Blockchain Assisted Task Distribution and Secure Deduplication using Adaptive Deep Reinforcement Learning in Cluster based Fog-IoT

Your Name,
Department, Address,
Email Address and Supervisor Name, Department, Address
*Corresponding author name-

***Abstract*— Presently fog-assisted Internet-of-Things (IoT) has expected more interest in research community. Since the number of IoT devices is bulky today. Still, IoT devices sense data and forward them to the respective data centers or end users. With the growth rate of IoT devices, there are more duplicate data transmitted over the Internet. For instance, smart healthcare monitoring system will transmits huge amount of duplicate data to data processing center. It wills increases storage capacity and lack of efficiency for monitoring system. In this paper, task distribution and secure deduplication is implemented over Cluster-based IoT. There are four layers presented: IoT Devices Layer, Fog Layer, Cloud Layer and Service Layer. In IoT devices layer, devices to sense air pollutants are deployed. To mitigate security threats, IoT devices are authenticate to cloud server via trusted authority using Edwards Curve based Elliptic Curve Cryptography (EC-ECC). For cluster-head (CH) selection at the first layer, Adaptive Rewards Optimized Deep Reinforcement Learning (ARO-DRL) is presented. In fog layer, SHA-3 is proposed for duplicate verification and CH chooses the best fog node using Emperor Penguin Optimization Algorithm). In fog node, Packet Scrutinization Algorithm is presented for analyze the packet features, which consists of the any DDoS attack packets. Then proxy server is deployed in between cloud server and fog node for queue modeling, which is M/M/C model. In cloud layer, Hybrid cloud environment is aimed at protecting organizations' data in a highly secure manner. The hybrid cloud environment is a combination of private and public cloud. Our IoT devices are divided into sensitive and nonsensitive devices. Sensitive devices generate sensitive data, such as healthcare data; whereas nonsensitive devices generate nonsensitive data, such as home appliance data. IoT devices send their data to the cloud via a gateway device. Herein, sensitive data are split into two parts: one part of the data is encrypted using RC6, and the other part is encrypted using the Advanced Encryption Standard (AES) encryption scheme. Nonsensitive data are encrypted using the Fiestel encryption scheme. This is endeavored to provide query search results for IoT users at the service layer. The overall environment is assumed to be the decentralized in which security is invoked to the IoT devices for provisioning QoS by avoiding the attackers in environment. Experiments conducted and analyzed using NS3 with Java programming. Then we compare with previous works. Finally, the simulation results prove towards enhancement in average latency, user satisfaction, network lifetime, energy consumption, and security strength.**

***Index Terms*— Fog Assisted Internet of Things, Task Allocation, Secure Deduplication, Secure Clustering and NS3 with Java, and Blockchain**

I. NTRODUCTION

INTERNET of Things have grown recently and Industrial Sector is one of the best application areas. In an industrial area, deployment of wireless sensor actuator networks (WSAN), and wireless sensor networks (WSN) helps to sensitive information such as Energy Efficiency, Air Quality Management, Fault Prediction, Resource Prediction, and Product Planning. Some of the important WSN-IoT applications are follows,

- Smart city application (Waste Management, Noise Monitoring, Smart Lighting)
- Smart home application (Energy Management, Remote Monitoring, Security Provisioning)
- Smart transportation application (Traffic Control, Smart Parking, Multimedia Service Provisioning)
- Disaster management application
- Smart grids and energy control systems (Microgrids, Power Management)
- Smart healthcare (Remote Monitoring, Medical Diagnosis)
- Urban terrain tracking and civil structural monitoring
- Smart agriculture (Precision Agriculture, Quality Control, Animal Tracking)
- Industrial IoT (Industrial Internet, Smart Retail, Supply Chain Management)

One of the major challenges is to provide energy efficiency when establishing connectivity among these momentous devices. To manage this massive growth of numerous IoT devices and sensors, diverse research works have been contributed in this research area. However, preserving energy efficiency without affecting communication among IoT entities is still challenging due to,

- Involvement of huge number of nodes
- Communication among multiple entities
- Establishment of multi-hop communication
- Dynamic topology of network

- Lack of optimized network design

This research work is motivated by aforesaid challenges in WSN-IoT. This thesis proposes novel methodologies to achieve the objective of energy efficiency in IoT-Fog-Cloud connected environment. The major factors that affect energy consumption and network lifetime are, (1). Idle Listening (2). Node Isolation, (3). Data Transmission, (4). Overhearing, (5). Redundant Data, (6). Collisions, and (7). Frequent Retransmissions. [2], [3]. Security is being one of the challenging tasks to be performed in the IoT environment which is also subjected to other challenging topic of complexity. Authenticating the IoT devices by means of a special system bring efficient results in security. However, the storage of security credentials is offered by integrating IoT with cloud. Each IoT device is manufactured with the following characteristics,

- Low power – Due to their work process, the devices are presented only to perform particular task, hence they are equipped with lesser power.
- Low resources – The IoT devices are pre-defined to do their operations and hence they are designed with sufficient RAM, CPU and processor.

The characteristics of IoT devices are efficient and hence they are integrated with cloud platform for positive results. IoT enabled cloud is a capable to provide ubiquitous computing for easier and faster access (Xu et al. 2015). This integration is studied in four tier architecture of IoT devices, network device, and edge computing and cloud layer. In recent days, the IoT users are demanding for multiple services from cloud which are provided by this integration. The elasticity of cloud has enabled this environment to be integrated for providing several applications. IoT is concerned to provide efficient communication for the connected devices that are participating in the network. Since, this network environment is spread wider the number of devices connected in the system requires to be properly maintained. This helps in providing permission only for legitimate devices into the environment by which the participation of attackers are minimized. Conventional algorithms and mechanisms were presented to resolve the problem existed in providing security over the developed system and enabled to fight against attackers.

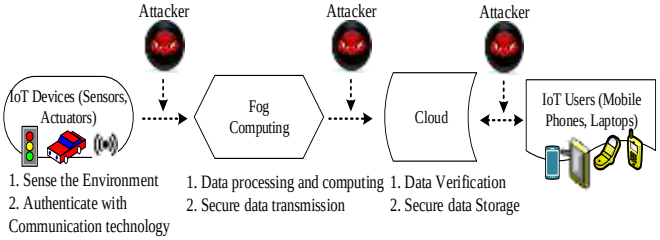

Fig.1. IoT integrated Fog Cloud

Fig 1 depicts the IoT enable cloud environment which is also subjected to attacker's involvement. The attackers can be of any type and also their goal into the system may vary. Whether the integration if IoT environment requires to be provided with security. Fog computing is an emerging paradigm that supports a wide range of applications in real-time. In this model, enormous fog nodes are deployed in various locations. The goal of fog computing model is to reduce the delay for users and the remaining traffic (require a large amount of processing time) is sent to the cloud data centers. Several optimization methods such as weighted sum, hierarchical and trade-off methods have been proposed [4]. Some of the characteristics of fog computing are the following: (1). Heterogeneity, (2). Online analytics, (3). Large Scale IoT applications support, and (4) Easy to interplay with cloud [5]. Healthcare, air pollution monitoring, smart grids, smart homes and smart vehicles are the best example applications that provide advantages via fog computing. Air pollution is the major key factor that affects human life, animals and plants. It happens due to harmful amounts of gases, dust/fumes or odor. Nitrogen dioxide (NO2), Carbon dioxide (CO2), Carbon monoxide (CO), Methane (CH4), Hydrogen sulfide (H2S), Hydrocarbons (Ethanol, Propane, Butane, Isobutane, and Toluene) and Ozone (O3) [6], [7]. In air pollution monitoring systems, there will be a huge amount of duplicate data forwarded to the data processing center. It increases the storage capacity and efficiency of the monitoring system. Hence data deduplication scheme is required in which the redundant or similar data is found out and eliminated. This will improve the storage requirement of the data. IoT devices submit the replicate sensing data that must be eliminated to store in it to the cloud server [8], [9]. On the other hand, task allocation is performed via fog computing for a number of IoT devices. It reduces latency, communication overhead and communication cost for IoT devices [10], [11], [12]. We jointly consider these two issues in this research. Clustering is one of the important processes that supported in fog enabled WSN. Cluster Head (CH) is selected on each cluster, and other nodes in the cluster member (CMs). It functions is mainly are follows: (i). It controls the random selection of CH for each round, (ii). It exploits the heterogeneity energy threshold to

avoid less residual energy nodes, which are nominated as CH in the next rounds, (iii). CH optimizes the minimum distance between the CHs and fog nodes, and (4). It increases energy efficiency for fog nodes and decreases the overhead. To choose the optimum CH. Fog nodes aggregate the sensing data and forward the computing results and then data upload to the cloud servers. Importance of fog computing for task allocation is followed [14], [15]:

- **Latency:** In many real world environments, low latency is a major constraint. To reduce the amount of time for task processing, fog nodes are deployed on the edge of the network.
- **Data confidentiality:** Large amount of data generated and forwarded in network that cause major attacks. To avoid malicious activity, data encrypted and forwarded to end users or other processing centers.

However, mining the data stored by enormous edge servers is difficult in a distributed computing environment. Cloud server or proxy server extracts this information and provide support to on-demand services to end-users. Cloud server outsourcing non-sensitive information to end-users, but it does not guarantee data security and integrity [16]. In the following, we decribed about the fog assisted cloud environment for IIoT applications.

The general architecture of Blockchain is illustrated in fig.2. Each block contains a list of transactions and a hash value of its own and previous block along with a timestamp. To ensure tamper-proof records, blockchain-based data provenance scheme known as ProvChain is introduced [12] which is the primary motivation behind our work. A privacy-preserving model for secure data storage is proposed using blockchain [13]. Blockchain-based forensic architecture is also utilized for vehicular network environment [14] to analyze the cases regarding accidents.

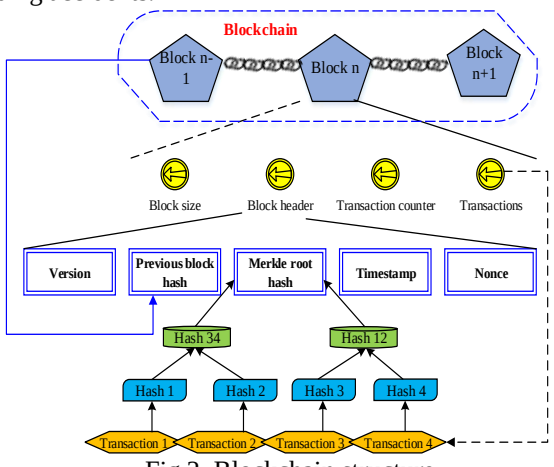

Fig.2. Blockchain structure

Remote monitoring and control system design for the application is motivated by several types of research. In IIoT each IoT device has a function for both sensing and actuation. In a cloud system, data storage occurs and is comprised of two functioning: visualization of historical information and remote monitoring of sensing environment. In Industry 4.0, fog nodes act as Gateways, which respond faster than cloud servers. This paper is provided with the new innovation of task allocation and secure deduplication in the combination of Industrial IoT, Fog and Cloud environments. Hereby we addressed the issues in this integration.

1.1 Motivation & Contributions

The main motivation of this paper is based on research challenges underlying in fog enabled IoT. IoT devices suffer from large latency from the cloud server. Despite mobile crowdsensing, cloud server does not provide services to users on time. The development of IoT is associated with its adaptation on a wide variety of applications. In general the IoT is built for the following purposes,

- Connecting multiple sensors and actuators
- Create decisions on real-time data criticality
- Monitor and report status for every fraction
- Analyzing a large scale data

On behalf of the motivation of IoT, it has been developed with multiple challenges. Among the challenges, security is one of the common needs in IoT while transferring sensitive data. For authenticating such IoT hardware devices, it is essential to consider multiple parameters as identity, password, pin number, one-time password and others. However, the IoT devices are resource constrained it is also required to mitigate the higher consumption of resources. Hereby in order to provide efficient authentication for IoT devices, the key solutions are,

- Multi-factor authentication
- Lightweight algorithms
- Blockchain technology
- Energy efficient clustering
- Scheduling tasks via queue management

The above mentioned three solutions are the key motivation of this proposed these for enabling authentication of IoT devices with the maintenance of lesser resources by deploying a lightweight algorithms. The effectiveness of incorporating lightweight algorithms will reduce the number of computations in the system for authentication of a particular device

The main contributions of this paper are summarized as follows:

- We firstly registers all IoT devices to cloud server via trusted authority using Edwards Curve Elliptic Curve Cryptography (EC-ECC).

