Chapter 2
Literature Review

The Literature review is about finding the relevant documents in the form of paper, book chapter, and journal and conference articles. These articles are critically reviewed, analyzed and well understood to get insights of the existing algorithms and their technological background need. Here the literature review done is formulating three problem statements that is suitability and adaptability of Fog Computing in health care, Optimization in Fog Computing and the security, which are in detail addressed in this thesis. The first literature review is related to implementing Health-as-a-Service in Fog Computing and its comparative analysis with Cloud Computing. After the analysis of HaaS, it is found that the Fog Node lags in terms of computation. So to improve Fog Node further the Computation Optimization in Fog Node is focused. This study is done in section two of literature review. The second literature review done is to get optimizations in Fog Computing. The third section of this chapter analyzes different lightweight encryptions algorithms in IoT and Fog domain. And based on the analysis what are the needs of the new encryption scheme is shown.

2.1 Health Care in Fog Computing

In recent years, the development and dissemination of smart healthcare system have a considerable attention by the convergence of a variety of IoT devices. The previous researchers attempt to mitigate the network delay while accessing the centralized cloud environment. The work [16] focuses on improving the network performance by grouping the distributed cloud resources into micro data centers based on the network latency, which ensures tolerable network latency. IoT-based Physical Activity Monitoring (PAMIoT) framework [17] utilizes the cloud services to handle and identify the physical activity information of a human body. It employs IEEE 802.15.4 and Bluetooth technologies to measure the dissolved oxygen in the blood, ECG, and number of steps. The CloudIoT architecture [18] eases the IoT

service delivery by introducing the virtual vertical service delivery based IoT PaaS platform. Also, it presents the domain mediation to provide the solution for domain-specific control applications.

Later, Fog Computing introduces a novel way of moving the cloud infrastructure, proximity to the IoT devices, which provides more opportunities to the IoT [19]. Fog micro data center [20] dynamically manages the resources for IoT deployments, which is a service oriented resource management framework. It predicts the utilization of the resources by the users and pre-allocates the resources by exploiting the knowledge of user behavior and the probability of resource usage in future. Ubiquitous Data Accessing method (UDA-IoT) in the IoT systems effectively provides the response to the emergency medical services by ubiquitously acquiring and processing the medical data, which improves the accessibility of IoT data resources [21]. Healthcare services mostly depend on the network connectivity and hence, to tackle interruption or delay by the network, the work [22] integrates the IoT with the Fog Computing offering the cloud resources with effective network performance. Dynamic resource estimation and pricing model [23] co-locates the smart gateways to create Fog-based micro data centers for IoT, which ensures the efficient and effective resource management in the IoT systems. To resolve the latency issue, the work [24] presents a smart fog gateway with Fog Computing in which the architecture performs pre-processing and trimming process before transforming the cloud of things data into the cloud server. The ECG feature extraction model [25] enhances the traditional health monitoring system by smart fog gateway incorporating embedded data mining, notification service, and the distributed fog data storage methods at the edge of the network.

Smart e-health gateway [26] ubiquitously offers local storage, embedded data mining, and real-time local data processing by exploiting the strategic position of gateways, which deals with the scalability, reliability, and energy efficiency issues while considering the burden of sensor networks and remote data centre. The iFogSim model [27] extends the cloud services to the edge of the network and decreases the network congestion by enhancing the resource management

techniques which performs real-time analytics and identifies the optimal place of applications on the edge devices. Architecture in [28] presents an IoT service delegation and resource allocation based on the linearized decision tree to diminish the latency for delay-sensitive applications based on collaboration between the fog and cloud environment. An extended work of [26] presents in [29], in which the smart e-health fog gateway supports the interoperability, reliability, and fog-based mobility support in the healthcare IoT systems. Dynamic Fog model [30] provides a service for time-sensitive healthcare applications involving large-scale, geospatially distributed, and latency-sensitive applications. It analyzes the most time-sensitive data of the Heart rate medical data to evaluate the performance of the dynamic fog model. Distributed analytics and edge intelligence model [31] explores the Fog Computing for pervasive health monitoring applications in terms of real-time fall detection.

