Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Cloud Server Data Optimization By Recurrence And Redundancy Reduction: A Comprehensive Review

Mr. Manjunath Singh. H[1] , Dr. Tanuja R[2]

[1]Assistant Professor, Department of Computer Science and Engineering, Uvce, Bangalore University, India
Email.id: mansh24.singh@gmail.com , mansh.singh@gmail.com
[2]Associate Professor, Department of Computer Science and Engineering, Uvce, Bangalore University, India
Email.id: tanujar.uvce@gmail.com.
Corresponding author: [1]Mr. Manjunath Singh. H, Assistant Professor, Department of Computer Science and Engineering, Uvce, Bangalore University, India
Email.id: mansh24.singh@gmail.com , mansh.singh@gmail.com

Cloud computing and the development of greater capacity data storage units present a big challenge to researchers today. The archiving of data and resources cannot be solved by increasing the resources relative to the data, nor can it be solved by expanding the resources. The data in a cloud computing environment is stored recursively and replicated multiple times. For data-intensive applications, reducing redundant information and reoccurrences is the key goal in the hunt for increased performance.

For example, concerns about data availability, dependability, security, bandwidth and the reaction time of data access will be addressed in order to meet this requirement. In spite of the widespread impacts and established status of cloud data replication, the author is unaware of any systematic, all-encompassing, or conclusive survey on the topic. With a view to better understanding existing research into IoT-based cloud computing storage systems, a comprehensive survey was conducted out to look for trends in redundancy reduction_ In addition, potential future directions taken by the research are discussed_

Keywords: Cloud storage, data redundancy, data reoccurrences, cloud data server optimization.

1. Introduction

Computing in the cloud is a blessing for modern computing and scientific data storage because it allows for larger amounts of data to be stored. It has been reported that there has been a significant increase in the amount of data that has been transferred and lied through the use of cloud computing infrastructure in the past decade_ Third-party service providers are responsible for the great bulk of data, according to a report issued by ICT [1]. The space utility [2,3] at Google, that is used by major services like Email, Google Drives, as well as File Share, has resulted in the limitation of the bandwidth available for access to free storage. However, despite the restrictions, the number of people signing up to create new user profiles has risen significantly. It is common for storage service providers to develop space optimization methods

in order to increase the complexity of storage devices and buffered spaces. The current trend in research [5] indicates that there is an increased demand for the design of an optimization algorithm for space and memory_ It is concerning to see how quickly storage warehouses are expanding their capacities. The warm temperatures of servers and warehouses are having an effect on the surrounding environment and contributing to the phenomenon of global warming [6]_ Figure—1

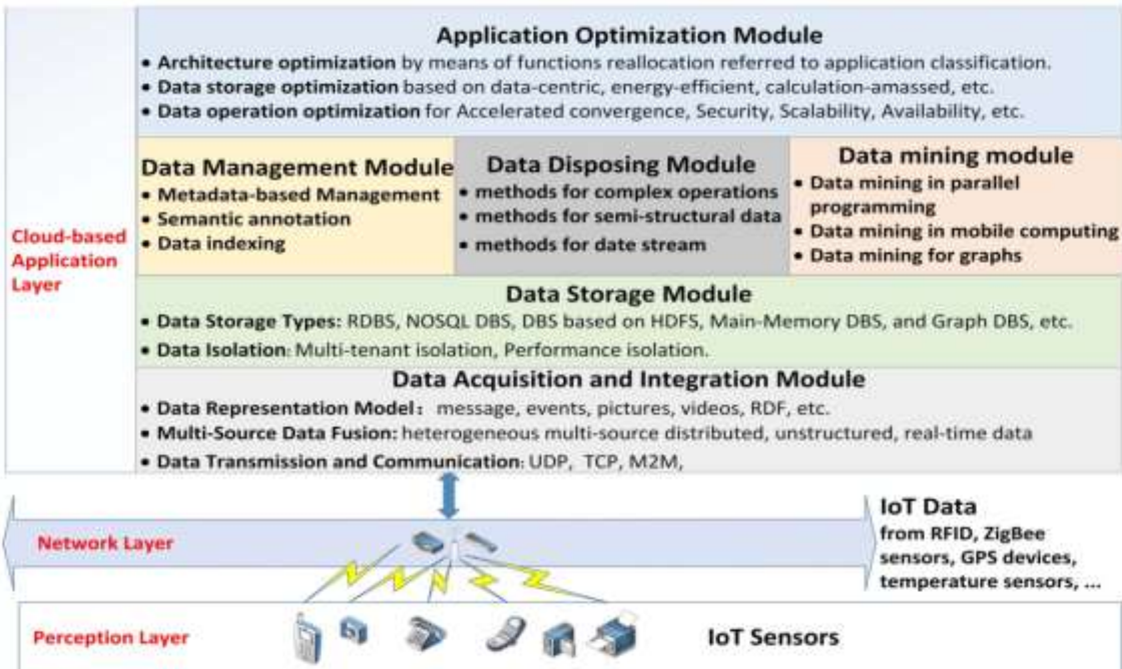


Figure 1: IoT Based Cloud Server System

A typical Internet of Things application framework consists of the perception layer, the network layer, and the application layer [2]. The application layer is vital to cloud-based loT storage systems since it consists of middlewares and business models. Cloud-based application layers have received a lot of attent1 for their abi!ity to process and analyse data quickly and effectively. This layer includes RFID (Radio Frequency Identification), wireless sensor networks (WSN), as well as other intelligent devices. Figure 1 depicts a cloud computing infrairture for 10T-based data storage_ Iot applications are the inspiration for this framework. Data acquisition and integration module, data storage module, database management module, data mining module, and application optimization module re all part of a framework that is comprised of a number of different modules. As shown in Fig. 1, associated technologies can be divided into a variety of functional modules, which will be discussed in more detail below.

• Module for Data Acquisition and Integration.- The system's overall structure faces a basic issue while trying to collect and combine heterogeneous data from scattered and mobile devices as an input module.

• Data Storage Module.- There are many different types of databases and file systems, alluding XML files in HDFS, RDBMS, and no SQL, when it comes to IoT data formats, sit as structured, semi-structured, and unstructured data in large quantities (NoSQL). A NoSQL database provides a framework for storing and retrieving data that is not structured using thiibular relationships seen in traditional relational databases_ The most efficient way to store data in the cloud is to use a combination of graph database management systems. • Data Management Module.- Diverse techniques to data management are realised on various platforms in order to find and retrieve data from a large amounts of data sources efficiently. Semantic relations and connected data are a few examples.

• Module far Data Processing! Mass data processing technologies like Map Reduce can be used to build parallel and decentralized data processing on the cloud_ Data searching and reasoning can be performed in a more flexible manner to better accommodate massive amounts of data.

• Data Mining Module.- Before being used for any desired application, high-level information must first be extracted, categorised, abstracted, and analysed. Considering that the data acquired from sensors always are raw and low-level data, this process is essential_ Therefore, 1viding end users with the results of extensive data analysis should be the major goal of the data mining process when applied to data from the Internet of Things.

• Application Optimization Module.- According to the analysis of the application, related algorithms or methods are necessary for analyzing IoT data in the cloud platforms that provide varying levels of required performance. These functional requirement include a decrease in 1/0 operations, an acceleration of converging, security, scalability, reliability, management, cost and price reductions, and other improvements_

Managing complex operations that require large quantities of memory is inefficient for IoT devices because of their limited computational, processing, and memory capabilities. However, despite the fact that virtualization in the cloud allows lot applications to access space, computing, and networking resources, certain end-user mobile applications still require low latency [84] in able to reach their stored personal information with geo-distribution, user mobility, and location-based information [85].

There is an urgent need to shift data storage closer to the user since loT and the cloud carat meet these requirements and because cloud services get their own limits. In this context, fog computing emerges as a useful extension of cloud computing, bringing cl services closer to the location where they are actually needed by users.

As can be seen in Figure 2,the Fog computing architecture is comprised of three primary layers of processing: the Cloud layer, the Fog layer.d the Edge layer. The fog layer is responsible for providing temporary storage to end devic e it ated at the network edge, while the cloud layer is in charge of providing permanent storage. Because the data is stored in close proximity to the end user, this architectua also gives users the ability to exercise control over their own data.

This, in turn, protects the confidentiality, integrity, and authenticity of user data from being modified or retrieved in an unauthorised iranner. The ability of the Fog layer to store data miles it possible to offload computational services to low-powered end devices. As a result, the Fog computing paradigm with its transient storage service near the edge of the network [94] is utilised in order to solve the issues that are associated with the storage of the user's data.

The ability of cloud computing to store data and perform computations has resulted in the modif1tion of various schema in the field of computer science [95,96]. But as a result of the rapid development of IoT technology, the number of services provided by IoT devices has multiplied many times over. These services consist of, among other things, smart homes [97], smart grids [98,99], smart cities [100,101]*,smart vehicles [102], edge routers [103], and so on_ The widespread application of these Internet of Things devices generates a massive amount of data, which is then sent to the cloud to be lived and processed. When this data is it used in latency-sensitive applications, it requires quick storage and processing, which presents a challenge for cloud-based storage and processing systems. Even tough cloud computing is able to store data on its large data centres that are located all over the world, without the Fog computing paradigm, which didn't come into existence until 2012 [97], There are issues with limited storage capacity, latency, limited bandwidth, computation overhead, which ultimately affects customer satisfaction with cloud service quality [98-100].

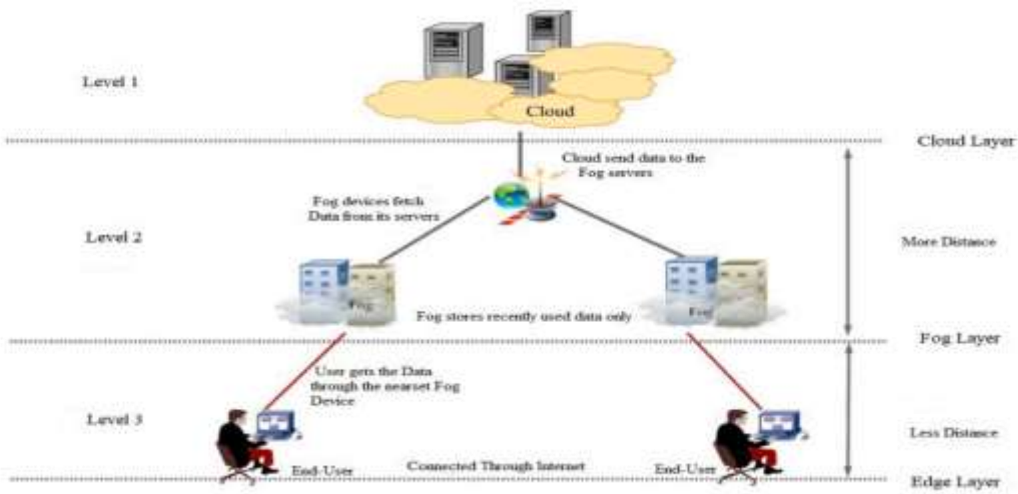


Figure 2: Overall Architecture of Cloud Storage System

2. Detailed Procedures for the Survey

The following is a list of the procedures that were carried out in order to carry out the extensive survey:

- Establish the purpose of the research_
- Establish the background for the study.
- construct the questions for the research.
- Gather the information from the web resource.
- Provide an overview of the findings from the survey in summary form.