- Then we cluster similar IoT devices using Adaptive Rewards Optimized Deep Reinforcement Learning algorithm based on node residual energy, node degree (no. of neighbor connections), and distance between nodes.
- For secure data deduplication, SHA-3 is presented in which hash values are generated for the verification of data deduplication. Before that, Jaccard Similarity is proposed to remove the redundant data packets. After hash generation, CH sends Duplicate Service Check (DSC) and Duplicate Service Response (DSR). Then we select the optimal fog node from the CH using Emperor Penguin Optimization Algorithm (EPO).
- Proxy server is deployed in the fog layer, which performs scheduling operations which schedules packets into two classes such as real-time and non-real time packets using M/M/C model.
- Before transmitted to the cloud server, data packets are encrypted using three lightweight encryption algorithms such as RC6, Fiestel and AEs.
- The experimental results show that the proposed scheme is outperforms than the previous works based on the QoS metrics: average latency, energy consumption, user satisfaction, network lifetime and security strength.

1.2 Paper Organization

The remaining part of the paper is structured as follows: section ii describes the state-of-the-art in the field of fog assisted iiot. in section iii, we present the major problem statements. in section iv, we briefly explain our proposed system design and architecture. In addition, describe each new idea is a well-organized way. In section v, experimental settings for the proposed system design are presented and also evaluate the comparison between the proposed as well as previous approaches. Finally, the conclusion and future work of this paper are presented and summarized in section vi.

Table1. Notations and Descriptions

Notation	Description
d_n	IoT devices in the system $n =$ 1,2,3, … …
ID_n	Identity of each IoT device $n =$ 1,2,3, … …
d_t	Type of IoT device
R	Random number
sk_n	Secret key of each IoT device $n = 1,2,3, \ldots \ldots$
$L_{n(x,y)}$	Location of each IoT device $n = 1,2,3, \ldots \ldots$
Pid	Generated new identity
$E_{K_F}(B)$	Encrypted PUF biometric
Pk_n	SRAM-PUF based public key of IoT device
Pr_n	SRAM-PUF based private key of IoT device
$S_i(D_n)$	Signature with data structure of n^{th} device
$b_{id}\ (d_n)$	Block identity of n^{th} device
$T_b\ (d_n)$	Block timestamp of n^{th} device

II. STATE-OF-THE-ART

In this section we describe latest literatures that have been done in the fog computing, cloud computing and IoT.

2.1 Works Related to Clustering in IoT / WSN

Authors in [16] have proposed a hyper round policy based clustering scheme to enhance the network lifetime. Authors have highlighted that the frequent re-clustering is controlled with the support of hyper round policy which is handled by fuzzy inference system. Initially, some random nodes have been chosen and CHs and cluster formation follows TDMA slots to form clusters. Then the re-clustering process is triggered by fuzzy inference system in order to improve the network lifetime. However, frequent clustering and re-clustering in each round increases energy consumption due to frequent exchange of control packets.

A fuzzy power optimized clustering algorithm was proposed to reduce energy consumption in WSN [17]. At first, the nodes have been classified into different categories based on node degree. Then optimal CH is selected based on multi-parameter iteration adaptively among same category of nodes. The considered parameters are degree of centralism and distance with base station. The major part of this work relies upon node density instead of other significant metrics which is not suitable for energy aware network. A rotating energy efficient clustering for heterogeneous devices (REECHD) has proposed with the concept of intra-traffic rate limit [18] In REECHD, the CH selection is performed based on probability value as follows

$$CH_{prob} = \max\left(\frac{C_{prob}}{K}\left(\frac{E_{Residual}}{E_{max}} + IW^{-1}\right), P_{min}\right) \quad (2.3)$$

Here the leader election probability (CH_{prob}) is computed in terms of predefined initial probability (C_{prob}), minimum probability value a CH must have (P_{min}), residual energy ($E_{Residual}$) and the constant value (K).However, this method is not suitable for large-scale network where node isolation problem will occur for the nodes located far away from the network.Whale optimization algorithm has been implemented with self-adaptiveness for optimal CH selection [19]. The major objective of the authors is to analyze the self-adaptive whale optimization algorithm (SAWOA) with benchmark optimization techniques. The fitness formulation considers residual energy, load and distance metrics for CH selection. Although optimization algorithms perform better than random approaches, it increases energy and consumption in every round for CH selection. Authors in [20] have proposed an optimal clustering algorithm for lifetime maximization (LiMCA) in WSN-IoT. Authors have presented their contributions in two-fold as: energy consumption is balanced among CHs and clusters are formed by LiMCA algorithm.

Initially, the network is divided into multiple slices and then clusters are formed in each slice. In every cluster CH selection is performed in a non-optimal manner which increases number of CH rotations and re-clustering. Further, many optimization techniques also focused in research works for cluster formation.

2.2 Works Related Task Allocation & Security Schemes

In [21] authors have proposed multi criteria based decision making approach for task allocation in several nodes, which is implemented at the edge nodes. In this paper, tasks implemented at nodes in local manner or presented in peer topology. The proposed decision making scheme were follows two decisions for optimal task allocation. Energy consumption is high due to high latency. Spatial crowdsourcing assisted task owners based task allocation and data aggregation have been proposed through fog computing. Server is enabled to collect sensed information from mobile users. Data aggregation is a specific task, which has drawn much attention in mining massive spatial crowdsensing data. Fog nodes are deployed in several regions, which can assist the server to distribute and aggregate data in a privacy aware manner. Privacy for mobile users are lacking due to inefficient cryptography scheme. In [22] IIoT-based fog computing technology was presented, which was implemented for smart factory application. In this paper, a hierarchical fog servers based deployment was invoked, which categorizes sensed data into two forms: high priority and low priority. High priority requests are scheduled firstly since it is based on emergency/urgent demands. Furthermore, a workload assignment algorithm was used to offload high traffic load of fog nodes to higher fog tiers. End-to-end delay is large due to large amount of workloads at fog tier. Adaptive configuration of fog nodes were implemented in [23] over IIoT environment. It offers several IoT services such as imminent failure detection, and automatic monitoring control at fog nodes. It improves the performance of industrial systems. Lyapunov optimization and parallel gibbs sampling methods were proposed in this paper for adaptive fog nodes configuration. However, it is not adopted for any real-time application. In [24] smart resources partitioning was proposed in fog assisted IIoT environment. Firstly authors have exploited Zipfs law to compute the relationship between popularity ranks of computing control layer from data processing layer of IIoT. Experimentation was conducted to show the successful achievement in terms of response time, response rate and delay time. It causes large complexity due to large number of fog nodes that are geographically distributed in the fog layer. In [25] hybrid approach such as reinforcement learning and fuzzy logic algorithms were proposed to minimize latency for healthcare application over IoT environment. The reason for this hybrid approach is to assign packets to different processors of virtual machines available for gateway. Limitation of this work is to high service latency in the application layer. Security is not primarily concentrated on recent works. It is essential for fog enabled cloud environment. In [28] matrix based key agreement and lightweight authentication model was proposed, which focused to make communication through fog computing and also it verifies identify of multiple parties. In this work, healthcare data was encrypted and uploaded into cloud server. Encryption and decryption time is high. Intelligent traffic control system application was implemented in fog based security framework. The proposed framework is referred as Intelligent Transportation Control System (FSF-ITLCS) that comprised of department of motor vehicles (DMV), road side units (RSUs), and vehicles. The proposed addresses various security attacks such as Sybil, DoS, Replay and Impersonation. Overall computation time is high.

2.2 Works Related to Smart Applications in IoT

Air pollution is a key factor that affects health life of the Human, Animals and Plants. There are several application areas related to air pollution monitoring include roadside pollution monitoring, industrial perimeter monitoring, site selection for reference monitoring stations, and indoor air quality monitoring [29], [30]. In [31] authors proposed post-process de-duplication method, which was used to determine redundant air pollutant data and removed it efficiently for manage cloud storage systems. This scheme was flexible for data access control and revocation process. A detailed content of all gases present in the air is important for accurate deduplication. With the rapid growth of Urbanization and Industrialization, air pollution monitoring system was presented in WSN. All sensor information is forwarded to Cloud Server (ThingSpeak an Open Source API) [39 for er analysis and compute air quality index (AQI), which is used to visualize location of air quality efficiently. Timely sensing and transmitting framework is required without processing delay in cloud storage. Authors in [35] we2e proposed air pollution system for monitoring global airs. The main contributions of this paper are three-fold: (1). Determine air pollutants in the given area from Gas Sensors, (2). Design portable interface and user friendly interface i.e. Android Application, in which remote users can access pollution range in the given area. In this work, delay is high when deal with huge amounts of sensors information and thus fog based routers/edge devices are required to reduce delay and also minimizes computation complexity in cloud computing. In [33] IoT based 3D air quality sensing system was presented, which was designed as a real-time, power efficient and fine grained architecture. It was designed with four layers: sensing layer (data collection), transmission layer (bidirectional communications support), processing layer (data processing and analysis), and finally presentation layer (provide graphical interface for users). However, data security was not considered on such air pollution monitoring applications. In order to protect air

quality monitoring system framework, authors in [34] were focused on data integrity and security for low cost air quality sensor, which are used to collect sensors information and manage such pollutants under three cases: Sensor in Physical Possession, Sensor MAC address knowing (geographical information) environment, and Automatic air pollution monitoring in large scale environment. In [35] mobile crowdsensing challenge (increases number of participating mobile users) was tackled through fog computing. User's task allocation is invoked by the selection of nearest fog nodes and also task assignment is done based on user mobility. Here fog assisted secure data deduplication scheme was proposed which ensures data confidentiality. Fog nodes detect and remove replicate data using BLS-obvious pseudorandom number function and chameleon hash function. These were used to hide users information to anonymous mobile users. Currently, air pollution is an emerging topic in IoT, such system sense huge volume of data. For such kind of application, this application may not suitable. Task allocation/assignment does not effective since it find fog nodes based on users local information (preferences and mobility patterns), Boneh–Lynn–Shacham (BLS) -oblivious pseudorandom number function is used for deduplication verification and unauthorized users can easily generate pseudorandom number of deduplication and also it is time consuming process. In Fog based IoT (sensors, devices) environment, energy efficient cluster-based routing is an essential thing that presented in [36], here authors have proposed a new P-SEP based fog computing model. It follows two procedures such as Fog based Energy Efficient Routing using ant colony optimization (FEAR) and Fog based Energy aware Cloud Routing (FECR). This new P-SEP reduces 9% and 8% of energy usage in FECR and FEAR, respectively. Similarly, 74% of network lifetime is increased by 74% and 83% in FECR and FEAR, respectively. The clustering process is not effective since a node with more residual energy is selected as CHs, and optimal fog node allocation does not investigate and implemented properly because it randomly selects the adjacent fog nodes (the major for invoking fog computing is that reduces processing delay and meets users while request application or services). Adaptive block compressed sensing was proposed in [37], which is based on sensor-cloud data acquisition method over fog environment. Adaptive block compressed sensing is an image compression technique, which was proposed in the lower WSN layer. The drawback of this scheme is large complex and it does not lightweight. In addition, it causes high energy utilization in fog nodes due to virtual clusters formation in lower WSN layer. In [38], fog enabled cloud environment was considered in IIoT. A large amount of data was generated in different sources. However IoT devices are vulnerable and insecure to several threats. AVL tree was constructed in cloud server for indexing different data sources, which causes high query processing cost and consumes more time to rebalance the tree. Secure KNN was proposed to ensure data confidentiality, which is very expensive and thus data searching time is high. Furthermore, KNN is not suitable for dense area and also it does not suitable for processing large amount of data, particularly in real-world dataset processing.

Our proposed scheme overwhelms th issues that are mentioned in the previous works.

III. Problem Statement

Security is major issue in fog computing (i.e.) processing unauthorized user task consumes more resources and degrades the overall performance. Task scheduling and queue management considers limited parameters and follows conventional FIFO policy results in large waiting time. In fog environment, task offloading is handled with either task or fog oriented metrics. But it is necessary consider both in order to achieve better efficiency. Overall, secure task management is still challenging in fog-cloud-IoT environment [39], [40]. Cluster formation was proposed using the particle Swarm Optimization (PSO) algorithm whose objective was to enhance energy efficiency. Using the procedure of traditional PSO leads higher time consumption for routing and clustering the sensor nodes in the network. The increase in time impacts on both clustering and routing process [34]. In [36], author presents hierarchical data fusion method for smart healthcare. All three levels consider the biosensor reading only for the patient's health status monitoring. However, environmental factors also play pivotal role in healthcare monitoring. CPE which is used in all three levels use some pre-defined rules to detect the patient status. But this is not suitable in practical scenario since each user has different signs and parameters. Thus, CPE based analysis is not suitable for real-time analysis. In [37], task scheduling and offloading is performed based on the task priority. Offloading decision by gateway increases time and complexity since the gateway performs offloading in a centralized manner for all tasks. All queues follow FIFS policy which increases the waiting time for the tasks with small execution time and slack time. To manage the task starvation, the priority value of the tasks in LP queue are provided with high priority and executed. However, increasing priority for LP tasks affects the computational time for HP tasks. In [38] author focuses on energy efficient offloading in fog-cloud environment for IoT applications.. Offloading objective function fully depends upon the task characteristics. However, for efficient offloading it is necessary to consider the fog node characteristics too. Because, offloading unaware of fog node characteristics results in overloading among fog nodes. Firefly algorithm is inefficient in local search. Thus, firefly algorithm is not suitable to select optimal fog for offloading. Besides, the optimal solution is affected by the firefly control parameters since it has large parameters to be tuned.