2.1.1 Research Gap

- Pervasive Health care applications generate vast amount of data, some intelligence needed to filter this data
- Effect on QoS after trimming and pre-processing the data coming from heterogeneous applications
- Simulation based SLA-aware data flow placements and resource scheduling is done but not in real time
- Need of real-time fog-based medical data analysis
- Cloud Computing services can be extended for health care applications
- Smart fog gateway is missing to filter the health care data
- Dynamic processing on the go is needed for Health care applications

2.2 Optimization in Fog Computing

Discussed here is the literature review of systems which have utilized or optimized systems relevant to Distributed Computing with respect to performance. The utilization and improvements within the system on account of response time, memory, number of cores, and CPU usage are highlighted as well. Further, in terms

of performance improvement, it targets work relevant to Fog Computing. These papers have been studied to verify that similar parameter optimization is possible in Fog Computing or not.

Concerns encountered while scheduling jobs parallel on clusters of distributed processors have been taken into consideration by [32]. Routing schemes of 2 kinds and scheduling techniques of 3 types have been considered. To evaluate the performance of these task scheduling algorithms for every one of the routing scenario is the purpose of this paper. Discussed and tabulated in this paper are the numerous other system parameters which required to be accounted for the purpose of task scheduling and job submission. This paper evaluates the feasibility of every scheduling algorithm in every case of routing along with the effect of system parameters on the job of task scheduling. The algorithms' performances in various load conditions of the system are evaluated via simulation. The effect of the various scheduling policies on the performance of the system can be understood by evaluating simulation.

A system that targets to enhance the functioning of a parallel multifrontal solver, MUMPS has been suggested by [33] and this approach to scheduling is based on memory. The slaves and/or associates of a processor are selected on the basis of memory constraints On the basis of their memory availability, the slaves are selected. It targets to minimise the utilized stack size at run time. A paper that attempts to improve the effective utilization of global memory has been suggested in [34]. Strategies for distribution of jobs are constructed appropriately. Additional load migrates to associates with availability for sufficient memory when a node does not have requisite memory to accept jobs. Page faults occur on account of unbalanced memory allocations and thus, the aim is reduction of the same to improve efficiency. The performance of memory-bound jobs is enhanced by the suggested policy for load sharing. A paper with proper description of distributed systems is suggested in [35]. For ensuring complete utilization of computational capacity, it is imperative to have a fair load-sharing policy in distributed systems, as this has a significant effect on performance. A major role in performance is also played by the memory of the system and thus, the available memory becomes the

basis of the load sharing systems. Memory-based load sharing system has better performance in the case of jobs that are memory-bound. Better performance is exhibited by the algorithm created here in load sharing systems as compared to FCFS and Round Robin algorithms. With respect to performance, memory-based load sharing systems adapt more and are sensitive towards the variance of memory. A memory-based hybrid Dragonfly algorithm is demonstrated in [36] for optimization.

A two-phased scheduling algorithm, H2GS has been suggested by [37] paper. With primary focus on heterogeneous systems, it functions on distributed systems. A heuristic list-based algorithm constitutes the first phase, utilizing which a greatly efficient schedule is produced. Shorter schedules are evolved in the second phase. Priorities are allotted to tasks that have to be scheduled. The task with highest priority and readiness is chosen for the purpose of scheduling. A processor is chosen in the subsequent phase. To reduce the execution time, a task is chosen and submitted to the processor. In the paper presented in [38], an algorithm for scheduling is implemented on processors whose numbers are predefined. It aims to improve performance and perform efficient scheduling both. The algorithm is called Heterogeneous Earliest Finish Time (HEFT). The highest upward rank value is selected and allotted to a processor at every step. The earliest time of completion is reduced on the basis of the insertion-based approach. This sturdy algorithm performs well with a broad array of graph structures.

Numerous techniques for scheduling/co-scheduling techniques utilized in distributed systems have been demonstrated in [39]. The 2 kinds of local scheduling presented here are Proportional-sharing scheduling and Predictive scheduling. For automatic environmental changes and adoption new architectures Predictive scheduling provides intelligence, adaptivity, and proactivity. New algorithms, architectures, and methods embedded in the system are studied by it. Prior inputs which exist in the form of a vector of performance information (CPU usage) are accumulated by the allocator into sets. Every set then corresponds to a scheduling decision. To maintain a limited demand for memory, the allocator divides or merges the sets. An algorithm for scheduling in the environment of Cloud

Computing is demonstrated by paper [40]. To determine the node on which job scheduling should take place, parameters such as processor status are used in this algorithm. The aim is to get an efficient scheduling method which would reduce the total processing time of all the loads by dividing them between all the available processors. The processor for assigning the present job is determined with the help of a formula which considers the link time and the processing power along with the history of scheduling.