2.1 Frame the research objective

• To optimise the data stored in the cloud by reducing instances of redundancy and recurrence, with the goal of improving bandwidth, latency, and storage costs as a result_

2.2 Setting the Research context.

Data replication's principal function nowadays is to make it easier to manage massive amounts of data over a distributed network. In addition to improving data availability, data redundancy also speeds up the process of accessing data and cuts dol. on the amount of time it takes to do so. Data replication in online backup clusters faces two major challenges: selecting the proper number of data replicas and establishing the placement of the duplicates to ensure that system operations can be completed as rapidly as possible.

Regarding the first issue, maintaining a large number of rigid replicas is not an approach that is appropriate nor does it maximise cost efficiency. Distributed data storage systems like Google's GFS, Amazon S3, and Hadoop's HDFS, for instance, replicate data usinid three-replicas approach by default In regard to the latter issue, load balancing is required in order to meet the expectations of the users, which necessitates an in-depth and methodical examination of the system [7].

2.3 Frame the research questions

The research questions (RQ) are framed with consideration given to the context of the proposed research work. The following are some of the research questions:

RQ1: In order to optimise the data stored on cloud servers, which strategies for removing redundant data are being applied?

RQ2: Which reoccurrence reduction methods have been investigated for use with cloud server data?

RQ3 Refers to the performance evaluation metrics that are utilised in the process of measuring the performance of the data stored on cloud servers.

2.4 Collect the data from web resource

After formulating the research questions, we developed a search string by making use of boolean expressions to link synonyms by using the 'OR' operator and essential search terms by using the 'AND' operator. Depending on the particular online library that was being searched, exclusion criteria were specified by using either "NOT" or "AND NOT.' The search string looked like this: traffic AND (predict OR forecast) NOT network NOT Internet NOT air NOT rail NOT rail When there was no option other than to fill out forms in order to search, the exact same search string was used as a reference. The following online library resources were utilised in the process of data collection for the review:

- IEEE Xplore
- Science Direct
- SpringerLink

RQ1; In order to optimise the data stored on cloud servers, which strategies for removing redundant data are being applied?

A phenomenon called as redundancy can emerge when sensor nodes are extensively distributed (either arbitrarily or systemically) to observe a same phenomena [106]. Redundancy is also known as the duplication of data When dealing with a hostile environment [1071 that is inaccessible to humans, such as the depths of the ocean, volcanoes, earthquakes, and so on, random deployment is the method that is recommended.

As a result, dense distribution may be expected, which finally result in duplicate information. Sensor nodes are often deployed in close vicinity to one another in a random. deployment method_ The crossing of their individual sensing ranges causes the overlapping that follows. All of the exchanging nodes will feel the same event and provide duplicate information if it occurs in a region that overlaps with another [108-120]. The concept of "spatial redundancy" refers to the fact that it is possible for multiple sensors to capture the same data about the same phenomenon simultaneously.

Based on the approach of constructing the network to satisfy the needs of certain applications and the geographic distribution of network deployment. Sensor node readings are made more reliable and the network's high reliability is improved by using this technology in a range of applications. Spatial diversity is both essential and practicable in loTs and WSNs, since sensor nodes are idely dispersed. This is because dense distribution of sensor nodes results in an enormous amount of redundant data [113]. The spatial redundancy method has attracted a lot of attention in the previous few decades.

For some purposes (such as military applicati.), high spatial redundancy is nearly preferable to low redundancy since it signals improved data accuracy, reliability, and network availability_ Improves network tolerability in the face of failures [111]. This is in contrast to the fact that a low spatial redundancy is undesirable, as this leaves the system incapable of coping with mistakes or defects and makes it more prone to them [114]. However, in order to function properly, applications like those used for forecasting the weather need only a moderate amount of spatial redundancy.

Spatial redundancy offers both advantages and downsides just like every coin. Processing redundant and irrelevant features at the underlying architecture is costly whenever there is a high incidence of spatial redundancy due to operational procedures and storage expenses.

Acquiring details about a certain occurrence occurring within the vicinity of sensor nodes that duplicate themselves raises the possibility of sending the same data to the data centers (e_g_ cloud, db).

Cloud storage services like Dropbox, Google Drive, and Baidu have been increasingly popular with the emergence of the internet and digital gadgets [8]. It enables users to outsource their data with increased flexibility and reliability. There is without a doubt a significant amount of duplication within the multi-user model of cloud storage_ These duplicate data significantly reduce the efficiency of the use of storage space and waste some of the bandwidth available on the network.

Consequently, a large number of researchers implement data deduplication technology during the processing of data storage in order to eliminate redundant data .

Data redundancy is the process of comparing hash values, including Rabin Fingerprint values, to find duplicate data. The hash value is the same for all of the same data.. Since the reduplication mechanism removes duplicates, no data is stored As a result, they should be included as co-first authors together with the other writers who made equivalent contributions to the project. According to the results of the detection of duplicate data, the Corresponding Author data was duplicated. In order to investigate further potential for redundancy within these data, the researchers chunk the massive amount of data into smaller pieces_ A content-defined algorithm is a logical next step in the progression of existing chunking algorithms, which can be seen in BS W [9], TTTD [10], Elastic Chunking [11], and Fast CDC [14].

As a result, the level of redundancy that can be achieved is directly proportional to the chunk size. The traditional method of data reduplication, on the other hand, is only able to identify rnpletely duplicate data and is unable to eliminate redundancy in data that is similar. Specifically, two chunks share a large amount of the information yet include a tiny number of separate bytes, resulting in differing hash values. Because their hash values are different, the traditional method of data deduplication treats them as separate chunks. It should come as no surprise that it is difficult to identify redundancy among chunks with similar content. In addition, there are a great deal of redundant components involved in the multiuser storage scenario. You may save even more bandwidth and storage space by deleting redundant data from pieces that appear to be the same.

Abbes et al. [22] looked into the possibility of running distributed applications using a virtualized version of the container concept. This prediction was made using the experimental data forecasting model that they proposed. In order to determine the nature of the connection that exists between the availability and the number of replicas, they made use of the regression method.

In the multilevel and diverse task approach of cloud - based systems, Ao et al. [23] suggested a framework to i stutahow duplicated resources might be assigned &minimize the time needed to perform a job. As a result of communication delays caused by ey-efficient replication,

Boru et al. [24] suggested their concept to improve service quality. On the basis of energy use and bandwidth duplication, a cloud datacenter model is proposed. Cloud users will be more satisfied with their interactions with these services if they are able to reduce communication latency.

In the cloud, Zou et al. [25] developed a method of isolation known as serializable snapshot isolation, which is more reliable than eventual consistency. To guarantee that every copy is snapshot isolated, it is now possible to create one-copy serializability in an SRDC using a controller method.

Building a prototype for SRDC was another method through which they looked for transaction read and write sets with the goal of enhancing the performance of certificati techniques. Using the ideal multiple copies and the proper partitions, the petimance data from the TPC-W combinations revealed that reaction rate could be improved.

In order to reduce the power consumption of data file access inside data centre1Alghamdi et al. [26] investigated the file duplication problem and introduced two successful heuristic file replication algorithms for energy and time. This was accomplished by working on the file replication problem_ A time-saving method of approximation was provided to ensure that energy use in the file replication was maintained. Profit was the driving force behind their strategy, and a piecewise function that can be successfully computed was added to it. An approach to replicating and retrieving data was given by Armknecht and colleagues [27]_

Using cloud technologies that were compatible with the current cloud system, the growing garden of building copies could be relocated to the cloud. They emphasise the system's resilience to both unauthorised cloud vendors and unwelcome intruders in their security-resistance demonstrations. In order to improve the cloud servers' data availability,

Nivetha et al_ [28] considered the possibility of placing replicas in a variety of data centres_ The creation of replication may be contingent on the cloud data centres having the highest possible degree of data that is usable or accessible. When selecting replicas, you should look for the copies that are most suited to being placed in more than one data centre. The authorisedesers are responsible for completing the configuration of the data centres so that they can supply the appropriate system to the customer. Multiple data servers can work together to find storage servers that can provide the fastest feasible response time to data requests from clients, according to Jayalakshmi et al. [98]. They endeavoured to maintain the scheme's target at the optimum number of replicas throughout their efforts.

Their system will send a files request to the nearest datacenter in order to save money on data transmission for the content that is being sought. This method has the ability to reduce completion time and data transmission costs by expanding the number of copies_ In point of fact, the increase in the number of replicas only applies to the files that are in high demand_ When a request is being processed, any files with a low processing count or a low frequency of access are destroyed. Due to its ability to reduce storage costs, this technique controls the number of clones to an ideal level over a particular length of time Data centres located close to clients can reduce response times for users and the degree of latency they encounter, both of which can be reduced by using appropriate replication data centres. When there are fewer replicas, the amount of energy that is consumed is reduced. Therefore, striking a balance between a number of competing priorities is an essential component of the replication control strategies.

The developers of [29] used a small number of copies in an attempt to increase data availability while decreasing the time consumers had to wait. To make this happen, the authors devised a model consisting of two stages: in the first stage, the choice of the Ira centre is made, and in the second stage, dynamic data replication takes place_ They tries to answer questions in their model of the system: What file should I copy? Is it permissible to copy it at all? What are the best places to go in order to get this job done efficiently? These innovative approaches, that include datacentre management and dynamic data replication, are aimed at providing the appropriate number of copies while also enhancing system performance_ These goals are similar to those of the older strategies. Dynamic replication of data in the cloud was shown by Rajalakshmi et al. in their study [30].

This method makes use of a management system that gives users the ability to manage replicas. This strategy emphasized the conceptual model for determining the optimal replication and the position of that duplication in order to optimise c loud-based data storage. Both the use of files as well as the replication of processes are integrated into their suggested method's features list. The first method encompasses the replication placement and creation processes, which are carried out with the aid of the catalogue and the index. When determining whether or not the destination has enough storage capacity to store a file, this is utilised. The goal of replication is to reduce the delay time, price, bandwidth consumption, and availability of the data by distributing the data among several servers.

Kumar et al. [31] presented a workload-attentive replication approach for data insertion in order to save resources. It was possible to decrease the number of systems required to execute a query or interaction by creating partitioning algorithms that took into account the anticipated workload and treated it as a graph. In additional to this, they exploited the fine-grained structures in order to limit the costs of dispersed updates, boost throughput, and respond to a range of varied loads.

