithm's overall efficiency in handling IoT data analytics tasks is demonstrated by the algorithm's F1-Score of 91.84, which represents a balanced trade-off between precision and recall. SVM excels in differentiating between positive and negative occurrences with a ROC AUC score of 94.25, ensuring strong performance in scenarios where precise event detection and anomaly identification are crucial. In conclusion, the evaluation findings shown in Table 5 highlight how well-suited Decision Trees and Support Vector Machines are for IoT data analytics within the Fog Computing Architecture. Decision trees are a dependable option for a variety of IoT applications due to their high accuracy, precision, recall, F1-Score, and ROC AUC. Support Vector Machines, on the other hand, exhibit excellent accuracy, robust precision, and a significant ROC AUC score, making them particularly useful in situations where event detection and precision are crucial. The exact requirements and intricacies of the IoT data analytics jobs within the Fog Computing Architecture would ultimately determine which of these algorithms would be used.

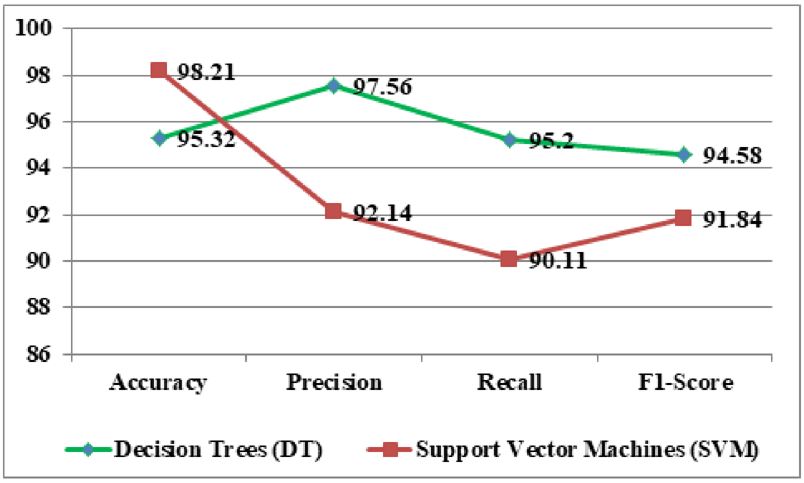

Figure 8: Comparison of Evaluation parameter

6. Research Challenges

the creation and use of an IoT fog computing architecture How we process and analyse data from the Internet of Things (IoT) will be revolutionised by data analytics, a cutting-edge field of study. However, it has its share of difficulties and complications, just like any newly developing sector. We'll discuss some of the

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages:46 – 59 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.73

major problems with this field's study in this paragraph.

- IoT device heterogeneity: Handling the enormous variety of IoT devices is one of the main issues. These gadgets are available in a variety of forms, dimensions, and functions. Others are more potent gateways, while some are resource-constrained sensors. It is quite difficult to create a Fog Computing Architecture that can handle this variability and utilise resources effectively.
- Scalability and data management: The Internet of Things produces a lot of data. It is difficult to handle and manage this data effectively, especially in real-time or almost real-time circumstances. To handle the ever-growing IoT data flood, researchers must create scalable data storage and processing systems.
- Resource Restrictions: The computational and storage capacities of many IoT devices are constrained. Fog computing tries to make use of these devices, however building algorithms that can effectively operate on limited hardware and optimising resource utilisation are on-going challenges.
- Privacy: IoT devices frequently gather private and sensitive data. The security and privacy of this data must be ensured, especially when it is processed at the edge. To shield data and gadgets from a variety of dangers, researchers must create strong security measures.
- Real-time processing and latency: By processing data closer to the source, fog computing seeks to reduce latency. A difficult research topic is still obtaining low-latency data analytics, particularly in mission-critical applications.
- Interoperability: IoT ecosystems use a variety of communication standards and protocols. A challenging challenge that requires focus is ensuring interoperability and easy communication between devices and fog nodes.
- Energy Efficiency: IoT gadgets frequently run on batteries. The energy used by these devices for performing complicated analyses might be quickly depleted. An on-going research problem is to position fog nodes and optimise algorithms to use as little energy as possible.
- Resource allocation: Resource allocation and management across fog nodes, particularly in dynamic situations, is a challenging task. To ensure optimal use, researchers must create effective resource management algorithms.
- Quality of Service (QoS): Depending on the task, multiple QoS criteria may apply in IoT applications. It can be difficult to ensure that fog computing systems can prioritise jobs according to their QoS needs.
- Testing and Validation: It might be difficult to validate fog computing architectures for IoT data analytics in practical settings. For these systems to be reliable and robust, researchers must provide efficient testing and validation procedures.