IV. System Model

In this section we present the system model for the proposed task allocation and secure deduplication via FaCIIoT. The proposed system architecture for the operations (task allocation and secure deduplication) is depicted in fig. 3.

4.1 System Design and Architecture

The proposed model consists of five entities such as IoT devices (Sensors), fog nodes (gateways), trusted authority (TA), proxy server (PS) and cloud server (CS). Our proposed system for air pollution monitoring contains of four layers namely, the IoT Devices Layer, the Fog Layer, the Cloud Layer, and the Service Layer. The Functioning of each layer is presented in the following:

In IoT devices layer, sensors and actuators are deployed and sensed data. For fast data transmission, clusters are formed at IoT devices layer based on node residual energy level, node degree and distance between nodes. Cluster Head (CH) aggregates data transmitted by Cluster Member (CM) and forward to fog nodes

- In Fog layer, function between CH and fog nodes enabled for verification of data either duplicate or not. It is supported by the hash generation of data.
- In Cloud Layer, proxy server maintains scheduling list for packets transmitted from IoT devices. In this layer, packets are encrypted and stored based on the sensitiveness.
- In Service layer, sensors information is provided for IoT users (authorized users).

We present the brief description of these four layers and design considerations are presented in the following sub-sections.

4.2 IoT Devices Layer

In this layer, IoT devices are deployed in the environment, which measures the concentration of all air pollutants and forwarded sensed data to cloud server for processing via optimal fog nodes. In this work, we consider foremost air polluting and healthcare related sensors such as CO, NOx, SO_2, PM, CO_2, and VOC. These IoT devices are authenticated to trusted authority using Edwards Curve based Elliptic Curve Algorithm (EC-ECA).

a. Authentication

Trust Authority plays a major role in intrusion prevention system that prevents the cloud user's data from the intruders. It provides secure access of cloud user data by generating One Time Signature to each legitimate user. Cloud users stored their data on cloud environment based on Elliptical Curve Cryptography (ECC) method. ECC is a public key encryption technique based on elliptic curve theory that can be used to create more efficient cryptographic keys. It generates the keys through the properties of the elliptic curve equation instead of the traditional method of generation as the product of very large prime numbers. Using this cryptographic method, every user has a pair of private and public key. Public key is used for encryption and signature verification and the private key is used for decryption and signature generation. The key size of ECC is 164 bits. The ECC provides higher security than other encryption process. It uses shorter keys compared to other cryptography method for higher security level.

ECC algorithm starts with the formation of elliptic curve E. The general form of the Edwards Curve E is given as,

$$y^2 = x^3 + ax + b \quad (1)$$

where,

a, b – Real Numbers

x, y – Points on Elliptical curve E

This equation is known as Edwards equation. The coefficients 'a' and 'b' are called characteristic coefficients of the curve, and they have the capability to determine what points must present on the curve. The elliptic curve also requires a non-singular curve. The following equation satisfies that it does not contain any singularities. The condition is explained as,

$$\Delta = 4a^3 + 27b^2 \quad (2)$$

where, the value$\neq 0$

Usually Points on the curve are represented with an x and y component similar to Euclidian coordinate system. This representation considers one exception that is one point in the curve is infinity.

$$A = \begin{pmatrix} a_x \\ a_y \end{pmatrix} \quad (3)$$

This equation is used for the point representation. With the help of this equation the upper case denotes the integers use the pseudo code for Elliptic curve cryptographic algorithm. The public key is generated using standard generator P (point in elliptic curve) and multiplying that point by random number S. Once the public key generated by ECC, it can be used for encrypt and signature verification process. The main goal of the ECC is protecting the user data from the intruders.

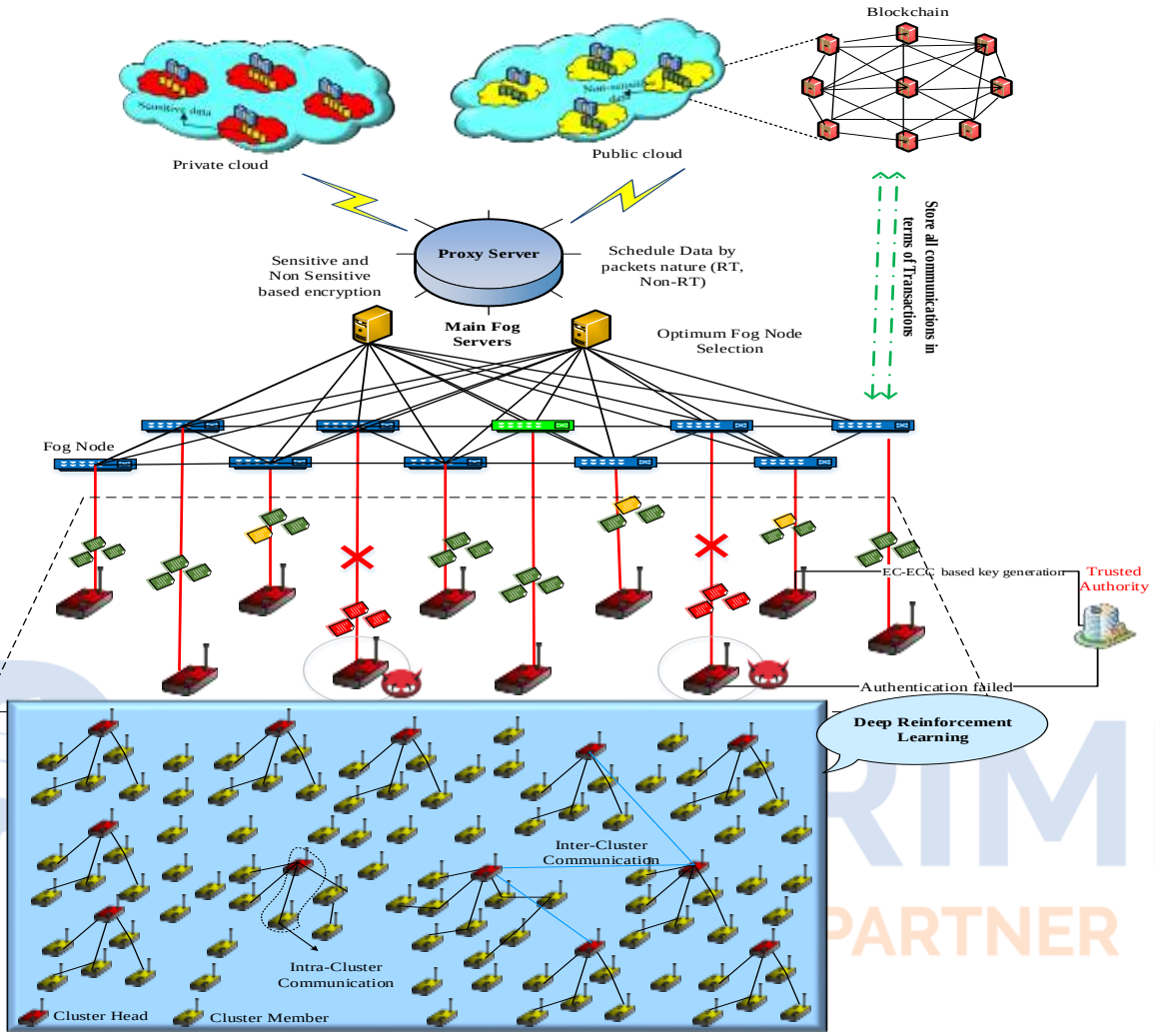

Fig.3. System Model

An elliptic curve is a non-singular projective algebraic curve which is presented over some field k with genus 1 and a specified point O. k does not have characteristic 2 or 3, this will be a smooth plane cubic curve with the point at infinity, and the curve as points satisfying the equation

$$y^2 = x^3 + ax + b, \quad (4)$$

where,a &b are Discriminant

$$\Delta = -16\,(4a^3 + 27b^2) \quad (5)$$

The group law on an elliptic curve is exploited for key selection in elliptic curve cryptography is depends upon the elliptic curve as an abelian group with points as elements. The group law is point additions which add two points P and Q.

a. Cluster Head Selection and Formation

At first, similar sensors are grouped according to three parameters: node residual energy, node degree and distance between nodes. Each cluster contains one CH and two or more cluster members. In data transmission, sensors can communicate within one cluster and also CHs can communicate via other CHs.

a) ***Residual Energy (RE):*** It represents the current energy level of sensor nodes. As assumed, all nodes have same initial energy (IE) and the energy level is varied over a time period. For node N_i the RE is computed as follows

$$RE(N_i) = IE - DE \quad (6)$$

Where DE represents the dissipated energy value over a time period.

b) ***Node Degree (D):*** It defines the connectivity of sensor nodes in the constructed graph. It is computed in terms of number of relative neighbors a node has in the graph.

c) ***Distance with sink node ($dis(N_i, Sink)$):*** It represents the distance between N_i and sink node. It is computed in terms of Euclidean distance as follows

$$dis(N_i, Sink) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (7)$$

Where (x_1, y_1) and (x_2, y_2) represents the coordinates of N_i and sink node respectively.

d) ***Hop count (HP) and mobility (M):*** *HP* defines the number of hops between N_i and sink node. Mobility of the node defines the current mobility speed of the node.

e) ***Link Stability (LS):*** It defines the stability of link between N_i and sink node. It can be expressed as follows,

$$LS = \frac{Radius}{(dis(N_i, Sink))} \quad (8)$$

Here $Radius$ represente the communication range of N_i.

f) ***RSSI***: It is computed in terms of power presented in the radio signal received by N_i from sink node. It is computed as follows,

$$RSSI = P_0 \left(\frac{dis(N_i, Sink)}{dis_0}\right)^{\sigma} \quad (9)$$

Here P_0 represents the reference power received at the distance of d_0 and σ denotes the path loss component.

By using all seven metrics optimal leader nodes are selected. At first, the weight value is computed for all nodes based on $RE, D, dis(N_i, j)$ as follows,

$$W(N_i) = RE + \left({}^{D}/_{dis(N_i, j)}\right) \quad (10)$$

The nodes are sorted in descending order based on weight value. Then the threshold value (μ) is computed based on the average weight value as follows,

$$\mu = \frac{(W(N_1) + W(N_2) + \cdots + W(N_n))}{n} \quad (11)$$

The nodes which have weight value higher than threshold $(W > \mu)$ are considered for second stage. Thus the number of nodes to be processed in next stage is reduced based on weight value.

In CH, Deep Reinforcement Learning is introduced to completely avoid energy consumption issue. Deep reinforcement learning is a new algorithm, which learns and interacts with real-world environment. The goal of this algorithm is to achieve the maximum reward in the current state of the environment. It is based on the finite markov decision process (f-MDP). A set of entities used in this algorithm is follows.

- S is the set of states
- A is the set of actions
- The state transition probability $p(S'|s, a)$. It is a probability distribution function on state space for a given action a for state s
- The discount factor is $\propto$, which range from 0 and 1
- Reward $\gamma\ (S * a)$ is computed using state and action (Set of Real Numbers)
- To get in easier, we assume rewards are discrete
- Use f-MDP when S and A are finite variables

Assume that the current state s and action a in environment is given, then the probability distribution function for next state s' is computed and also the next reward R is expressed as follows:

$$p(s', r|s, a) = P_r(S_{T+1} = S'.\gamma_{t+1} = r|S_T = s.A_T = a) \quad (12)$$

The state transition probability is computed according to reward function (if γ is discrete) and it is expressed as follows:

$$p\ (s'|s, a) = \sum_{r \in Y} p(s', r|s, a) \quad (13)$$

An expected reward is computed for the current s and a is computed by:

$$r(s, a) = E[\gamma_{T+1}|S_T = s, A_T = a] = \sum_{r \in Y} r \sum_{s' \epsilon S} p(s', r|s, a) \quad (14)$$

Then we define state value function, which is described by specific policies since future r is based on the agent current actions. In following we compute the state-value function and action-value function.