An intelligent swarm optimization model was used in Shaoet al's [47] proposal for a method of data placement that is both reliable and cost-effective, and it collaborates the edge and cloud environments_ They approached the issue of placement as a binary programming problem and modelled it

For the replication optimal placement, Zhang [32] proposed a time-efficient, multi-objective solution with the aim of decreasing the cloud storage system's reaction time to a minimum while also taking quality of service constraints into consideration.

Mansouri et al. [33] studied the cost-effectiveness, reaction time of applications, and cloud storage balancing of cloud storage. They explained how they used the metrics described previously in the paragraph in their model. They investigate several parameters associated with the metrics discussed earlier while using the CloudSim simulator to test their methodology. These measures led their model to determine that the ideal place for storing copies was the one they had chosen_ Additional to that, they gave a technique of replacing a replication depending on the provided of the document, din-lost recent access time for the shared data rate, as well as the size of the replicas.

Huang et al. [34] provided a model-based strategy for discovering relationships between cloud storage data items. Utilization of less energy while improving efficiency was a primary objective of this approach. In addition to that, they incorporated a replica placement and backup strategy into their proposed mining method. When it comes to distributed cloud storage systems,

Atrey and colleagues [35] have devised an efficient method for distributing large amounts of data_ Their model incorporated two algorithms in order to handle this problem, increase efficiency and speed up processing. They added these techniques since scaling their partitioning approach for huge workloads necessitates a lot of computation.

Replica placement optimization was addressed by Ebadi et al. [49] in their study. An technique based PS0 and a local search technique known as Tabu Search were devised to solve this issue. Their goal was to achieve reasonable performance from a system. The HDFS framework was provided by ko et al. [36], along with an explanation of how fault tolerance is achieved through the use of erasure-coded replication.

The physical location of the EDFS replications is an extremely important factor in both the dependability and performance of HDFS. This tactic puts the fundamental principle behind maintaining hashing consistency into practise. They assume the least possible and most efficient use of storage space so that they can evaluate the performance of their framework. Because cloud storage relies on the dependability of the information that is being saved, fault tolerance is a vital need. [36] Bioinformatics data replication was discussed by Da Silva et al. in the cloud computing context. This strategy was developed with the intention of reducing the amount of time required for the application of bioinformatics. It is possible to demonstrate the rich scalability of data replication systems through the usage of localised blocks.

A replication technique based on the frequency of access to a shared folder was developed by Janpet et al. [37] to improve the quality of cloud storage. This allowed for greater flexibility in how the data could be accessed.

This approach reduced the amount of time it took to access information, with both the duration of effort saved variable depends on the measured available bandwidth and the distance of the data storage site. This method established a new duplication scheme by introducing a network model complete with access frequency modelling, delay modelling, and budget modeling._

QoS-aware data replication (QADR) was developed by Lin et al. in reaction to the high Quality - of - service first replication technique, which is a greedy strategy that cannot lower the cost or the number of QoS breaches that arise during data replications. [38] The second technique specifies the least maximum-flow (MCMF) problem to solve the QADR issue in polynomial time, even though the first method requires more computing effort.

Yi et al_ [39] presented Cadros, a cloud-based data duplication and storage technique, in their research with the goal of enhancing data accessibility for dispersed online social networks. According to whether or not they were duplicated using complete or removal coding, the released data is divided into distinct groups in Cadres. It will be easier to find the best data division if there is sufficient data access to justify the additional expense of removing any unnecessary coding. MinCopysets is a scalable replication algorithm that was proposed by Cidon et al_ in their paper [40].

In order to increase the data's longevity while keeping the advantages of randomness in load balancing, this algorithm was designed Load balancing and data replication were separated because they planned to utilise random nodes for both, but they used non-random nodes for data redundancy. They aimed to separate themselves from one other_ As open-source distributed storage systems RAM Cloud and HDFS, they were able to demonstrate that their suggested algorithms result in a negligible increase in transaction time. Article [41] offered an approach to data replication with the purpose of saving money on data storage and making workflow applications more easily transferable_ According to their system, the types of datasets are divided into three categories, while the amount of data storage space is divided into two categories.

In the paper [42], Djebbar et al. propose an effective method for the placement of data as well as the scheduling of tasks. The K-means algorithm is used to replicate data and implement a classification-based investing strategy. As a result, both the reaction time and the cost have been improved.

According to Ibrahim et al. in their work [43], the cloud storage system may be improved by using intelligence while storing data. Fair provisioning of distributed replicated in cluster nodes takes into account the outcomes of load balancing. This does not require any load balancing utilities to be used.

Using Hadoop Distributed File System (HDFS), the authors developed a replication placement approach to strengthen the system's fault tolerance properties in their article [44]. They also demonstrated a homogeneous cluster system, in which all of the nodes in the cluster had the same computing capacity. There has been work done by _Mang et al_ [45] to design a cost-live technique for the deployment of replication to enhance application response times and load balancing in cloud storage.

The bidding replication procedure is initiated when the data availability does not fulfil the specified parameters. Replica properties and bidding mode settings are combined in this way_ When Khalajzadeh and his colleagues published the study referred to as [46], they used an algorithm based on genetics to optimise the placement and replication of social media data on cloud servers with variable access rates. When compared to other placement policies, their method is able to achieve a manageable placement policy while simultaneously guaranteeing the delay. Using a duplication placement and migration strategy, the authors Li et al_ [47] suggested that edge cloud environment apps impose a balanced demand on the data nodes. To solve the multiObjective issue, they used the genetic algorithm they had suggested.

It was presented by babas et al. [51] to increase the availability of cloud storage system replicas by using a dynamically data management strategy. In the paradigm of the system which they have designed, users that need data copies would visit one of multiple datacentres located over a vast geographic region.

A cost-aware replication management strategy was proposed by Li et al. [52] as a method for solving difficulties in edge-cloud computing systems connected with issues related to energy consumption and fault tolerance.

In addition to this, they had a recovery strategy in place in the event that any of the nodes failed. When a node in a cloud data storage system fails, the data is still accessible and secure because to the use of an all-or-nothing transform and fountain codes, according to Bacis et al [48]_ That's what we did_ One strategy used encryption techniques to guard against tampering, while the other described a mechanism to avoid data becoming unavailable in the case of transmission failure.

[49] Khalili Azirni has proposed a bee colony-based information management technique for cloud storage systems that selects and prioritises the data that should be duplicated as well as how often and where it should be replicated. Selecting the data for replication, the time of replication, and where the duplicate will be located are all part of this method.

A multi-objective offline optimization solution was presented by Long et al. [50], and it made use of some latency factors in order to make replication decisions_ File unavailability was one of the many causes that contributed to delay. Their response also incorporates the use of an enlarged artificial immune approach to the choice of a layout. Their suggested algorithm incorporates a few strategies that were derived through the use of cloning, mutation, and selection processes.---p14

The state-of-the-art works on detecting redundant data and using delta compression to eliminate the superfluous component of the data are N-transform [13] and Finesse [14]. These two papers can be found here and here. Only the content that is completely unique is saved. The N-transform will generate the feature for each chunk so that it can guarantee a high level of efficiency in the similarity detection process.

In addition, the feature distance is the determining factor in determining whether or not the chunk shares a similarity with the others. If the feature distance between the two chunks is smaller, then the chunks are more similar to one another. The Rabin fingerprints [15] of a chunk are obtained by N-transform in their entirety. There are N-dimensional hash sets generated for each and every fingerprint digit. Finally, the features include the N **most** significant values of each of the N dimensions, one for each. However, because the linear transformation process causes feature calculations, the N-transform is affected negatively by these calculations. Finesse uses a grouping strategy to calculate the chunk features, which allows for a further acceleration of the resemblance detection process [14].

Rabin fingerprints connected with each sub-chunk are therefore extracted into a number of continuous sets of same size after the chunk is broken up into sub-chunks. Using these values, the characteristic is then built by grouping them together based on how important each set is. When it comes to resemblance detection, the objective of the Finesse algorithm is to achieve performance that is superior to that of the N-transform method. The Metadata element is **cruc**ial to decluplication speed, yet it is ignored by the existing similarity detection technique, including Finesse_ Despite the reality that Finesse takes less time to complete the transformation than N-transform, this is still the case.

Some researchers make use of delta compression, a method for the reduction of data, to achieve the highest possible compression ratio [16]_ This allows them to identify and get rid of any similar chunks that contain duplicate parts. To record the difference between chinks that are otherwise similar, the delta algorithm must be used in conjunction with the delta format. In delta files, only the differences between two versions are saved. When determining which chunks can benefit from delta compression, however, delta compression does not perform well. Because of this, the initial stage in delta compressing is referred to as similarity detection and is tasked with figuring out how similar different chunks are.

Aronovich et al. [17] suggested a new kind of similarity signatures that can be used in the deduplicati**ion** system. This uses byteby-byte comparisons and hash-based identity techniques to combine similarity matching with byte-by-byte comparisons.

An efficient deliplication technique for data management systems is proposed by Xu et al_ [18], which uses byte-level encoding to reduce duplicate data There are a number of different methods for detecting similarities at a finer granularity level [19,20]. These methods can create a large number of false positives since they extract the features from non-overlapping blocks. As recent research, such as N-transform and Finesse, has been conducted, efforts have been rrie to enhance the accuracy and performance of resemblance detection by making use of the grouping features mechanism. Both of these methods discover data chunk-level similarity patterns.

The N-tinsform approach is used to get the chunk's top k biggest Rabin fingerprint values. After that, a super feature of this chunk can be computed by combining several features in the same way. Finesse also identifies which sub-chunk includes the greatest Rabin fingerprint level and then arranges the remaining Rabin fingerprint values into features as per the ranking of these values as well. However, according to the findings of previouirearch [21], the metadata factor is essential to the performance of the deduplication system. When the average chunk size is smaller, the amount of metadata that is generated is proportionally greater. Data deduplication with similarity detection will exacerbate the problem even further. When it comes to matching two images, the present state-of-the-art approach is computationally intensive as well as data-heavy_ The chunk feature, the validation code, and the chunk sequence can all be included in this metadata. The present approach of pattern matching does not take metadata into account while performing data deduplication.

RQ2: Which reoccurrence reduction methods have been investigated for use with cloud server data?