An article concerning the numerous scheduling algorithms utilized in Distributed Computing in Cloud Computing is presented in [41]. The integration of virtualized systems is the noticeable distinction between cloud-based computing and Distributed Computing. For reinforcing the server, Virtual Machine (VM) allocation is carried out. In this case, the scheduler pool is designated as the software which is taken into account for scheduling. Hardware requirements (number of cores in the system along with their relevant statistics of utilization, etc.) are utilized for the purpose of scheduling by the scheduler. Varied gang scheduling algorithms have been examined in. [42] and their corresponding efficiencies for clusters constituting multi-core systems are analyzed. The scheduling of gangs in multi-core cluster systems occurs via the proposed migration structure. Outcomes regarding the performance of the system are provided using an evaluation model. Fragmentation occurs in gang scheduling when the gangs do not fit in idle cores owing to their sizes. By dynamic migration of parallel jobs, adjustable and flexible schedules are created. As the load balancing requirement is satisfied by migrations, due importance is given to them.

The main focus in [43] paper has been the development of a job scheduling task for mobile users in Fog Computing. The Bees Swarm algorithm along with the total amount of memory and CPU execution time is utilized by them. For an essential aspect, which is saving network bandwidth in communication, in [44] authors have proposed compression of raw data followed by resending it, and its subsequent decompression for the purpose of processing on the receiver end. While such compression and decompression techniques save the network bandwidth, they increment the time of response in health care which is extremely crucial.

2.2.1 Research Gap

- To extend the Distributed Computing in the Fog Computing to support latency-sensitive IoT based health care applications
- To design the smart Fog Computing cluster with smart job allocation to satisfy the Service Level Agreements (SLAs) in terms of ensuring the optimal response time and resource utilization
- To develop an algorithm, to dynamically decide node capability, that enables to identify the node that can be assigned the appropriate task for health care applications in Fog Computing
- To develop an optimized Fog Computing based performance model in the health care domain to serve the community better

2.3 Need of Lightweight encryption and multi-level Fog scenario

The authors in [45] presented the problems that exist with cloud and IoT amalgamation and differentiated the concepts of cloud and Fog Computing. The authors present how Fog Computing accomplish using parameters such as delay in processing, the cost required and the consumption of power. A universal architecture is presented which consists of numerous IoT systems linked to the Fog Node which is then connected to the Cloud. It helps to illustrate how the Fog Node is positioned. The main purpose of Fog Computing is to supply resources to the nodes lying at the lowest possible layer. These nodes are important as they generate data. The presence of Fog in the architecture helps to create services that are more refined as they lie close to the IoT devices [46]. The architecture consists of multiple IoT devices connected to the Fog Node. These IoT devices include healthcare related services, smart home devices and devices in a factory. Fog Nodes play an important role by adding more layers between the IoT devices and the Cloud. Hence, they help in several applications such as processing and analysis of data and providing security for data which is more sensitive and needs to be secured. The performance measures help them to evaluate several parameters. It shows several factors such as low processing delay, low latency and more security.

Although the authors discussed about the importance of Fog Computing there are several challenges that still need to be addressed. Horizontal and vertical scaling should be implemented between underlying nodes and Fog Nodes and data can be sent to the cloud for value added services. The Fog Node should be capable of processing what kind of tasks are to be processed locally and what needs to be send to the cloud so that the Fog Node can help in managing important requests and optimum resource utilization is obtained. The architecture consists of a single Fog Node. A single Fog Node may not be capable of carrying out tasks for efficient functioning of the system as the number of users may increase and the Node will not be capable of forwarding all the user requests directly to the Cloud. Some other challenges include support for multiple Fog Nodes in a single Scenario such that there are multiple levels and the security of user's information and private data is ensured at different levels. Hence, we have implemented a Fog Computing architecture which tries to overcome such problems and consists of multiple nodes which are able to communicate with each other with the help of horizontal and vertical communication and encryption complexity increases as the level increases.