State Value Function:

The policy π for the state s of value is computed by the expected return, which is represented as $V_\pi(S)$, which is computed towards the current state s. It is computed below.

$$V_\pi(S) = E_\pi\ [G_T|S_T = s] \quad (15)$$

$$V_\pi(S) = E_\pi\ [\sum_{k=0}^{t-T-1} r^k \gamma_{T+K+1}|S_T = s] \quad (16)$$

Action Value Function:

The policy π for the action Z of value is computed by the action a in state s, which is represented as $Z_\pi(s, a)$. It is computed below.

$$Z_\pi(s, a) = = E_\pi\ [G_T|S_T = s,\ A_T = a] \quad (17)$$

$$Z_\pi(s, a) = E_\pi\ [\sum_{k=0}^{t-T-1} r^k \gamma_{T+K+1}|S_T = s, A_T = a\ ] \quad (18)$$

In IDP agent, deep reinforcement learning algorithm is applied with update each switch stage by two metrics: Flow Duration and Packet Inter Arrival Time. Flow duration is the time difference between 1st and last packet at time t and t+n, whereas the inter packet arrival time is the time difference between two succeeding data packets. In DNN, we given the input variables of all switches and it's current state. From the input variables, hidden layer will compute the weight value and then find the present state. In this work, switches state is computed for each incoming request from the host.

In this algorithm, the IDP agent intends to increase the reward obtained from the environment. In this work, reward function is the major objective function, which indicates the performance of specific switch and this objective function is computed based on flow duration and packet inter arrival time. IDP Agent is mainly consists of two goals and these are considered for reward function. The first goal of an IDP agent is to assign benign packet_in message to switch and the second goal is to malignant packet_in message must be

avoided and eliminate from network and hence we can minimize the attack traffic percentage.

b. Secure Deduplication

After authorization of nodes to TA, cluster formation and cluster head selection. The node separates the received data based on the *Region ID* and then similarity is estimate. We have used Jaccard Similarity for computing the similarity between the data. The formulation of the Jaccard Similarity is given as,

$$Sim(P_1, P_2) = \frac{|P_1 \cap P_2|}{|P_1 \cup P_2|} \quad (19)$$

In this method for determining the similarity using Jaccard, the data packets are converted into shingles. The similarity is between data packets $P_1$and P_2 are estimated from equ (5.13) for data packet 1 P_1 and data packet 2 P_2. The lesser values of $Sim(P_1, P_2)$ similarity will ensure that the data packets $P_1$and P_2 are similar in the data. Hence from these two packets $P_1$and P_2, either P_1or P_2 will be dropped. Here, we drop only the data and include the sensor node ID in the packet to notify that the particular sensor have sensed and given similar data. If the value of similarity is higher, then it is identified that the two data packets are not similar. So both the packets will be transmitted to CH by the intermediate sensor nodes. The computation of similarity is processed by intermediate nodes and hence the time consumption is not larger.

Once, the redundant data is eliminated, the intermediate sensor node transmits the received data to CH. Hence it is ensured that there is no redundant data packets present in CH. For duplication verification, SHA-3 is used to hash the aggregated data of CH. Currently, SHA-3 becomes a significant algorithm for duplication verification. It ensures data integrity while transmits data packets to the cloud server through fog nodes. In SHA-3, the input of data packet is arbitrary length and output is messages digest or hash values. For a hash generation, 512bits are used and the properties of SHA-3 are depicted in table.2

Table.3.Properties Of SHA-3

SHA3(512bits)	
Parameters	*Variants*
Block size (bits)	576
Capacity	1024
Word size (bits)	64
Rounds	24
Operations	AND, OR, Rot, and Not
Security strength	256
Output size (bits)	512

After a generation of hashes, CH sends DSC_request to near and optimum fog nodes. Fog node checks whether this data packet is received or not. If packets at stored in temporary storage of fog nodes, it will immediately send DSC_response. Then CH will store the file to a cloud server via fog nodes.

4.3 Fog Layer

In this layer, fog server selects optimum (nearest) fog node via fog server using EPC algorithm, which follows the procedure of Fast Optimization algorithm.

(a). Optimum Fog Node Selection

EPC Depiction: The above mentioned benefits are obtained by employing the EPC optimization algorithm in the optimizer block. This work is the first in utilizing the EPC optimization algorithm in the IoT framework betterment. The proposed EPC algorithm is developed in the year of 2019 by (harifi et al. 2019). In general, the heuristic algorithm depends either on the swarm behaviour or nature inspired behaviour. In contrast, the proposed EPC algorithm pursues both swarm and nature inspired behaviour. The behaviour of the penguin is organized through the thermal radiation and spiral like movement. The proposed optimization algorithm perceives an optimal solution in the fourth iteration itself. This way of obtaining the solution reduces the execution time of the mining system.

The proposed EPC algorithm gets input as the set of fog nodes along with its current status. Using the received input, the EPC algorithm creates the initial inhabitants array of the emperor penguin (item). It computes the fitness function for fog node based on fog node current residual energy, distance and buffer state. Each penguin in the EPC algorithm estimates its own heat radiation, spiral movement and attractiveness. In addition, it also determines the new position for the next moving direction.

Pseudocode for Optimizer operation

Require : *optimum fog node*
Ensure : $< buffer\ state, energy\ \&\ distance >$
// *Optimizer function*
Generate population P= $[P_1, \dots P_n]$;
For all $P \in [P_1, \dots P_n]$ **do**
  Compute → fitness $f(i)$ for P_i;
  If $(f(i) > f(j))$
    $\mathcal{S}(P_i) \leftarrow f(i)$;
  End If;
End For
emit $(K_n \rightarrow P, V_n \rightarrow \mathcal{S}(P))$
End

The heat radiation ($\mathcal{P}_{hr}$) for each penguin is computed using the below expression

$$\mathcal{P}_{hr} = \mathfrak{s}_a \varepsilon \sigma {T_a}^4 \quad (20)$$

Here, the $\mathfrak{s}_a$ represents the surface area, T_a denotes the absolute temperature, σ denotes the Stefan boltzman constant and ε represents the emissivity. The attractiveness ($\mathcal{P}_{\mathcal{A}}$) of the each penguin is estimated with the aid of the upcoming equation,

$$\mathcal{P}_{\mathcal{A}} = \varsigma_a \varepsilon \sigma {T_a}^4 e^{-\mu d} \quad (21)$$

Where, μ denotes the attenuation co-efficient and d represents the distance between the two linear resources. The penguin spiral movement is computed as follows:

$$x_h = a \cos \theta_k \, e^{b\theta_h} \quad (22)$$

$$y_h = a \sin \theta_k \, e^{b\theta_h} \quad (23)$$

Here, x_h and y_h indicates x and y components of the penguin position 'h'. The spiral moving behaviour of the penguins in the EPC algorithm provides the searching speed effectually. With the use of aforesaid expressions, EPC estimates the fitness function for each penguin which is signified as follows:

$$f(i) = \sum_{i=1}^{n} \mathcal{P}_{hr} \mathcal{P}_{\mathcal{A}} x_h \, y_h \quad (24)$$

Using the above equation, fitness function is estimated for each penguin which defines the optimal value with less amount of time. Since, it converges fastly with optimal solution compared to the other traditional algorithm like PSO, GA and so on.

(b). Packet Scrutinization in Fog

In intrusion detection phase, fog server plays significant role to detect the attack packets in the system. The packets from various IoT devices at different locations are collected by fog nodes via gateway. Each fog nodes are connected to fog server to monitor the behaviour of packets. Due to the dynamic movement of IoT devices, the packet traffic arises in the fog environment.. To solve this complication, fog server assigns the threshold value to each fog nodes. When packet arrival is above than threshold value, the fog node migrates the packets from heavy traffic fog node to idle fog node.

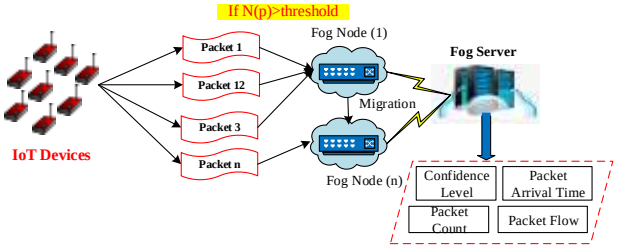

Fig.4. Fog node with packet scrutinization algorithm

Here the proposed packet scrutinization algorithm to analyze the packets which are collected by fog server. The packet scrutinization algorithm analyze the packets based on packet arrival time, packet flow, packet count and their confidence level. Here packet arrival time is defined as the time when the packets entered in the fog node. The sequence of packets arises from source called as packet flow. Confidence level also called as trust value which is defined as the frequency of appearances of attributes in the packet flows and the packet counting is performed based pon its header. The confidence level is calculated based on single attribute and pair of attriburtes follows as,

(i) Confidence level for single attributes,

$$C\ (A_i = a_{i,j}) = \frac{N\ (A_i = a_{i,j})}{N_n} \quad (25)$$

where, i = 1,2,3…n and j = 1,2,3..m_i

(ii) Confidence level for attribute pairs,

$$C\left(A_{i1} = a_{i1,j1}, A_{i2} = a_{i2,j2}\right) = \frac{N\ (A_{i1} = a_{i1,j2}, A_{i2} = a_{i2,j2})}{N_n} \quad (26)$$

where, i_1 = 1,2,3…n, i_2 = 1,2,3…n, and j_1 = 1,2,3..m_1, j_2 = 1,2,3..m_2, N – the number of attributed that are considered to overcome the folding attacks and DDoS attacks, N_n – Total number of packets on packet flow in one time interval, A_i – i^{th} attribute in packet, $N\left(A_{i1} = a_{i1,j2}, A_{i2} = a_{i2,j2}\right)$ – Number of packets whose attribute Ai1 has the value ai1, j1 and Ai2 422 has the values ai2, j2 423 in packet flow in one time interval (t). Using above equation, calculate the confidence level for every packet. If confidence level of the packet is low, then the corresponding packet is discarded. Fog node only allows the packets which have high confidence level than threshold level.

Flowchart for packet scrutinization algorithm

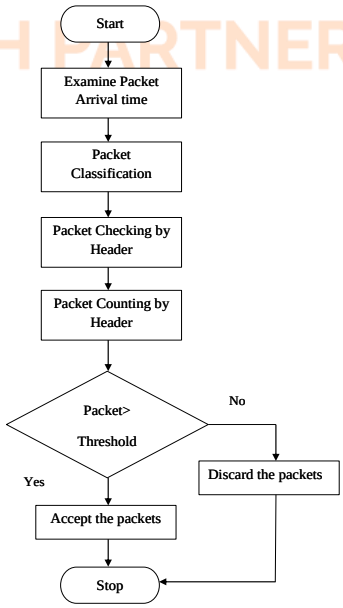

Fig.5. Flowchart for packet scrutinization algorithm

Fig 5 shows the clear view about the working process of packet scrutinization algorithm

Algorithm for Packet Scrutinization

Step 1: Start
Step 2: Examine the arrival time of every packet from all Cloudlets
Step 3: Classify the packets based on arrival time and its Flows
Step 4: Check packets according to its header
Step 5: Count the packets according to its header
Step 6: Check confidence level using (3.4) and (3.5)
If (packet > threshold level)
Accept packet
else
Discard packet
Step 7: End

Thus using this algorithm, we can easily detect and remove the initial flooding attack and port scanning attack.

4.4 Cloud Layer

Proxy server is deployed before cloud servers. In this server, we construct queues for data forwarded by different IoT devices. Proxy server act as primary node deployed between fog node and cloud server.

(a). Queue Modeling

After classification, the normal packets are further processed into fog node whereas the intruder packets are deleted from the fog node. In order to provide efficient processing to the packets, the research work introduces a queue modeling named M/M/C which performed based on the packet prioritization. Usually queuing system is characterized with four basic components such as Queue Discipline, Arrival rate, Service Channel and Service Rate

The proposed queuing model estimates the arrival time based on packet entering into environment. Then service channels are specified with multiple s that can estimate different packets and service rates are defined as that multiple packets are executed with different Proxy Servers at a time. The prioritization of the packet is based on the type of rules which are represented with processing time and arrival time. Fig 6 illustrates the proxy server allocation process performing by the $M/M/C$ queue modelling.

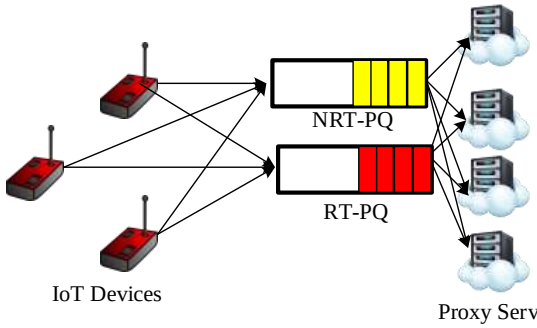

Fig.6. $M/M/C$ queue modelling

Fig 7 shows the working process of $M/M/C$ queue modelling. In this method, multi-users and multi-servers are involved to allocate the packets in specific virtual machine for further execution. The state space model of the $M/M/C$ queue modelling is shown in fig 7.