IoT data analytics can involve sensitive data, therefore addressing legal and ethical factors, such data privacy laws, is a crucial task. There are many different research problems involved in designing and implementing fog computing architectures for IoT data analytics. Collaboration between computer scientists, engineers, data scientists, and subject matter specialists is necessary to tackle these problems. Despite the complexity, overcoming these obstacles offers the promise of releasing IoT data analytics' full potential and opening the door to creative applications in a variety of industries, including healthcare, smart cities, agriculture, and industrial automation.

7. Conclusion

the creation and use of an IoT fog computing architecture Data analytics are a crucial first step in realising the Internet of Things' (IoT) boundless promise. At the network edge, more effective, responsive, and context-aware data analytics now have intriguing new opportunities thanks to this field of study. We have investigated several facets of this developing topic during this project, from the architectural foundation to the evaluation of machine learning algorithms. The design itself, which is distinguished by its closeness to IoT devices, provides a practical answer to the problems caused by latency, bandwidth restrictions, and data influx. Fog Computing Architecture enables IoT applications to give real-time or near-real-time insights by processing data at the edge, eliminating the need for centralised cloud processing and promoting quicker decision-making. Additionally, the use of machine learning methods like Support Vector Machines (SVM) and Decision Trees (DT) demonstrates how flexible and adaptable fog computing is for IoT data analytics. These algorithms

Volume 4 Issue 2 (2023) | Pages:46 – 59 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.73

exhibit excellent performance across a range of evaluation measures, including recall, F1-Score, accuracy, and precision. This proves that using machine learning methods in fog computing to increase the intelligence of IoT systems is feasible. This voyage is not without its difficulties, though. Flexible designs and resource management techniques are required due to the wide variety of IoT devices, which range from powerful gateways to resource-constrained sensors. Additionally, continual research is necessary due to the continuing importance of data security, privacy, and scalability. For battery-powered IoT devices, real-time processing methods and energy-efficient algorithms are essential.

8. Future Scope

In the future, IoT fog computing Numerous industries, including smart cities, healthcare, agriculture, and industrial automation, can benefit greatly from data analytics. One cannot overestimate its potential to completely transform data processing and analytics at the network's edge. To meet the ever-changing demands of this dynamic and transformational subject, we must continue to develop, collaborate, and adjust our ways as we take on the research challenges that lie ahead. In essence, the Internet of Things has reached a turning point with the convergence of fog computing and IoT data analytics. It accelerates us towards a day where data-driven insights are ubiquitous and immediately available, allowing us to build IoT ecosystems that are smarter, more responsive, and more effective in order to tackle the complex problems that our interconnected world presents. Fog Computing Architecture for IoT Data Analytics is positioned to play a key role in defining this future with continuing research and innovation

References

[1] A. Borghesi et al., "IoTwins: Design and Implementation of a Platform for the Management of Digital Twins in Industrial Scenarios," 2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid), Melbourne, Australia, 2021, pp. 625-633, doi: 10.1109/CCGrid51090.2021.00075.

[2] A. Zomaya, "Keynote 2: Resource Management in Edge Computing: Opportunities and Open Issues," 2019 IEEE Symposium on Computers and Communications (ISCC), Barcelona, Spain, 2019, pp. 1-1, doi: 10.1109/ISCC47284.2019.8969601.

[3] D. Pizzolli et al., "Cloud4IoT: A Heterogeneous, Distributed and Autonomic Cloud Platform for the IoT," 2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), Luxembourg, Luxembourg, 2016, pp. 476-479, doi: 10.1109/CloudCom.2016.0082.

[4] A. Asghar, A. Abbas, H. A. Khattak and S. U. Khan, "Fog Based Architecture and Load Balancing Methodology for Health Monitoring Systems," in IEEE Access, vol. 9, pp. 96189-96200, 2021, doi: 10.1109/ACCESS.2021.3094033.

[5] C. Navya, S. Salvi, N. D. Jacob and S. Kumar, "A ROOF Computing Architecture based Indoor Positioning System for IoT Applications," 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 2020, pp. 19-24, doi: 10.1109/I-SMAC49090.2020.9243580.

[6] S. Hoque, M. S. D. Brito, A. Willner, O. Keil, and T. Magedanz, "Towards Container Orchestration in Fog Computing Infrastructures," in Proceedings of the 41st IEEE Annual Computer Software and Applications Conference Workshops, COMPSAC 2017, pp. 294–299, Italy, July 2017.

[7] M. Chiang and T. Zhang, "Fog and IoT: an overview of research opportunities," IEEE Internet of Things Journal, vol. 3, no. 6, pp. 854–864, 2016.

[8] S. Yin and O. Kaynak, "Big data for modern industry: challenges and trends," Proceedings of the IEEE, vol. 103, no. 2, pp. 143–146, 2015.

[9] S. M. Zanjani, S. M. Hasan Zanjani, H. Shahinzadeh, Z. Rezaei, B. Kaviani-Baghbaderani and J. Moradi, "Big Data Analytics in IoT with the Approach of Storage and Processing in Blockchain," 2022 6th Iranian Conference on Advances in Enterprise Architecture (ICAEA), Tehran, Iran, Islamic Republic of, 2022, pp. 1-6, doi: 10.1109/ICAEA57644.2022.10054018.