The authors in [47] illustrated the importance of Fog Computing and it be used as a gate-way between Cloud Computing and the Internet of Things. They examine the superiority of Fog Computing over Cloud Computing which comprise of numerous advantages such as upgraded awareness of the locality, strengthened quality of services to the mobile device users and competence to the network. The presence of smart devices on the boundary of the network helps the Fog in acting as supplement to Cloud Computing.

The authors propose an overview of different applications of the IoT with the use of Fog Computing. Four different scenarios have been discussed to show the implementation. In case of an Office, it is shown as an example of a general relation. However, in the case of a Factory which is smart, it is shown as an application of Internet of Things which is industrial. The authors give an example of real-life scenario which involves the Traffic which is smart. They show that data can be collected and analyzed at the Fog Nodes instead of communicating with the cloud server. Furthermore, in case of unavoidable circumstances such as an ambulance

stuck in heavy traffic can be given way to ensure that there is no further delay. This takes place with the help of numerous IoT devices.

The general idea to propose different applications with different scenarios is to ensure that there are numerous devices to work in unison so that there is minimal processing delay. A response is principal to deal with problems stated above. The scenarios mentioned above have multiple IoT devices that are linked to a Fog Node. The Fog Node is then connected to the cloud sever. Hence the scenario does not consist of multiple Fog Nodes to ensure security, scalability, low latency, vertical communication which have been implemented in the proposed architecture. The authors also present different security implications of Fog Computing with IoT devices which include privacy, confidentiality and authentication. Hence, we have proposed lightweight algorithms to deal with security and privacy.

The authors in [48] gave an outline of the concepts of Fog Computing and the Internet of Things. They talked through about various advantages of Fog Computing and how it is proficient of supporting many IoT applications to provide superior services to users. The challenges faced by the IoT devices such as scalability, complexity, dynamicity, heterogeneity, latency and security that needs to be overcome in order to have a successful development of fog architecture. The fog environment involves numerous Fog devices thus the computation is distributed and can be energy efficient as compared to the centralized cloud model of computation. Furthermore, the authors addressed several issues of using Fog Nodes with the IoT devices. This included communication of fog with the cloud, communication between fog severs, parallel computation. To ensure that there is high performance and low latency between the nodes is important and needs to be ensured so that there is a proper communication system between the Fog Node and the Cloud [49]. Our proposed architecture overcomes these limitations by using multiple Fog Nodes to ensure that computation is distributed and having different levels of encryption to preserve the confidentiality of the data.

The authors in [50] discussed about the importance of Fog and Edge Computing paradigms in order to the increase the data capacity which is being transmitted

over the network. They discussed about the influence of Fog Computing and Edge Computing on IIoT (Industrial Internet of Things). One application proposed by the paper was the Supply Chain Management of industries, where numerous operations are used in order to manage, observe and organize the movement of different items from their manufacturing center to their distribution hubs in a systematic way. The authors described about the logistics network of blackberry (fruit) as proposed in [51], to explain how companies can enlarge their logistics network systems by leveraging edge and Fog Computing. The nodes that are stationed at the farmlands can help in surveying of the blackberries. The data collected by the sensors is passed on to the Fog Nodes which can anticipate and inform the farmer about the harvesting of blackberries. Edge computing can also be used to process data of temperature, light and humidity which is captured using sensors (IoT devices). The authors also discussed about some common challenges faced by Fog and Edge Computing which needs to be addressed. With the use of Fog Computing we need to design optimized methods for Distributed Computing across nodes and ensuring low latency. This pliability and distribution of computing gives rise to various security concerns which needs to be handled.

The authors in [52] proposed an IoT-enabled Smart Home Architecture having 6 phases. A scenario of living room is considered where the sofa and chairs that are fitted with sensors are used to detect the existence of a person in the room. The data collected by the sensors will be associated with the control unit of TV so that it is turned off whenever there is no one in the living room for a certain period of time. Thus, Horizontal communication can take place between the sofa and chairs which sends the data to the Fog Node(IoT Gateway) whereas the Smart TV control unit can be switched off with the help of Vertical communication taking place between the IoT device and Fog Node.