In today's world, a significant portion of economic activity is predicated on the collection,processing, and utilisation of data by businesses at every stage of the supply chain. The amount of data that companies generate and then analyse is growing at an exponential rate. According to the findings of a recent study, the amount of data that is being recorded across the globe is growing at a rate of between 30 and 40 percent each year [53]. A sizeable amount of the data that has been saved is also currently being actively processed. While customers are regularly utilizing online services like social media, video streaming, and cloud storage, businesses are mining vast amounts of customer data in order to provide increased levels of personalization. It is anticipated that we will have processed five zettabytes of data on a global scale by the year 2025 [54], if we continue to grow at this rate. Cloud computing is becoming increasingly popular among individuals and businesses alike as a means of storing and processing ever-increasing amounts of data. When it conies to storing and processing data,cloud computing provides a number of important advantages, including high performance, resource elasticity, cost efficiency, and high availability_ Cloud users are able to adjust the amount of resources they use up or down depending on the changing compute and storage requirements of their applications, and they only have to pay for the resources they actually use at each time interval. As a result of cloud operators sharing their large-scale platforms with multiple tenants, the expense of maintaining highly available computing and storage infrastruclie is amortised over time. However, users will only be able to reap the benefits of the cloud in terms of improved performance and cost efficiency if theiloud platform is able to provide each application with the appropriate type and quantity of compute and storage resources_ Particularly important for the efficiency of data-intensive cloud applications as well as their overall cost is the selection of the storage medium_ For instance, when using local NVMe Flash storage, a Spark SQL equijoin query on two 128 GB tables can be finished 8.7 times faster and costs 8 times less than when using remote disc volumes on the Amazon EC2 public cloud [112].

Fog and edge computing systems' communication robustness has been the subject of various research [68-72]. Several research have focused on the copying of data through one node to another, which needs an increase in IoT compute resources [68,70]. Self-healing strategies such as fresh fog node allocation and data redundancy or relocation [71] are considered by Jeong et al. in case the fog fails. A load balancing approach proposed by Al-Khafajiy et al. [71] may be used to distribute the massive volumes of data collected from end-node devices amongst fog nodes_ This experiment does not contain any procedures for identifying and recovering defects if the network between fog and cloud is severed, in comparison to our Fog-DaRe system. DTNs have the capacity to hold data until a predetermined time after a disconnection, and then transmit it via the network when the link is accessible [72]. Research [1-81] link DTN with fog-based IoT systems, although these studies primarily address latency or packet loss metrics, do not include fog resource restrictions and only cover a limited number of application scenarios [73-85]. IBR-DiN is a MQTT with DTN architecture presented by Luzuriaga et al. to analyse the connection between a Raspberry Pi gateway as well as sensor boards [38], They put three alternative disconnect situations to the test: one where there is no disconnect, one where the disconnect lasts six minutes, one where the disconnect lasts twelve minutes. The corniunication channel fault is varied by 0%, 25%, 50%, and 75% in each scenario. Although just seven sensor boards are included in the experiment, it is noted that Il3R-DIN has no persistent storage and instead relies on RAM to store data.

DTN has been used in smart farming settings using mobile nodes, which broadcast data anytime they make contact [76,77]. As part of a real-world experiment, Kulatunga and colleagues monitored cows, each of which generates around 100kB of data every five minutes and transmits it to a wearable devisor gateway station. Using this method, researchers may replicate real-world settings. [75] In this study, the authors examine the link between the distance travelled, network latency, and device energy use. An architecture that takes use of DTN and mobile nodes is examined by Castellano et al in their paper. Regardless of the fact that they do not apply these processes, they remark that fog is capable of cleansing data by filtering and aggregation [76]. With the use of a simulator, they've been able to determine how long it will take to transport a bundle (packet) as well as how likely it is that it will arrive on time.

In addition to this, the same metrics are evaluated at 1, 2, 4, and 6 second intervals during the connection time. They came to the conclusion that because this architecture replicates packets, it uses more RAM than other architectures.

Edge network devices are able to "aggregate and synthesise useful information from the received raw data," according to Moura and Hutchison's [78] research. This his reduce the high data volume that is produced by the Internet of Things. Even though they don't take into account the possibility of the Ica stages becoming disconnected from one another, some researchers [74, 80] still take data reduction strategies, such "compression, into account when studying IoT systems. In an Internet of Things context, Spiegel et al. suggest the RLBE algorithm, which would be a version of the RLE approach. In the conclusion, the authors claim that RLBE is preferable to other lossless data compression algorithms since it saves 60% of the energy necessary for the data to get to the end nodes (thing stage). In contrast, little consideration is given to the influence on resilience in the studies.

Routray et al. provide an intellectual comparison of various compression algorithms used in the Internet of Things. They point out that situations with low bandwidth present a challenge when trying to improve the performance of the network. One strategy for achieving this objective is to cut down on the transmission of information that is not relevant. They claim that data should be compressed via edge computing since the nodes themselves lack the processing capability to handle massive amounts of data. The Zstandard, LZ4, and LZO are mentioned as algorithms that need minimal bandwidth.

Lossless compression techniques are tested on a range of edge/fog computing devic1 by Gia et al_ These include LZ4, LZW, Huffman, and Zstanclard compression algorithms (Raspberry Pi 3B, Intel UP, Intel i5, and UP gateway). The Zstandard method was thought to have a higher compression rate than the inventors realised. Nevertheless, the compression rate is the sole parameter taken into consideration.

As well as the speed of compression rather than the impact of compression for IoT resiliency. Chandak et al. and Blalock et al. present techniques for compressing multivariate data time series based on prior data samples [87,88]. They compare their algorithms to those of other compression methods using datasets obtained from Internet of Things use cases in order to determine the compression and decompression ratios as well as the speed_ Data compression was not suggested by either Chandak et al. or Blalock et al. for building a robust Internet of Things flow of data. Fu et al. [82] suggested a data management solution that addresses data fault tolerance and subsampling across fog and multi-clouds. F2MC is an abbreviation for fogto-multicloud data management. In spite of this, the authors do not test F2MC in a scaled situation where hundreds of sensors are delivering real-time data to the system. In addition, they do not take into account work device limits such as LPWAN. For the purpose of collecting and condensing the massive amounts of data generated by thousands of sensors, Sinaeepourfard et al. Zip compression may be used to save storage space in loT systems by deleting unneeded data and compressing it, according to their research. There are also factors of resilience that they don't take into consideration, such how data reduction strategies affect transmission rate after a cloud disconnect and fog computing resources_

Storage, retrieval, and preprocessing functiirlities were all integrated into a single framework by Fu et al_ [100] for processing data in the Industrial Internet of Things (HOT). Fog architecture and cloud infrastructure were used to create this framework. In this design, the embedded device will only hold time-sensiie data after pre-processing raw data; the rest of the data will be transferred to the cloud. To limit the amount of data that must be transferred to the cloud and the amount of storage space required, pre-processing has been implemented_

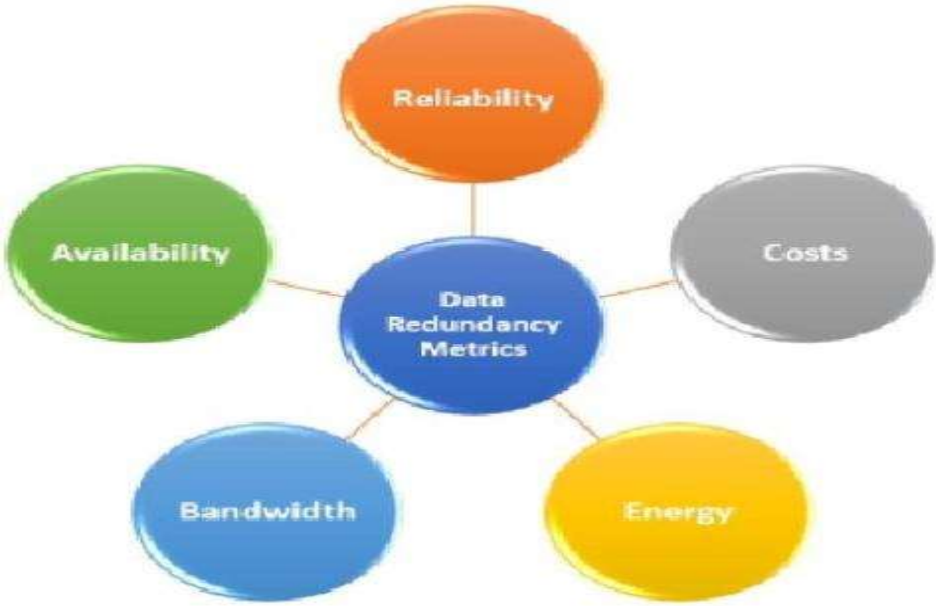
An accurate and effective method of retrieving the data that has lin stored has been developed by the authors in the form of a retrieval feature tree. The use of an index encryption technique based on the kNN algorithm has also been proposed to facilitate private searches. These constraints must be solved in order for flexible data search techniques and the invention of a new indexing system for storage to be encouraged within this framework_ Mining data while

Pseudonym certificates for smart devices in a fog-enabled Internet of Things system were presented by Guan et al. [105] as an anonymous, privacy-preserving authentication technique (APPA). In order to safeguard the privacy of individual devices, pseudonyms are processed at selected Fog nodes using the paillier algorithm. The computational difficulty and the communication expense are both reduced as a result of this strategy. A cloud-based system would not have this problem, as data aggregation takes place at fog nodes rat than in the clouds Zhang et al [105] presented a system for offloading compute jobs from terminal nodes to Fog nodes equipped with transitory rage, which also minimises the power consumed and the time required to accomplish a task. Offloading of tasks to chosen Fog nodes in accordance with the fairness metric is accomplished through the utilisation of the PTO scheme that has been proposed. As a consequence of implementing the scheme, the amount of energy used and the amount of time needed to complete tasks were both reduced in the Fog environment. However, the scheme does not account for the security of the offloaded data, which is something that could be investigated in the future.

RQ3: Refers to the performance evaluation metrics that are utilised in the process of measuring the performance of the data stored on cloud servers

Overall, the data replication strategy's potential to improve reliability is a plus [47]. For this reason, data replication might be seen as enhancing the service's dependability in the event of assaults or disasters [48]. For this reason, data replication is an effective approach to expand data sharing principles over a larger region_ As a direct consequence of this, fault tolerance in these kinds of cirinstances will be increased. In order to keep the level of dependability at an acceptable level, it is necessary to increase the number of replicas, which results in an increase in expenses [49]. As a result, striking a balance between the need to ensure reliability and the desire to minimise associated costs is a requirement that calls for careful management.