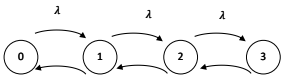

Fig.7. State space diagram of $M/M/C$ Queue modelling

In this figure, λ refers to packet arrival time and 2μ represents the service rate of the packets. Our proposed queue model has multi-user packets (∞) and multi-server (C) "$M/M/C$" and we propose four priorities such as A, B, C and D which is depicted as follows:

- **Class 1:** When a packet has short waiting time and its request type is urgency then packet gets the first priority on queue for processing.
- **Class 2:** A packet with long waiting time and has urgency request then we furnish the second priority for the packet.
- **Class 3:** A packet with short waiting time and has no urgency gets the third priority for processing.
- **Class 4:** A packet with long waiting time and has no-urgency for processing, then we furnish the fourth priority for the packet.

Based on these conditions, the research work allocates the normal packets to the virtual machine to further execution that improves the QoS of our proposed system. Processing steps of the queue modelling is described as follows.

Steps for queue modeling
Step 1: Start *Step 2:* Proxy Servers *Step 3:* if (P is RT: (Urgency && WT- Short) PT: "Class 1" End if *Step 4:* if (P is RT: Urgency && WT - Long) PT: "Class 2" End if *Step 5:* if (P is RT: No Urgency && WT - Short) PT: "Class 3" End if *Step 6:* if (P is RT: No Urgency && WT: Long) PT: "Class 4" End if Step 7: P → Q *Step 8:* Q → Proxy Servers *Step 9:* End

In above steps, P is the packet, Q represents FIFO queues, RT specifies request type, WT denotes waiting time and PT illustrates priority type (1, 2, 3, and 4) of packets.

(b). Sensitive & Non-Sensitive Data Encryption

This data encryption is the first process handled for security guarantee. For data encryption, the message block is classified

into two sub-blocks, in which one sub-block is encrypted with AES and the other sub-block is encrypted using RC-6 algorithm. The conversion of plain text into cipher text using keys is the process of encryption. As per this proposed encryption, the plain texts from sensor nodes are split into two block, each individual block is considered to be of 128 bits.

First Block:

Let p_i [0: N/2-1] and P_i [N/2: N-1] be the two divided block for the plain text, here N is not an integer number which has a fraction. From this division the first block N/2 is encrypted using AES as mentioned above. The size of this block is 128 bits and having the generated key K and length L.

$$p_i = \sum_{i=0}^{i=\frac{n}{2}-1} B_i \; 0 \leq i \leq n/2 - 1 \quad (27)$$

$$C_i = e_{AES}(K, B_i) \quad (28)$$

From equ (6.8), the plain text of the first block is converted into cipher text that is denoted as C_i and e_{AES} is the encryption function that is the function used in AES algorithm. This encryption is followed with the processing of next 128 bit block. Here the entire data packets are split into equal halves for easier and faster processing of data. This division enabled to provide security of the data.

Second Block:

For the second block of the plain text, we follow RC-6 encryption to secure the data. The second block P_i [N/2: N-1] using RC-6 encrypts the data at faster speed with the growth of the security level.

$$P_i = \sum_{i=n/2}^{i=n-1} B_i \quad n/2 \leq i \leq n-1 \quad (29)$$

$$c_i = e_{RC6}(K, B_i) \quad (30)$$

Using the above equation, the second 128-bit block is encrypted. Two cipher texts are generated for single plain text. Two different encryptions are used especially for the providence of security. If any third party identifies the encryption algorithm, only a part can be retrieved**.**

For non-sensitive data encryption, Fiestel algorithm is used to encrypt another part of the divided data i.e. S_{ib}. In cryptography, Fiestel cipher is a symmetric technique used in construction of block ciphers. In this encryption scheme, encryption and decryption operations are similar that tends to easy operation in converting data into cipher text. The proposed Fiestel algorithm splits the data into two parts that tends to improve the security of the encrypted data. Pseudo code for fiestel encryption scheme is illustrated in above table 1. Fiestel encryption scheme encrypts the data at fast rate which is significant for encryption operations. The encryption process of fiestel algorithm has multiple rounds of handling raw data where each round has substitution process which is monitored by the permutation process. Virtually all rounds in fiestel encryption process is same structure.

Let $\mathbb{F}$ be the round function of the fiestel cipher and $K_0, K_1, \ldots K_n$ be the sub-keys for the rounds 0,1,..n respectively. At first, data S_{ib} is split into two equal pieces that are S_{ib_L} and S_{ib_R}. For each round r=0, 1….n compute,

$$S_{ib_L}(r+1) = S_{ib_R}(r) \quad (31)$$

$$S_{ib_R}(r+1) = S_{ib_L}(r) \oplus \mathbb{F}\,(S_{ib_R}, K_i) \quad (32)$$

Where $\oplus$ represents the XOR operator and K_i represents the key value. Then the cipher text attained as $S_{ib_{R+1}}$ and $S_{ib_{L+1}}$. The one of the advantageous of fiestel encryption scheme is round function is doesn't need to be invertible

4.5 Services Layer

In this layer, IoT data are retrieval after the verification of user's authentication. If registration is successful for users in TA, then the searching result is provided for users.

The procedure of Blockchain

Step 1: IoT user/ device requests for a transaction from Blockchain.
Step 2: A new block that denotes the particular transaction is created.
Step 3: Then the particular block will be disseminated to all the other nodes participating in the network.
Step 4: Further all the nodes that received the transactions, will validate the currently received transaction.
Step 5: After verification, the particular block is included into the Blockchain
Step 6: later the transaction is verified and executed.

Using asymmetric cryptography of elliptic curve, pair of keys is generated as public key and private key for IoT device. This PoW is a blockchain authentication method followed in blockchain. This consensus algorithm is more helpful in supporting resource constrained devices. The use of asymmetric cryptography in blockchain enables to provide incorruptible data storage in the blockchain network. The use of PoW is approximately 200 times faster.

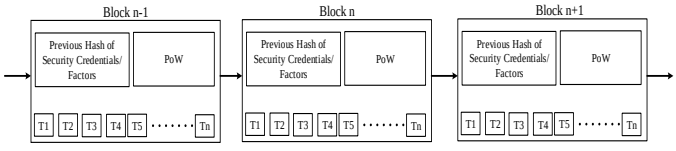

Fig.8. PoW Consensus

The blocks in the blockchain are authenticated using PoW that is represented in fig 8. The nodes present in the network

are enabled to record the distributed ledger and so the transaction information of the nodes can be followed properly. Each block is authenticated and for every successful validation the node will be credited increment in its trust values. Here, the use of asymmetric ECC for key generation with 256 bits attains 128 bits in security level which is effective and it sustains to be protected in the system.

V. Experimental Results and Discussion

In this section we well described the simulation part and also discussed the performances of the proposed scheme using several metrics. Table 4 shows the simulation parameters.

Table.4.System Configuration

Name	Description
Simulation tool	NS-3.26
Development toolkit	JDK-1.8
Operating System	Ubuntu 14.04 LTS
Development platform	Netbeans 8.0
Processor	Pentium (R) Dual-Core CPU E5700@3.00 GHz
Installed memory	2GB RAM

Table.5 Network Environment

Simulation parameters	Values
Number of nodes	100 IoT devices
Number of fog nodes	5
Number of cloud server	1 (Hybrid cloud)
Number of simulated tasks	10, 20, 30, 40, and 50
Number of Smart Gateway	1
Simulation area	1000m×1000m
Task arrival rate	[0,5]
Simulation time	100seconds
Initial energy of a node	5J
Traffic type	CBR
Packet interval	0.1s
Learning rate	0.2

Table.6 Packet information

Mobility Configuration Metrics		
Mobility of MUs Mobility model of MU Interval Time		300 ms Random way point model 0.1s
Packet Configuration Metrics		
Packet Interval Bit Rate		100 ms 2Mbps
Attack Configuration metrics (PPS – Packets per Second)		
Attack Rate (Per Attacker)	High (pps)	1000
	Low (pps)	20
Cumulative Attack Packet Rate	High (kbps)	1000-1200
	Low (pps)	6-70
Cumulative Traffic Rate	High (mbps and pps)	3.6
	Low (mbps and pps)	6-70kbps
Protocol Configuration Metrics		
Protocol Used Latency (processing)		IPv6 10μs
Deep Reinforcement Learning Configuration Metrics		
Reward rate		0.9
Batch size		100
Learning rate		0.001 -0.1
The number of hidden layers		3
The number of nodes at input layer		2
The number of nodes at output layer		2
Activation function		ReLU, and Linear
Optimizer		ADAM

5.1 Simulation Environment

Simulation of the proposed model is presented in this section, which is implemented using NS3 with Java. It is useful for simulation and modeling of resource management in IoT, edge computing and fog computing paradigms. It is an open-source Java based network simulator released by cloud computing and distributed systems (CLOUDS) laboratory at University of Melbourne. NS3 scope is to simulate environment consists of huge number of IoT devices (e.g. sensors, devices and actuators) and fog nodes. Fig 9 (a) and (b) shows the proposed Cloud-IoT environment with IoT devices, data users, gateway devices cloud servers and TA. In this, IoT devices are sensing their surrounding in the range of 100m. IoT devices are sensing field and encrypts the data using lightweight encryption scheme. Encrypted data are stored in cloud server via gateway devices. TA provides security to the stored data via providing access only to the authorized user that avoid malicious user accessing file from the cloud server.

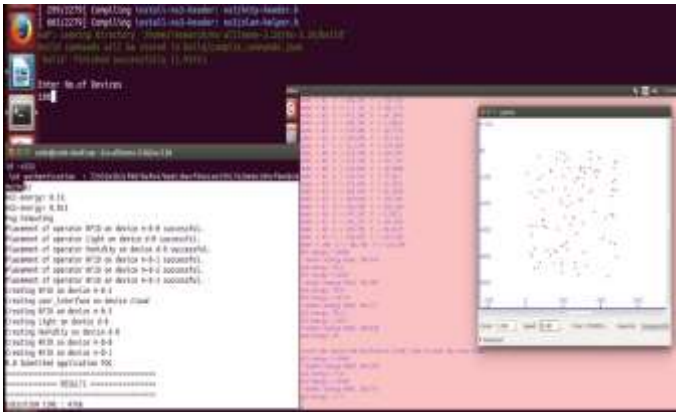

Fig.9 (a). Simulation Details

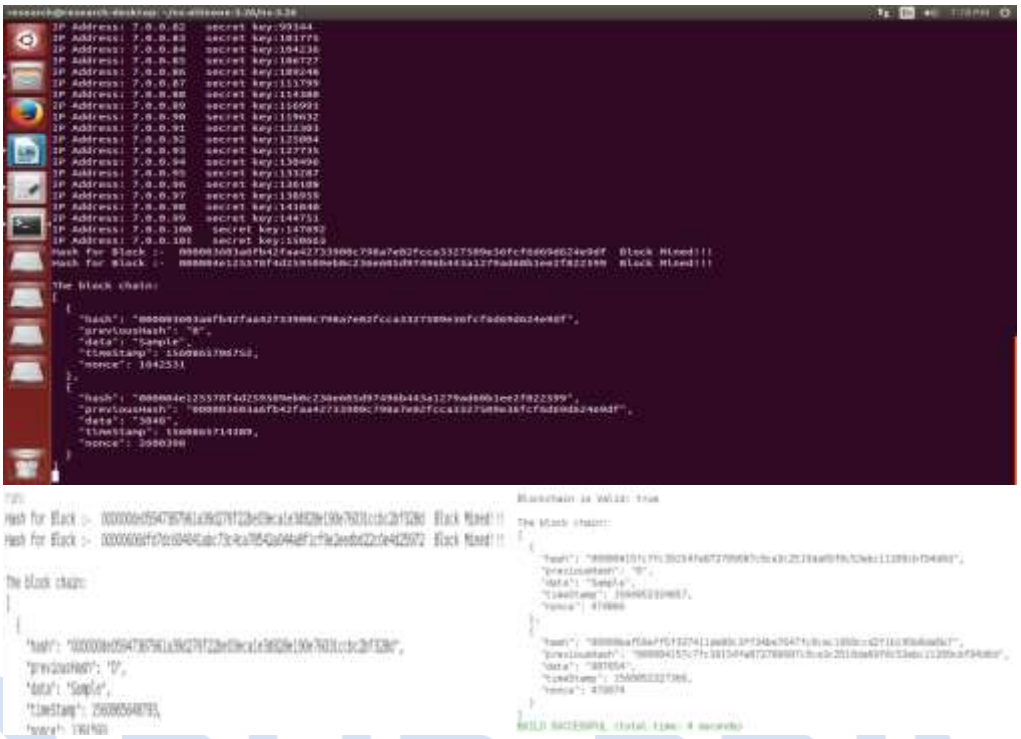

Fig.9 (b). Blockchain Details

5.2. Case Study: Remote Health Monitoring System

The proposed scheme is tested over Remote Health Monitoring application. Currently, several industries and vehicles closer to the city that omits different huge level of air pollutants and redundant is here so it is a need to monitor and control duplicated data transmitted over fog environment, whereas healthcare information must be kept as private for storing and retrieving. Duplication is now happening in massive places, which lead to poor QoS and cause severe issues in cloud serves. Healthcare is harm to the environment so it must be reduce for protecting people, but it is a challenging and important task in fog enabled IoT. In IoT, there are several sensors are deployed to monitor environment based information and healthcare related information. Fig.5. shows the simulation topology for healthcare monitoring in IoT. In this work we categorized air pollutants into three classes that are given below:

- **Primary air pollutants:** Generally, it is produced from several gas sensors include carbon dioxide, sulfur oxides, nitrogen oxides, carbon monoxide, volatile organic compounds, radioactive pollutants, etc.