Another scenario is mentioned where unusual incidents are used to identify specific incidents of scrutiny such as earthquake, tsunami, wildfires etc., or for observing the movements of aged people or people with persistent sickness at home. The function of the Fog Node is significant in the case of any abnormal

incident or outside the threshold reading/value. If an abnormal incident takes place then a notification is activated, and the appropriate individuals are informed.

The paper compared several legacy cryptographic algorithms like BlowFish, AES128-CBC, DES3, AES256-CTR etc., with lightweight algorithms like CLEFIA and TRIVIUM. The paper also discussed how Replay Attacks, DoS/DDoS Family Attacks and many more can be mitigated by using lightweight encryption algorithms to encrypt the data. Privacy is becoming a crucial constituent in the field of Internet of Things. Although IoT devices are mainly used to gather information about the daily routines of humans, surrounding conditions, etc., they are exposed to several security and privacy threats. The traditional privacy and security techniques are not appropriate for IoT devices due to their inadequate energy, storage size, and computing ability; thus, we need lightweight cryptographic algorithms to handle these limitations [53]. In [54] a composite encryption method is proposed which incorporates both symmetric and asymmetric lightweight encryption algorithms to set up a secure communication in fog-to-things computing. Here the user's message is encrypted using symmetric cipher and then the public key is encrypted using the asymmetric cryptographic cipher, which is then added along with the ciphertext that is transferred to the Fog Node. The proposed PRE (proxy re-encryption) scheme consists of five algorithms for System Setup, Key Generation, Data Owner Encryption, Fog Re-encryption and User Decryption. Although the author used lightweight cryptography for encryption of the user data the implementation of five algorithms is quite complex and it also does not guarantee the optimal usage of the system resources.

The authors in [55] discussed about the need of security and privacy of data in IoT devices in order to build confidence among users and use IoT technology at a large scale. As IOT devices remain unsupervised for a long time it is extremely open to attacks. Also due to wireless transmission of data, attacks like eavesdropping becomes quite easy to execute. Furthermore, IOT devices have low processing capabilities and limited memory. Thus, the execution of traditional encryption algorithms like AES, DES which are computationally costly will hamper the proper functioning of such constrained devices that has limited resources. The authors

therefore proposed a lightweight encryption algorithm SIT which is a block cipher algorithm having 64-bit cipher key and plain text. Here the cipher key is split into 4 bits and then initial substitution I performed on 4-bits to convert it to 16-bits. These 16-bits then operates on f-function which performs linear and non-linear transformations. The output of f-function is ordered in a 4X4 matrix and then converted into four arrays of 16-bits each to acquire the round keys (K1, K2, K3, K4). The fifth key is obtained by performing XOR on four round keys. For Encryption process, the plain text is divided into four parts each of 16-bits. Bitwise XNOR, XOR operations are being used for encryption purpose. Here also f-function is used which is like Key Generation process. Here the algorithm has 5 rounds for Encryption of Plain Text to Cipher Text. Although the SIT algorithm is developed to decrease the memory and power consumption and augment the speed of encryption, it requires to refine some operations by adding more complicated procedures in the design and should also increase the size of the key in order to enhance the privacy level by considering the utilization of memory and power simultaneously [56].

The author in [57] introduced a system architecture for Fog Computing with Device-to-Device communication support, which can be used to preserve the privacy of data in IoT devices. In traditional Fog Computing framework, when an Edge Device(ED) and Network Access Device (NAD) want to have a reliable connection, then for validating each other, they might require the help of Centralized Cloud Server (CCS), and that harms the working of the system by having an excessive delay. Thus, in order to have less delay and secure handover it is vital to have a Device-to-Device connection technology. Furthermore, due to the rise in the number of IoT devices NADs can be completely overburdened. Thus, to solve this problem NADs might require connecting with each other to discharge some of the computational overhead. Therefore, there should be a Fog Computing framework with collaborative Device-to-Device communications that allows EDs to connect with one another and even assist one another to verify without requiring CCS.