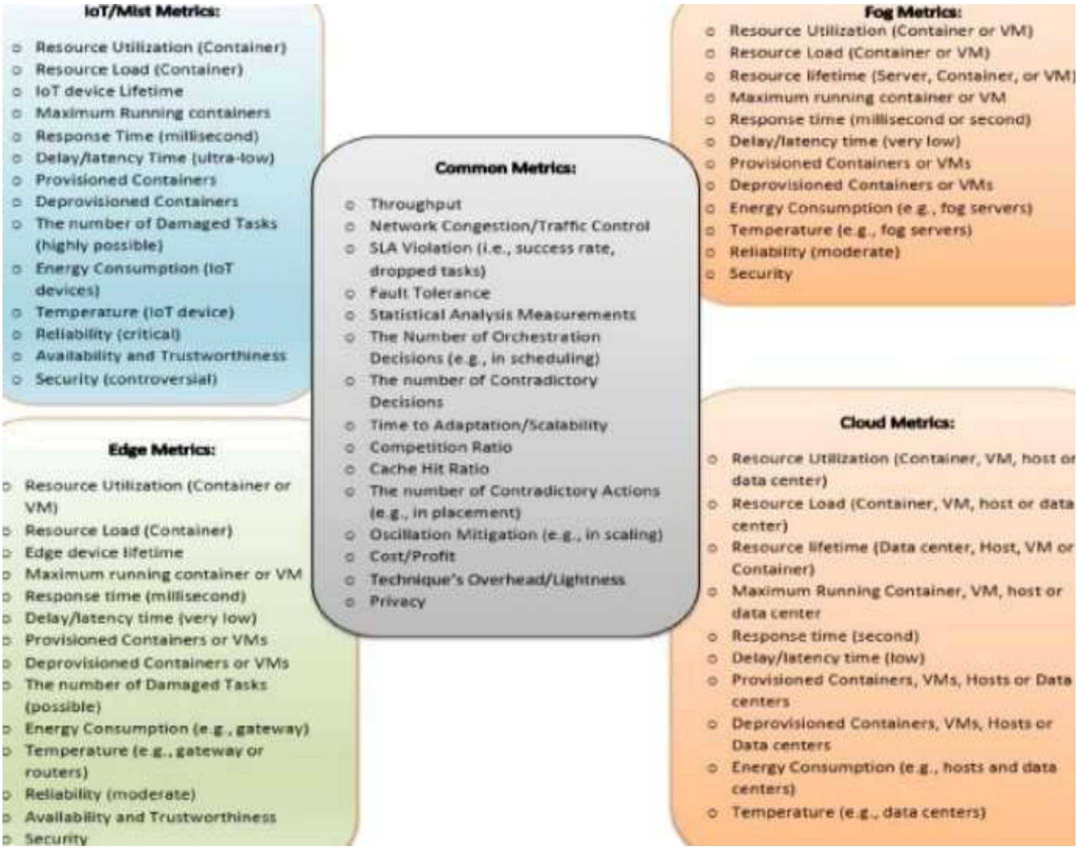
Figure 2: Data Redundancy evaluation metrics in cloud server system



To ensure that data is accessible, the service must be available in the cloud infrastructure, which is a challenging task in an environment that changes so frequently_ Thiixtent to which the system is consistent is determined by its accessibility_ In order to satisfy operational demands for the essential service, the system has to be scalable and flexible in order to do so in a timely way. Adding this feature will raise the system's original and continuing expenses, in addition to its inherent complexity. A delicate balance must be struck between meeting demand for availability while also preserving consistency [50].

The computing model or layer in which the loT application is orchestrated has a significant impact on the performance metrics that a1 derived from it For example, the availability of Internet of ' gs devices to take care of offloaded tasks is a major concern in a mist model, whereas in a oud model or layer, the availability is guaranteed. Instead, the evaluation of response times is of the utmost importance in the cloud model_ Not only are availability and response time essential in the cloud and the mist, but they are also essential in the edge and the fog_ On the other hand, the weight that is gin to each performance metric varies depending on the context. The performance metrics of new computing paradigms, such as fog and edge computing, may be similar to cloud computing since these new paradigms inherit some of the cloud computing properties_ Several measures are similar to all levels, while some have special needs_ When it comes to measuring how much resources are being used, containers, virtual machines (VMs), or a combination of the two might be used. Each of these indicators has its own groupings to make it easier to keep track of them.

Figure 2 provides a taxonomy of performance measurements that may be applied io all computer environments. This was motivated by the previous statement. This taxonomy elucidates the measurable metrics that are unique to each computing layer, in addition to the metrics that are measurable on a universal level across all layers. After that, the cloud layer is broken down in order to find measurable performance metrics that are specific to each model of cloud individually.



•During the monitoring phase, the performance of the entire system, including the underlying infrastructure and how it affects users, can be monitored. CloudWatch, which is offered by AWS, is a good illustration of this.
•The term "resource utilisation " refers to the proportion of available resources that are used up by the new amount of work that must be done. To put it another way, it reveals how busy the CPU currently is. A resource could be a central processing unit, random access memory, memory, bandwidth, etc.

The term "Resource Load" refers to the p0centage of work that is being done by a physical host, a virtual machine (VM), a container, etc_ It is a measurement of the number of tasks that are currently running as well as those that are waiting in line to be executed on the central

processing unit queue. Only the latter was meant to be referred to by the previously metiined CPU utilisation_
The term throughput" refers to the ratio of the number of tasks that arrive to be processed to the number of tasks that are processed in a given amount of time.

The term "VM Lifetime" refers to the amount of time that the virtual machine (VM) is rented It is defined as the maximum number of resources that are currently being run_ The maximum running resource.
The term "response time" refers to the amount of time that elapses between the dispatching of a request to a server and the conclusion of the task's execution..

Delay time, also known as latency, is defined as the amount of time that elapses between the actual completion of a task and the time that was desired for its completion.

- It is defined as the incoming load that can be indicated by the number of incoming requests to it during the time, and the term "network congestion" or "traffic control" is often used interchangeably.

- The number of tasks that were damaged can be defined as the number of requests that were not successful in receiving an answer or at the very least an answer that wasprovided in a timely manner.
- SLA Violation is defined as the number of tasks, or the percentage of total tasks, that experience, delay time that was longer than what was conceded in the agreement. Additional parameters, such as the availability of resources, may also be included in the Service Level Agreement (SLA).
- The term "fault tolerance" refers to the ratio of the number of faults that have been found to ti total number of faults that are currently present. Errors could be caused by either the software or the hardware [75].

Energy Consumption is defined as the amount of energy that is consumed by a resource in order for it to complete the execution of a workload [75].

- The term "temperature" refers to the degree to which or the intensity of the heat produced by the underlying infrastructure, such as a data centre, when tasks are being carried out [76].

- The term "reliability" refers to the capacity of a mechanism to carry out its intended function satisfactorily under specified conditions, despite the presence of some problems [771.
- It is defined as the existence of the system at any time that users make a request [75]. Availability is also known as accessibjy.
- Trustworthiness refers to the extent to which the optimization technique can deal with devices that cannot be trusted.

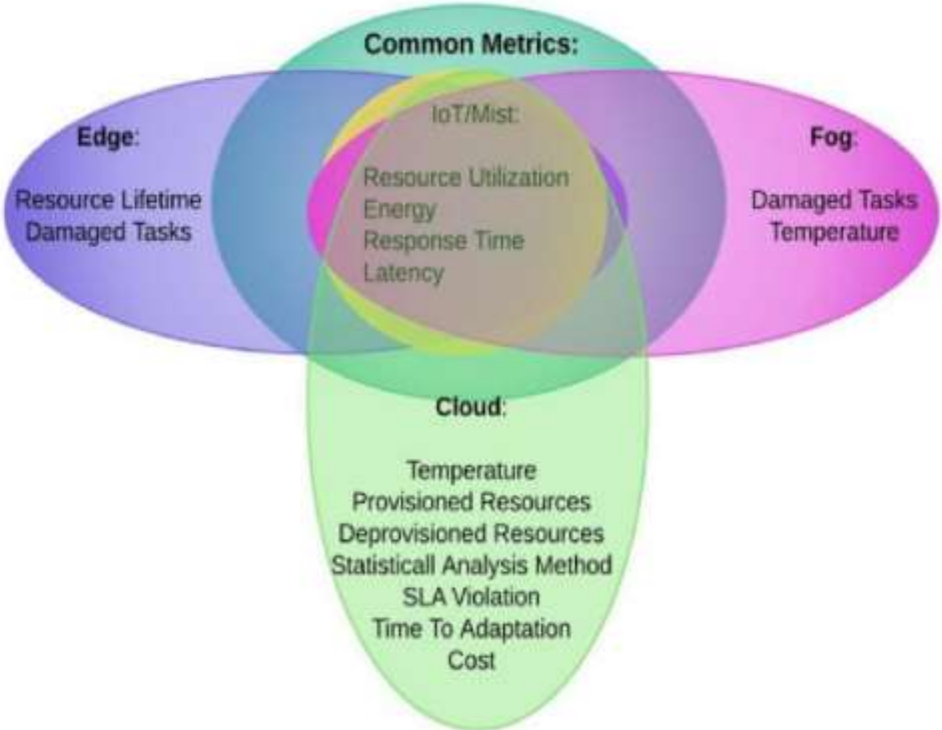


Figure 4: Supported by Existing Simulators' Performance Metrics

- 3. Summarise the outline on the findings from survey SLA standards are being worked on by companies like Amazon and Google, however they are still in the early stages and therefore do not take into consideration a variety of indicators. With the Amazon EC2 service, for example, each and every instance is guaranteed an hourly.
- uptime of 90% or more. If the 90 percent uptime guarantee is not met, they will not charge the user for the service. This is the only commitment that they make available. Even if the instance is accessible, there is still a possibility that there will be questions regarding its performance. Amazon will not issue a refund for the money, and any losses incurred as a result of the unavailability will not be taken into account. Failures of this type occurring on a regular basis will not have any impact on Amazon's SLA violation policies; instead, the company will simply disregard the penalty for the usage of the instance during the preceding hour. The management of resources in fog computing, such as the sharing and discovery of storage, presents a significant challenge when trying to improve application performance. According to the principles of fog computing, centralised data centres and pervasive mobile devices are expected to collaborate and share the diverse resources and services at their disposal. [36, 82]

- **4. Future Avenues.**

● The Internet of Things generates dataflows at a breakneck pace. Electric gadgets, including smartphones, sensors, and tablet devices, are constantly generating data streams at a greater pace. This data may be accessed and used in a variety of ways [55]_ The proliferation of mobile and other forms of communication technology are primarily responsible for making this possible. For instance, in intelligent traffic systems, a variety of cameras are used for the purpose of monitoring traffic_ When a camera has a higher resolution, it is possible to analyse more information, which results in a significant rise in the amount of bandwidth that is required by the system. However, because of developments in manufacturing and production techniques, the performance of each sensor, and the volume of sensor-deployed devices, has substantially grown and iirroved. This is due to the fact that there are now more sensors. This is true even for complex devices, such as cameras and monitoring systems, which are experts in the transmission of large-scale video streams. It is anticipated that the data generated in real time will provide ',more comprehensive reflection of the practical circumstances; Moreover, these information also pose demanding needs for the computing and storage resources that are available. The observation of flow of traffic and congestion in real time, for example, is useful to prompt assessment and assessment of road conditions. This allows citizens to make knowledgeable decisiod and appropriate plans. From the other hand, the maximum speed creates many obstacles for providing real-time processing and response, such as in the areas of data security, transmission, and processing.