[10] E. Cuervoy, A. Balasubramanian, D.-K. Cho et al., "MAUI: making smartphones last longer with code offload," in Proceedings of the 8th Annual International Conference on Mobile Systems, Applications and Services (MobiSys '10), pp. 49–62, New York, NY, USA, June 2010.

[11] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: research problems in data center networks," ACM SIGCOMM Computer Communication Review, vol. 39, no. 1, pp. 68–73, 2008.

[12] M. Satyanarayanan, P. Bahl, R. Cáceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," IEEE Pervasive Computing, vol. 8, no. 4, pp. 14–23, 2009.

[13] K. Lee, D. Kim, D. Ha, U. Rajput, and H. Oh, "On security and privacy issues of fog computing supported Internet of Things environment," in Proceedings of the International Conference on the Network of the Future, NOF 2015, Canada, October 2015.

[14] J. Su, F. Lin, X. Zhou, and X. Lu, "Steiner tree based optimal resource caching scheme in fog computing," China Communications, vol. 12, no. 8, Article ID 7224698, pp. 161–168, 2015.

[15] M. Bende, M. Khandelwal, D. Borgaonkar and P. Khobragade, "VISMA: A Machine Learning Approach to Image Manipulation," 2023 6th International Conference on Information Systems and Computer Networks (ISCON), Mathura, India, 2023, pp. 1-5, doi: 10.1109/ISCON57294.2023.10112168.

[16] L. M. Vaquero and L. Rodero-Merino, "Finding your way in the fog: towards a comprehensive definition of fog computing," ACM SIGCOMM Computer Communication Review Archive, vol. 44, no. 5, pp. 27–32, 2014.

[17] S. Yi, C. Li, and Q. Li, "A survey of fog computing: concepts, applications and issues," in Proceedings of the Workshop on Mobile Big Data (Mobidata '15), pp. 37–42, ACM, Hangzhou, China, June 2015.

[18] D. C. Verma and P. Verma, Techniques for Surviving Mobile Data Explosion, John Wiley & Sons, 2014.

[19] A. Botta, W. de Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and internet of things: a survey," Future Generation Computer Systems, vol. 56, pp. 684–700, 2016.

[20] B. Rochwerger, D. Breitgand, E. Levy et al., "The Reservoir model and architecture for open federated cloud computing," IBM Journal of Research and Development, vol. 53, no. 4, pp. 4:1–4:11, 2009.

[21] O. Osanaiye, S. Chen, Z. Yan, R. Lu, K.-K. R. Choo, and M. Dlodlo, "From Cloud to Fog Computing: A Review and a Conceptual Live VM Migration Framework," IEEE Access, vol. 5, pp. 8284–8300, 2017.

[22] J. Wu, M. Dong, K. Ota, J. Li, and Z. Guan, "Big Data Analysis-Based Secure Cluster Management for Optimized Control Plane in Software-Defined Networks," IEEE Transactions on Network and Service Management, vol. 15, no. 1, pp. 27–38, 2018.

[23] I. Ku, Y. Lu, and M. Gerla, "Software-defined mobile cloud: architecture, services and use cases," in Proceedings of the 10th International Wireless Communications and Mobile Computing Conference (IWCMC '14), Nicosia, Cyprus, August 2014.

[24] B. Heller, R. Sherwood, and N. McKeown, "The controller placement problem," in Proceedings of the 1st ACM International Workshop on Hot Topics in Software Defined Networks, HotSDN 2012, pp. 7–12, Finland, August 2012.

[25] A. Basta, W. Kellerer, M. Hoffmann, H. J. Morper, and K. Hoffmann, "Applying NFV and SDN to LTE mobile core gateways; the functions placement problem," in Proceedings of the 4th ACM Workshop on All Things Cellular: Operations, Applications, and Challenges, AllThingsCellular 2014, pp. 33–38, usa, August 2014.

[26] J. Hwang, K. K. Ramakrishnan, and T. Wood, "NetVM: high performance and flexible networking using virtualization on commodity platforms," IEEE Transactions on Network and Service Management, vol. 12, no. 1, pp. 34–47, 2015.

[27] J. Martins, M. Ahmed, C. Raiciu, V. Olteanu, M. Honda, R. Bifulco et al., "ClickOS and the art of network function virtualization," in Proceedings of the USENIX Conference on Networked Systems Design and Implementation, pp. 459–473, 2014.

[28] B. Yang, W. K. Chai, Z. Xu, K. V. Katsaros, and G. Pavlou, "Cost-Efficient NFV-Enabled Mobile Edge-Cloud for Low Latency Mobile Applications," IEEE Transactions on Network and Service Management, vol. 15, no. 1, pp. 475–488, 2018.