2.3.1 Research Gap

- IoT devices remain unsupervised for a longer duration of time, so they need the data and access security protocols.
- In specific scenarios, IoT-IoT devices also need to be communicated hence it should have data sharing security.
- By considering the computational capabilities of the IoT device, a lightweight encryption scheme is needed.
- Single Fog Nodes may lag in terms of computation and decision-making time, so we need multiple Fog Nodes at the same horizontal levels to support it.
- An efficient security scheme is needed for sharing data across multiple Fog Nodes at the same and at different levels to achieve parallel and Distributed Computing among the Fog Nodes to facilitate big data analytics.
- To reduce the cloud traffic, the multilevel Fog Node scenario is suggested, and also its security solutions are provided.
- For multiple levels, we need multiple security algorithms which makes the system more complex and vulnerable because every Algorithm is covering one or the other aspect of the security or computation.

2.4 Problem definition

Nowadays, IoT-driven healthcare applications play a vital role in the distributed environment. IoT constantly generates the massive amount of stream data, which leads the complexity in handling this huge amount of data streams in the IoT devices itself. Since, the IoT devices are resource-constrained devices with the limited storage and processing capability, especially network resources. Moreover, the integrated cloud and IoT technology also impose several challenges for the end-users, network, and the terminals associated with the problem of high congestion, fast battery consumption, and low scalability. Since, long distance between the smart IoT devices and the cloud server creates the gap in providing the response, which leads to latency issue. Accordingly, the latency issue creates a greater negative impact on the healthcare applications as healthcare applications are the delay-sensitive applications in real-world. Even though Fog Computing paradigm

provides the opportunities to the end-users, Cloud-Fog interface encompasses several challenges such as context-based resource allocation, workload imbalance, and service overhead. It consumes more time to identify the available VMs from the distributed fog environment to centralized cloud environment, which degrades the performance the service when dealing with the delay-sensitive healthcare applications. Hence, there is an essential need of satisfying QoS, ensuring quick response time and better resource utilization. Moreover, Fog Computing does not have the ability to perform the compute-intensive process, to provide the massive storage, and to establish the wide area connectivity. Also, dividing the computing of the application in the fog and sending the compute-intensive process to the centralized cloud is arduous task due to the occurrence of high network latency. Thus, this work targets on providing the smart fog gateway for delay-sensitive healthcare applications by smart partitioning and allocation. i.e, Fog Node receives continuous stream of the healthcare data, first it needs to break the data stream into the chunks of data. Further these data chunks are processed by Fog in order to take actions like whether to forward this data to cloud or to process the data chunk by decision tree which will take an action on real-time basis.

To enable early decision-making for time-critical applications, Distributed Computing offers quicker computations. As Fog Computing has shortage of computation power, presentation of Distributed Computing in Fog Computing enhances the IoT based latency sensitive healthcare applications. The algorithm is required for smart job allocation as it makes sure of the optimal resource utilization and response time for the smart Fog Computing cluster, which further helps in attaining the SLA considering timely decision making. To enable identification of the Fog Node which can be allocated the appropriate task for health care applications in Fog Computing, the algorithm should determine the capability of the node. Thus, we can obtain an optimized Fog Computing -based performance model for serving the community better considering the health care domain.

IoT devices remain unsupervised for longer duration of time so they need the data and access security protocols. In certain scenarios, IoT-IoT devices also needs to be

communicated hence it should have the data sharing security. By considering the computational capabilities of the IoT device, the lightweight encryption scheme is needed. Single Fog Nodes may lag in terms of computation and decision making time, so to support it we need multiple Fog Nodes at the same horizontal levels. Efficient security scheme is needed for sharing data across multiple Fog Nodes at the same and at the different levels to achieve parallel and Distributed Computing among the Fog Nodes to facilitate big data analytics. To reduce the cloud traffic, the multilevel Fog Node scenario is suggested and also its security solutions should be addressed. For multiple levels multiple security algorithms are needed which makes the system more complex and vulnerable because every algorithm is covering one or the other aspect of the security or computation. So a unique style lightweight algorithm is suggested to satisfy all the security needs at all the levels in IoT-Fog context.

2.5 Problem Statement

The IoT and Cloud Computing paradigm is delay-sensitive for critical real-time health care applications. IoT-Cloud architecture is also sending redundant data on the Cloud, which is utilizing more network bandwidth. Fog Computing lags in terms of computation power if more real-time data streams are subjected to it. So, there is a need of a computation power optimization and better techniques in Fog Computing to handle and support real-time data in more efficient manner. Most of the times IoT devices remain unattended, which makes it more vulnerable and this requests security. Traditional security algorithms are not handled by IoT devices due to low processing capabilities.