Real-time surveillance of flow of traffic and congestion is beneficial to prompt asseitent and evaluation of road conditions in the context of intelligent transportation systems_ In order to ease the burden of data processing and transmission, distributed loT architectures such as fog computing, edge computing, and others utilise processing data closer to sensors.

Fog computing and other dispersed loT systems make advantage of these architectures. However, this practise will eventually lead to problems, such as an increased risk to the data's security and the use of a singular processing model_ Real-time methods aiming at achieving increased data stream in intelligent traffic have been the subject of numerous research As an example, Rathore et al. [55] established a framework for multiple discriminant analysis of smart traffic using Giraph and Spark_ Real-time results were the goal of the researchers. The investigators Nallaperuma et al. [56] utilized online learning instead of asynchronous and progressive learning to handle the massive amounts of newly entering data. One of the most important strategies for managing high-speed dataflow is the utilisation of edge nodes [57, 58].

● Data reduction techniques such as delta compression can be used to get the maximum compression ratio [65], [66]. This allows them to identify and get rid of the duplicate part that is present among the similar chunks. To record the difference between chunks that are otherwise similar, the delta algorithm must be used in conjunction with the delta format. In delta files, only the differences between two versions are saved. Delta compression, on the other hand, not very good at answering the question of which chunks should be considered candidates for delta compression. Resemblance detection is the initial phase of delta compression, and its goal is to discover how similar chunks are_ Aronovich et al_ [61] suggest a new kind of similarity signatures that can be used in the deduplication system. In order to achieve greater savings, For database management systems, the work of Xu et al. [68] suggests a similarity-based deduplication solution based on byte-level encoding.

- There are also some other approaches to the detection of similarities at a coarser grain level [69,70]_ These methods can create a large number of false positives ince they extract the features from non-overlapping blocks. As the most recent works, such as N-transform [65] and Finesse [66], attempt to improve the accuracy and performance of resemblance detection by making use of the grouping features mechanism, this article will discuss those works_ Both of these approaches detect similarities at the chunk level in the data. After that, a superfeature of this chunk can be computed by combining several features in the same way. However, according to the findings of previous research [67], the metadata factor is essential to the performance of the deduplication system. When it comes to similarity detection, the current state-of-the-art work is computationally intensive and generates an enormous quantity of metadata. The chunk feature, the validation code, and the chunk sequence can all be included in this metadata_ The existing method for similarity identification during data deduplication doesn't really take into consideration metadata [68].

5. Conclusion.

In recent years, the explosion in the amount of digital information has resulted in data occupying an ever-increasing amount of storage space [1]. It has been discovered that the amount of redundant information contained in the data saved on the cloud server by the application system can reach as high as 60 percent, and that the amount of redundant information continues to grow as more time passes. Conventional file storage system is able to minimize of repetitive stored data by improving data density as well as, as a result, the amount of space that is dominated by the data [2, 3]. This is achieved via coding mapping which is organised as per the inter - relation of the data. Additionally, the typical data storage technique is only able to eliminate data redundancy that is included within the document itself; it is helpless to handle knowledge that is repeated in numerous files_ It is abundantly evident that the conventional digital storage technology and system management have had a difficult time keeping up with the demands of data in terms of quickness, quantity, storage efficiency, and safety. These prerequisites can be divided down into the following categories: As a consequence of this, it is absolutely necessary to carry out research on effective means of storing data.

References

[1] Dasgupta K, Mandal B, Dutta P, Mandal JK, Dam S (2013) A genetic algorithm (ga) based load balancing strategy for cloud computing. Procedia Technol 10:340–347.

[2] Wood T, Ramakrishnan KK, Shenoy P, Van der Merwe J (2011) CloudNet: dynamic pooling of cloud resources by live WAN migration of virtual machines. ACM Sigplan notices, vol 46. ACM, New York

[3] Kim W (2013) Cloud computing architecture. Int J Web Grid Serv 9(3):287–303

[4] Litoiu M, Woodside M, Wong J, Ng J, Iszlai G (2010) A business driven cloud optimization architecture. In: Proceedings of the 2010 ACM Symposium on Applied Computing pp. 380–385.

[5] Yang Y, Hu T, Ye Y, Gao W, Zhang C (2019) A knowledge generation mechanism of machining process planning using cloud technology. J Ambient Intell Humaniz Comput 10(3):1081–1092

[6] Alapati S, Kumar P, Palani GS, Ravindran R, Sadasivam SK (2014) U.S. patent no. 8,914,515. U.S. Patent and Trademark Ofce, Washington

[7] Milani, B.A., Navimipour, N.J.: A comprehensive review of the data replication techniques in the cloud environments: major trends and future directions. J. Netw. Comput. Appl. 64, 229–238 (2016)

[8] P. Sharma, R. Jindal, and D. B. Malaya, "Blockchain technology for cloud storage: A systematic literature review," ACM Comput. Surv., vol. 53, no. 4, pp. 89:1–89:32, 2020.

[9] A. Muthitacharoen, B. Chen, and D. Mazieres, "A low-bandwidth ` network file system," in Proceedings of the 18th ACM Symposium on Operating System Principles, SOSP 2001, Chateau Lake Louise, Banff, Alberta, Canada, October 21-24, 2001, K. Marzullo and M. Satyanarayanan, Eds. ACM, 2001, pp. 174–187.

[10] K. Eshghi and H. K. Tang, "A framework for analyzing and improving content-based chunking algorithms," Hewlett-Packard Labs Technical Report TR, vol. 30, no. 2005, 2005.

[11] W. Tian, R. Li, Z. Xu, and W. Xiao, "Does the content defined chunking really solve the local boundary shift problem?" in 36th IEEE International Performance Computing and Communications Conference, IPCCC 2017, San Diego, CA, USA, December 10-12, 2017. IEEE Computer Society, 2017, pp. 1–8.

[12] W. Xia, Y. Zhou, H. Jiang, D. Feng, Y. Hua, Y. Hu, Q. Liu, and Y. Zhang, "Fastcdc: a fast and efficient content-defined chunking approach for data deduplication," in 2016 USENIX Annual Technical Conference, USENIX ATC 2016, Denver, CO, USA, June 22-24, 2016, A. Gulati and H. Weatherspoon, Eds. USENIX Association, 2016, pp. 101–114.

[13] P. Shilane, G. Wallace, M. Huang, and W. Hsu, "Delta compressed and deduplicated storage using stream-informed locality," in 4th USENIX Workshop on Hot Topics in Storage and File Systems, HotStorage'12, Boston, MA, USA, June 13-14, 2012, R. Rangaswami, Ed. USENIX Association, 2012.

[14] Y. Zhang, W. Xia, D. Feng, H. Jiang, Y. Hua, and Q. Wang, "Finesse: Fine-grained feature locality based fast resemblance detection for postdeduplication delta compression," in 17th USENIX Conference on File and Storage Technologies, FAST 2019, Boston, MA, February 25-28, 2019, A. Merchant and H. Weatherspoon, Eds. USENIX Association, 2019, pp. 121–128.

[15] M. Rabin, Fingerprinting by Random Polynomials, ser. Center for Research in Computing Technology: Center for Research in Computing Technology. Center for Research in Computing Techn., Aiken Computation Laboratory, Univ., 1981.

[16] P. Shilane, M. Huang, G. Wallace, and W. Hsu, "WAN optimized replication of backup datasets using stream-informed delta compression," in Proceedings of the 10th USENIX conference on File and Storage Technologies, FAST 2012, San Jose, CA, USA, February 14-17, 2012, W. J. Bolosky and J. Flinn, Eds. USENIX Association, 2012, p. 5.

[17] L. Aronovich, R. Asher, E. Bachmat, H. Bitner, M. Hirsch, and S. T. Klein, "The design of a similarity based deduplication system," in Proceedings of of SYSTOR 2009: The Israeli Experimental Systems Conference 2009, Haifa, Israel, May 4-6, 2009, ser. ACM International Conference Proceeding Series, M. Allalouf, M. Factor, and D. G. Feitelson, Eds. ACM, 2009, p. 6.

[18] L. Xu, A. Pavlo, S. Sengupta, and G. R. Ganger, "Online deduplication for databases," in Proceedings of the 2017 ACM International Conference on Management of Data, SIGMOD Conference 2017, Chicago, IL, USA, May 14-19, 2017, S. Salihoglu, W. Zhou, R. Chirkova, J. Yang, and D. Suciu, Eds. ACM, 2017, pp. 1355–1368.

[19] F. Douglis and A. Iyengar, “Application-specific delta-encoding via resemblance detection,” in Proceedings of the General Track: 2003 USENIX Annual Technical Conference, June 9-14, 2003, San Antonio, Texas, USA. USENIX, 2003, pp. 113–126.

[20] G. Forman, K. Eshghi, and S. Chiocchetti, “Finding similar files in large document repositories,” in Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, Illinois, USA, August 21-24, 2005, R. Grossman, R. J. Bayardo, and K. P. Bennett, Eds. ACM, 2005, pp. 394–400.

[21] E. Kruus, C. Ungureanu, and C. Dubnicki, “Bimodal content defined chunking for backup streams,” in 8th USENIX Conference on File and Storage Technologies, San Jose, CA, USA, February 23-26, 2010, R. C. Burns and K. Keeton, Eds. USENIX, 2010, pp. 239–252.

[22] Abbes, H., Louati, T., Ce´rin, C.: Dynamic replication factor model for Linux containers-based cloud systems. J. Supercomput. 76, 7219 (2020)

[23] Ao, W.C., Psounis, K.: Resource-constrained replication strategies for hierarchical and heterogeneous tasks. IEEE Trans. Parallel Distrib. Syst. 31(4), 793–804 (2019)

[24] Boru, D., Kliazovich, D., Granelli, F., Bouvry, P., Zomaya, A.Y.: Models for efficient data replication in cloud computing datacenters. In: 2015 IEEE international conference on communications (ICC) (pp. 6056–6061). IEEE (2015)

[25] Zou, X., Pan, J., Du, W., Chen, S.: Elastic database replication in the cloud. In: International conference on algorithms and architectures for parallel processing (pp. 667–681). Springer, Cham (2015)

[26] Alghamdi, M., Tang, B., Chen, Y.: Profit-based file replication in data intensive cloud data centers. In: 2017 IEEE International Conference on Communications (ICC) (pp. 1–7). IEEE (2017)

[27] Armknecht, F., Barman, L., Bohli, J.M., Karame, G.O.: Mirror: enabling proofs of data replication and retrievability in the cloud. In: 25th {USENIX} security symposium ({USENIX} security 16) (pp. 1051–1068) (2016)

[28] Nivetha, N.K., Vijayakumar, D.: Modeling fuzzy based replication strategy to improve data availabiity in cloud datacenter. In: 2016 international conference on computing technologies and intelligent data engineering (ICCTIDE’16) (pp. 1–6). IEEE (2016)

[29] Jayalakshmi, D.S., Ranjana, T.R., Ramaswamy, S.: Dynamic data replication across geo-distributed cloud data centres. In: International conference on distributed computing and internet technology (pp. 182–187). Springer, Cham (2016)

[30] Jayalakshmi, D.S., TP, RR, Srinivasan, R.:, Dynamic data replication strategy in cloud environments. In: 2015 fifth international conference on advances in computing and communications (ICACC) (pp. 102–105). IEEE (2015)

[31] Rajalakshmi, A., Vijayakumar, D., Srinivasagan, K.G.: An improved dynamic data replica selection and placement in cloud. In 2014 international conference on recent trends in information technology (pp. 1–6). IEEE (2014)

[32] Kumar, K.A., Quamar, A., Deshpande, A., Khuller, S.: SWORD: workload-aware data placement and replica selection for cloud data management systems. VLDB J. 23(6), 845–870 (2014)

[33] Zhang, T.: A QoS-enhanced data replication service in virtualised cloud environments. Int. J. Netw. Virtual Organ. 22(1), 1–16 (2020)

[34] Mansouri, N.: Adaptive data replication strategy in cloud computing for performance improvement. Front. Comp. Sci. 10(5), 925–935 (2016) 104. Huang, Y., Huang, J., Liu, C., Zhang, C.: PFPMine: a parallel approach for discovering interacting data entities in data-intensive cloud workflows. Futur. Gener. Comput. Syst. 113, 474–487 (2020)

[35] Atrey, A., Van Seghbroeck, G., Mora, H., De Turck, F., Volckaert, B.: SpeCH: a scalable framework for data placement of data-intensive services in geo-distributed clouds. J. Netw. Comput. Appl. 142, 1–14 (2019)
[36] da Silva, G.H.G., Holanda, M., Araujo, A.: Data replication policy in a cloud computing environment. In: 2016 11th Iberian conference on information systems and technologies (CISTI) (pp. 1–6). IEEE (2016)
[37] Janpet, J., Wen, Y.F.: Reliable and available data replication planning for cloud storage. In: 2013 IEEE 27th international conference on advanced information networking and applications (AINA) (pp. 772–779). IEEE (2013)
[38] Lin, J.W., Chen, C.H., Chang, J.M.: QoS-aware data replication for data-intensive applications in cloud computing systems. IEEE Trans. Cloud Comput. 1(1), 101–115 (2013)
[39] Fu, S., He, L., Liao, X., Huang, C.: Developing the Cloudintegrated data replication framework in decentralized online social networks. J. Comput. Syst. Sci. 82(1), 113–129 (2016)
[40] Cidon, A., Stutsman, R., Rumble, S., Katti, S., Ousterhout, J., Rosenblum, M.: MinCopysets: derandomizing replication in cloud storage. In: Proc. 10th USENIX Symp. NSDI (pp. 1–5) (2013)
[41] Xie, F., Yan, J., Shen, J.: Towards cost reduction in cloud-based workflow management through data replication. In: 2017 fifth international conference on advanced cloud and big data (CBD) (pp. 94–99). IEEE (2017)
[42] Djebbar, E.I., Belalem, G.: Optimization of tasks scheduling by an efficacy data placement and replication in cloud computing. In: International conference on algorithms and architectures for parallel processing (pp. 22–29). Springer, Cham (2013)
[43] Ibrahim, I.A., Dai, W., Bassiouni, M.: Intelligent data placement mechanism for replicas distribution in cloud storage systems. In: 2016 IEEE international conference on smart cloud (smartcloud) (pp. 134–139). IEEE (2016)
[44] Dai, W., Ibrahim, I., Bassiouni, M.: A new replica placement policy for hadoop distributed file system. In: 2016 IEEE 2nd international conference on big data security on cloud (bigdatasecurity), IEEE international conference on high performance and smart computing (HPSC), and IEEE international conference on intelligent data and security (IDS) (pp. 262–267). IEEE (2016)
[45] Zhang, H., Lin, B., Liu, Z., Guo, W.: Data replication placement strategy based on bidding mode for cloud storage cluster. In: 2014 11th web information system and application conference (pp. 207–212). IEEE (2014)
[46] Khalajzadeh, H., Yuan, D., Grundy, J., Yang, Y.: Improving cloud-based online social network data placement and replication. In: 2016 IEEE 9th international conference on cloud computing (CLOUD) (pp. 678–685). IEEE (2016)
[47] Li, C., Wang, Y., Tang, H., Luo, Y.: Dynamic multi-objective optimized replica placement and migration strategies for SaaS applications in edge cloud. Futur. Gener. Comput. Syst. 100, 921–937 (2019)
[48] Bacis, E., di Vimercati, S.D.C., Foresti, S., Paraboschi, S., Rosa, M., Samarati, P.: Dynamic allocation for resource protection in decentralized cloud storage. In: 2019 IEEE global communications conference (GLOBECOM) (pp. 1–6). IEEE (2019)
[49] Khalili Azimi, S.: A Bee Colony (Beehive) based approach for data replication in cloud environments. In: Fundamental research in electrical engineering: the selected papers of the first international conference on fundamental research in electrical engineering (pp. 1039–1052). Springer Singapore (2019)
[50] Long, S.Q., Zhao, Y.L., Chen, W.: MORM: a multi-objective optimized replication management strategy for cloud storage cluster. J. Syst. Architect. 60(2), 234–244 (2014)

[51] Dabas, C., Aggarwal, J.: Delayed replication algorithm with dynamic threshold for cloud datacenters. In: Applications of computing, automation and wireless systems in electrical engineering (pp. 625–637). Springer, Singapore (2019)

[52] Li, C., Wang, Y., Chen, Y., Luo, Y.: Energy-efficient faulttolerant replica management policy with deadline and budget constraints in edge-cloud environment. J. Netw. Comput. Appl. 143, 152–166 (2019)

[53] Mark Lantz. Tape storage mounts a comeback. IEEE Spectrum, 55(9):32–37, Sep. 2018.

[54] David Reinsel, John Gantz, and John Rydning. The digitization of the world: From edge to core. https://www.seagate.com/files/www-content/our-story/trends/files/idcseagate-dataage-whitepaper.pdf, Nov 2018.

[55] J. C. Yang, C. F. Ma, J. B. Man, H. F. Xu, G. Zheng, and H. B. Song, Cache-enabled in cooperative cognitive radio networks for transmission performance, Tsinghua Science and Technology, vol. 25, no. 1, pp. 1–11, 2020.

[56] M. M. Rathore, A. Ahmad, A. Paul, and G. Jeon, Efficient graph-oriented smart transportation using internet of things generated big data, in Proc. 11th Int. Conf. Signal-Image Technology and Internet-Based Systems (SITIS), Bangkok, Thailand, 2015, pp. 512–519.

[57] D. Nallaperuma, R. Nawaratne, T. Bandaragoda, A. Adikari, S. Nguyen, T. Kempitiya, D. De Silva, D. Alahakoon, and D. Pothuhera, Online incremental machine learning platform for big data-driven smart traffic management, IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 12, pp. 4679–4690, 2019.

[58] J. Li, M. Siddula, X. Z. Cheng, W. Cheng, Z. Tian, and Y. S. Li, Approximate data aggregation in sensor equipped IoT networks, Tsinghua Science and Technology, vol. 25, no. 1, pp. 44–55, 2020.

[59] D. Kim, J. Son, D. Seo, Y. Kim, H. Kim, and J. T. Seo, A novel transparent and auditable fog-assisted cloud storage with compensation mechanism, Tsinghua Science and Technology, vol. 25, no. 1, pp. 28–43, 2020.

[60] P. Shilane, G. Wallace, M. Huang, and W. Hsu, "Delta compressed and deduplicated storage using stream-informed locality," in 4th USENIX Workshop on Hot Topics in Storage and File Systems, HotStorage'12, Boston, MA, USA, June 13-14, 2012, R. Rangaswami, Ed. USENIX Association, 2012.

[61] P. Shilane, M. Huang, G. Wallace, and W. Hsu, "WAN optimized replication of backup datasets using stream-informed delta compression," in Proceedings of the 10th USENIX conference on File and Storage Technologies, FAST 2012, San Jose, CA, USA, February 14-17, 2012, W. J. Bolosky and J. Flinn, Eds. USENIX Association, 2012, p. 5.

[62] L. Aronovich, R. Asher, E. Bachmat, H. Bitner, M. Hirsch, and S. T. Klein, "The design of a similarity based deduplication system," in Proceedings of of SYSTOR 2009: The Israeli Experimental Systems Conference 2009, Haifa, Israel, May 4-6, 2009, ser. ACM International Conference Proceeding Series, M. Allalouf, M. Factor, and D. G. Feitelson, Eds. ACM, 2009, p. 6.

[63] L. Xu, A. Pavlo, S. Sengupta, and G. R. Ganger, "Online deduplication for databases," in Proceedings of the 2017 ACM International Conference on Management of Data, SIGMOD Conference 2017, Chicago, IL, USA, May 14-19, 2017, S. Salihoglu, W. Zhou, R. Chirkova, J. Yang, and D. Suciu, Eds. ACM, 2017, pp. 1355–1368.

[64] F. Douglis and A. Iyengar, "Application-specific delta-encoding via resemblance detection," in Proceedings of the General Track: 2003 USENIX Annual Technical Conference, June 9-14, 2003, San Antonio, Texas, USA. USENIX, 2003, pp. 113–126.

[65] G. Forman, K. Eshghi, and S. Chiocchetti, "Finding similar files in large document repositories," in Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge

Discovery and Data Mining, Chicago, Illinois, USA, August 21-24, 2005, R. Grossman, R. J. Bayardo, and K. P. Bennett, Eds. ACM, 2005, pp. 394–400.

[66] P. Shilane, G. Wallace, M. Huang, and W. Hsu, “Delta compressed and deduplicated storage using stream-informed locality,” in 4th USENIX Workshop on Hot Topics in Storage and File Systems, HotStorage’12, Boston, MA, USA, June 13-14, 2012, R. Rangaswami, Ed. USENIX Association, 2012.

[67] Y. Zhang, W. Xia, D. Feng, H. Jiang, Y. Hua, and Q. Wang, “Finesse: Fine-grained feature locality based fast resemblance detection for postdeduplication delta compression,” in 17th USENIX Conference on File and Storage Technologies, FAST 2019, Boston, MA, February 25-28, 2019, A. Merchant and H. Weatherspoon, Eds. USENIX Association, 2019, pp. 121–128.

[68] M. Rabin, Fingerprinting by Random Polynomials, ser. Center for Research in Computing Technology: Center for Research in Computing Technology. Center for Research in Computing Techn., Aiken Computation Laboratory, Univ., 1981.

[69] E. Kruus, C. Ungureanu, and C. Dubnicki, “Bimodal content defined chunking for backup streams,” in 8th USENIX Conference on File and Storage Technologies, San Jose, CA, USA, February 23-26, 2010, R. C. Burns and K. Keeton, Eds. USENIX, 2010, pp. 239–252.

[70] S. S. Gill, I. Chana, M. Singh, and R. Buyya, “CHOPPER: an intelligent QoS-aware autonomic resource management approach for cloud computing,” Cluster Comput., pp. 1–39, 2017.22 Preprint submitted to Internet of Things

[71] S. Singh, I. Chana, M. Singh, and R. Buyya, “SOCCER: self-optimization of energy-efficient cloud resources,” Cluster Comput., vol. 19, no. 4, pp. 1787–1800, 2016.

[72] S. S. Gill et al., “Transformative effects of IoT, Blockchain and Artificial Intelligence on cloud computing: Evolution, vision, trends and open challenges,” Internet of Things, vol. 8, p. 100118, 2019.

[73] A. Jonathan, M. Uluyol, A. Chandra, J. Weissman, Ensuring reliability in geodistributed edge cloud, in: Resilience Week (RWS), 2017, pp. 127–132, http://dx.doi.org/10.1109/RWEEK.2017.8088660.

[74] Y. Harchol, A. Mushtaq, J. McCauley, A. Panda, S. Shenker, CESSNA: Resilient edge-computing, in: MECOMM’18, Association for Computing Machinery, New York, NY, USA, 2018, pp. 1–6, http://dx.doi.org/10.1145/3229556.3229558.

[75] T. Jeong, J. Chung, J.W. Hong, S. Ha, Towards a distributed computing framework for fog, in: IEEE Fog World Congress (FWC), 2017, pp. 1–6, http://dx.doi.org/10.1109/FWC.2017.8368528.

[76] M. Al-khafajiy, T. Baker, A. Waraich, D. Al-Jumeily, A. Hussain, IoT-Fog optimal workload via fog offloading, in: IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion), 2018, pp. 359–364, http: //dx.doi.org/10.1109/UCC-Companion.2018.00081.

[77] K. Fall, A delay-tolerant network architecture for challenged internets, in: Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, in: SIGCOMM ’03, Association for Computing Machinery, New

[78] York, NY, USA, 2003, pp. 27–34, http://dx.doi.org/10.1145/863955.863960. [38] J.E. Luzuriaga, M. Zennaro, J.C. Cano, C. Calafate, P. Manzoni, A disruption tolerant architecture based on MQTT for IoT applications, in: 14th IEEE Annual Consumer Communications Networking Conference (CCNC), 2017, pp. 71–76, http://dx.doi.org/10.1109/CCNC.2017.7983084.

[79] C. Kulatunga, L. Shalloo, W. Donnelly, E. Robson, S. Ivanov, Opportunistic wireless networking for smart dairy farming, 19, (2) 2017, pp. 16–23, http: //dx.doi.org/10.1109/MITP.2017.28,

[80] G. Castellano, F. Risso, R. Loti, Fog computing over challenged networks: A real case evaluation, in: IEEE 7th International Conference on Cloud Networking (CloudNet), 2018, pp. 1–7, http://dx.doi.org/10.1109/CloudNet.2018.8549354.

[81] J. Moura, D. Hutchison, Fog computing systems: State of the art, research issues and future trends, with a focus on resilience, J. Netw. Comput. Appl. 169 (2020) 102784, http://dx.doi.org/10.1016/j.jnca.2020.102784.

[82] S. Chandak, K. Tatwawadi, C. Wen, L. Wang, J. Aparicio Ojea, T. Weissman, Lfzip: Lossy compression of multivariate floating-point time series data via improved prediction, in: Data Compression Conference (DCC), 2020, pp. 342–351, http://dx.doi.org/10.1109/DCC47342.2020.00042.

[83] D. Blalock, S. Madden, J. Guttag, Sprintz: Time Series Compression for the Internet of Things, Vol. 2, New York, NY, USA, 2018, http://dx.doi.org/10.1145/3264903.

[84] Fu, X. Qiu, J. Wang, F2mc: Enhancing data storage services with fogtomulticloud hybrid computing, in: IEEE 38th International Performance Computing and Communications Conference (IPCCC), 2019, pp. 1–6, http://dx.doi.org/10.1109/IPCCC47392.2019.8958748.

[85] A. Sinaeepourfard, J. Garcia, X. Masip-Bruin, E. Marin-Tordera, A novel architecture for efficient fog to cloud data management in smart cities, in: IEEE 37th International Conference on Distributed Computing Systems (ICDCS), 2017, pp. 2622–2623, http://dx.doi.org/10.1109/ICDCS.2017.202.

[86] Y. Guan, J. Shao, G. Wei, M. Xie, Data security and privacy in fog computing, IEEE Netw. 32 (5) (2018) 106–111.

[87] Salim Bitam, Sherali Zeadally, Abdelhamid Mellouk, Fog computing job scheduling optimization based on bees swarm, Enterp. Inf. Syst. 12 (4) (2018) 373–397.

[88] Sourav Kunal, Arijit Saha, Ruhul Amin, An overview of cloud-fog computing: Architectures, applications with security challenges, Secur. Priv. 2 (4) (2019) e72.

[89] YuXin Liu, Anfeng Liu, Shuang Guo, Zhetao Li, Young-June Choi, Hiroo Sekiya, Context-aware collect data with energy efficient in cyber–physical cloud systems, Future Gener. Comput. Syst. (2017).

[90] Chao Qu, Ming Tao, Ruifen Yuan, A hypergraph-based blockchain model and application in internet of things-enabled smart homes, Sensors 18 (9) (2018) 2784.

[91] D. He, N. Kumar, S. Zeadally, A. Vinel, L.T. Yang, Efficient and privacypreserving data aggregation scheme for smart grid against internal adversaries, IEEE Trans. Smart Grid 8 (5) (2017) 2411–2419.

[92] Debiao He, Neeraj Kumar, Jong-Hyouk Lee, Privacy-preserving data aggregation scheme against internal attackers in smart grids, Wirel. Netw. 22 (2) (2016) 491–502.

[93] N. Kumar, A.V. Vasilakos, J.J.P.C. Rodrigues, A multi-tenant cloud-based DC nano grid for self-sustained smart buildings in smart cities, IEEE Commun. Mag. 55 (3) (2017) 14–21.

[94] P.K. Sharma, N. Kumar, J.H. Park, Blockchain-based distributed framework for automotive industry in a smart city, IEEE Trans. Ind. Inf. 15 (7) (2019) 4197–4205.

[95] S. Garg, A. Singh, S. Batra, N. Kumar, L.T. Yang, UAV-Empowered edge computing environment for cyber-threat detection in smart vehicles, IEEE Netw. 32 (3) (2018) 42–51.

[96] K. Kaur, S. Garg, G.S. Aujla, N. Kumar, J.J.P.C. Rodrigues, M. Guizani, Edge computing in the industrial internet of things environment: Softwaredefined-networks-based edge-cloud interplay, IEEE Commun. Mag. 56 (2) (2018) 44–51.

[97] Tomoya Enokido, Dilawaer Doulikun, Makoto Takizawa, An energy-aware load balancing algorithm to perform computation type application processes in a cluster of servers, Int. J. Web Grid Serv. 13 (2) (2017) 145–169.

[98] Ziluan Liu, Jiangsheng Li, Yanru Wang, Xin Li, Shanzhi Chen, HGL: A hybrid global-local load balancing routing scheme for the Internet of Things through satellite networks, Int. J. Distrib. Sens. Netw. 13 (3) (2017) 1550147717692586.

[99] Tiago Rogerio Mück, Zana Ghaderi, Nikil D. Dutt, Eli Bozorgzadeh, Exploiting heterogeneity for aging-aware load balancing in mobile platforms, IEEE Trans. Multi-Scale Comput. Syst. 3 (1) (2016) 25–35.

[100] Fan Jiang, Yao Liu, Benchao Wang, Xianchao Wang, A relay-aided device-todevice-based load balancing scheme for multitier heterogeneous networks, IEEE Internet Things J. 4 (5) (2017) 1537–1551.

[101] Simar Preet Singh, Anand Nayyar, Rajesh Kumar, Anju Sharma, Fog computing: from architecture to edge computing and big data processing, J. Supercomput. 75 (4) (2019) 2070–2105.

[102] J. Fu, Y. Liu, H. Chao, B.K. Bhargava, Z. Zhang, Secure data storage and searching for industrial IoT by integrating fog computing and cloud computing, IEEE Trans. Ind. Inf. 14 (10) (2018) 4519–4528.

[103] Rajeev Kumar Bedi, Jaswinder Singh, Sunil Kumar Gupta, MWC: an efficient and secure multi-cloud storage approach to leverage augmentation of multi-cloud storage services on mobile devices using fog computing, J. Supercomput. 75 (6) (2019) 3264–3287.

[104] Y. Shu, M. Dong, K. Ota, J. Wu, S. Liao, Binary reed-solomon coding based distributed storage scheme in information-centric fog networks, in: 2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks, CAMAD, 2018, pp. 1–5.

[105] Kanza Hassan, Nadeem Javaid, Farkhanda Zafar, Saniah Rehman, Maheen Zahid, Sadia Rasheed, A cloud fog based framework for efficient resource allocation using firefly algorithm, in: International Conference on Broadband and Wireless Computing, Communication and Applications, Springer, 2018, pp. 431–443.

[106] 123Ajay Kattepur, Harshit Dohare, Visali Mushunuri, Hemant Kumar Rath, Anantha Simha, Resource Constrained Offloading in Fog Computing, Association for Computing Machinery, New York, NY, USA, 2016.

[107] 124 Zhitao Guan, Yue Zhang, Longfei Wu, Jun Wu, Jing Li, Yinglong Ma, Jingjing Hu, Appa: an anonymous and privacy preserving data aggregation scheme for fog-enhanced iot, 125 (2019) 82–92.

[108] 125 Survey on fog computing: architecture, key technologies, applications and open issues, J. Netw. Comput. Appl. 98 (2017) 27–42.

[109] 126 Takayuki Nishio, Ryoichi Shinkuma, Tatsuro Takahashi, Narayan B. Mandayam, Service-oriented heterogeneous resource sharing for optimizing service latency in mobile cloud, in: Proceedings of the First International Workshop on Mobile Cloud Computing & Networking, Association for Computing Machinery, 2013, pp. 19–26.

[110] 127 S. Xie, Z. Chen, Anomaly detection and redundancy elimination of big sensor data in internet of things, arXiv:1703.03225(2017).

[111] 128 R. Priyadarshi, B. Gupta, A. Anurag, Deployment techniques in wireless sensor networks: a survey, classification, challenges, and future research issues, J. Supercomput. (2020) 1–41.

[112] 129 H. El Alami, A. Najid, Ech: an enhanced clustering hierarchy approach to maximize lifetime of wireless sensor networks, IEEE Access 7 (2019) 107142–107153.