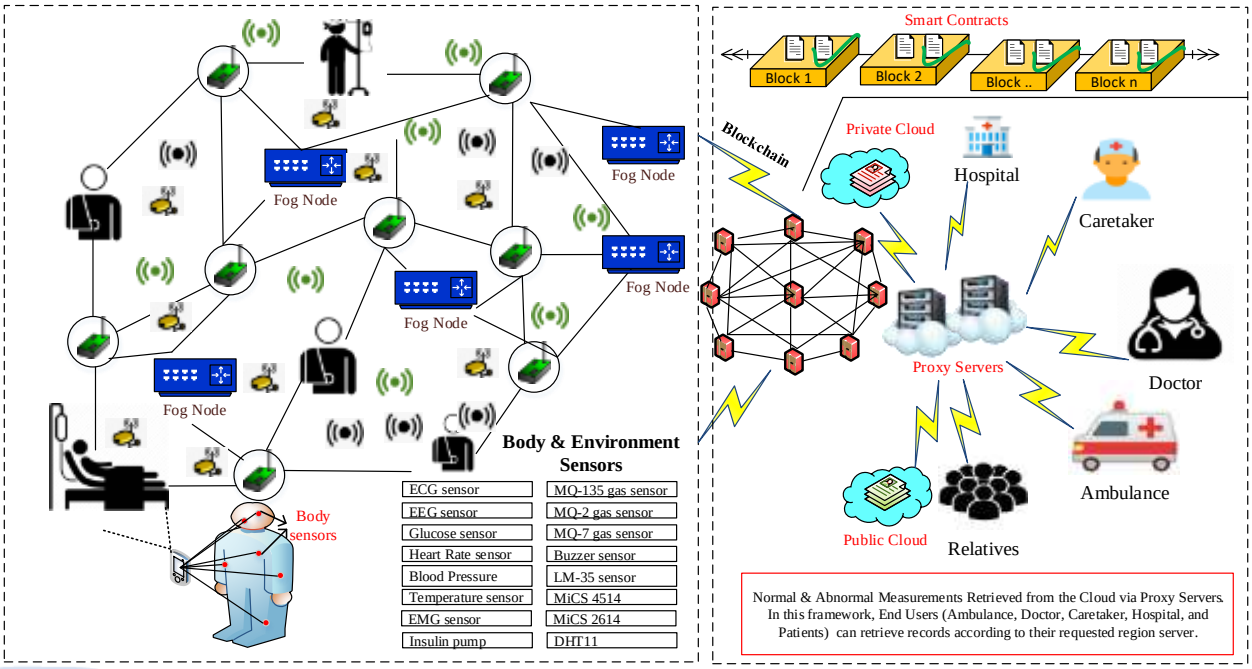

Fig.7.Simulation Topology

- **Secondary air pollutants**: It is generated by communications made from primary air pollutants include ground level ozone, peroxyacetyle nitrate, smog, etc.
- **Others:** It covers minor hazardous and organic persistent air pollutants.

Table 4 shows list of IoT sensors used in this paper for monitoring remote healthcare monitoring system. Sensors are MQ-135, MQ-2, MQ-3, Buzzer sensor, LM-35, MiCS4514, MiCS2614, and DHT11 are deployed for measuring environment information in this area.

Here, the body sensor nodes are deployed in IoT layer and the aggregated sensed data is transmitted to data processing unit. The body sensors can be heartbeat sensor, temperature sensor, pressure sensor, oxygen level sensor and motion sensor. Then the processed data is delivered to doctors, caretakers, ambulance and relatives based on severity level. This intelligent healthcare system mitigates all related issues in conventional healthcare such as delay, inaccurate system and so on. Functionalities of each sensor are illustrated in table.4

Table.7. (a). Sensors and Functionalities

Environment Sensor type	Functionality
MQ-135 gas sensor	For measuring air quality
MQ-2 gas sensor	To detect CO, Alcohol, Smoke/Propane, H2, LPG, and CH4
MQ-7 gas sensor	Detecting CO, and suited sensing concentrations CO in the air
MQ-3 gas sensor	Detects Benzine, Hexane, CO, Alcohol
Buzzer sensor	For giving alarm to inform about Unhealthy Air or Exceeding chemical values on each sensor
LM-35 sensor	For measuring temperature inputs
MiCS 4514	For measuring NO_2 and CO
MiCS 2614	For measuring O_3
DHT11	For measuring Humidity and Temperature

Table.7. (B). Sensors and Functionalities

Body Sensor type	Functionality
ECG sensor	For measuring heart rate
EEG sensor	To detect brain actions and recording nerves activity
Glucose sensor	Detecting the glucose content in the body
Heart Rate sensor	Detects heart rate accurately
Blood pressure sensor	This sensor detects the blood pressure level
Temperature sensor	For measuring temperature inputs
EMG sensor	For measuring muscles information
Insulin Pump	For measuring pumps

Based on HCI, we can measure impact of health on people due to healthcare. It is categorized in to good, satisfactory moderate, poor, very poor, and severe. One of the primary air pollutants is CO, which must be controlled and monitored concurrently.

5.2 Evaluation Measures

Average latency, user satisfaction, network lifetime, network lifetime, energy consumption and security strength are mainly concerned in the combination of IoT, fog and cloud computing paradigms. The definition of these important measures can be following.

(i). Average Latency: It is the time required to respond to the user's given request at a time. The average latency is defined

as the sum of time taken to process all requests given by the IoT device. It is written by:

$$A_l = {min + max}/{2} \quad (33)$$

Where A_l is the average latency and its unit is milliseconds (ms). It is computed on minimum and maximum amount of time. Minimum time is zero and the maximum time is the time require for processing single request.

(ii). User Satisfaction: It is a metric that finds how well a service response from the fog/cloud will satisfy user's requirement. It is not same for users with requests specific service. Hence it differs based on user's service request arrival time and distance to the fog/cloud servers. It is written by:

$$U_s = S_{RT} + S_{IT} + S_Q \quad (34)$$

Where U_s is the user satisfaction, S_{RT} is the service response time, S_{IT} is the service initiated time, and S_Q is the service quality.

(iii). Network Lifetime: It is defined as the amount of time during which the sensor network is fully operative. It can be defined as the maximum duration of operational time of the network while the network perform specific task. It is expressed as,

$$NL = \frac{E_0 - E[UU]}{P + \delta E[Rep]} \quad (35)$$

Where E_0 represents the initial energy consumed by all sensor nodes, $E[UU]$ is expected wasted energy, $E[Rep]$ represents the expected reporting energy and δ is the average sensor reporting rate. The network lifetime is measured in time duration or in number of rounds.

(iv). Energy Consumption: It is defined as the amount of energy consumed to perform processes such as sensing, data transmitting and data receiving. Energy consumption of the network is represented as follows

$$E_C(N) = \sum_{i=1}^{n} [E_{Tx}(N_i) + E_{Rx}(N_i) + E_{sensing}(N_i)] \quad (36)$$

Where $E_{Tx}, E_{Rx}, E_{sensing}$ energy consumed for transmission, reception and sensing by a sensor node N_i.

(v). Security Strength: It is essential to analyze the designed system for task allocation and deduplication. It ensures the user satisfaction and supported for massive data storage at cloud servers. It is computed by the following.

$$S_{st} = K_s + M_s + E_t + D_t \quad (37)$$

Where S_{st} is the security strength, K_s is the key size, M_s is the message size, E_t is the encryption time, and D_t is the decryption time.

(vi). **Detection rate (D):** The detection rate (D) is defined as the percentage of correctly detected attack records of the intrusion detection system. It also defined as the ratio between numbers of attack detected in the system to the number of attacks appeared in the system.

$$\mathrm{D} = \frac{\text{Number of detected attacks}}{\text{Number of attacks present}} \times 100\% \quad (38)$$

Detection rate in terms of TP and FN,

$$\mathrm{D} = \frac{\mathrm{TP}}{(\mathrm{TP}+\mathrm{FN})} \quad (39)$$

(vii). Speedup ratio: The speed up ratio is defined as that supports increasing the performance between two nodes. Here the average execution time take for training the data is considered as speed up ratio.

$$\text{Speedup ratio} = \frac{\text{Time taken for intrusion detection}}{\text{Overall execution time}} \times 100\% \quad (40)$$

(viii). Throughput: Throughput is defined as the rate that packet finishes successful processing of packets. It is also defined as the ratio between the number of successfully executed packet to the total number of packets in the intrusion detection system.

$$\text{Throughput} = \frac{\text{Number of successfully processed packets}}{\text{Total number of packets}} \times 100\% \quad (41)$$

(ix). Packet loss ratio: Packet loss is defined as the ratio between the numbers of packets lost to the total number of packets sent in the intrusion detection system.

$$\text{Packet loss ratio} = \frac{\text{Number of lost packets}}{\text{Number of sent packets}} \times 100\% \quad (42)$$

5.3 Comparative Analysis

As earlier mentioned, different performance evaluation measures are evaluated and compared with four previous works, namely, task allocation and secure deduplication (TA & SD) [35], fog-based energy efficient routing protocol (FEER) [36], adaptive block compression sensing (ABCS) [39], and secure data storage and searching in IoT (SDSS-IIoT) [38].

5.3.1 Average Latency

Some of the IoT applications such as fire accidents, healthcare and healthcare requires very severe constraints in latency i.e. 10's of ms. When enabling data collection and processing features include clustering and classification of data at the device layer or fog layer, we can reduce the latency. Fig 10 shows performance of the average latency with respect to number of IoT devices.

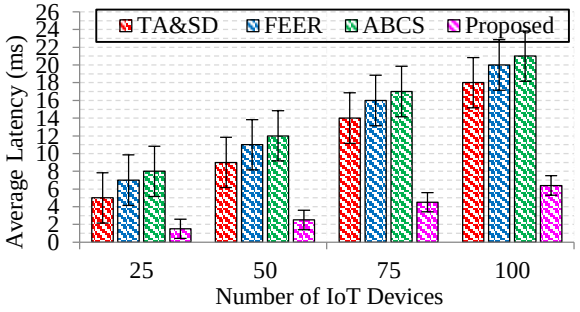

Fig.10. Average latency vs. No. of devices

Fog layer lies in the cloud layer is to minimize the latency. However, previous mechanisms (TA&SD, FEER, ABCS) were required large amount of time for processing data sensed by different IoT devices. The average latency for twenty IoT devices for the proposed scheme is 2.7ms which is minimum then previous works such as TA&SD, FEER, and ABCS for 8.9ms, 10.54ms, and 11.54ms, respectively. When compared to FEER and ABCS, TA&SD, requires minimum average latency due to avoiding redundant copies in fog layer.

Mathematical computations of these three works are more and processing time for data transmission, and collection is large. We proposed EPO for optimum fog node selection, which takes less waiting time for data transmission. In addition, we proposed SHA-3 (512bits) for hash generation, which eliminates duplicate data in the fog layer. Before that, packets are removed when its repeatedly sensed by IoT devices.

5.3.2 User Satisfaction

The most suitable service has provided for the user with attained best QoS shows the system has achieved better performances. It is based on the consideration of each QoS parameter, which determines user service quality. Fig 11 indicates the performance of user satisfaction with respect to number of IoT devices. This metric affects the performance of QoS parameter since user satisfaction is important while design secures data storage and task allocation for industrial applications.

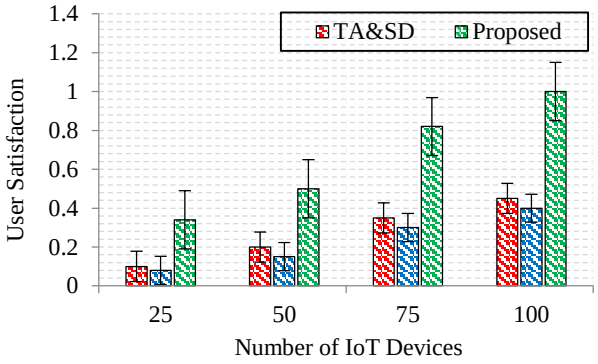

Fig.11.User satisfaction vs. No. of devices

The proposed scheme satisfies user with less service response time and high service quality. The average user satisfaction for the proposed scheme is 0.5, than TA&SD (0.21) and SDSS-IIoT (0.17), respectively. User satisfaction rate in TA and SD is lesser due to improper management of cloud servers. In addition, users requests to cloud server, which increases latency and also suffers from producing high QoS.

5.3.3 Network Lifetime

Improve network lifetime in fog assisted IIoT is a challenging task. In this paper, we evaluated network lifetime for comparison. Fig 12 shows the performance of the network lifetime with respect to number of IoT devices.

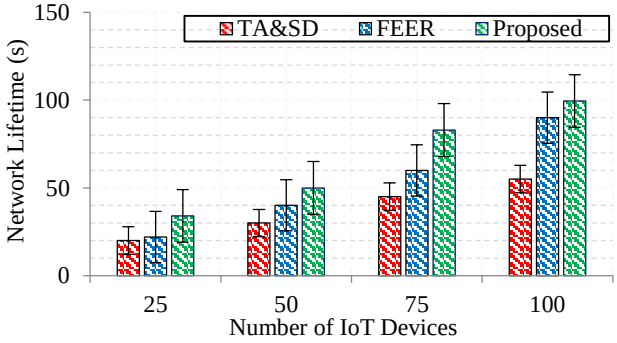

Fig.12.Network lifetime vs. No. of devices

This network lifetime is well-defined as the operating time of the nodes in the network for the maximum duration in order to perform a particular task. Fig.12. depicts the evaluation result for network lifetime metric with respect to the increase in number of sensor nodes. As per the increase in number of nodes, the network lifetime also increases i.e. upto 100 nodes the network lifetime is increased in the proposed protocol. Network lifetime metric is inversely proportional to the consumption of energy. The reduction in energy consumption creates its impact over network lifetime. Hereby network lifetime is increased while comparing with previous research works existed in WSN. On undergoing this comparison, it is identified that the EPO and DRL sustains only for 10 rounds, which is due to poor design.

Our proposed scheme reduces rate of energy consumption and improves network lifetime In FEER, network traffic is reduced, scalability of the network is improved and also latency is minimized, but network lifetime is less due to routing among fog nodes. On the other hand, TA&SD holding less network lifetime, which is primarily due to poor fog nodes selection. Data transmission from IoT device to the fog node is based on the local information. Here authors were not focused on optimal fog nodes selection, which leads to high energy consumption at IoT devices. In general fog nodes and IoT devices are resource-constrained.

5.3.4 Energy Consumption

Energy consumption is the mainly concern in fog enabled IoT applications. IIoT sensors/devices/actuators are energy-constrained while some operations require huge amount of energy. This high energy consumption problem can be addressed by proposing energy efficient tasks e.g. clustering. In clustering, CH transmits data to the near fog nodes and hence mathematical computations are reduced and require only minimum energy consumption in IoT devices. Fig.13 shows the performance of energy consumption with respect to number of devices.

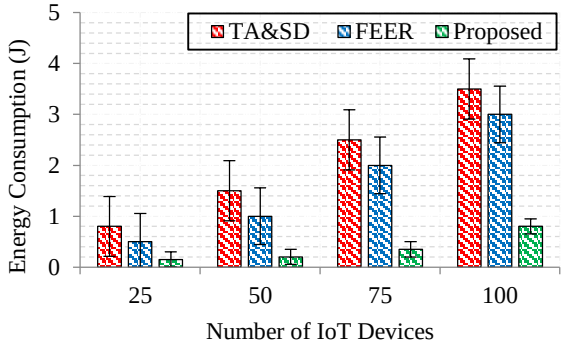

Fig.13. Energy consumption vs. No. of devices

From the graph, it can be seen that the proposed scheme has attained less energy consumption than the previous works such as TA&SD, and FEER. In FEER ACO based routing consumes more energy and also there is no assurance to get always the shortest path from the source (fog node) to the destination node (fog node). Despite of ACO routing, FEER consumes more energy and TA&SD verifies data deduplication at fog nodes in randomly. For these large computations are required to send packets request from sensor to fog nodes, which leads to more energy consumption at both

IoT devices and fog nodes. The average energy consumption of the proposed scheme is 0.26J, and the previous works such as TA&SD and FEER are 1.5J and 1.21J, respectively. Our proposed system architecture of task allocation and secure deduplication in FaCIIoT consumes less energy with the help of best fog node selection, and cluster formation.

5.3.5 Security Strength

Investigation of security for any real-time application is important, particularly in healthcare monitoring in IIoT. Security strength is computed based on several metrics such as key size, message size, encryption and decryption time. Fig 14 shows the performance of the security strength with respect to key size, message size, encryption time and decryption time. We compare the proposed scheme with some previous works based on key size (in bits) starting from 64bits to the 2048 bits. When key size increases security strength may increases or decreases. Our proposed scheme has attained better security strength, which is depicted in Table 8, whereas previous works such as TA&SD and SDSS-IIoT was attained less security strength due to ineffective security algorithms.

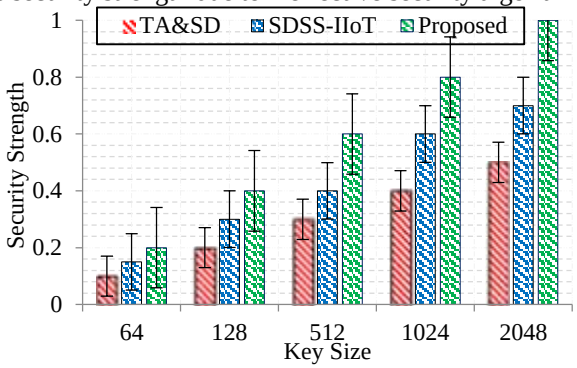

Fig.14.Security Strength vs. Key size

Table.8.Security strength for proposed vs. previous works

Key size	Security Strength		
	TA&SD	SDSS-IIoT	Proposed
64bits	0.1	0.15	0.2
128bits	0.2	0.3	0.4
512bits	0.3	0.4	0.6
1024bits	0.4	0.6	0.8
2048bits	0.5	0.7	1.0
Average	**0.25**	**0.358**	**0.5**

In [38] authors were proposed BLS-Pseudorandom function, which is not strong in terms of security. Hence, security is leaked. In SDSS-IIoT, secure KNN algorithm is proposed to improve data confidentiality, which increases storage capacity and also cause privacy data leakage.

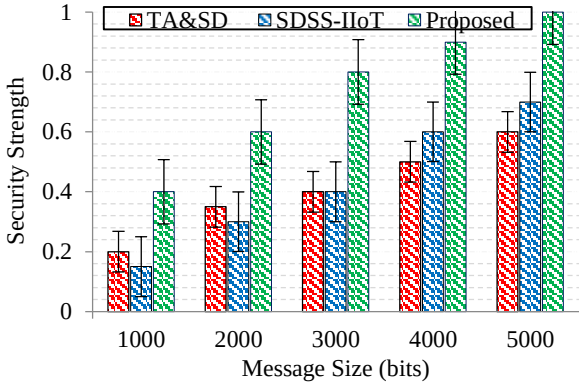

Fig.15.Security Strength vs. Message size

Table.9.Security strength for proposed vs. previous works

Message Size	Security Strength		
	TA&SD	SDSS-IIoT	Proposed
1000bits	0.2	0.15	0.4
2000bits	0.35	0.3	0.6
3000bits	0.4	0.4	0.8
4000bits	0.5	0.6	0.9
5000bits	0.6	0.7	1.0
Average	**0.341**	**0.358**	**0.61**

Fig 15 and table 9 shows the performance of the security strength with respect to message size (bits). The average security strength for the proposed scheme is 0.61, which is higher than previous works such as TA&SD, SDSS-IIoT since its obtained 0.341, and 0.358, respectively. We proposed ECC-HM, which is lightweight cryptographic algorithm, which gives high security strength when message size increases. It consumes minimal amount of time for encryption and decryption. Fig 12 13 and table 10 and 11 shows the performance of security strength with respect to time of encryption and decryption. Pseudorandom function and secure KNN algorithm are not lightweight cryptography and thus it takes high processing time.

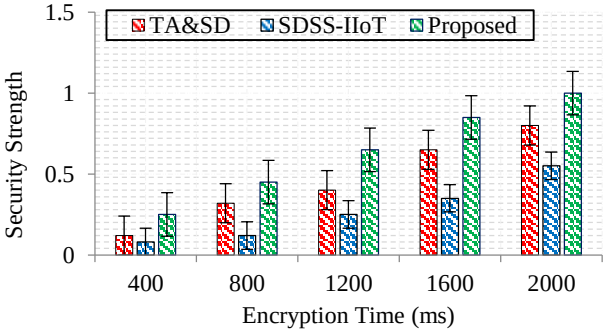

Fig.16.Security Strength vs. Encryption time

Table.10.Security strength for the proposed vs. previous works

Encryption Time	Security Strength		
	TA&SD	SDSS-IIoT	Proposed
400ms	0.12	0.08	0.25
800ms	0.32	0.12	0.45
1200ms	0.4	0.25	0.65
1600ms	0.65	0.35	0.85
2000ms	0.8	0.55	1
Average	**0.381**	**0.225**	**0.533**

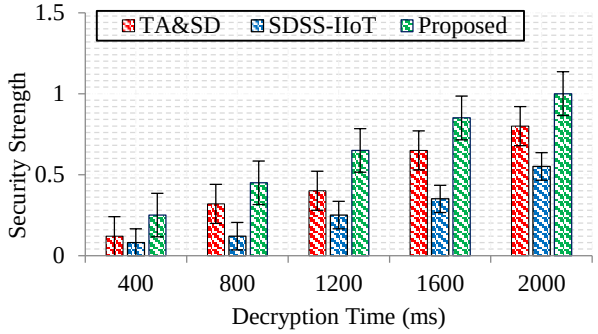

Fig.17.Security Strength vs. Decryption time

Table.11.Security strength for the proposed vs. previous works

Decryption Time	Security Strength		
	TA&SD	SDSS-IIoT	Proposed
400ms	0.06	0.04	0.02
800ms	0.16	0.06	0.03
1200ms	0.2	0.12	0.08
1600ms	0.35	0.15	0.011
2000ms	0.4	0.55	0.012
Average	**0.195**	**0.153**	**0.0255**

5.3.6 Detection Rate

In this paper, we proposed a new combination of algorithms for classifying packets in different aspects. We firstly classify packet into normal or attack. If the packet is attack, then we identify whether attack is frequent or rare attack. Fig 18 indicates the performance of DR with respect to number of attacks. We compare our proposed model with previous works in fog Cloud environment. DR can be varied according to number of packets and number of nodes arrived in the network. Our proposed model reaches high detection rate for any type of class (normal/attack). The average DR is 99.4%, which is relatively higher than the previous works, such as 97.5%, 94.52%, and 97.86% for existing works.

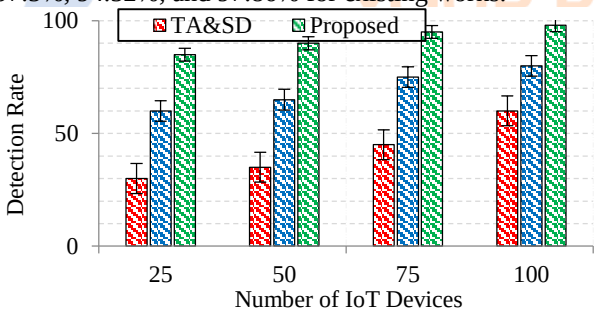

Fig.18.Detection Rate vs. Number of IoT Devices

In this paper, we invoke trusted authority (one-way hash function) for intrusion prevention, which restricts the access of malicious nodes. It helps to improve DR when presence attackers. In previous works, legitimate nodes can be easily compromised by intruders and get packets and have full of rights to access the system.

5.3.7 Throughput

It is defined as the successful packets transmission rate than previous works. It is a positive indicator so it must be higher to show the system has obtained better performance. Fig 19 shows the result for throughput with respect to number of nodes.

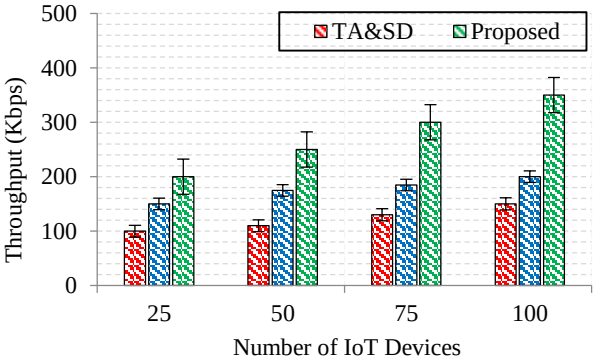

Fig.19. Throughput vs. Number of IoT Devices

In previous work, authors proposed DNN for attack detection which result higher throughput, which is the first better existing work, compared to our proposed model. We combine and proposed lightweight algorithms for effective classification. In other previous works, throughput decreases and does not suitable for intrusion detection under large scale network environment. Experiment results shown that the proposed model has obtained the average of throughput in 220kbps which is higher than previous works.

The experimental results shows that the proposed scheme is outperforms than the previous works such as TA & SD, FEER, ABCS, and SDSS-IIoT in terms of average latency, user satisfaction, network lifetime, security strength, and energy consumption. In this paper we address the following research questions:

- How can we allocate fog nodes to different IoT users efficiently?
- Improper organization of storage files result high energy consumption and delay, which does not satisfy the user application requirements?
- How to protect entire Fog assisted Cloud based IoT environment against attackers?
- How to achieve better performances by proposing robust blockchain based IoT architecture for task distribution and secure deduplication.

VI. Conclusion and Future Work

Healthcare is one of the major problems in the industrial sector in which deduplication is the critical factor to minimize storage capacity and latency of cloud server and fog nodes, respectively. The main purpose of this application consideration is that today air pollutants range exceeds its threshold range. Therefore it causes severe health issues for people. To overwhelm this issue, we designed this paper over the Fog assisted Cloud environment for IIoT, which is carried out two processes, namely Task Distribution and Secure Deduplication. For task distribution from IoT devices to fog nodes, we selected optimal CH, which generates a hash using for aggregated data using SHA-3. After SHA-3, DSC_request is sent to optimal fog node which process request and send DSC_response message to CH. To reduce latency for data transmission, we determined optimum fog nodes emperor penguin optimization algorithm. A proxy server is deployed between the cloud servers and fog nodes, which is employed

to schedule user queries. To achieve data confidentiality, we proposed a security algorithm lightweight algorithms are proposed for data encryption. Finally, simulation is conducted to implement the proposed as well as previous works comparison in terms of average latency, user satisfaction, energy consumption, and network lifetime and security strength. Our proposed scheme has proved that it's outperforming than previous works.

In the future, we have planned to concentrate on other real-time applications and use any fault detection mechanisms for error-ness correction.

REFERENCES

[1] Aazam, M., Zeadally, S., & Harras, K. A. (2018). Deploying Fog Computing in Industrial Internet of Things and Industry 4.0. IEEE Transactions on Industrial Informatics, vol. 14, no. 10, pp. 4674-4682, 2018

[2] Sharma, S., & Saini, H. (2019). A Novel Four-Tier Architecture for Delay Aware Scheduling and Load Balancing in Fog Environment. Sustainable Computing: Informatics and Systems, 100355.

[3] Wu, C., Li, W., Wang, L., & Zomaya, A. (2018). Hybrid Evolutionary Scheduling for Energy-efficient Fog-enhanced Internet of Things. IEEE Transactions on Cloud Computing, 1–1.

[4] Haider, F., Zhang, D., St-Hilaire, M., & Makaya, C. (2018). On the Planning and Design Problem of Fog Computing Networks. IEEE Transactions on Cloud Computing, 1–1.

[5] Verma, P., & Sood, S.K. (2018). Fog Assisted-IoT Enabled Patient Health Monitoring in Smart Homes. IEEE Internet of Things Journal, 5, 1789-1796.

[6] Dautov, R., Distefano, S., & Buyya, R. (2019). Hierarchical data fusion for Smart Healthcare. Journal of Big Data, 6, 1-23.

[7] Naghshvarianjahromi, M., Kumar, S., & Deen, M.J. (2019). Brain-Inspired Intelligence for Real-Time Health Situation Understanding in Smart e-Health Home Applications. IEEE Access, 7, 180106-180126.

[8] Zhu, T., Colopy, G.W., MacEwen, C., Niehaus, K.E., Yang, Y., Pugh, C.W., & Clifton, D.A. (2019). Patient-Specific Physiological Monitoring and Prediction Using Structured Gaussian Processes. IEEE Access, 7, 58094-58103.

[9] Leu, F., Ko, C., You, I., Choo, K.R., & Ho, C. (2017). A smartphone-based wearable sensors for monitoring real-time physiological data. Computers & Electrical Engineering, 65, 376-392.

[10] Bhatia, M., & Sood, S.K. (2018). Exploring Temporal Analytics in Fog-Cloud Architecture for Smart Office HealthCare. Mobile Networks and Applications, 1-19.

[11] J.Li, J. Jin, J. D. Yuan, H. Zhang (2018). Virtual Fog: A Virtualization Enabled Fog Computing Framework for Internet of Things. IEEE Internet of Things Journal, vol. 5, pp. 121-131

[12] P. Wu, E.W. Ngai, Y. Wu, (2018). Toward a real-time and budget-aware task package allocation in spatial crowdsourcing, Decision Support Systems, vol. 110, pp. 107-117.

[13] P.G.V. Naranjo, M. Shojafar, H. Mostafaei, Z. Pooranian, E. Baccarelli (2016). P-SEP: A prolong stable election routing algorithm for energy-limited heterogeneous fog-supported wireless sensor networks. The Journal of Supercomputing, vol. 73, no.2, 733–755

[14] V. Moysiadis, S. Panagiotis, I. Moscholios, (2018). Towards Distributed Data Management in Fog Computing, Wireless Communications and Mobile Computing, vol.2018, pp.1-14

[15] M. Lavassani, S. Forsstrom, U. Jennehag, T. Zhag, (2018). Combining Fog Computing with Sensor Mote Machine Learning for Industrial IoT, Sensors, vol. 18, no. 1532, 2018

[16] G. Peralta, P. Garrido, J. Bilbao, R. Aguero, P.M. Crespo, (2019). On the Combination of Multi-Cloud and Network Coding for Cost-Efficient Storage in Industrial Applications, Sensors, vol. 19, no. 7, pp. 1-19

[17] Y. Yu, L. Xue, Y. Li, X. Du, M. Guizani, B. Yang, (2018). Assured Data Deletion with Fine Grained Access Control for Fog-based Industrial Applications, IEEE Transactions on Industrial Informatics, vol. 14, no. 10, pp. 4538-4547, 2018

[18] F.H. Tseng, M.S. Tsai, C.W. Tseng, Y.T. Yang, C.C. Liu, L.D. Chou, "A Lightweight Auto-Scaling Mechanism for Fog Computing in Industrial Applications", IEEE Transactions on Industrial Informatics, vol.14, no.10, 4529-4537, 2018

[19] Y. Liu, K.A. Hassan, M. Karlsson, O. Weister, S. Ghong, "Active Plant Wall for Green Indoor Climate based on Cloud and Internet of Things", IEEE Access, vol. 6, pp. 33631-33644, 2018

[20] Neamatollahi, P., Naghibzadeh, M., & Abrishami, S. (2017). Fuzzy-Based Clustering-Task Scheduling for Lifetime Enhancement in Wireless Sensor Networks. IEEE Sensors Journal, 17, 6837-6844.

[21] Li, J., Hou, X., Su, D., & Munyemana, JDD. (2017). Fuzzy poweroptimised clustering routing algorithm for wireless sensor networks. IET Wireless Sensor Systems, 7(5), 130–137.

[22] Cacciagrano, D., Culmone, R., Micheletti, M., & Mostarda, L. (2019). Energy-Efficient Clustering for Wireless Sensor Devices in Internet of Things. Performability in Internet of Things, 59-80.

[23] Reddy, M.P., & Babu, M.R. (2017). A hybrid cluster head selection model for Internet of Things. Cluster Computing, 1-13

[24] Halder, S., Ghosal, A., & Conti, M. (2018). LiMCA: an optimal clustering algorithm for lifetime maximization of internet of things. Wireless Networks, 1-19.

[25] Li, G., Wu, J., Li, J., Wang, K., & Ye, T. (2018). Service Popularity-based Smart Resources Partitioning for Fog Computing-enabled Industrial Internet of Things. IEEE Transactions on Industrial Informatics, 1–1.

[26] Shukla, S., Hassan, M. F., Jung, L. T., & Awang, A. (2018). Architecture for Latency Reduction in Healthcare Internet-of-Things Using Reinforcement Learning and Fuzzy Based Fog Computing. Recent Trends in Data Science and Soft Computing, 372–383.

[27] Miao, D., Liu, L., Xu, R., Panneerselvam, J., Wu, Y., & Xu, W. (2018). An Efficient Indexing Model for the Fog Layer of Industrial Internet of Things. IEEE Transactions on Industrial Informatics, 1–1.

[28] Shen, J., Yang, H., Wang, A., Zhou, T., & Wang, C. (2018). Lightweight authentication and matrix-based key agreement scheme for healthcare in fog computing. Peer-to-Peer Networking and Applications.

[29] K. Bashir Shaban, A. Kadri, E. Rezk (2016). Urban Air Pollution Monitoring System with Forecasting Models, IEEE Sensors Journal, vol. 16, no. 8, pp. 2598–2606.

[30] K. Hu, A. Rahman, H. Bhrugubanda, V. Sivaraman (2017). HazeEst: Machine Learning Based Metropolitan Air Pollution Estimation From Fixed and Mobile Sensors. IEEE Sensors Journal, vol. 17, no. 11, pp. 3517–3525.

[31] S. Beulah, F. R. Dhanaseelan, (2018). An Optimal Method for Duplication Detection and Elimination from Air Pollution Data of Wireless Sensor Network, International Journal of Environment and Waste Management, vol.21, no. 2/3.

[32] S. Dhingra, R. B. Madda, A. H. Gandomi, R. Patan, M. Daneshmand, M. (2019). Internet of Things Mobile - Air Pollution Monitoring System (IoT-Mobair), IEEE Internet of Things Journal, 1–1.

[33] Z. Hu, Z. Bai, Y. Yang, Z. Zheng, K. Bian, Song, L. (2019). UAV Aided Aerial-Ground IoT for Air Quality Sensing in Smart City: Architecture, Technologies, and Implementation, IEEE Network, vol. 33, no. 2, pp. 14–22

[34] L. Luo, Y. Zhang, B. Pearson, Z. Ling, H. Yu, X. Fu, (2018). On the Security and Data Integrity of Low-Cost Sensor Networks for Air Quality Monitoring. Sensors, vol. 18, no. 12, 4451.

[35] Adhikari, M., Mukherjee, M., & Srirama, S.N. (2019). DPTO: A Deadline and Priority-aware Task Offloading in Fog Computing Framework Leveraging Multi-level Feedback Queueing. Internet of Things Journal.

[36] Adhikari, M., & Gianey, H.K. (2019). Energy efficient offloading strategy in fog-cloud environment for IoT applications. Internet of Things, 6, 100053.

[37] Ni, J., Zhang, K., Yu, Y., Lin, X., & Shen, X. S. (2018). Providing Task Allocation and Secure Deduplication for Mobile Crowdsensing via Fog Computing. IEEE Transactions on Dependable and Secure Computing, 1–1.

[38] Borujeni, E. M., Rahbari, D., & Nickray, M. (2018). Fog-based energy-efficient routing protocol for wireless sensor networks. The Journal of Supercomputing.

[39] Liu, Z., & Li, S. (2018). Sensor-cloud data acquisition based on fog computation and adaptive block compressed sensing. International

Journal of Distributed Sensor Networks, vol. 14. Issue. 9, 155014771880225.
[40] Fu, J., Liu, Y., Chao, H.-C., Bhargava, B., & Zhang, Z. (2018). Secure Data Storage and Searching for Industrial IoT by Integrating Fog Computing and Cloud Computing. IEEE Transactions on Industrial Informatics, 1–1.
[41] Bathiya, B., Srivastava, S., & Mishra, B. (2016). Air pollution monitoring using wireless sensor network. 2016 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE).