Ministry of Higher Education and Scientific Research
University of Babylon
College of Science for Women
Department of Computer Science

Efficient Resources Management in IoT Environment Based on Fog Computing

A Thesis

Submitted to the Council of College of Science for women at the University of Babylon in Partial Fulfilment of the Requirement for the Degree of Master in Science \ Computer Science

Submitted by

Rawaa Nadhum Saeed

Supervised by

Asst. Prof. **Asst.Prof.**

Dr. Muhammed Abied Mahdi. **Dr.Mahdi Abed Salman.**

2022 A.D. **1444 A.H**

بسم الله الرحمن الرحيم

﴿هُوَ الَّذِي بَعَثَ فِي الْأُمِّيِّينَ رَسُولًا مِنْهُمْ يَتْلُو عَلَيْهِمْ آيَاتِهِ وَيُزَكِّيهِمْ وَيُعَلِّمُهُمُ الْكِتَابَ وَالْحِكْمَةَ وَإِنْ كَانُوا مِنْ قَبْلُ لَفِي ضَلَالٍ مُبِينٍ﴾

صدق الله العلي العظيم

الجمعة : اية 2

Supervisors Certification

We certify that this thesis entitled "Efficient Resource Management in IoT Environment Based on Fog Computing" is done by (Rawaa Nadhum Saeed) under our supervision.

Signature:

Name: Asst. Prof. Dr. Muhammed Abaid Mahdi

Date: / /2022

Address: University of Babylon/College of Science for Women

Signature:

Name: Asst. Prof. Dr. Mahdi Abed Salman.

Date: / /2022

Address: University of Babylon/College of Science for Women

Head of the Department Certification

In view of the available recommendations, I forward the thesis entitled "Efficient Resource Management in IoT Environment Based on Fog Computing" for debate by the examining committee.

Signature:

Name: Dr. Saif Mahmoud Khalaf.

Date: / /2022

Address: University of Babylon/College of Science for Women

Dedication

To the spring of tenderness, My Dear

Parents.

To whom were my support and shadow when fatigue hits me,

My faithful husband.

To the seed of the heart and the hope of tomorrow,

My daughter, Ruqayya.

Acknowledgments

First, praise and thanks to Allah for giving me the strength to complete this work. I would like to express my thanks to my supervisors Dr. Mahdi and Dr. Muhammed for their continuous guidance and keen supervision throughout this study. They have taught me the methodology to carry out the research and to present it as clearly as possible. It was a great honor to work under their supervision. I am extremely grateful for what they have offered me.

I would like to express my appreciation to the University of Babylon and the College of Science for Women-Department of Computer Science for providing all sorts of support in conducting this thesis.

At last, I am grateful to my parents, my husband, family member, and friends for their patience, support, and encouragement during this period.

Abstract

A significant amount of data has been generated as a result of the widespread use of the IoT and the rise in internet-connected gadgets. Streaming such data to the cloud data centers for processing, analysis, and storage is the typical procedure. Such a stream is characterized by redundancy. Redundancy causes the wasting of computing resources at different levels.

The term "stream data" refers to an ordered sequence of d-dimensional data points.Which can be read-only once due to restricted processing capabilities such as time and memory.That are often processed in blocks referred to as windows with a given size known as the window size. The estimation of window size is affecting the resources required for the algorithm and its performance which is measured by accuracy and reduction rate.

This thesis investigates an effective resource management approach. Where data is gathered from various IoT devices at the fog layer and based on a clustering data reduction technique is implemented on a fog node.We used Genetic Algorithm (GA) to optimize window size for the reduction process. GA combined with Fuzzy Subtractive Clustering (FSC) as a fitness function is the proposed method to choose an effective window size. The experimental results show the impact of window size on the accuracy and reduction rate for each type of IoT data. However, the optimal window size makes the decision where each process should be implemented (IoT, fog, or cloud).

A model has been proposed based on optimal Windows Size that gives the required resources such as memory, storage, energy, and bandwidth that

used to determine the suitable layer. Using optimal window size at chosen layer the online reduction process is carried and sends results to the cloud. We used different datasets, results show a high reduction rate approximately (99%) and high accuracy (90%) which have been obtained by using Mean Square Error (MSE) as evaluation metrics.

Table of Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms

AEPs	Application entry points.
CBDR	Compression-based data reduction
COR	Correctness ratio.
CSP	Chaotic Time Series Predication
DB	Davies and Bouldin index.
DRCT	Data Reduction is based on the Compression Technique
FSC	Fuzzy Subtractive Clustering
FzGASCE	Fuzzy Clustering, Genetic Algorithm, Subtractive Clustering and Bayesian cluster.
GA	Genetic Algorithm
IoT	Internet of Thing
KFCM	Kernel-based fuzzy C-means.
LDA	linear Discriminant Analysis
MSE	Mean Square Error.
NIST	National Institute of Standard and Technologies.
PCA	Principle Components Analysis.
PSO	Particle Swarm Optimization.
QoS	Quality of Service
RM	Resource Management
SI	Silhouette Index
SLA	Service-level Agreements
SSE	SUM of Square Error.
TTDR	Two Tier Data Reduction
WCSS	Within Cluster Sum of Squares.

CHAPTER ONE

GENERAL INTRODUCTION

Chapter One

General Introduction

1.1 Introduction

The volume of stream data reported by the Internet of Things (IoT) devices and internet-connected gadgets has increased significantly. These data are frequently referred to as "huge data" [1]. In addition, because IoT data is typically redundant, cloud storage will be misused to keeping unnecessary or useless information. Therefore, data reduction and preprocessing must be performed close to the data source [2]. IoT is an Internet extension that collects, analyzes, and distributes data from IoT devices. Many IoT applications demand quick data processing, which provides a new challenge in existing IoT design, notably in terms of bandwidth requirements, low latency, and real-time interaction [3, 4]. Despite efforts to augment IoT applications with cloud computing capability, there are still unresolved challenges, such as mobility support, geographical resource distribution, real-time interaction, and low latency [5]. Consequently, there is a new trend in computing paradigms to relocate resources such as communication, computation, and storage to the network edge.

Data generated by IoT devices increases the requirement for processing and storage resources, which must be accounted for when evaluating the typical resource limitations of IoT devices. Few high-end nodes in the IoT ecosystem, such as edge devices or smart gateways, are required by three-tier designs [6]. System resources are classified as either physical or virtual. Physical resources include memory, network

bandwidth, and energy, just a few examples. The virtual resources comprise storage, encryption, and data fusion techniques and procedures.

Fog computing is a relatively computing paradigm that provides a distributed infrastructure for data processing near to-end devices. It is imperative to investigate how an intermediary layer of fog nodes between IoTs and cloud computing may enhance Quality of Service (QoS) [7]. One of the methods to efficiently economize resources is reducing communication and processing. Reducing the amount of data transmitted over the network eliminates redundancy from data. However, many techniques are implemented for data reduction, such as compression, prediction, clustering, classification, etc [8].

Clustering is a technique for data reduction and one of the most prevalent unsupervised learning approaches in data mining [9]. Clustering operates under the assumption that data tuples are objects [10]. It organizes a collection of things into a class of identical objects. Instead of the raw data, cluster representations of the data, such as centroids, are utilized for data reduction [9]. As a performance metric, clustering approaches must be optimized to obtain high accuracy and reduction rate. The Genetic Algorithm is one of the optimization techniques employed.

Genetic Algorithms (GAs) are well known as a kind of effective and robust optimization algorithms well-suited for multimodal functions [11]. The genes are the basic data structure of GAs used to represent the solutions to a problem.There is a wide range of ways among algorithms for performing adaptive search processes to find the nearest optimal solutions for optimization problems.This approach is a search method that is based on natural selection and genetic concepts.

In this thesis, we are interested to optimize the clustering algorithm for each sensor data to produce the best window size. Using a Genetic

Algorithm and clustering-based data reduction to drop the redundant IoT streaming data, knowing the amount of resources required for the optimal window size we need to determine where each process can be carried out (IoT, fog, or cloud).

1.2 Problem Statement

1) IoT devices produce huge data that is streamed to servers for processing.
2) Stream data is characterized by redundancy. Which causes wasted in resources such as storage, memory, processing and energy.
3) Fog computing-based architecture can play an important role in efficient resource management, by selecting the suitable layer (IoT, fog, or cloud) for processing individual data streams based on data type, variation rate, and data rate. The data reduction technique is achieve a high reduction rate while preserving data accuracy. In addition, preprocess performance is highly affected by how resources are managed.

1.3 Aims of the Study

1. Optimizing Window Size for each sensor data to produce optimal Window Size.
2. Eliminating redundancy in data streams by using one of the common data reduction techniques (clustering) with preserving accuracy.
3. Knowing the amount of resources required for the optimal window size lets us decide where each process should be carried out (IoT, fog, or cloud) .

1.4 Thesis Contribution

IoT devices produce huge data that are characterized by redundancy. Which causes wasted in resources such as energy, memory, storage, and bandwidth. So that we optimized the window size by using one of the optimization algorithms is Genetic Algorithm. Using one of the data reduction techniques is the fuzzy subtractive algorithm (FSC) as a fitness function to obtain optimal window size. While obtaining a high reduction rate and preserving accuracy through computing Mean Square Error (MSE) as evaluation metrics. Then knowing the amount of required resources for optimal window size let us decide where each process can be carried out (IoT, fog, or cloud) through the minimum difference between availability and required resources for each optimal window size. Real time reduction process implemented in chosen layer based on amount of resources that applied online FSC on stream data and send representative data to the cloud for processing.

1.5 Related Works

In this study, we are review the recent works that use optimization clustering.

The work in [12], combines the fuzzy clustering approach with a Genetic Algorithm (GA), subtractive clustering (SC), and Bayesian cluster validation to build a unique clustering method called fzGASCE. This approach identifies the correct number of clusters to form a given dataset and does it efficiently. They find that it performs better when comparing the results of fzGASCE against those of other strategies that use a similar approach. COR, EVAR, and EMIS used as measures of performance.

The work in [13], discover the SC algorithm's parameter ra (raduis), a Genetic Algorithm may use an objective function to estimate the

parameter's excellent value. Genetic Algorithms (GAs) can predict the best value for ra, even in noisy situations, because they are well-known as effective and resilient optimization methods. GA is used, and a new goal function is formed. This paper is utilized CSP as a performance metric. Our work is comparable to this in employing a Genetic Algorithm to optimize the parameters in the clustering methods.

The work in [14], provides a GA-based clustering method. Using a revolutionary initial population selection technique, this strategy can automatically determine the requisite number of clusters and uncover the appropriate genes. Allowing the initial seeds used in K-Means to adapt themselves in line with the requirements of the clustering process might lead to a better clustering result. Using them innovative fitness function and gene rearrangement approach, they create high-quality cluster centers to accomplish this. A sign test analysis showed that the results of the experiments showed that them approach is statistically superior to five more current methods on twenty natural data sets applied in this study based on six evaluation criteria, including Xie-Beni (XB), Sum of Square Error, COSEC, F-measure , entropy, and Purity (a measure of disorderliness).

The work in [15], provides a kernel-based (KFCM) fuzzy c-means (FCM) clustering method for the goal of optimizing clustering. Genetic Algorithm (GA) optimization is used in conjunction with an updated genetic algorithm and a kernel technique in this approach to optimization (GAKFCM). This technique optimizes the initial clustering center using an enhanced adaptive genetic algorithm. The KFCM approach is then utilized to guide classification, improving the FCM algorithm's performance. The first clustering center is optimized using both of these approaches. A Matlab simulation is used to evaluate the performance of the FCM method,

KFCM algorithm, and GAKFCM algorithm on test datasets. The results demonstrate that the GAKFCM algorithm can successfully overcome the drawbacks of FCM and greatly improve clustering performance. This study makes use of precision and recall as evaluation metrics.

The work in [16], is reduce the number of iterative steps required for clustering the data so that critical information may be obtained in less time. A genetic algorithm is now being used to minimize the total number of steps. The standard k-means method's step count can be lowered by employing GA, as has been found. Evolutionary computation is used to speed up the k-means process and also minimize the time it takes. The method improves the overall efficiency and reduces the complexity of the offered methodology. Genetic algorithm principles are used in our study to improve the clustering method's parameters. This study makes use of MSE.

The work in [17], Evolutionary Genetic Method (GA) based k-means algorithm was used for cluster analysis. The k-means algorithm is used to initialize the GA population by the methodology described thus far. By employing the GA operators, a new population will be formed. A new mutation can only develop at clustering sites with the greatest distance between them. Using this strategy, a variety of problems can be resolved. According to the results, the novel proposed method enhanced cluster-based data analysis. Various exam difficulties can be solved utilizing the demonstrated strategy. The sum of square errors (SSE) is utilized as a performance metric. This work employs Genetic Algorithm principles to optimize the clustering method's parameters.

The work in [18], proposes an unsupervised clustering strategy based on a genetic algorithm that seeks out the optimum cluster centers based on a k-means data framework. Using the genetic algorithm, the k-means clustering approach is less vulnerable and less likely to converge on a local

minimum when centers are randomly begun. Adding two clustering validity indices to the selection technique makes it possible to determine how many clusters should be produced automatically. Sixteen datasets from diseases and four from single cells are used to demonstrate the algorithm's usefulness. According to the results, this technique performs substantially better than the algorithms deemed to be highly advanced on the greater part of the datasets. Performance metrics used are WCSS, DB, and SI.

The work in [19], presents FZGPS as a unique algorithm that was built by combining FCM with the Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and the Subtractive Clustering (SC) techniques. They show that FZGPS operates well on gene expression datasets that are created as well as those that are collected via studies. This new algorithm efficiently and successfully decides how many clusters to build from a given dataset. The performance metric used was accuracy. This work is similar to this in using Genetic Algorithm concepts to optimize the parameter in the clustering algorithm.

The work in [20], have presented automatic genetic fuzzy c-means as a new algorithm that calculates the initial centroids and modifies the number of clusters as the study proceeds. The proposed algorithm is genetic but incorporates extra operators for crossover, mutation, and modified tournament selection. Further, three distinct cluster validity indices intercluster distance, intra-cluster, and correct ratio are the basis for a novel fitness function. To demonstrate the algorithm's utility in data quality, actual data sets are used.This work is similar to this in using genetic algorithm concepts to optimize the parameter in the clustering algorithm.

The work in [21], attribute selection was performed by the use of a genetic algorithm that was designed at K-Nearest Neighbor (KNN) for the

aim of the classification task. The purpose of the genetic algorithm is to organize features according to rank, with the notion being that an attribute's relevance to the classification task grows as its value increases. The research was done on the Indian dataset, which comprised 768 data. As a result of the test, they acquired the optimal combination with 1 attribute pick. A decrease in 3 and 4 characteristics from the K-Nearest Neighbor (KNN) accuracy before and after was lowered to 76.52 percent and 76.96 percent in this combination. Characteristics 1 and 4 may be divided by selecting these traits. The accuracy of the K-Nearest Neighbor (KNN) method before it is lowered is judged to be 76.52 percent, and after attribute, the reduction is found to be 79.57 percent. An assessment metric was utilized to determine how accurate the results were achieved by subtracting characteristics from the results obtained by maximizing outcomes after eliminating attributes. The evolutionary algorithm principles used to optimize the clustering algorithm's parameters are similar to those in this study. We combined the genetic algorithm with a fuzzy subtractive clustering algorithm to optimize window size as input data with obtained a high reduction rate and high accuracy.

Table (1.1) provides a summary of related works that discussed the optimization by using a Genetic Algorithm with data reduction methods (clustering), variables optimization, and the evaluation metrics in each research, arranged by the publication year.

Table 1.1: A Summary of Related Works

References	Reduction Technique (clustering)	Optimization variables	Evaluation metrics
(Le, Altman et al. 2012),[12]	Combine FCM with (GA), SC, and Bayesian cluster	Input data	COR, EVAR, EMIS
(Shieh, Chang et al. 2013)[13]	GA with FSC	Parameter	CSP
(Rahman and Islam 2014)[14]	GA with K-means	Input data	Xie-Beni (XB), Sum of Square Error, COSEC, F-Measure, Entropy, and Purity
(Ding and Fu 2016)[15]	GA with FCM	Input data	Precision and Recall
(Irfan, Dwivedi et al. 2017)[16]	GA with K-means	Input data	Mean Square Error
(El-Shorbagy, Ayoub et al. 2018)[17]	GA with K-means algorithm.	Input data	Sum of square error
(Nguyen, Louis et al.)(2019) [18]	GA with K-means	Input data	Within cluster sum of squares(WCSS), Davies and Bouldin (DB) index, The silhouette index (SI)
(Le and Vu 2020)[19]	FCM with (GA), (PSO), and (SC) algorithms	Input data	Accuracy

(Jebari, Elmoujahid et al. 2020)[20]	GA with FCM	Parameter	Inter-Cluster distance, Intra Cluster, and Correct Ratio(RC)
(Ihsan, Zarlis et al. 2021)[21]	GA with KNN	Parameter	Accuracy
Our work	Combined GA with FSC algorithm	Input data	Accuracy and Reduction Rate

1.6 Thesis Organization

There are five chapters in this thesis, one of them is the general introduction. The following is a summary of the remaining chapters:

Chapter Two: Theoretical Background

All basic principles are well covered by resource management in an IoT environment.

Chapter Three: Proposed Work

This chapter describes the experimental stages of the proposed method in optimization clustering.

Chapter Four: Experimental Results and Evaluation

Here, we describe and evaluate the proposed system's installation and performance and the outcomes of the proposed system's performance.

Chapter Five: Conclusions and Future Works

This chapter summarizes the study's results and makes recommendations for future research.

CHAPTER TWO

THEORETICAL BACKGROUND

Chapter Two

Theoretical Background

2.1 Introduction

The Internet of Things (IoT) connects devices to each other and the rest of the internet to generate more meaningful interactions between objects. Typically, the connecting procedure connects sensing, actuating, and control devices. Furthermore, these devices follow the appropriate communication protocols that are compliant with industry standards. In various efficient ways, IoT can achieve the goal of smart identifying, discovering, following, and controlling things [22].

The definition of the resource will dictate the resource management procedure. In computing, resources are everything that can be assigned to the system. Reducing communication and processing is one approach for efficiently economizing resources [23].

Data redundancy is frequently removed from data to reduce the amount of data delivered over the network. However, numerous data reduction techniques, such as compression, prediction, clustering, classification, and others [8], are used.

One of the data reduction strategies is clustering. Instead of raw data, cluster representations of the data, such as centroids, are employed for data reduction [9]. It is the most widely used unsupervised learning technique in data mining [9]. Data tuples are treated as objects in clustering [10]. It categorizes a group of objects into a category of comparable objects.

However, clustering techniques must be optimized to obtain further accuracy and reduction rate as a performance measure. Two directions of

optimization are usually proposed for clustering: algorithm parameters and data selection.

This chapter emphasizes the theoretical concepts, and some other concepts related to IoT, resource management, clustering, and optimization.

2.2 Internet of Things (IoT)

The Internet of Things (IoT) is a worldwide network connecting billions of heterogeneous smart devices and sensors, most of them are embedded, translucent, or invisible [24]. Sensors are crucial components of smart objects.

Sensors for the Internet of Things (IoT) are compact, inexpensive, and consume less electricity [25]. Most Internet of Things (IoT) devices require sensors to gather environmental or self-state information. Sensors for temperature, light, sound, motion, air quality, and other variables can be included in IoT systems to carry out specific functions. IoT allows objects to interact, share data, and make decisions. IoT uses ubiquitous and pervasive computers, communication technologies, sensor networks, and internet protocol standards to make common objects smart [26].

Huge data is a term used to describe a significant amount of structured, unstructured, or semi-structured data produced by IoT applications [27]. "Huge data" refers to a large volume of data that is hard to store, handle, and analyze traditionally [28]. Volume, velocity, and variety describe it. The amount of data produced from diverse sources, which continues to rise, is referred to as the volume of data. The amount of data is enormous, necessitating extensive storage capabilities [27, 28]. Variety refers to the fact that IoT data can be structured, semi-structured, or unstructured and can originate in various sizes and sources. The term

"velocity" refers to the high speed at which data is generated, which is required to deliver effective processing or analysis techniques [29].

2.2.1 IoT Architecture

In IoT architecture, there are three primary layers which IoT devices, fog, and cloud, as illustrated in figure 2.1.

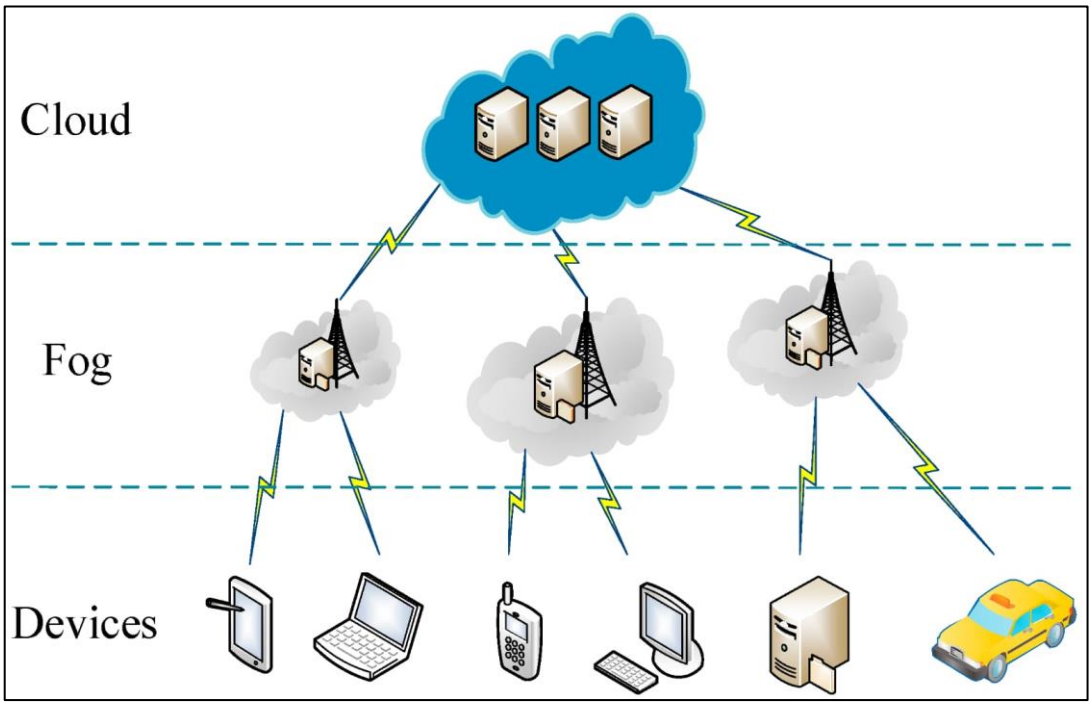


Figure (2.1): IoT Three-Tier Architecture [30]

As seen in figure (2.1), fog computing cooperates with cloud computing to form together a three-tier architecture (IoT-devices to fog to the cloud) [31]. As previously mentioned, fog computing does not ignore the role of cloud computing. Fog computing depends on cloud computing in performing complex processing [32]. Fog and cloud computing work together to provide high performance in such an IoT system.

The following is a description of the three layers of architecture:

1)Thing layer: This layer is closest to the user and the real sensor's surroundings. It is also called the perception layer. This layer has connected IoT sensors like environmental sensors and heart rate sensors. It also hosts

networked devices and allows data to be gathered and shared. The main job of these sensors and gadgets is to pick up data from physical objects or events in the area and send it to a higher layer (like the fog layer) to be processed. IEEE 802.15.4, Wi-Fi, Bluetooth, MQTT, and other communication protocols are used by each device or sensor. This communication protocol is needed for things to send data processing requests to other levels based on the data they have made or sensed. [33].

2)Fog layer: The essence of the fog layer is its proximity (one hop) to the clients and its distributed platform, which permits distributed processing across numerous fog nodes adopting fog computing standards [34]. Fog computing provides processing, communication, and storage capabilities at the Internet's edge by acting as an intermediary layer between customers and cloud data centers [35]. Fog computing brings "cloud-based services" closer to IoT data-handling end devices [36]. Cloud computing [37, 38] is not used as a central component. As a result, it can be utilized to meet requirements unrelated to cloud objectives [32]. Fog computing is made up of fog nodes that are widely dispersed. Fog nodes are servers that are placed close to IoT objects. These nodes are capable of a wide range of computational, storage, and networking tasks. Fog nodes allow fog computing to distribute cloud-based services across a vast geographical area [37]. Fog computing also allows for minimal latency, real-time interaction, and scalability, etc [35].

3)Cloud layer: Cloud computing provides large-scale resources, "e.g., logical resources such as databases or physical resources such as CPUs " or that clients may access on-demand and over the internet [39]. The National Institute of Standards and Technology (NIST) provides the following definition of cloud computing [40]: "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared

pool of computing resources such as networks, servers, storage, applications, etc. That can be rapidly provisioned and released with limited management effort or service provider interaction."

Cloud computing offers several financial and technological benefits, the first relieving business owners and businesses of the need to spend money on infrastructure and upkeep. Because cloud service providers do not supply capacity based on peak load, resources are released when the demand for service is low; cloud computing reduces operating expenses. Furthermore, renting resources is based on necessity, i.e., you only pay for what you use ("pay-per-use"). Finally, moving infrastructure services to the cloud shifts risk from the business to the infrastructure provider, who is typically well-equipped to handle it. In terms of the latter, cloud computing provides several technological benefits, including improved hardware and software resource utilization, performance "isolation," and flexibility [41].

Cloud computing services come in a wide range of forms, the most prevalent of them is storage (Storage as a Service). The ability to provide storage capacity for data through the internet is referred to as storage [42]. The cloud is the most convenient and cost-effective instrument for dealing with data created by IoT because of its almost unlimited, low-cost, long-term, and on-demand storage capacity [41, 42]. This solution allows data owners to migrate their data from traditional computers to the cloud. For a variety of reasons, cloud storage is preferable. The first is cost-effectiveness, which is beneficial to small and medium-sized firms. Data owners can save money on infrastructure installation and maintenance by storing their data in the cloud [43]. The data owners will only have to pay for the required storage space [44]. The second advantage of storing data in the cloud is that users can access it anywhere at any time. Users and small businesses cannot typically keep their servers as stable as the cloud.

Meanwhile, cloud storage suffers a security challenge, as unauthorized users may access to data owners [43]. As a result, fog computing is a suitable solution to this challenge.

2.2.2 IoT Data Stream

Stream data can be described as dynamic data. That is, data that changes fast over time is large, sequential, continuous, and real-time. Stream data has its characteristics when compared to traditional data. Traditional data is stable and static and may be retrieved and processed multiple times. On the other hand, the most prevalent methods for dealing with stream data are data mining approaches. Stream data mining techniques such as clustering, in general, may only produce approximate results and must adhere to the following constraints [45]:

1) **Single-pass:** In contrast to the conventional data mining techniques, which are continually applied to standard data (static data), stream mining techniques test each data object in the stream at most once and cannot be traced back.
2) **Real-time response:** Numerous time-sensitive applications must reply in real-time, i.e., the processing and decision-making time must be extremely quick.
3) **Bounded memory:** Stream data comes in an enormous amount, so it is infeasible to store it. Thus, processing these data, storing only summarized data, and possibly removing the rest is important.
4) **Processing:** Stream data have limited processing power and storage; interpreting and storing the voluminous IoT data creates a resource management challenge [48].
5) **Storage**: to be stored for processing algorithms; therefore, an IoT device's processing capacity implies that it also needs storage capacity to receive process stream data from IoT devices [46].

6) **Energy:** Resources may be managed more effectively through power-efficient hardware, low-power consumption devices, and sleep modes. In order to maximize resource utilization, energy is crucial. For networking to function, a sufficient amount of energy must be available. Data, protocols, packet size, processing speed, transmissions, and reception consume energy [47].
7) **Bandwidth:** Data streams are limited in terms of capacity. However, bandwidth management in IoT systems reduces energy, storage, and processing needs [47].

2.2.3 Data Stream Handling

As mentioned earlier, stream data are continuous, so only part of the total data streams can be processed, and approximate results may be obtained. This portion is treated as a data object time window. W [i, j] $=(x_i + x_{i+1}, \ldots\ldots, x_j)$, where i<j and i is start point, j end point. Different time-window styles are used to deal with stream data; examples [45]:

1) Landmark window: In this type, all data in the stream is important from starting time instant 1 to the last time instant tc. This type defines the window as W [1, tc]. All data objects in the window have the same level of importance, i.e., there is no difference between the old and current data.

2) Sliding window: In the sliding window, the window is defined as W [tc-w+1, tc]. This style focuses on recent data, where only the most recent data objects in the window are important, and the rest are removed. The mining result is based on the window size; when the window size is large, the window may have old information. In case of window size is small, the window may have incomplete data, and the model may have big variances. Hence model accuracy reduces.

3) Fading window: In this type, each data object is given a different weight depending on its arrival time, whereas the new data objects obtain higher weights. Using this type, the old data objects will have little influence or importance in the mining results [49, 50].

2.3 Resource Management

Resource management, which assures optimal resource use, and load balancing, reduces SLA violations, and improves system performance by lowering operational costs and energy consumption, is one of the primary challenges of cloud-based IoT environments. Resource discovery and sharing are crucial for IoT applications because they directly affect services and QoS. Significant issues arise as a result of the dynamic nature of IoT nodes in terms of communication and data capture, as well as the restricted computational and storage capabilities available in the fog compared to the cloud. In an IoT ecosystem, the massive amount of data generated by sensor-instrumented real-world items will place a high demand on processing and storage resources in order to be turned into meaningful information or services. Some applications will be sensitive to delay, while others would require complex processing, such as historical data and time-series analytics. Given the typical resource restrictions of IoT nodes, it's impossible to imagine a real-world, ultra-scale IoT ecosystem without a cloud platform or at the very least certain powerful devices, such as Smart Gateways [50] or edge/fog nodes [51]. The notion of resources in this complicated IoT edge cloud scenario might range from physical resources like memory, CPU, network bandwidth, electricity, and so on to software resources. Software resources include procedures for doing information fusion, detecting a complex event, and performing virtualization. The physical and virtual resources in the IoT ecosystem are summarized in table 2.1.

Table (2.1): Lists Physical and Virtual IoT Resources [33]

Platform	Physical Resource	Virtual Resource
IoT Devices	Temporary storage, Processing, Energy, Bandwidth.	Algorithms and Protocols used for data aggregation, and processing.
Fog	Medium Storage, Processing, Energy, Bandwidth.	Algorithms and Protocols are used for data aggregation, processing, encryption, and virtualization.
Cloud	High Storage, High-end Processing.	Algorithms and Protocols used for data aggregation, processing, encryption, virtualization, etc.

2.3.1 Key Activities for Resource Management in IoT

To describe the core activities of a generic framework for resource management in IoT, there are three tiers in an IoT ecosystem (figure 2.2) ,in which: (i) the bottom tier encompasses the things (IoT devices/nodes/smart objects), (ii) the top tier encompasses cloud nodes and (iii) an optional middle tier consists in Smart Gateways or edge nodes.

Applications connect to the system through a variety of application entry points (AEPs), issuing requests to the IoT infrastructure that are anticipated to be fulfilled. These entry points could be physical devices from the IoT bottom tier, such as cellphones (a resource-intensive IoT

device), personal computers running Web portals (and so accessing the system via the cloud), or even a gateway or edge node.

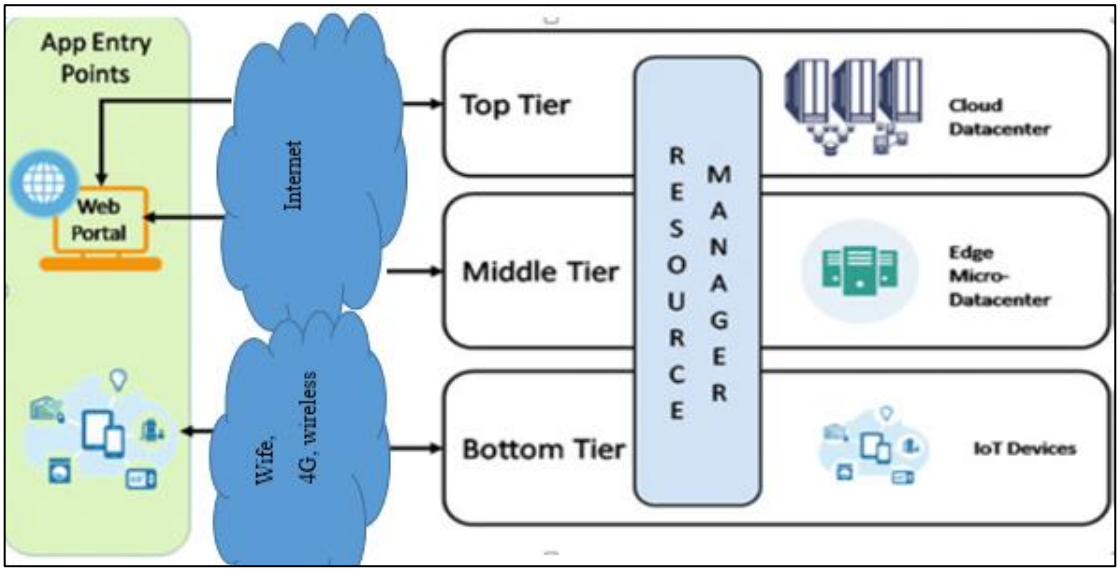


Figure (2.2): The Resource Management [33]

Besides this hypothetical physical model with two or three tiers, IoT systems have a logical tiered design. The RML is in charge of all actions relating to the system's resource management. The RML will be implemented as a resource manager (RM) subsystem that will be distributed among various hardware components in the IoT system's two or three tiers. As previously stated [53], in the context of cloud computing, the main difficulty for a resource management layer (RML) is to automate resource supply. In IoT, the same goal is pursued, but with extra requirements, as previously stated. The RML's ultimate purpose is to determine the optimal resource allocation scheme based on the most up-to-date system information in order to maximize the usage of system resources. This function necessitates a variety of ways for enlisting resources that fulfill the applications formally or informally stated QoS requirements. Although resource allocation is at the heart of an RML, there are additional actions that support it in order to ensure the system's correct and continuous operation.

Determining the layer of this process (IoT, Fog, and cloud) based on the availability and requirement of resources is a decision in resource management. The development of resource use is how to reduce data for the process and storage, one of the resource management methods is data reduction.

So that we used many equations to compute the resources that include energy, space complexity (memory), storage, and bandwidth. There are resources competed in the equations illustrated below:

1) **Energy** $= \mathrm{E}_{\mathrm{elec}} \times \mathrm{l} + \mathrm{E}_{\mathrm{amp}} \times \mathrm{l} \times d^2$ (2.1)[92]

Where $\mathrm{l} = \text{window size} \times \text{number of bits}$

$\mathrm{E}_{\mathrm{amp}}$ Signify the energy usage of multipath fading.

$\mathrm{E}_{\mathrm{amp}} = 100 \times 10^{-12}$

$\mathrm{E}_{\mathrm{elec}}$ Signifies the energy usage of the electronic circuit.

$\mathrm{E}_{\mathrm{elec}} = 50 \times 10^{-9}$

While the value of d (distance) we assumed 10 m.

2) Space complexity (Memory)

$$\text{Memory} = \sum_{\mathrm{i}=1}^{\mathrm{n}} \text{variable}_{\text{size}} * \text{number}_{\text{of}_{\text{variable}_{\mathrm{i}}}} \quad (2.2)$$

Where n=number of features, variable_size = number of bits required to store each variable (for python, a float is used. Which is 32 bits), number_of_variables = number of variables in feature.

3) Storage: Storage has no effect because it is always widely available, the clouds have a large storage and the fog also, and even IoT sensor uses the flash to store for a period of time, but it was set to know how many values.

$$\text{Storage} = \text{number}_{\text{variables}} * \text{variable}_{\text{size}} \quad (2.3)$$

4) Bandwidth: Bandwidth has been excluded from resources due to insufficient information such as data rate, number of IoT devices, and number of fog nodes.

2.4 Data Reduction

Dealing with large data volumes for analysis or processing is time-consuming and difficult. The practice of acquiring a lesser volume of data to represent the actual phenomena that provide the same analytical conclusions is known as data reduction. Horizontal and vertical data reduction techniques are also used, as well as dimensionality reduction, data compression, and numerosity reduction [60]. Dimensionality reduction, also includes PCA, Attribute Subset Selection, and wavelet transforms, is the process of reducing the number of variables (attributes) under consideration. There are many data reduction techniques such as:

1) **Data compression technique:** is designed to produce a compressed or reduced version of the raw data. The data reduction is termed lossless if the original data can be recovered from the compressed data without any loss of information.Whereas lossy data retrieval is defined as getting only an approximation of the original data (some data is missing). Data compression techniques can reduce data size and storage requirements by compressing data more efficiently [57]. Data reduction is based on the compression technique (DRCT), which works at the level of IoT sensor nodes [54]. A Two-Tier Data Reduction (TTDR) technique is proposed to work at two tiers of the network that are: sensor nodes and the gateway by using compression [55].
2) **Numerosity reduction:** is a data reduction technique that substitutes the original data volume with smaller, alternative representations of the data. This method might be parametric or non-

parametric. A model is used to represent data in parametric. The model is used to estimate data, with only the data parameters (rather than the actual data) being required to be retained. This category includes regression and log-linear models [8]. Nonparametric methods stores small representations of data and includes histogram, sampling and data cube aggregation [66].

3) **Feature extraction:** Feature extraction can reduce the amount of data. The process of extracting features entails transforming essential features into more significant ones. Extraction features can reduce complexity and offer a simple data representation for each variable in feature space that acts as a linear combination of important input variables [46]. The two most popular methods for data recovery are principal component analysis (PCA) and linear discriminant analysis (LDA) [58].
4) **The clustering technique:** is a data compression technique. It is one of the most often employed unsupervised learning techniques in data mining [9]. In clustering, data tuples are viewed as objects [10]. It organizes a set of things into a class of related objects. Clustering splits datasets into many groups based on the similarity of items (distances). When the distance between items inside the same cluster is the smallest, and the distance between clusters is the greatest, we are refer to the group of objects as a cluster. For data reduction, cluster representations of the data, such as centroids, are employed rather than the raw data [9].

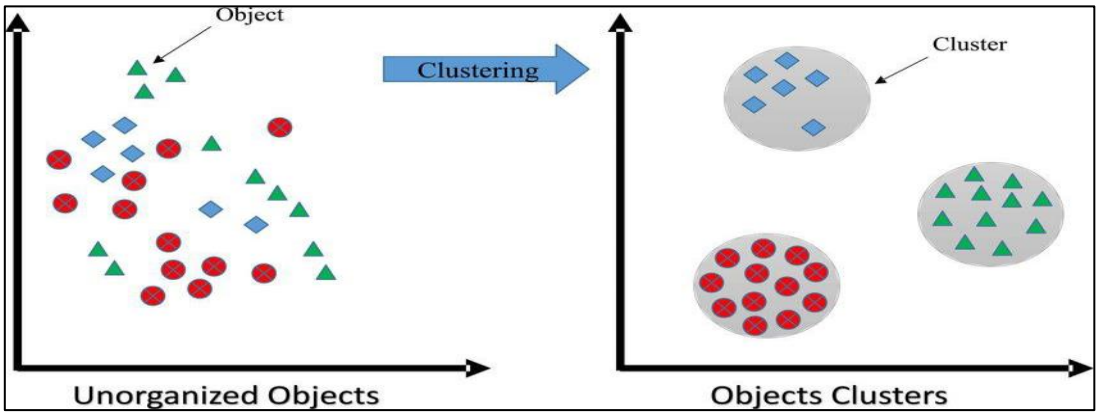


Figure (2.3): Clustering Process [59]

However, most clustering algorithms can be divided into Partitioning, or Hierarchical [61]. Hierarchical algorithms partition data items into layered clusters, resulting in a tree structure of clusters. Hierarchical approaches can be categorized as either agglomerative or divisive. In agglomerative, each data object is initially set up as a separate cluster, and subsequently, these clusters are merged into bigger clusters until all of the objects form a massive cluster. In the first step, division methods group all data items into a single cluster. Subsequently, the cluster is subdivided until each data object gets its cluster [63]. Propose a two-layer data transmission reduction strategy that capitalizes on both temporal and geographical data redundancy and is energy efficient [60]. Alternately, partitioning information is provided below.

a) **K-means algorithm**: This partitioning method divides the data into non-overlapping clusters, each representing a different cluster. One item is required for each cluster, and one object may only belong to one cluster. The algorithm tries to divide the dataset into a predetermined number of clusters using this method [62].

b) KNN (K-nearest neighbor) algorithm: One of the data reductions is data classification, which uses the Modified k-Nearest Neighbors' approach to classify the acquired sensed data. Classification is based on the perceived data that is most similar across all classes. The Data Transmission (DaT) protocol has been suggested for reducing the data transmission cost inside each sensor node by getting rid of the redundant data [70].

c) It is not necessary to know how many clusters there are to begin, but rather to identify clusters in a sorted sequence, such as the first huge cluster, find the second cluster, and so on. This type of clustering includes **subtractive and mountain clustering** [65]. As a result, SCA is the most commonly employed clustering algorithm [8].

Normalization, the first step in clustering, unites or scales data from many ranges into a single range. Its goal is to give all data attributes the same range of values. An attribute is normalized when its values are scaled to a single range, such as [0.0-1.0]. Normalization prevents qualities within a big range from outstripping those within a small range in distance-based approaches like clustering. Min-Max normalization is the most often used form of normalizing. The initial data collection is subjected to a linear transformation, but the relationship between them is preserved using Min-Max normalization. max_i and min_i are the attribute's highest and lowest values. According to this equation, A Min-Max normalization maps a value (v) of attribute i to (v̀) in the range [newmax_i, newmin_i] (2.4) [66]

$$\grave{v} = \frac{v_i - min_i}{max_i - min_i} (new\ max\ i - new\ min\ i) + new\ min\ i \quad (2.4)$$

Where min_i and max_i are the original minimum and maximum attribute values, respectively. While newmax_i =1.0 newmin_i= 0.0.

2.4.1 Subtractive Clustering Algorithm (SCA)

SCA is a one-pass adaptive algorithm [67]. The data set is first normalized in this algorithm. SCA does not require a predetermined number of starting clusters. It assumes that any point in the data collection could be a cluster center candidate [68]. For clarification, let $\mathbf{x_i}$ be a vector of n data points where $\mathbf{x_i} = \{\mathbf{x_1}, \mathbf{x_2}, \dots \dots \mathbf{x_n}\}$ the potential ($\mathbf{p_i}$) of $\mathbf{x_i}$ is computed using the following equation [69]:

$$P_i = \sum_{j=1}^{n} e^{-\alpha||x_i - x_j||^2} \qquad (2.5)$$

Where,

$\| \|$: is the Euclidian distance.

α: is a constant, where

$$\alpha = \frac{4}{r_a^2}$$

ra: is a radius that is a positive constant that defines a cluster center range. The radius (ra), represents the circle where neighboring data points are located, the data points beyond this circle will have little influence [70]. Data points' potential or density is a function of their distances from all other data points [68]. The density of its surrounding points determines each data point's potential to be a cluster center; if the data point has too many neighbors, its potential will be maximized. As a result, the likelihood of becoming a cluster center is increased [70]. This means that the initial cluster center will be picked based on the data point with the greatest potential [71]. The high-potential data point will be chosen as the first cluster center once each data point's potential has been calculated. Following that, the equation will be used to update the potential of each data point (2.6) [69]:

$$P_i = P_i - P_1^* e^{-\beta||x_i - x_1^*||^2} \quad \dots\dots\dots\dots\dots (2.6)$$

Where,

x_1^*: is the center of the first cluster and P_1^* It is potential.

β is a constant, where

$$\beta = \frac{4}{r_b^2}$$

rb: is a constant positive neighbor with an observable decrease in their potential measure [65].

Typically, rb is set (1.5 ra) to prevent acquired close cluster centers [88]. According to the equation (2.6), the potential of each data point is subtracted as a function of its distance from the original cluster center. No second cluster will exist since the data point nearest to the first cluster has a significantly smaller potential [68]. The data point with the highest remaining potential becomes the second cluster center after the potential of all data points has been updated by (2.7). After discovering the kth cluster center, the potential of each data point will be recalculated according to the equation (2.7) [69]:

$$P_i = P_i - P_k^* e^{-\beta||x_i - x_k^*||^2} \qquad (2.7)$$

Where x_k^* is the center of the K_{th} cluster and P_k^* their potential value.

Overall, Cluster radius (ra), accept ratio (ε⁻), reject ratio (ε_), and squash factor (rb) [72] are the four factors that influence the method. The radius is the most important factor in determining the number of cluster centers. Hence it must be carefully chosen. There is no best way to determine the radius value. According to [65], the radius value is considered to be in the range [0.1-0.9], with the radius value determined by the acquired error, specifically the Root Mean Square Error (RMSE). After that, the best radius value is picked as the one with the lowest RMSE. Many cluster centers derive from a small radius value and vice versa. As a cluster center, the accept and reject ratios are simply thresholds for accepting or denying points. When the accept (reject) ratio is set to a high value, the number of clusters increases (decreases), and vice versa [72]. As previously

stated, each data point is a potential cluster center that must be tested against a set of criteria to determine if it is a center point or not [71, 72].

Algorithm(2.1): General Steps of FSC algorithm

Step1: Normalized data set by the equation (2.4)

Step2: by equation (2.5) compute the potential (p_i) of all data points

Step3: Selecting center i by finding the value x_i with max potential pi

Step4: Revise the potential value of each data point by equation (2.6)

Step5: Testing whether the data point is a cluster center or not by the following conditions:

1) If $P_k^* > \varepsilon^- P_1^*$

Accept X_k^* as a center and continue

2) If $P_k^* < \varepsilon\_ P_1^*$

Reject X_k^* and terminate the clustering process

Suppose dmin is the minimum distance between X_k^* and all previously found centers

3) If $\frac{d_{min}}{r_a} + \frac{P_k^*}{P_1^*} \geq 1$

Accept X_k^* as a center and continue; otherwise, Reject X_k^* and set the potential at X_k^* equal to zero then, select the data point with the next maximum potential as new X_k^* and re-test.

2.5 Clustering Optimization

Clustering may be optimized using a variety of approaches, tools, and procedures. A Genetic Algorithm is an essential technique for reducing the number of parameters or input data that need to be optimized.

1) **Parameter optimization**: several studies are used to optimize parameters, such as ra (radius parameter in fuzzy subtractive clustering) optimized by a Genetic Algorithm [13]. K parameter in KNN (K-nearest neighbor) by the expectation Maximation algorithm (EM) in [74]. Combining FCM with the Genetic Algorithm (GA), the Particle Swarm Optimization (PSO), and the Subtractive Clustering (SC) techniques [75]. Attribute selection was performed using a genetic algorithm implemented at K-Nearest Neighbor (KNN) for the classification task [21].
2) **Input data optimization**: estimating the number of clusters and cluster centers by using PSO (Practical swarm Optimization) with FSC (Fuzzy subtractive clustering) [19]. Decrease the number of iterative steps necessary for clustering using GA with K-mean [16].The work in [20] applied GA with FCM. We use FSC with GA optimized in input data.

2.5.1 Genetic Algorithms (GAs)

GA is a search and optimization process guided by natural selection and genetics concepts [11]. Some basic notions from genetics are artificially taken and used to create search algorithms that are both robust and require minimal problem knowledge. Selection and recombination operators have well-defined roles in GAs. The selection operator directs the search, while the recombination operator generates new search regions. There is much inherent parallelism in genetic algorithms. GAs search

through complicated, vast, and multimodal landscapes for near-optimal solutions to an optimization problem's objective or fitness function. The parameters of the search space in GAs are recorded as strings (called chromosomes), and a population is a collection of such strings. A random population is formed at first, representing various points in the search space. Each string has a goal and fitness function that expresses the degree of goodness of the string. A handful of the strings are chosen based on the survival of the fittest criterion, and each is given many copies to go into the mating pool. To create a new generation of strings, biologically inspired operators like crossover and mutation are used for these strings. The selection, crossover, and mutation process continue for a predetermined number of generations or till a termination condition is satisfied [77]. One of the Genetic Algorithm types is simple Genetic Algorithm that used in this work:

1. Initial population is randomly generated.
2. The fitness function value is calculated for each individual of population by using FSC algorithm as fittnes function.
3. Select parents from the population.
4. Genetics operators (Crossover and Mutation) are applied to establish children and keep the best.
5. Replecment population.
6. Stopping criterion.

Implementation Overview applied in GA [78]:

1) I**nitialization**: In most cases, the algorithm begins with a randomly generated population. The nature of the problem determines the population size. We can encode with 0s and 1s. We shall, however, represent each gene with uniformly distributed numbers.

2) Selection operator: "Survival of the fittest" is the guiding concept in this case. Chromosomes are chosen for reproduction by this operator from the general population. An assessment function is used to find the better chromosomes, which are more likely to be picked for reproduction. Based on the situation at hand, the implementation of this fitness operator can be tailored. An objective function that applies to all chromosomes or a subjective function that applies certain criteria in a non-uniform manner can be used to evaluate which chromosomes are superior. A parent's chances of being selected are influenced by their fitness. In a single GA run, the best chromosome may not be selected. In contrast, if the GA is performed numerous times, the probability of selection will converge to its mathematically expected value. There are three primary approaches to identifying the most suitable individuals for breeding and selecting.

a) **Random selection**: is the most basic and efficient method of choosing parents. We shuffle the population using permutation and choose the first two individuals as breeding parents. This strategy is not advised because it contradicts "Darwin's Theory of Natural Selection," which states that individuals are chosen based on their fitness rather than randomly.
b) **Tournament selection**: This method is based on the chance of each individual being chosen. We run many tournaments among a randomly selected group of individuals, select one member from each group as the winner, and then repeat the process by grouping winners from the first iteration. In each iteration, the best member of each group has the highest chance of being chosen.
c) **Roulette wheel selection:** is the most popular and effective method for choosing parents. We are all familiar with how a

roulette wheel works at a casino: drop the ball, spin the wheel, and wait for the wheel to stop to see where the ball lands.

3) Crossover operator: This operator picks a random point in the string. The subsequences preceding and after that point are then swapped between the strings. As a result, two children's strings, i.e., chromosomes, are formed. Given two strings representing chromosomes, e.g., 11000 and 00111, a crossover operator will randomly select a position, e.g., second position. The resulting children of the crossover will be 11111 and 00000[78].

Parents: 11|000 and 00|111 Children: 11111 and 00000 Multi-point crossovers are simply crossovers with more than 1 position where crossover will occur.

There are three methods to perform crossover.

a) **Single-point crossover**: In this method, the residual sections of both parent chromosomes are exchanged to form two new offspring chromosomes cut at the same random place.
b) **Two-point crossover**: A method similar to the single-point crossover, the parent chromosomes are severed at two random locations, which is the only variation.
c) **Uniform crossover**: The method begins by selecting genes at random from both the parent chromosomes and genes that are not inherited. The gene that should be passed down is coded as 1, whereas the gene that should not be passed down is marked as 0. Then, as 0s and 1s, model them.

4) Mutation operator: This operator randomly changes one or some bits in the string representation of a chromosome. For example, a string 10101010 could be mutated in its first position to give 00101010. The

mutation operator controls the probability of mutation and the position of the mutation. Before mutation: **10101010**

After mutation: **00101010**.

5) Stopping Criteria:

a) The value of the termination criterion: The maximum number of repetitions is preferred over this option if we have an idea of the final criterion. As a starting point, below are a few examples of factors for dismissal: Remaining within the given time frame after receiving a fitness boost is acceptable if the population's variety falls below a particular point.

b) The population (generation) size: The population size should be proportional to the number of local maxima. The number of local maxima in some circumstances is unknown. As a result, a bigger population sample would be the best method.

A Genetic Algorithm's most essential parameters are outlined in the following sections:

a) Crossover probability: Crossover spots are randomly chosen within the chromosomes.

b) Number of crossover points: Multiple crossing locations are advantageous if the chromosome is lengthy.

c) Mutation probability: A gene's change in chromosomal value from 0 to 1 or vice versa is represented by this likelihood.

d) The maximum number of iterations: Genetic algorithms must have a maximum number of generations or a point at which they must cease iterating because of the constraints of computational resources. Limiting the number of iterations is advantageous if we do not know the end conditions.

2.6 Performance Metrics

There are several measures used to evaluate the performance, each of these terms is described in detail:

1) Reduction Rate: Subtractive clustering method is used for each window to minimize the amount of data. Equation (2.8) is used to calculate this:

Reduction rate = $1-\frac{\sum_{k=1}^{w}(\#C_i)}{w*windowsize}$……. (2.8)

Where: C is the number of centers resulting from clustering. [8]

And w is the number of windows (blocks) (w) = $\frac{dataset\ size}{window\ size}$.

2) Accuracy: Accuracy measures the ability of the representative points to reflect the original data. That is computed through mean square error. MSE can be expressed as in equation (2.9)

Accuracy =$\frac{\sum_{i=1}^{n}\sqrt{(|X_i-\hat{X}_i|)^2}}{\text{n}}$…… (2.9)

Where n is the number of items in the original dataset

X_i is the item i in the original dataset

$\hat{X}_i$ is the item i in the restored data in the dataset[89].

There are many evaluation metrics used to optimize clustering algorithms such as [79-89].

2.7 Summary

Some ideas related to the Internet of Things (IoT), clustering, and genetic algorithms are introduced in this chapter. First, this chapter presents a comprehensive introduction, including IoT's importance, architecture, stream data, and window styles. The concept of resources management and key activities for resource management are explained. As well as the overview of data reduction, clustering, and the work of FSC are introduced.We shown optimization using Genetic Algorithms, including selection, crossover, and mutation, and some of its advantages and services are also covered. The performance metrics are illustrated, including reduction rate, accuracy, and others.

CHAPTER THREE

EFFICIENT RESOURCES MANAGEMENT APPROACH

Chapter Three

Efficient Resource Management Approach

3.1 Introduction

One of the resource problems is the data management method. Resources are not equal in IoT layers: end device, fog, and cloud. Therefore knowing the number of resources required for each process lets us decide where each process should be carried out. IoT devices produce data that is streamed to servers for processing. Such a stream is characterized by redundancy. Redundancy causes the wasting of resources. To get rid of redundant data streams, one of the data reduction techniques is used. We showed in chapter two that one of the efficient reduction techniques is clustering namely the Fuzzy Subtractive Clustering (FSC) algorithm since it required no pre-knowledge about the number of estimated clusters in the dataset. It detects groups of similar items in the dataset (blocks of stream). Then, a set of centroids are considered as the representative data for the whole data block to be sent to next process of the given application.

Moreover, clustering could be optimized to obtain the best centroids and consequentially the required accuracy and the reduction rate. One of the critical parameters of clustering is the size of the data block usually referred to as widow size.

Many optimization techniques are known to researchers and are also reviewed in chapters one and two. However, since we are dealing with a single parameter for simplicity, we suggest using the Genetic Algorithm GA for this study.

We propose to use the reduction technique (clustering) as a fitness function to optimize window size within GA. Buffering the right size block will suggest the best layer to run the process, since block size will affect required memory, communication, and energy. This study shows the impact of suitable window size on the accuracy and reduction rate for each type of sensor. This chapter explains the proposed method, its main modules, and the data flow.

3.2 The Proposed Method

The proposed method consists of two dependent processes: window size detection and real-time reduction activation. The first process is implemented offline periodically and relatively with long intervals (every week, month, or when required) that employs GA and FSC on a collected dataset executed on a fog node. The objective is to obtain the optimal window size for each sensor providing high accuracy and a high reduction rate. Later, it calculates the required resource for the reduction process of each sensor based on obtained window size.

The second process (the actual data reduction) is implemented online in real-time with a determined window size. It is deployed at all layers but activated or deactivated at a given layer by process one. The process buffers data of a given IoT device in blocks with sizes equal to the determined window size. Then, applying the reduction technique to choose representative data that will be sent to the next application process of that IoT device.

Figure (3.1) shows the diagram of the proposed method of managing IoT platform resources. However, knowing the resources required to determine the suitable layer to carry out (IoT, fog, cloud) is based on the resources available in each layer. Resources include memory capacity, storage, bandwidth, energy, and others.

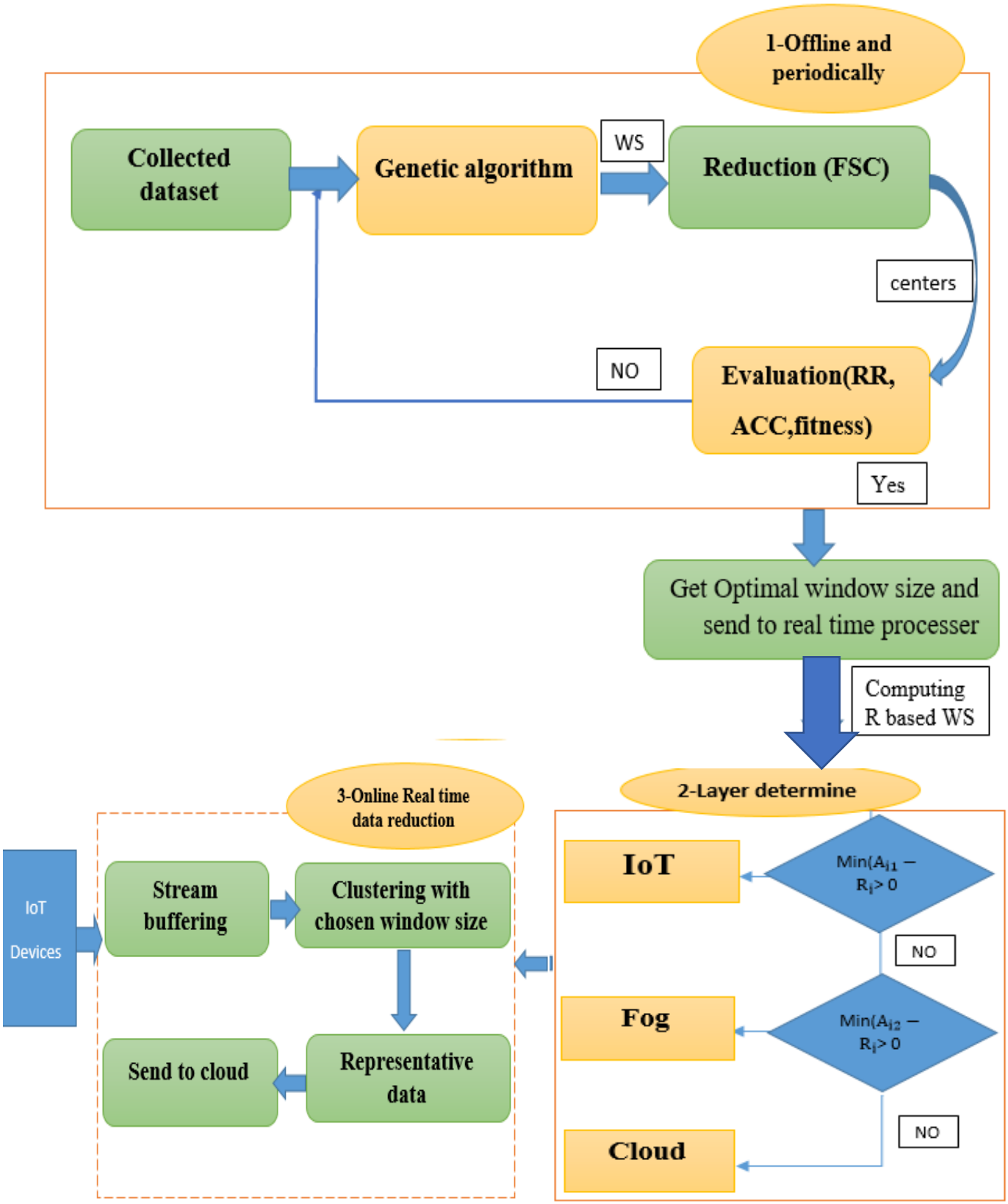


Figure (3.1): The Block Diagram of the proposed method.

Where RR represent reduction rate, ACC is accuracy, and R represent required resources computing by equations allustrated in chapter 2. While A represents the availability resources based on type of devices.
Window size effects on resources determine that need an algorithm. Then select the layer to implement the algorithm. The algorithms of each process will be explained in detail in the next sections.

3.2.1 Fog Process

The process in the fog node is essential for optimal management of IoT data and resource management. It consists of several sub-processes:

1) Data collection

In a fog node (server), a process is deployed to collect IoT data for long durations (for days for example). This data is stored in a database for analysis and detection optimal block size should be used in real-time process.

2) Genetic Algorithm

The GA is used to detect the optimal window size by applying the reduction technique to different window sizes on the same dataset. It gradually evolves optimal value through its operations. In our proposed method, GA performs the following process with given settings:

a) **Representation**: the required solution is the window size. It is an integer value ranging between two boundaries minimum value and maximum values, i.e. a chromosome represents one solution. In this study, the decimal value of the window size is represented as a string of genes with a given length n encoded as binary values (0s and 1s).
b) **Initialization**: the binary values of each chromosome's genes are generated randomly upon the beginning of GA.

c) **Evaluation:** the fitness function of each chromosome is evaluated by :

   1) Decoding the window size of the given chromosome into decimal value.

   2) Applying the FSC technique using the given window size for the whole dataset.

   3) Gathering the representative centers from each data block.

   4) Compute evaluation metric: reduction rate and accuracy as explained in section 2.6.

   5) Compute fitness function using the equation 3.1:

   $$\textbf{Fitness} = 2 - (\alpha \times \textbf{reduction rate} + \beta \times \textbf{accuracy}) \quad (3.1)$$

   Where α and β are two weighting parameters to make a tradeoff between reduction rate and accuracy.

d) **Stop GA** if the Termination condition is met; otherwise, continue. This study uses the stability of fitness value as a termination condition, i.e., when not all fitness values evolve for several iterations.

e) **Selection:** this study uses tournament selection as a selection strategy for selecting the fittest candidates from the current generation for reproduction. The selected candidates are copied for the mating pool.

f) **Recombination:** combining elements from two candidates from the mating pool to create new solutions (i.e., offspring). This study uses a one-point crossover with a given crossover probability determined empirically.

g) **Mutation:** mutations involve one change in particular gen of each chromosome with a small mutation rate (Pm) that is also determined empirically.

h) Replacement: Because of selection, recombination, and mutation, the offspring population replaces the original parental population keeping the fittest parent moving across generations.

Algorithm (3.1) explain the employed GA for our proposed method.

Algorithm (3.1): The Proposed Genetic algorithm

Input:

IoT device dataset.

Set α and β such that α + β=1

Set Pc ≈1

Set Pm ≈0

Output: *an optimal window size for reduction of the dataset.*

Steps

1: *Randomly generate a population. Each describes a window size solution.*

2: *Compute the fitness value.*

3: *If the stop criteria are met, go to step 8.*

4: *Do tournament parent selection.*

5: *Apply the crossover and mutation operators with the probability Pc and Pm respectively.*

6: *Replace population*

7*: Go to step 2.*

8: *Select the 'best' chromosome (window size) from the population as the optimal solution.*

3) Reduction Process

The data reduction technique is carried out twice for different purposes: one is periodically performed offline in a fog node to detect optimal block size. The other is carried online where it is needed based on the decision made by the first one. However, as a process, it consists of several steps as explained below**:**

a) Data normalization

IoT Stream Data falls within different ranges. Thus, a normalization technique is used to scale it into one range. This converts all the features the dataset to the same range. The normalization process is performed on each sensor block by equation (2.4) mentioned in Chapter2 (Section.2.4). The Algorithm (3.2) demonstrates the normalization process.

Algorithm (3.2): Normalization

Input: *data block*
Global Max value and Min value for the given IoT device data
Output: *Normalized data block*
1: *Set new upper and lower range boundaries (m,n) //* ***specific range***
2: For *each item in the data block,* ***do***

$$Y = \frac{X - X_{min}}{X_{max} - X_{min}} * (m - n) + n$$

3: End for

b) Clustering process

This study uses clustering as a data reduction technique (clustering). Namely, Fuzzy Subtractive Clustering (FSC) is used to cluster data blocks and obtain each cluster center as a representative item for other items in that cluster. FSC does not require a predefined number of centers. Each data point in the dataset must be regarded as a potential cluster center. Accepted cluster centers are selected by computing the potential of all data points.

Then, the highest data point potential will be selected as the first center. The details of this algorithm have been presented in Chapter 2 (Section.2.4.1). FSC algorithm is illustrated by the algorithm (3.3).

Algorithm (3.3): Fuzzy Subtractive Clustering

Input*: data block, Radius, accept Ratio, reject Ratio*

Output: *cluster centers*

Initialization:

Let accept ←True

1*: distance ←adjuncy matrix of data block*

2: *pot ←Potential (distance, ra)* ***// ra is a radius that defines a cluster center range***

3: *index ← 0*

4: *p1←pot[index] //* ***p1 is the maximum potential***

5: *currentMax←p1*

6: *current Index ←index*

7: ***While*** *accept and currentMax> 0* ***do***

8: *potentialRatio ← currentMax / p1*

9: *accept ←False*

10: ***If*** *potentialRatio> ep1* ***then // ep1 is the accept ratio***

11: *accept ←True*

12: ***Else***

13: ***If*** *potentialRatio< ep2* ***then // ep2 is the reject ratio***

14: *Reject point and stop*

15: ***Else***

16: d_{min}*←Distance[index][currentIndex]* ***// dmin is minimum distance***

17: ***For*** *i←1 to length (centersVector)* ***do***

18: *dist ←Distance[centersVector(i)] [currentIndex]*

```
19:                 If d_min<dist then
20:                         d_min ←dist
21:                 End if
22:         End for
23:   End if
24:   End if
25:    If d_min / ra + potentialRatio ≥1 then
25:             accept ←True
26:   Else
27:             pot[index] ←0
28:             index← currentIndex
29:            currentIndex← maxindexPotential(pot)
30:            currentMax ←Pot[currentIndex]
31:   End if
32:   If accept then
33:          centerVector.add (currentIndex)
34:         updatePotential (currentIndex, ra)
35:         currentIndex← maxindexPotential(pot)
36:         currentMax← Pot[currentIndex]
37:   End if
38:   End while
39:   return  centerVector
```

c) Evaluation Metrics

Reducing the amount of stream data need not scarifying the semantics of the data consequentially determining where a process is recommended to carry out is impacting the usages of available local resources and

reducing the communication which in turn economizes both bandwidth and energy. Two measures have been used and explained in the following:

1) **Reduction rate:** The reduction rate is the percentage of data reduced by applying FSC reduction techniques.That is illustrated in chapter 2 section 2.6.
2) **Accuracy:** Accuracy assesses the ability of the representative points to reflect the original data accurately. This measures the difference between the raw and data restored using only representative data to see if the reduction process leads to the loss of information. As illustrated in chapter 2 section 2.6.

Data restoring is copying backup data from secondary storage and restoring it to its original location or a new location. Restoring algorithm is applied based on the computing distance between centers and original data. So we need two arrays, the first for the original data and the other for the center data. The difference between the two arrays represents the restored data amount. Then we compares each data point with the center. If this value is close to the center, it will be summed in a group. Then, all the original values are replaced with the center value relative to the asset of values. This step is used to know the location of the centers in the original dataset and to determine the amount of loss in data.

Algorithm (3.4): Data Restoring

Input:

D the original data block,

C set clusters centers

Output: D' the restored data block

1: *let* $n \leftarrow |D|$

2: *let* $m \leftarrow |C|$

3:***For*** *each* $d_i \in D$ *{i=0..n }*

4: *Set* $k \leftarrow 0$;

5: *set* $d_0 \leftarrow distance(C_0, D_i)$

6: ***For*** *each* $c_j \in C$ *{j=1..m }*

7: *set* $d_j \leftarrow distance(C_j, D_i)$

8: ***If*** $(d_j < d_0)$:

9: $k \leftarrow j$

10: $d_0 \leftarrow d_j$

11: ***End if***

12: ***End for***

13: $\acute{D}_i \leftarrow C_k$

14: ***End for***

15: ***Return $\acute{D}$***

3.2.2 Layer Determination

Layer determination is a decision of activation of the real-time reduction process at a given node (IoT, fog, or cloud). The space complexity depends on the window size. After detection of the optimal window size by the first process. It becomes easy to calculate the required

resources. Hence, the following algorithm is used to obtain the minimum required resources vector R.

On the other hand, the available resources in each node are priory known. The decision to activate the real-time process at some layer node by comparing required and available resources which are default values.we apply normalization for both vectors and calculate the distance, which represents the minimum difference between them. If resultant value is positive then first vector is larger, otherwise the second is larger. Based on this comparation determine where process carred out (IoT, fog, or cloud). The required resources include computing memory, storage, and energy. That is illustrated in chapter 2 section (2.3.1).

Algorithm (3.5): Layer Determination

Input:

Sensor type T

Window size W

Output

minimum required resources vector R(window size)

L: Which platform layers (i.e. IoT devices, Fog, or Cloud).

Let A available resources matrix

1: R[1]: *energy complexity, using equation (2.1)*

2: R[2]: *space complexity, using equation (2.2)*

3: R[3]: *storage complexity, using equation (2.3)*

4: If (Min($A_{i1} - R_i > 0$)):

5: L= IoT

6: Else

7: If (Min ($A_{2i} - R_i > 0$)):

8: L= Fog

9: Else

10: L= cloud

11: End if

12: End if

13:Publish{topic:R, WindowSize:W, Sensor type:T, layer:L}

Publishing either using web page or broker(1).

3.2.3 Real-time Reduction

After determining the layer according to resources required to window size at fog node.It is publishe the choosen layer for process data of given IoT data. Hence, it is assumed that real time reduction process is deployed every where (IoT, fog, or cloud). Known the determined layer is made by all processes frequently query fog node. Only process at determined layer are activated.

We used the same fuzzy subtractive clustering on stream buffering with optimal window size value that is obtained from fog after using a genetic algorithm as an optimization method. As a result, representative data obtained from clustering send to the cloud for processing. This process is implemented online in real-time.

(1) Broker: logically a centralized component that handles all the accesses between different applications .it is means to find out where doucments can be stored best .that collects data on resource usage and availability for anumber of nodes that are in each others proximity will allow to quickly select a node with sufficient resource.

3.3Summary

In this chapter, we introduced an efficient resource management approach. This process is implemented in a fog node to select the optimal window size and includes sub-processes such as collecting data from sensing devices and then stored in the buffering database. A Genetic Algorithm is used to obtain the optimal window size. The reduction process is applied to normalization data using fuzzy subtractive clustering as a fitness function. Then, it computes evaluation metrics such as reduction rate and accuracy. The layer determination process based on the resource amount required for each window size results from the sensors implemented offline. Finally, the reduction process is executed on each window size, by using the same fuzzy subtractive clustering algorithm. Then sending the representative data results from clustering to the cloud for processing.

CHAPTER FOUR

RESULTS AND DISCUSSION

Chapter Four

Results and Discussion

4.1 Introduction

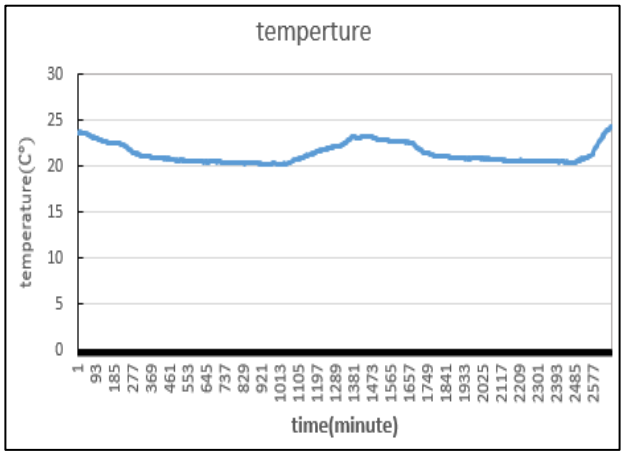
This chapter presents the results obtained from implementing the proposed method that is described in chapter 3 and discusses the performance. Also, a justification for all the results obtained from all implementation. This work involves using a personal computer hp that has specifications such as Windows 10 Pro, Intel(R) Core(TM) i7-3632QM CPU @ 2.20GHz, and memory size (RAM):6.00 GB and 64-bit Operating System. The software used is Python version 3.6.9 executed under google Colab platform; used with Python to implement code easily. The method relies on open source libraries such as Open. The NumPy basic package is utilized. Pandas is a data structure tool that is utilized. This chapter involves section one illustrating different datasets. Then represent the results after obtaining the suitable window size for each dataset, the genetic algorithm applied results and determined the layer for each window size based on the required resource amount. Then show results of evaluation metrics such as reduction rate and accuracy.

4.2 Dataset

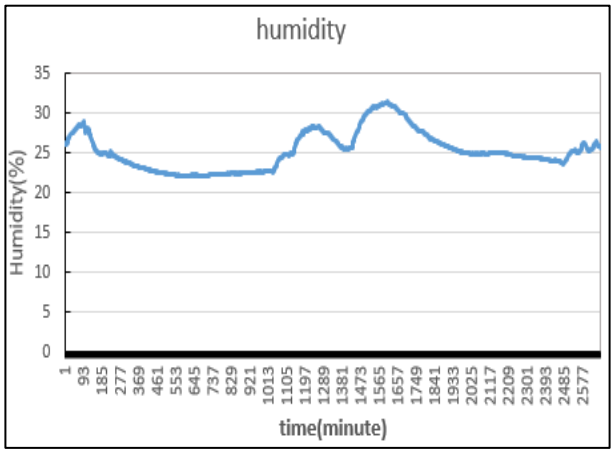
Several datasets were used to test the proposed method, such as:

1) **Occupancy room [90]**: this dataset consists of 2666 rows and four columns: temperature, humidity, light, and CO2. So, the temperature in Celsius, relative humidity in percentage (%), light in lux, and CO2 in ppm. Here, the time-stamped images of environmental factors, including temperature, humidity, light, and CO2, were acquired every minute to determine the ground truth occupancy. The range of

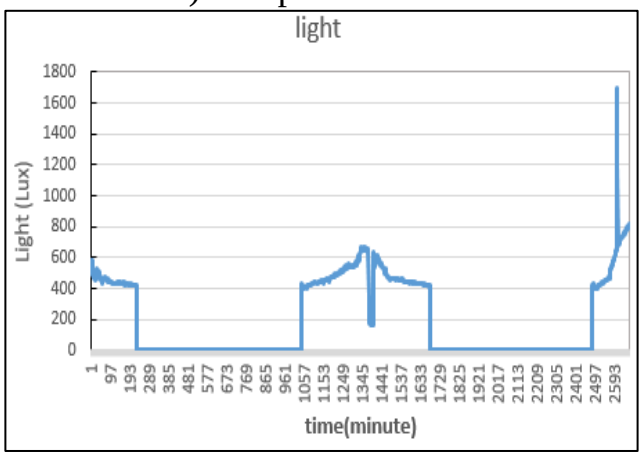
temperature data (0-30), range of humidity (0-35), range of light data (0-1800), and range of CO2 (0-1600). The y-axis represents the temperature data plotted against time in minute as the x-axis. That illustrated in figure (4.1).



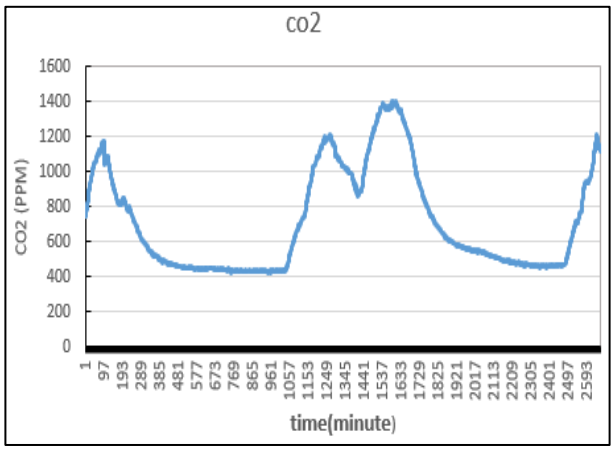
a) Temperature data



b)Humidity data



c)Light data

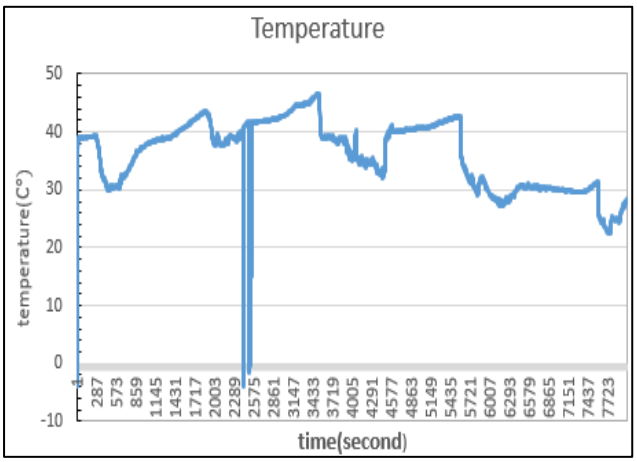


d)CO2 data

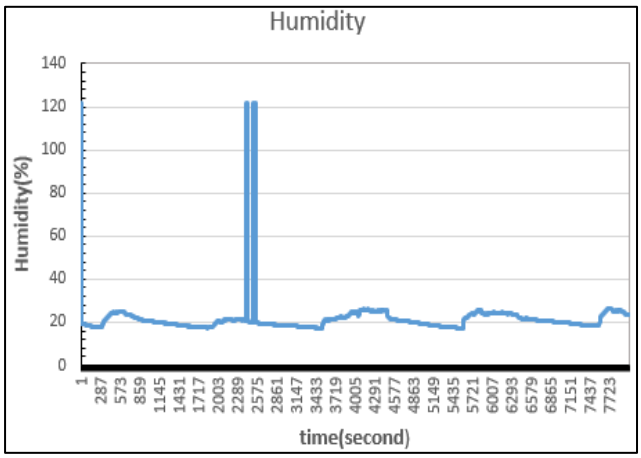
Figure (4.1): Description of Occupancy Room Dataset

2) **Intel Berkeley Research Lab Dataset [56]:** contains information about data collected from 54 sensors deployed in the Intel Berkeley Research Lab between February 28th and April 5th, 2004. Mica2Dot sensors with weatherboards collected time-stamped topology information and humidity, temperature, light, and voltage values once every 31 seconds. Data was collected using the TinyDB in-network query processing system, built on the TinyOS platform. It includes a log of about 2.3 million readings collected from these

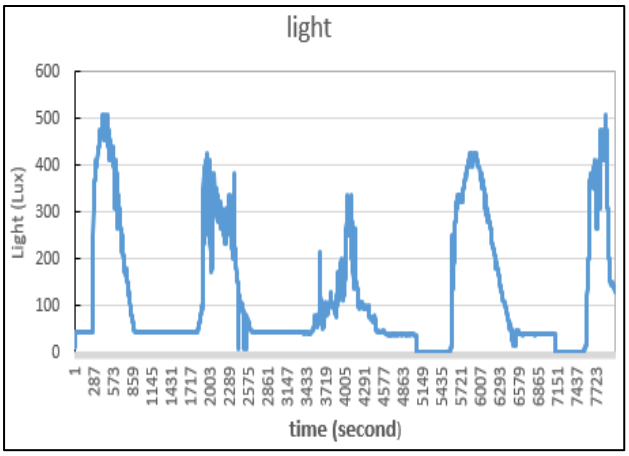
sensors.The temperature is in degrees Celsius.The temperature corrected relative humidity, ranging from 0-100%. Light is in lux (a value of 1 Lux corresponds to moonlight, 400 Lux to a bright office, and 100,000 Lux to full sunlight). Voltage is expressed in volts, ranging from 2-3; the batteries, in this case, lithium-ion cells that maintain a fairly constant voltage over their lifetime. Note that variations in voltage are highly correlated with temperature. Dataset volume used is 8000 rows and four columns as showen in figure (4.2).



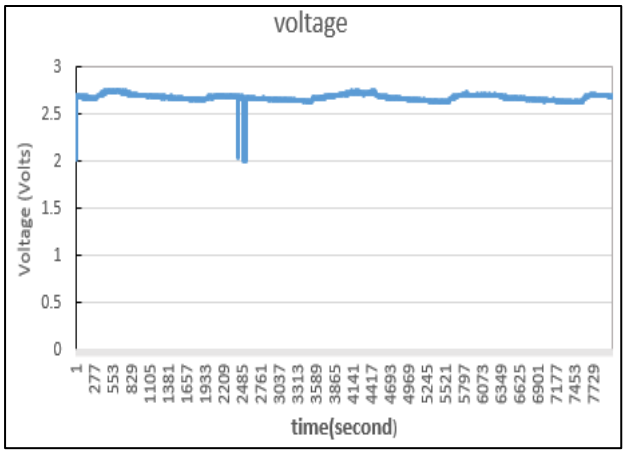
a) Temperature data



b) Humidity data



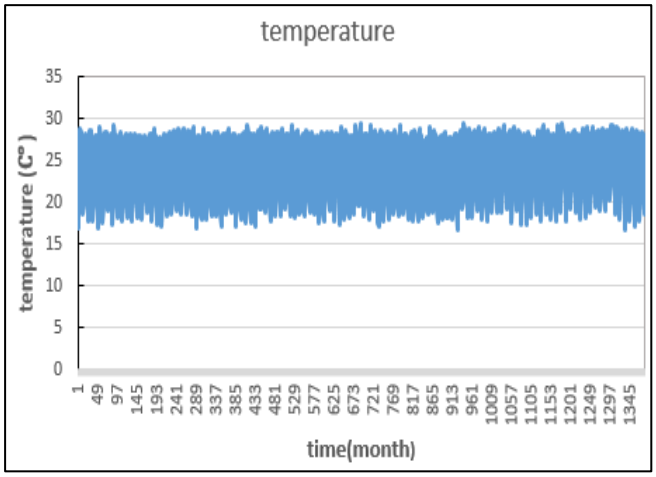
c) Light data



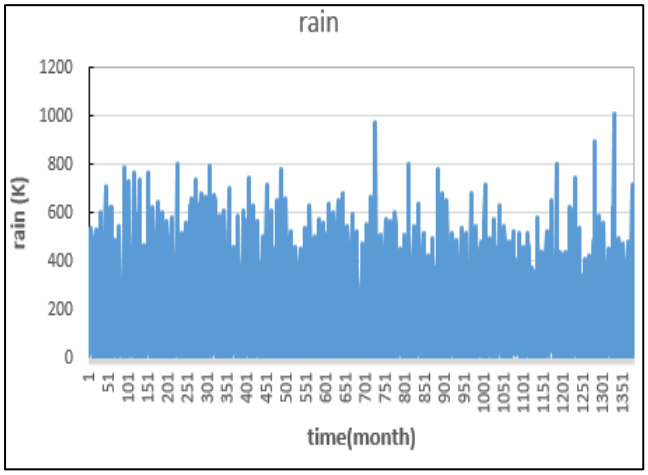
d) Voltage

Figure (4.2): Description of Intel Berkeley research lab datase

3) **Weather Dataset [76]:** This dataset contains the monthly average value of Bangladesh's temperature and rain from 1901 to 2015. It consists of two-column temperature and rain and contains 1381 rows representing data for each month. Temperature ranges (0-35) in Celsius while rain ranges (0-1200) in kilo as showen in figure (4.3).



a)Temperature data

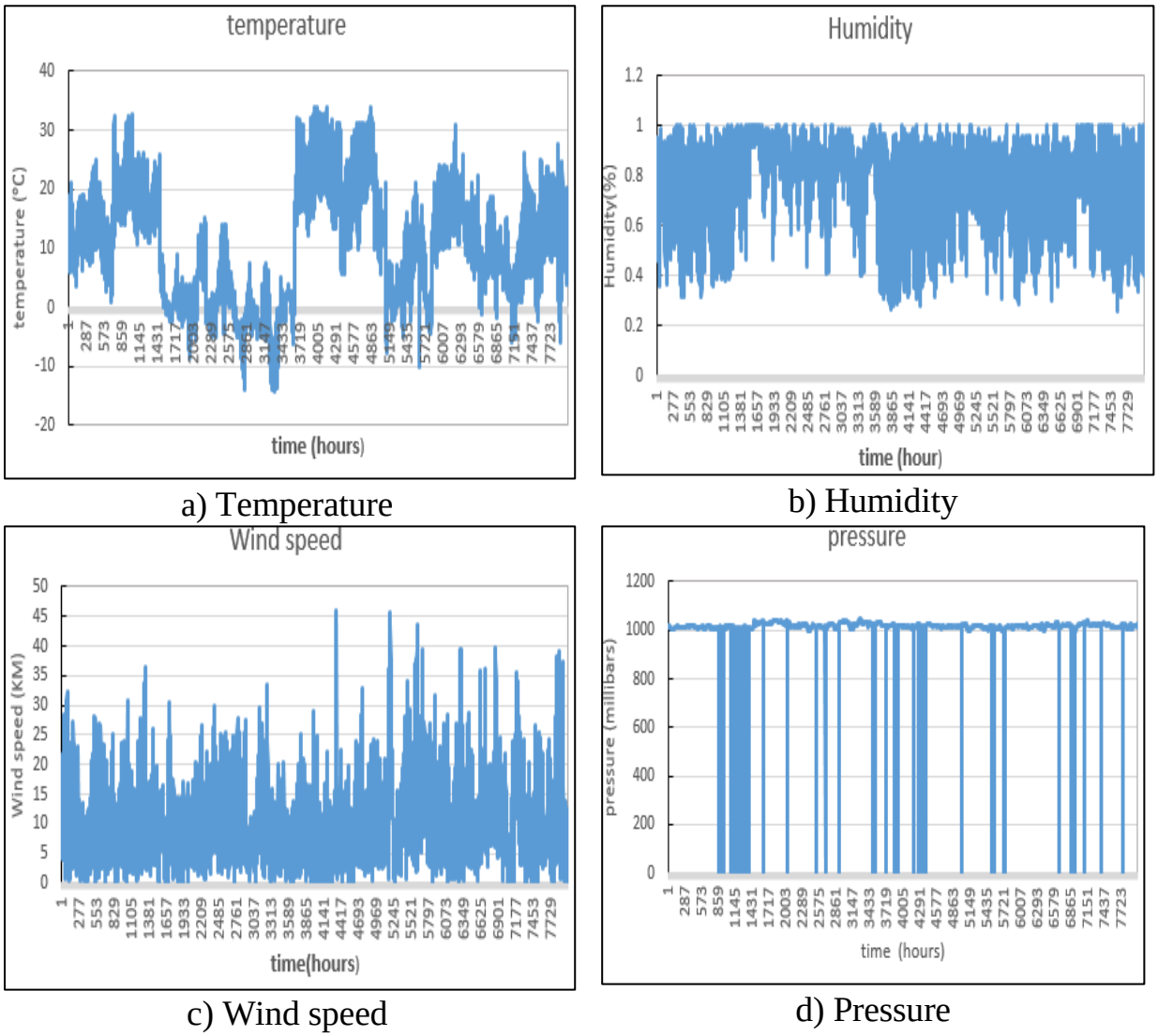


b) Rain data

Figure (4.3): Description of Weather Dataset

4) **Weather History [73]:** This dataset consists of four columns temperature in Celsius, humidity sensor, Wind Speed (km/h), and

Pressure (millibars). That contains 8000 rows, which represent data for each hour as showen in figure (4.4).



a) Temperature b) Humidity

c) Wind speed d) Pressure

Figure (4.4): Description of Weather History Dataset

4.3 Results

This section presents the results obtained from implementing the optimization of window size by using a genetic algorithm using one of the data reduction techniques (clustering), such as the Fuzzy Subtractive Clustering for experiments each with a different dataset: occupancy room,

Intel Berkeley research lab, weather dataset, and weather history dataset. These results obtained when α=0.7 and β=0.3 for all dataset.

We need to demonstrate genetic algorithm parameters to find the ideal window size. These parameters are illustrated in the table (4.1), which consists of population size is taken value through experiments, mutation, and crossover probability is 0.01 and 0.9 through try and error to reach the suitable results, and stopping condition is fitness stability for five generations in case no change occurs in fitness value or maximum fitness experimentally.

Table (4.1): Genetic Algorithm parameters are used to obtain the optimal window size.

Parameters	Value (s)
Population Size	10
Reproduction operator	Crossover and mutation
Crossover type	Single point
Selection type	Tournament selection
Mutation type	One change
Mutation rate	0.01
Crossover rate	0.9
Stopping criteria	Fitness stability for 5 generation or maximum iteration.

The resources in each layer are distributed within ranges illustrated in the table (4.2). This table involves three layers (IoT, fog, cloud) and resources (energy, memory, storage, and bandwidth). That involves the properties of the layers generally. These values are defaults and change depending on the device used.

Table (4.2): The Resources availability at each layer.

Resource	IoT layer	Fog layer	Cloud layer
Energy	4.8 J	110 J	2020 J
Memory(space complexity)	1-10 MB	15-17MB	8-64 GB
Storage	1MB	14-20MB	75.6 TB
Bandwidth	2MBps	30-73MBps	10 Gps

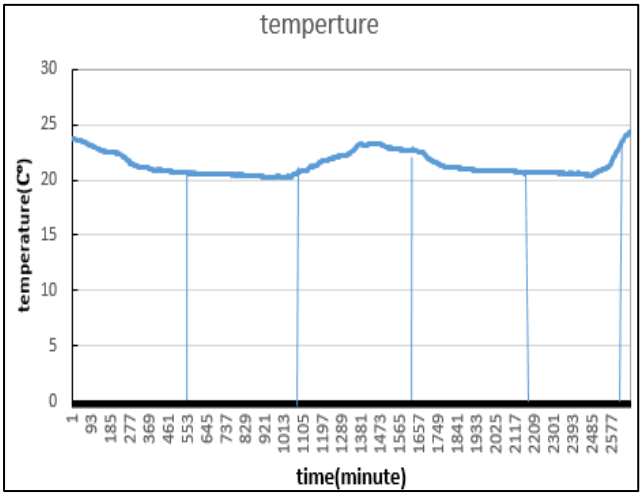
4.3.1 Results of Occupancy Room Dataset

The description of the reading for each sensor is presented in figure (4.5), which involves four sensors (temperature, humidity, light, and CO2).

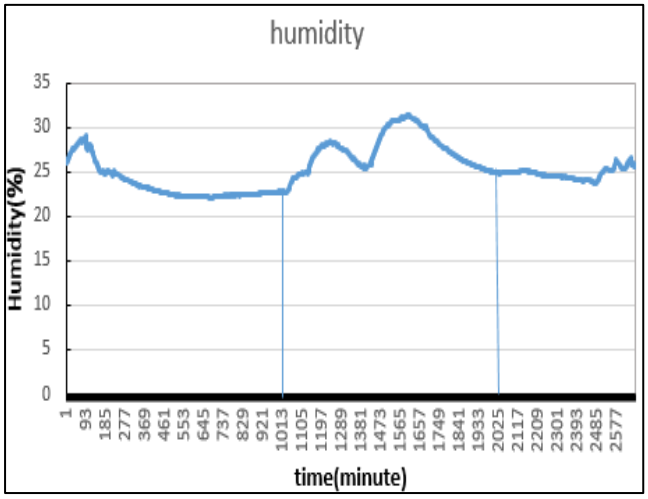
Figure (4.5)(a) shows that a room's temperature sensor varies over a minute but keeps a relatively constant range. The change in the temperature readings is very slight and relatively increasing in spaced intervals. It represents the relationship between the change occurring during the day and time as it rises and falls. That means the period of the beginning and the end is the same. The genetic algorithm discovered similar periods from the data. The results show a correlation between temperature data and time of the day, so the window size value of temperature is 552.

Figure (4.5)(b) shows humidity data after obtaining on window size value. The window size value is 1015. The time takes a consistent pattern in temperatures and others. Usually, this data is for a room in an office, and this time represents the duration of its occupancy, which means that its once occupied and once not. Therefore, there is a change in the window size value for temperature and humidity. In temperature, the change is rapidly

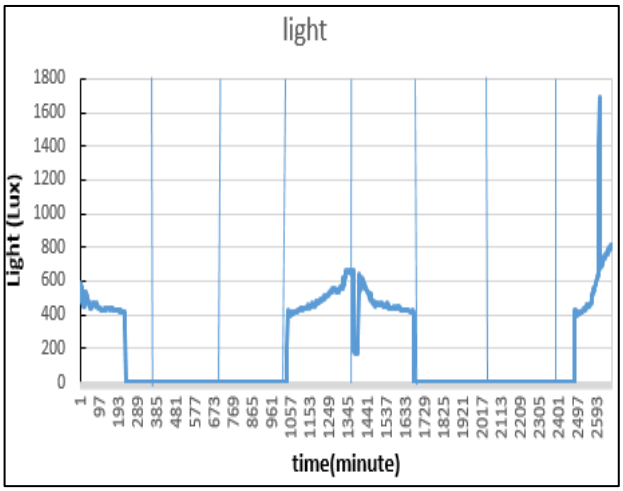
rising or falling, while in humidity, it is gradual. So the drawing for two periods repeats the same. Each line represent window size value.



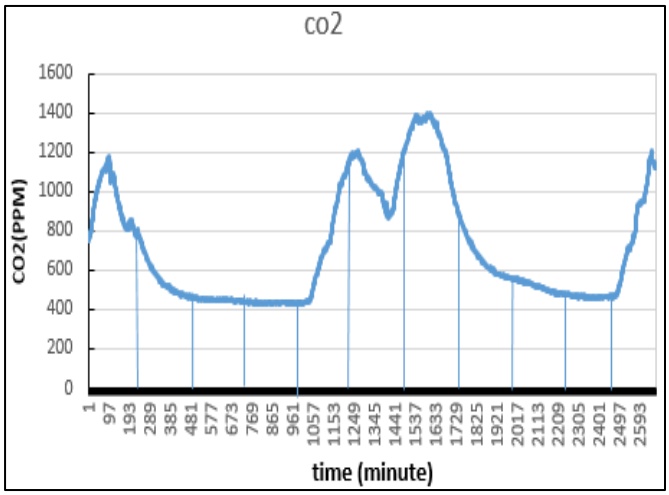
a) Temperature



b) Humidity



c) Light



d) CO2

Figure (4.5): Window Size of Occupancy Room Dataset

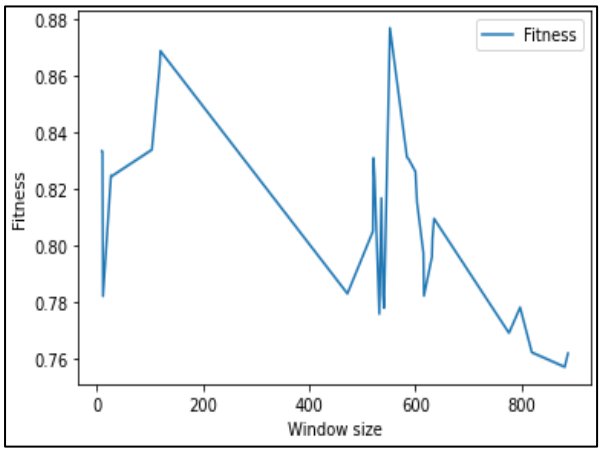
As illustrated in figure (4.5) (c), the light of a room continuously changes over the day. Compared with temperature and humidity sensors, the changes in light sensor readings are within equal intervals. The low values represent the night, while the high values represent the daytime. The beginning and ends of the curve are the same. So when window size=343. Then time takes a consistent pattern.

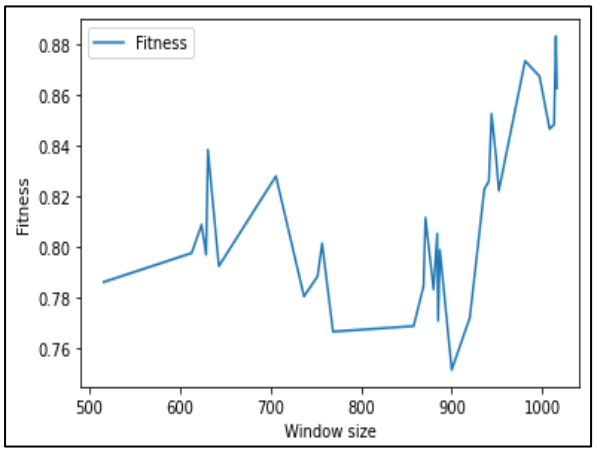
Figure (4.5) (d) illustrates that the window size value of CO2 is 247. This indicates that the resulting CO2 data is correlated with time that changes over the day. As a result, we used the curve of all sensors to represent the fitness value based on window size values.

Figure (4.6) represents the curve of the occupancy room dataset based on window size and fitness value.The y-axis represents fitness value while the x-axis represents window size value.

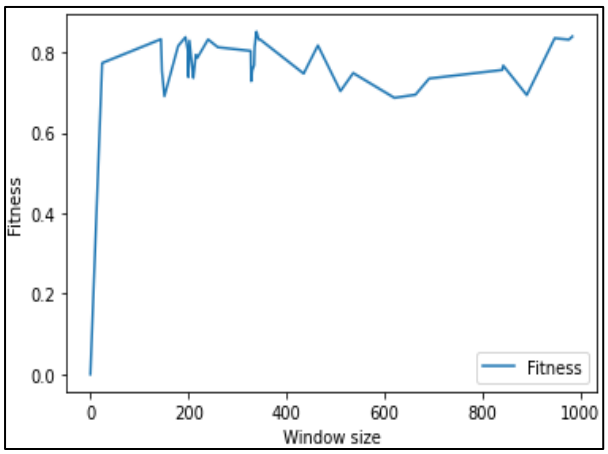
Figure (4.6) (a), shows the relation of the window size with fitness. The color curve represents the values for each fitness resulting from the implementation each time for the temperature sensor, so the curve appear up and down with different values, where each gradient represents a specific value. The down curve mean low values, and the up means a high value. This is useful in knowing the percentage of data spread resulting from implementing a Genetic Algorithm to obtain a suitable window size.

The humidity data and other sensors can be represented using the curves to show the Genetic Algorithm effect with data spread based on window size value. The same process was applied to represent light and CO2 sensors to show window size and fitness values, as illustrated in figure (4.6) (c, d). Down curve shows a low fitness value based on window size obtained from the implemented Genetic Algorithm in figure (4.6) (b).

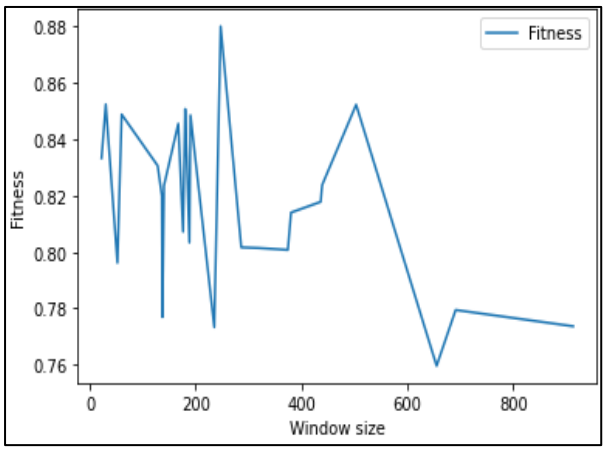




a) Temperature



b) Humidity

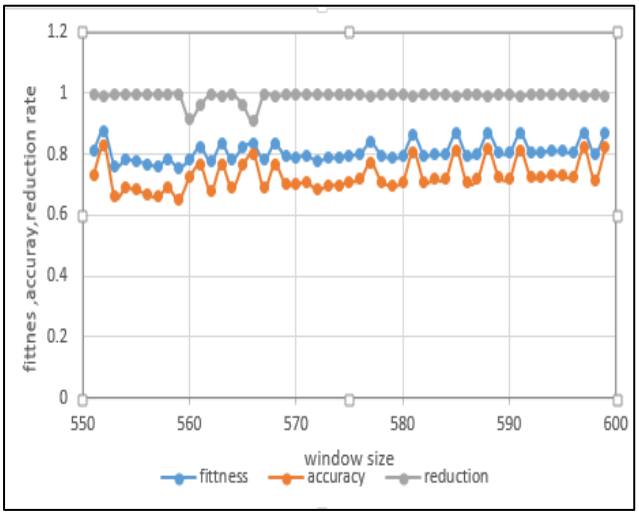


c) Light data

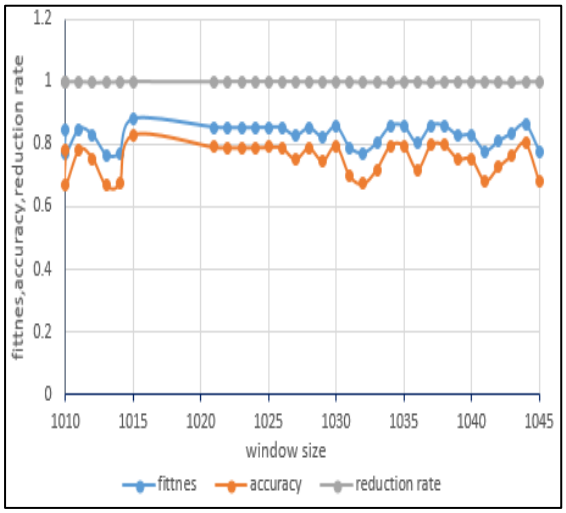
d) CO2 data

Figure (4.6): Occupancy Room Dataset

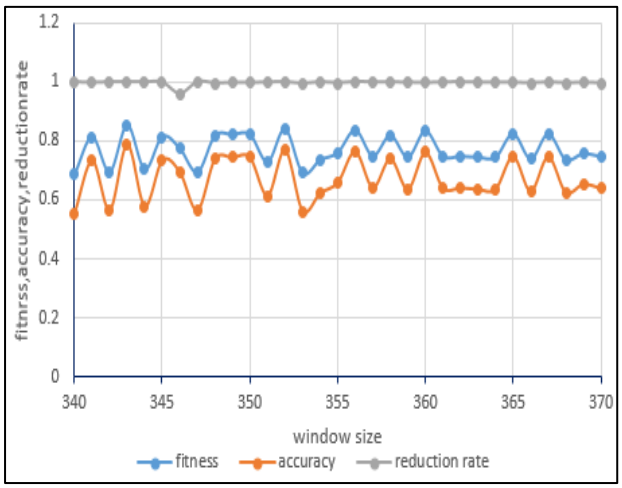
Through this figure (4.7), the change in the reduction rate is almost constant because the data is similar, while the accuracy goes up and down depending on the number of centers, so the fitness is affected by the accuracy. Then the size of the window depends on the amount and rate of change in the state of the data.



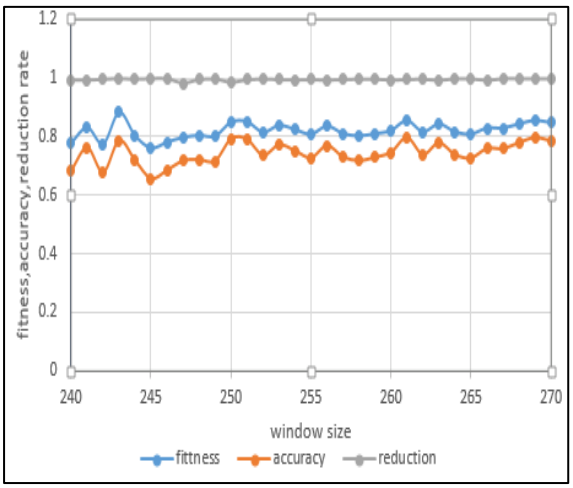
a)Temperature



b)Humidity



c)Light



d)CO2

Figure (4.7): The Relation of Window Size with Reduction Rate, Accuracy, and Fitness Based on Occupancy Room Dataset.

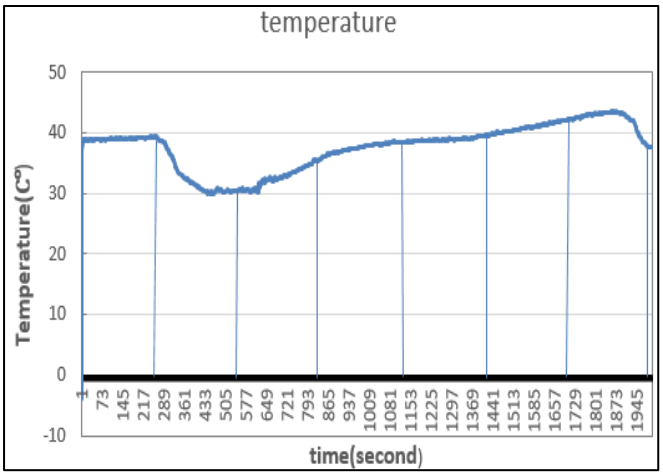
In another step, window size determines a suitable layer (IoT, fog, or cloud) to implement this process. That is based on computing resources required such as space complexity (memory), bandwidth, energy, and storage by using equation illustrated in chapter 2 section (2.3.1). Compute resources for each window size; then vector results compare with the resources availability in the table (4.1). Comparison between two vectors through applied normalization for both and compute minimum difference between them. Minimum values are decided where each process to implement in any layer (IoT, fog, or cloud).

Table (4.3): The Suitable Layer for Each Sensor Based on Required Amount of Resources in Occupancy Room Dataset.

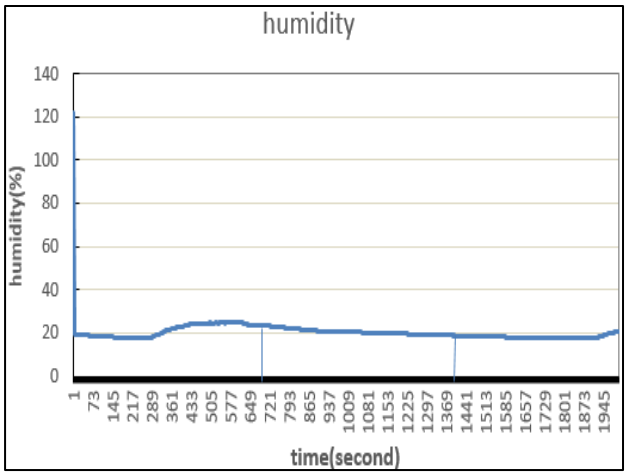
Sensor type	Optimal Window size	Reduction rate	Accuracy	Resources amount needed	Layer
Temperature	552	99%	82%	[9.4J,2212B, 0.001MB]	Fog
Humidity	1015	99%	83%	[55.8J,4064B, 0.003MB]	Cloud
Light	343	99%	78%	[4.68J,1376B, 0.001MB]	Fog
CO2	247	99%	83%	[5.39J,992B, 0.0003MB]	IoT

4.3.2 Results of Intel Berkeley Research Lab Dataset

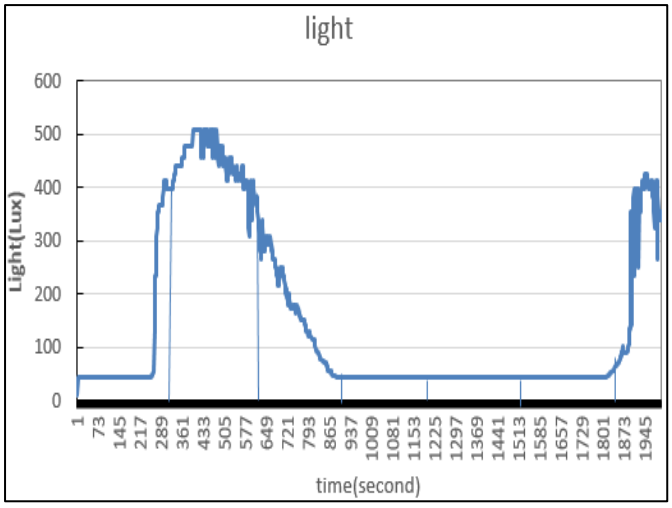
We show the results for the Intel Berkeley lab dataset by applying the same process. Here we show the main role of the genetic algorithm in optimizing window size after obtaining window size that shows relation with time based on the dataset. When the window size value for temperature =279, the window size of the humidity sensor is 686, and light is 305, and voltage is 116. This time shows a correlation between the data of the sensor and time. Because of the density of the data shown in the figure, we will take a part of the data to show the window size values, such as in figure (4.8).When each line represent window size values.



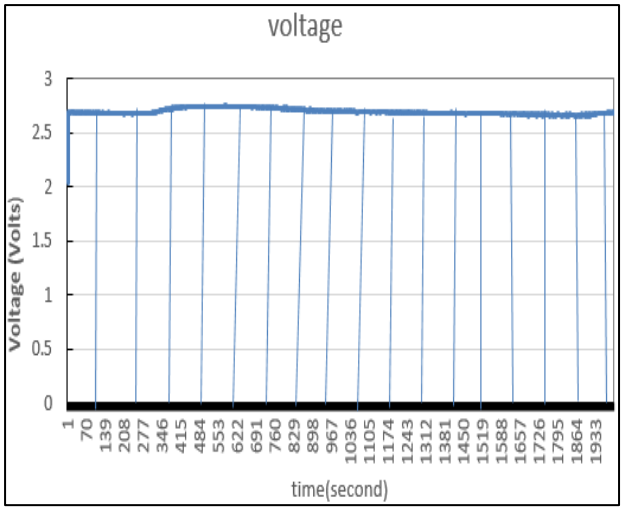
a)Temperature



b)Humidity



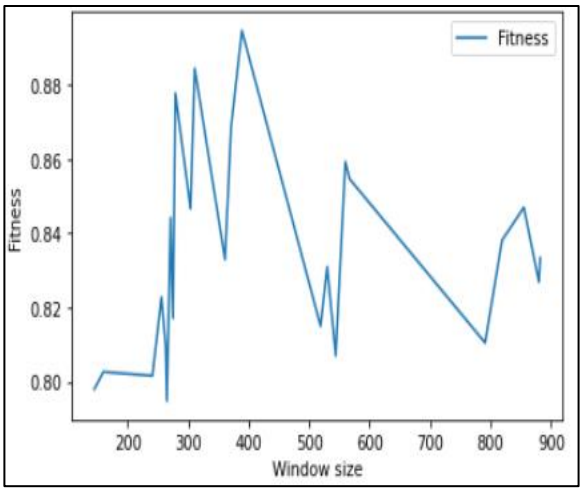
c)Light



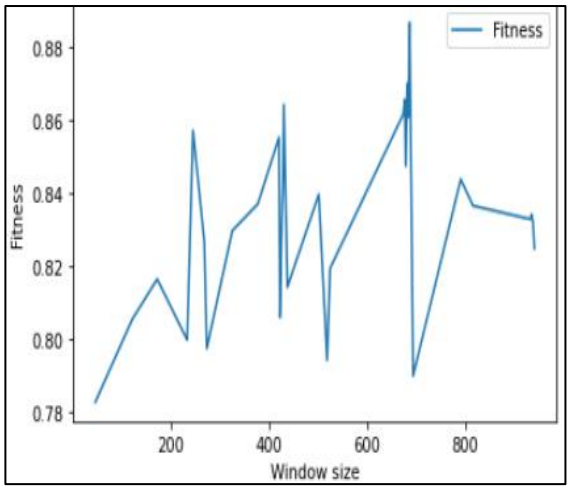
d)Voltage

Figure (4.8): Window Size of Intel Berkeley Lab Dataset

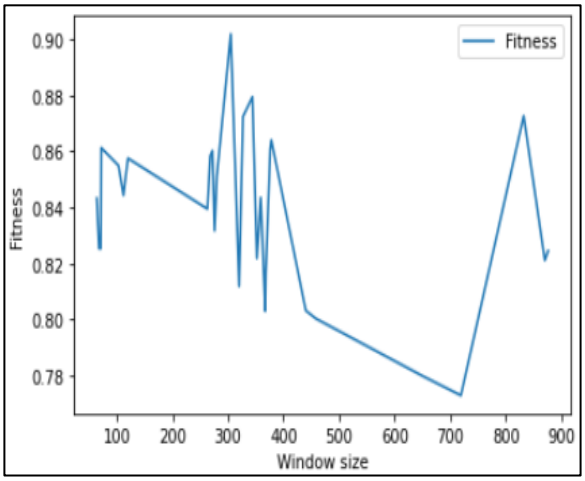
Figure (4.9) shows curves for all sensors (temperature, humidity, light, and voltage). That represents the fitness value based on window size. Color curve represents fitness. This Figure helps to understand the role of genetic algorithms in data spread.



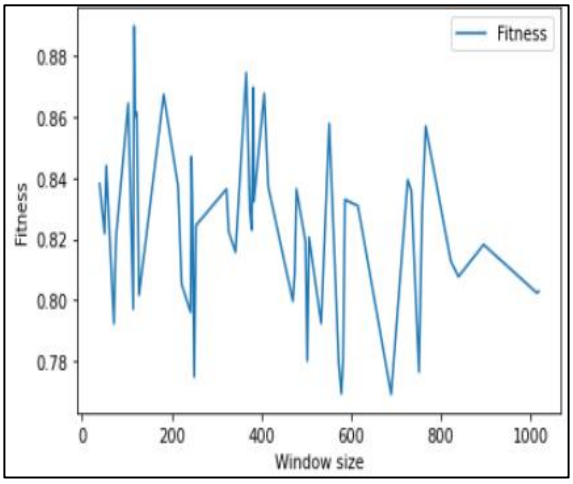
a)Temperature



b)Humidity



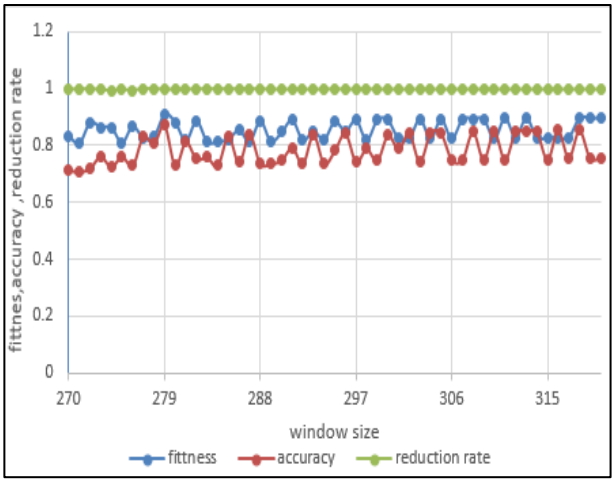
c)Light



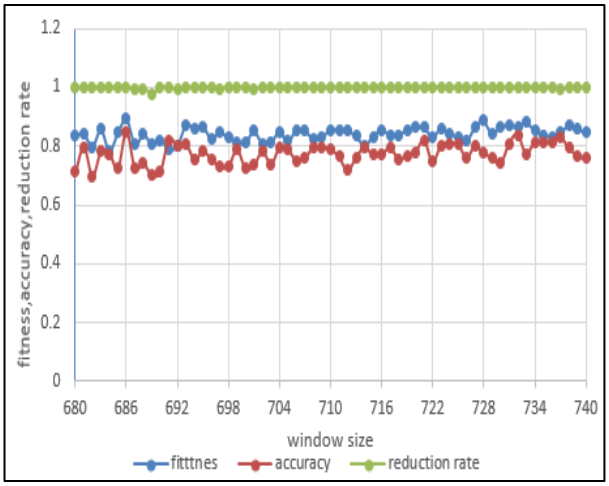
d)Voltage

Figure (4.9): Intel Berkeley Lab Dataset

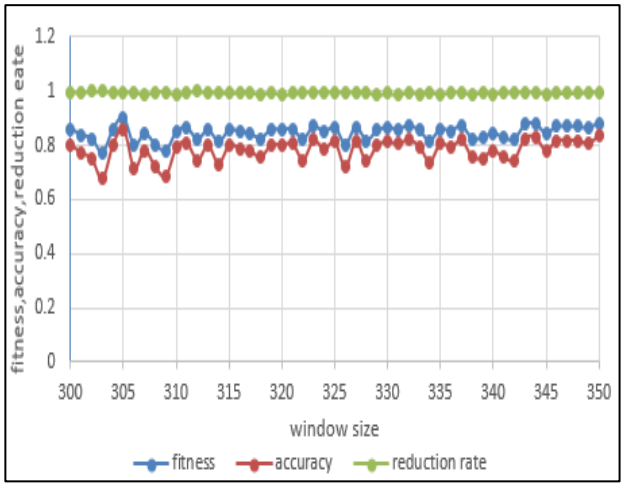
Through the figure (4.10), the change in the reduction rate is almost constant because the data almost is similar, while the accuracy goes up and down depending on the number of centers, so the fittnes is affected by the accuracy.



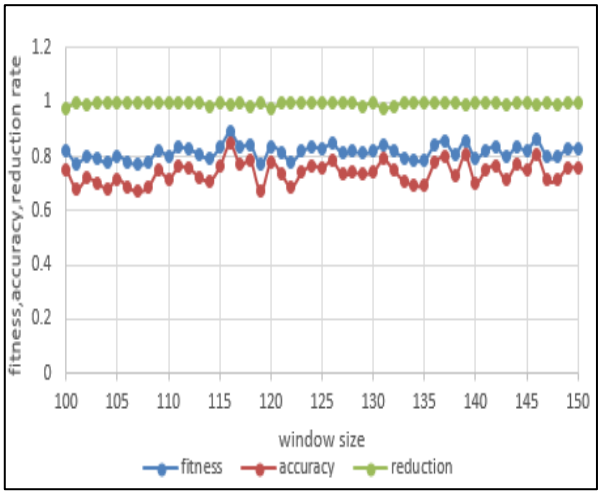
a)Temperature



b)Humidity



c)Light



d)Voltage

Figure (4.10): The Relation of Window Size with Reduction Rate, Accuracy, and Fitness Based on Intel Berkely Research Lab.

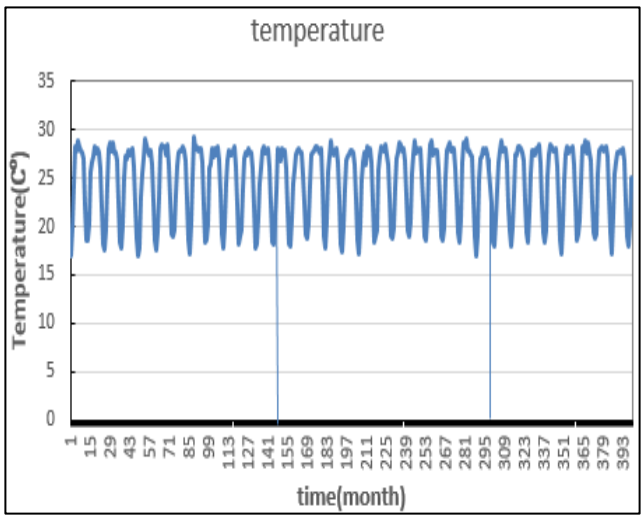
Determining the amount of resources for each sensor to decide the suitable layer to implement (IoT, fog, or cloud) based on window size value. This is done by comparing the vector of the available resources with the vector of the required resources.

Table (4.4): The Suitable Layer for Processing Data of Each Sensor Based on The Required Amount of Resources in Intel Berkely Lab.

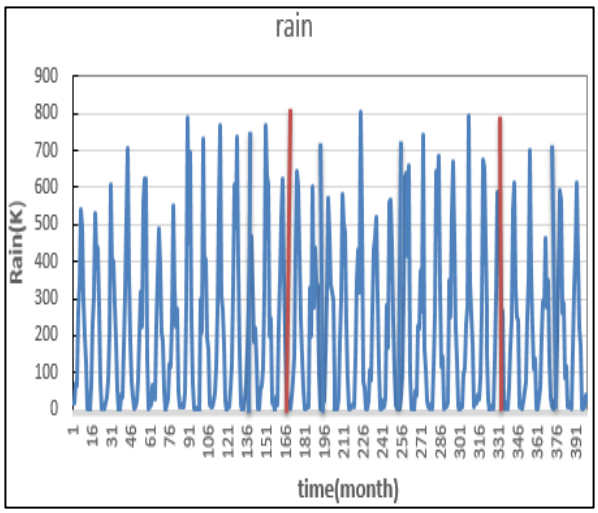
Sensor type	Optimal Window size	Reduction rate	accuracy	Resources amount needed	Layer
Temperature	279	99%	87%	[5.8J,1120B, 0.012MB]	IoT
Humidity	686	99%	84%	[24.29J,27447B, 0.002MB]	Cloud
Light	305	99%	86%	[1.9J,1224B, 0.001MB,]	Fog
Voltage	116	99%	84%	[3.2J,468B, 0.0005MB	IoT

4.3.3 Results of Weather Dataset

We used another dataset to show the results. After obtaining the optimal window size by applying a genetic algorithm that illustrates the correlation between data and time, we obtain the results of window size for temperature is 148, and rain is 166, which takes a consistent pattern in temperatures and others. Because of the density of the data shown in the figure, we will take a part of the data to show the window size values, such as in figure (4.9).



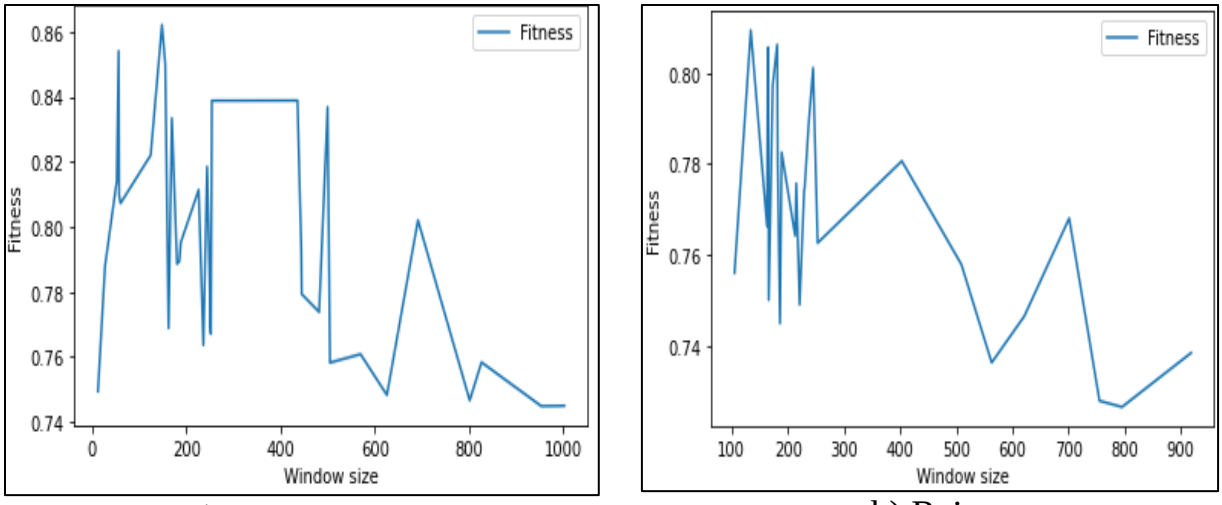
a)Temperature



b) Rain

Figure (4.11): Window Size of Weather Dataset

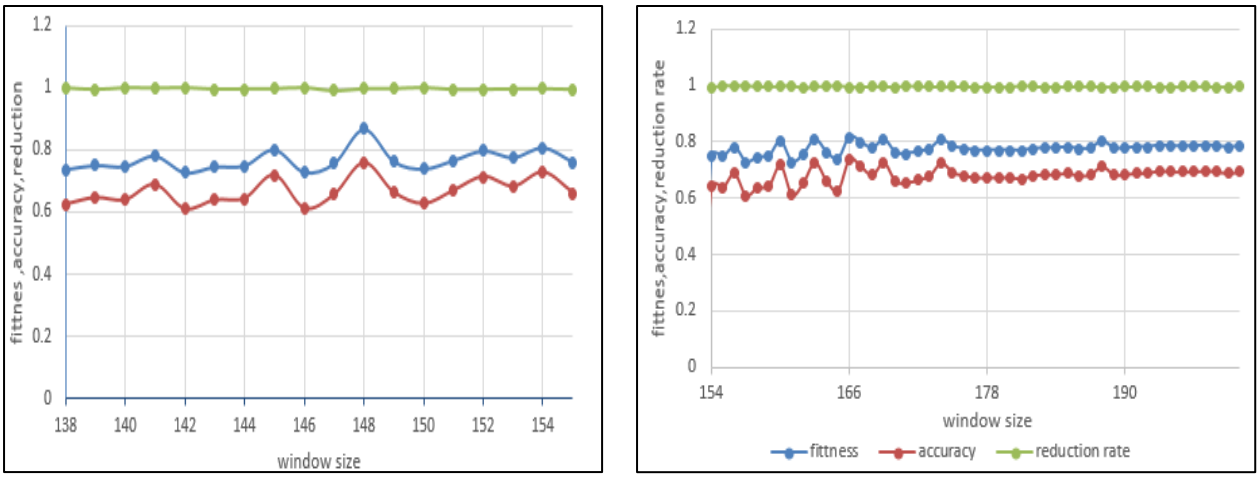
This is useful in knowing the percentage of data spread resulting from the implementation of a genetic Algorithm to obtain a suitable window size. Figure (4.12) illustrates relation window size with fittnes for temperature and rain data by using curves showing that Genetic Algorithms discovered similar periods from the data. When each gradient represents a value.



a) Temperature b) Rain

Figure (4.12): Weather Dataset

Figure (4.13) illustrated that change in fitness and accuracy while reduction rate almost consistant.



a) Temperature b) Rain

Figure (4.13): The Relation of Window Size with Reduction Rate, Accuracy, and Fitness Based on Weather Dataset.

The curves are useful in knowing the percentage of data spread resulting from implementing a genetic algorithm to obtain a suitable window size. The down curve means a low value, and the upper means a high value.

Determine the layer for each sensor to implement the process by computing resources required for each window size through comparing the vector of the available resources with the vector of the required resources; this is illustrated in table (4.5).

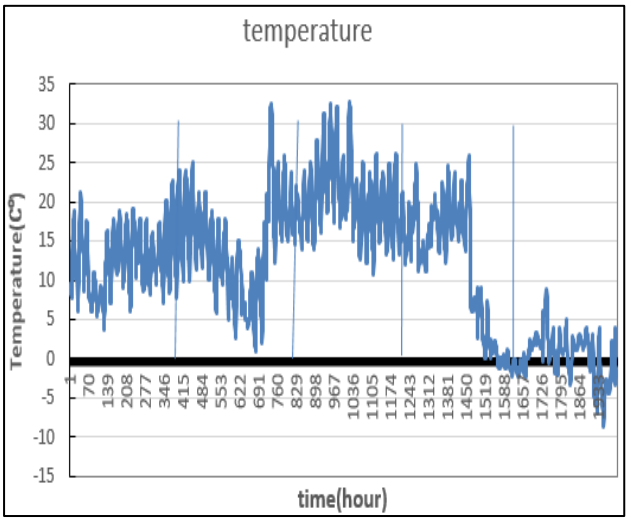
Table (4.5) : The Suitable Layer for Processing Data of Each Sensor Based on Required Amount of Resources in Weather Dataset.

Sensor	Optimal Window size	Reduction rate	accuracy	Resources amount needed	Layer
Temperature	148	99%	80%	[3.1J,596B, 0.004MB]	IoT
Rain	166	99%	73%	[3.8J,688B, 0.005MB]	IoT

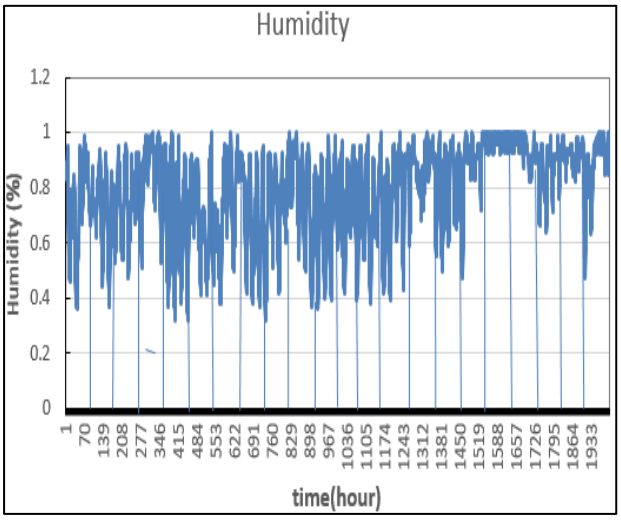
4.3.4 Results of Weather History Dataset

We are showing the results obtained after applying the same process to another dataset. The dataset represents five days of data.

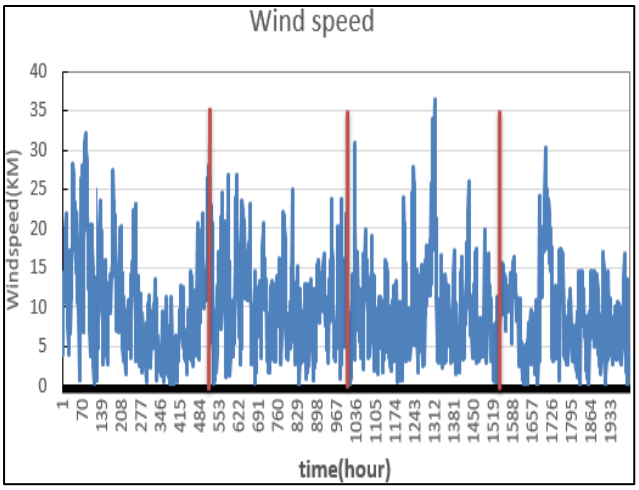
When the window size for temperature data is 409, 91 for humidity, 515 for wind speed, and 141 for Pressure. The time takes a consistent pattern in temperatures and others**.** The genetic algorithms discovered similar periods from the data. Because of the density of the data shown in the figures, we will take a part of the data to show the window size values, such as in figure (4.14).



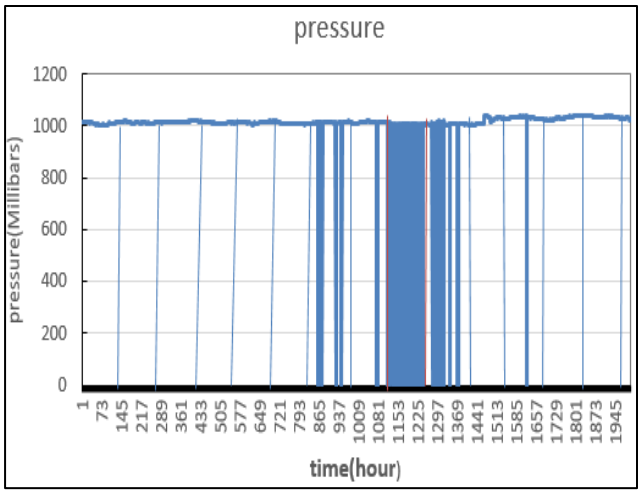
a) Temperature



b) Humidity



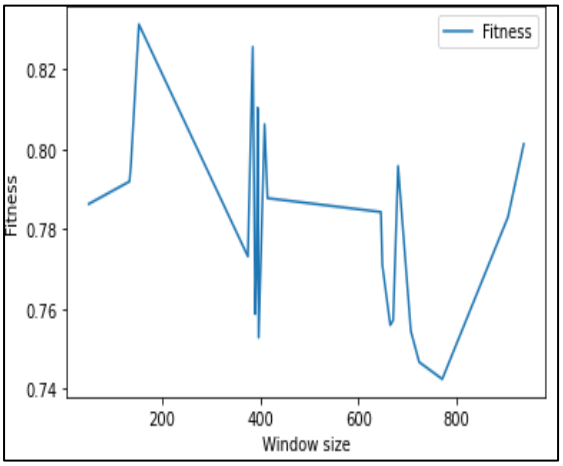
c)Wind Speed



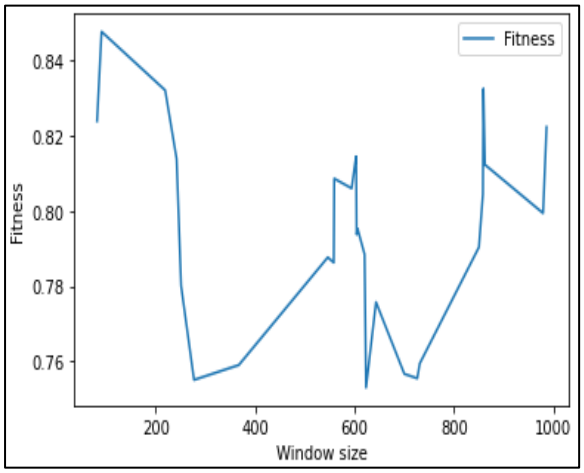
d) Pressure

Figure (4.14): Window Size of Weather History Dataset.

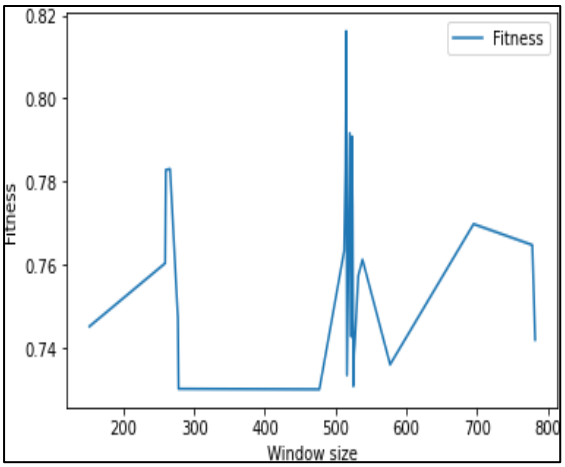
Figure (4.15) illustrates the weather history dataset when y-axis represents fitness value while the x-axis represents window size values.



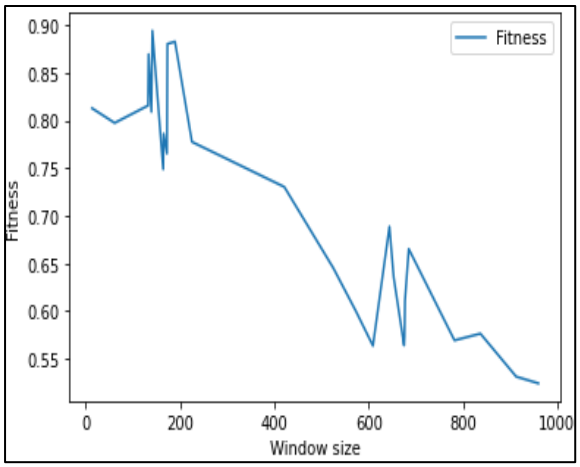
a)Temperature



b)Humidity



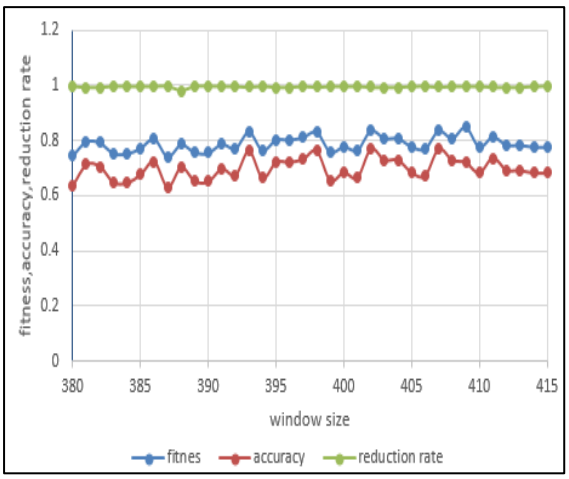
c)Windspeed



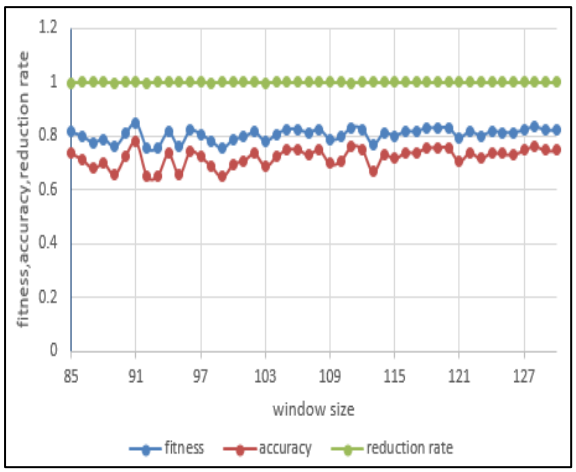
d)Pressure

Figure (4.15): Weather History Dataset

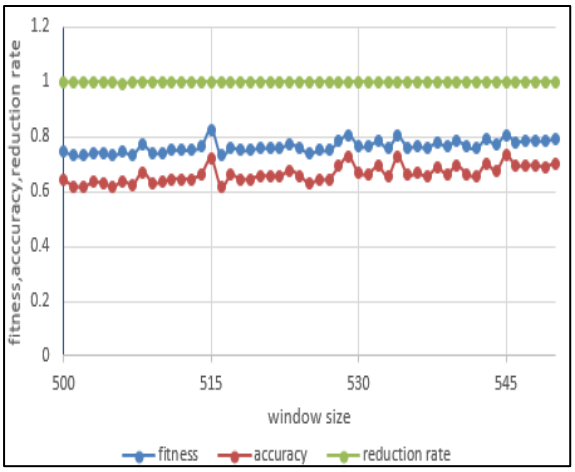
Through this figure (4.16), the change in the reduction rate is almost constant because the data is similar, while the accuracy goes up and down depending on the number of centers, so the fitness is affected by the accuracy**.**



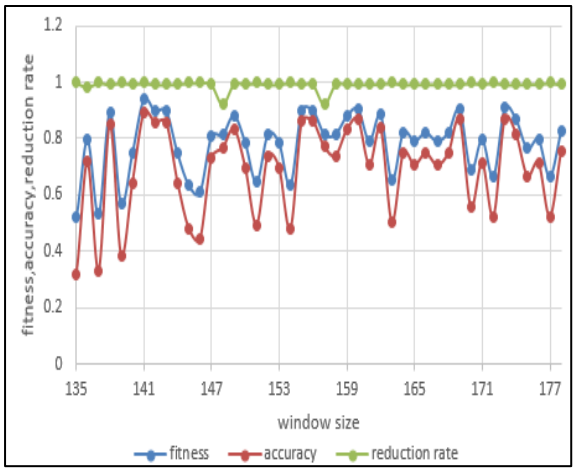
a)Temperature



b)Humidity



c)Windspeed



d)Pressure

Figure (4.16): The Relation of Window Size with Reduction Rate, Accuracy, and Fitness Based on Weather History Dataset.

After that, we determines a suitable layer for processing data of each sensor based on the amount of resources involved (energy, memory, storage, and bandwidth) illustrated in table (4.6).

Table (4.6): The Suitable Layer for Processing Data of Each Sensor Based on Required Amount of Resources in Weather History Dataset.

Sensor type	Optimal Window size	Reduction rate	Accuracy	Resources amount needed	Layer
Temperature	409	99%	82%	[7.4J,164B, 0.001MB]	Fog
Humidity	91	99%	86%	[5.4J,368B, 0.0008MB]	IoT
Windspeed	515	99%	80%	[14.6J,2064B, 0.001MB]	Fog
Pressure	141	99%	87%	[1.8J,568B, 0.0001MB]	IoT

4.4 Evaluation Metrics

The proposed method's performance was assessed using reduction rate and accuracy. The mean square error measure achieves accuracy. Below we present the results obtained from applying these measures to the datasets.

4.4.1 Reduction Rate

The results indicate that weight value made a trade-off between accuracy and reduction rate. So that it is not necessary to increase the size of the window, it will give more center. They may keep the same centers, but the size of the cluster is larger, so the loss becomes more, meaning that the accuracy is less, but the reduction rate does not change.

Table (4.7) shows the reduction rate and accuracy used to compute fitness. The reduction rate depends on the number of centers that result from each window after applying the clustering. The average reduction approximately is **99%.**This table shows the temperature sensor as an example to illustrate the results.That consist of experimental values for α and β such as (0.7, 0.3), (0.6, 0.4), (0.5, 0.5), (0.4, 0.6) and (0.3, 0.7). The fitness function illustrated in chapter 3 section (3.3.2) is based on reduction rate and accuracy values .Window size values obtained from each implementation.

Table(4.7):On-Demaind Reduction rate and Accuracy Results

α	β	Fitness value	Reduction rate	Accuracy rate	Window size
0.7	**0.3**	**0.96**	**0.99**	**0.87**	**211**
0.6	**0.4**	**0.91**	**0.99**	**0.85**	**288**
0.5	**0.5**	**0.95**	**0.99**	**0.90**	**552**
0.4	**0.6**	**0.95**	**0.99**	**0.90**	**253**
0.3	**0.7**	**0.93**	**0.98**	**0.92**	**85**

As a results, increasing required accuracy β redusing window size and vice versa for α so this affects on resources.

4.5 Comparison with Prior Studies

This section shows a comparison with some previous works. We have compared our proposed approach with the works presented in **[8, 60**]. The comparison is based on the reduction rate achieved regardless of the data reduction technique used. In **[8],**they investigates a data processing model for a real experimental environment in which data is collected from several IoT devices on an edge server where a clustering-based data reduction model is implemented. The subtractive clustering algorithm is employed for the first time for streamed IoT data with high efficiency. This work obtains of **97 %** reduction rate and an accuracy of **87%.** To reduce the amount of data communicated to the base station and eliminate duplicate comparable data sets received from the sensor devices, they suggested and implemented a clustering-based Dynamic Time Warping (DTW) with simple encoding in the latter. In **[60],** the energy-efficient Data Transmission and Aggregation Protocol (EDaTAP) is suggested for fog computing using periodic sensor networks (PSNs). The periodic EDaTAP protocol completes its work at the sensor device and fog gateway processing levels.This work has been explained in Chapter2.The reduction rate (compression ratio) achieved in this work is **97.4**%, and the accuracy is **98%** compared with the proposed approach, which achieved a reduction rate of **99%,** and the accuracy is **90%.** As showen in table (4.8) comparsion our work with prior studies based on reduction rate and accuracy metrics.

Table (4.8) Comparison with prior studies

Referencess	Reduction rate	Accuracy
[8]	97%	87%
[60]	97.4%	98%
Our work	99%,	90%

4.6 Summary

This chapter illustrates that high redundancy in IoT data was eliminated by applying a subtractive clustering algorithm to the window size of the data stream.The proposed solution achieves a high reduction rate while preserving data accuracy. This study uses a Genetic Algorithm for optimizing window size and fuzzy subtractive clustering as a reduction technique. It proposes a fitness function by computing reduction rate and accuracy. This study shows that the chosen window size highly impacts the seasonality of the data stream. Window size effects on resources determine the need for an algorithm. Then determine the layer to implement the algorithm. Finally, this chapter discusses the performance and justifies all obtained results from several datasets.

CHAPTER FIVE

CONCLUSIONS AND FUTURE WORKS

Chapter Five

Conclusions and Future Works

5.1 Conclusions

An efficient resources management approach is computed. This method uses fog computing to collect data from various IoT devices. In the fog node, a Genetic Algorithm is employed to improve the window size parameters, and clustering-based data reduction techniques are applied as a fitness function. After that calculate evaluation measures such as accuracy and reduction rate. After determining the ideal window size, choose the appropriate layer to be used (IoT, fog, or cloud) based on the amount of resources required for each window size. Then, rather than sending raw data to a cloud-hosted service, only representative data is delivered.

The conclusions drawn from this work are summarized as follows:

1. Some sensors need a few amount of resources so the suitable layer is the IoT layer to be implemented.
2. Some sensors need a middle amount of resources so a suitable layer is the fog layer to be implemented.
3. Some sensors need a high amounts of resources so the suitable layer the cloud layer to be implemented.
4. This approach is not suitable for sensitive data such as medical data because it eliminate much of original data and the accuracy is not 100% in such cases we advice to use compression.
5. Increasing required accuracy β reducing window size and vice versa for α so this affects on resources.

5.2 Future Works

The following points may be good ideas for future works:

1) Using the Genetic Algorithm to adaptively choose another parameters such as squash facter , Accept Ratio, and Reject Ratio parameters of the Fuzzy Subtractive Clustering (FSC) to find the best value of the parameters.

2) Making use of a service to determine the response time at the cloud and fog level.

3) Examine other optimization algorithms for selecting the optimal window size.

References

References

[1] A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, et al., "All one needs to know about fog computing and related edge computing paradigms: A complete survey," Journal of Systems Architecture, vol. 98, pp. 289-330, 2019.

[2] W. M. Ismael, M. Gao, A. A. Al-Shargabi, and A. Zahary, "An in-networking double-layered data reduction for internet of things (IoT)," Sensors, vol. 19, p. 795, 2019.

[3] T. Yu, X. Wang, and A. Shami, "A novel fog computing enabled temporal data reduction scheme in iot systems," in GLOBECOM 2017-2017 IEEE Global Communications Conference, 2017, pp. 1-5.

[4] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," IEEE internet of things journal, vol. 3, pp. 637-646, 2016.

[5] S. Yi, C. Li, and Q. Li, "A survey of fog computing: concepts, applications and issues," in Proceedings of the 2015 workshop on mobile big data, 2015, pp. 37-42.

[6] S. Singh and I. Chana, "QoS-aware autonomic resource management in cloud computing: a systematic review," ACM Computing Surveys (CSUR), vol. 48, pp. 1-46, 2015.

[7] A. Yousefpour, G. Ishigaki, and J. P. Jue, "Fog computing: Towards minimizing delay in the internet of things," in 2017 IEEE international conference on edge computing (EDGE), 2017, pp. 17-24.

[8] R. M. Obaise and M. A. Salman, "Data Reduction Approach Based on Fog Computing in IoT Environment," in 2020 7th International Conference on Electrical Engineering, Computer Sciences and Informatics (EECSI), 2020, pp. 65-70.

[9] S. García, J. Luengo, and F. Herrera, Data preprocessing in data mining vol. 72: Springer, 2015.

[10] J. Han, M. Kamber, and D. Mining, "Concepts and techniques," Morgan Kaufmann, vol. 340, pp. 94104-3205, 2006.

[11] A. Lambora, K. Gupta, and K. Chopra, "Genetic algorithm-A literature review," in 2019 international conference on machine learning, big data, cloud and parallel computing (COMITCon), 2019, pp. 380-384.

[12] T. Le, T. Altman, and K. J. Gardiner, "A fuzzy clustering method using Genetic Algorithm and Fuzzy Subtractive Clustering," in

Proceedings of the International Conference on Information and Knowledge Engineering (IKE), 2012, p. 1.
[13] Shieh, H.-L., P.-L. Chang, and C.-N. Lee. *An efficient method for estimating cluster radius of subtractive clustering based on a genetic algorithm*. In *2013 IEEE International Symposium on Consumer Electronics (ISCE)*. 2013. IEEE.
[14] M. A. Rahman and M. Z. Islam, "A hybrid clustering technique combining a novel genetic algorithm with K-Means," Knowledge-Based Systems, vol. 71, pp. 345-365, 2014.
[15] Y. Ding and X. Fu, "Kernel-based fuzzy c-means clustering algorithm based on genetic algorithm," Neurocomputing, vol. 188, pp. 233-238, 2016.
[16] S. Irfan, G. Dwivedi, and S. Ghosh, "Optimization of K-means clustering using genetic algorithm," in 2017 International conference on computing and communication technologies for smart nation (IC3TSN), 2017, pp. 156-16.

[17] M. A. El-Shorbagy, A. Ayoub, I. El-Desoky, and A. Mousa, "A novel genetic algorithm based k-means algorithm for cluster analysis," in International Conference on Advanced Machine Learning Technologies and Applications, 2018, pp. 92-101.

[18] H. Nguyen, S. J. Louis, and T. Nguyen, "MGKA: A genetic algorithm-based clustering technique for genomic data," in 2019 IEEE Congress on Evolutionary Computation (CEC), 2019, pp. 103-110.

[19] T. Le and L. Vu, "A novel fuzzy clustering method based on GA, PSO and Subtractive Clustering," in 2020 International Conference on Computational Science and Computational Intelligence (CSCI), 2020, pp. 331-337.

[20] K. Jebari, A. Elmoujahid, and A. Ettouhami, "Automatic genetic fuzzy c-means," Journal of Intelligent Systems, vol. 29, pp. 529-539, 2020.

[21] M. A. Ihsan, M. Zarlis, and P. Sirait, "Reduction Attributes on K-Nearest Neighbor Algorithm (KNN) using Genetic Algorithm," 2021.

[22] M. Sudhakara, K. D. Kumar, R. K. Poluru, R. L. Kumar, and S. B. Bhushan, "Towards Efficient Resource Management in Fog Computing: A Survey and Future Directions," in Architecture and Security Issues in Fog Computing Applications, ed: IGI Global, 2020, pp. 158-182.

[23] A. S. Tanenbaum and A. S. Woodhull, "Operating system concepts," Learning, vol. 2, 1992.

[24] S. Vashi, J. Ram, J. Modi, S. Verma, and C. Prakash, "Internet of Things (IoT): A vision, architectural elements, and security issues," in

2017 international conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2017, pp. 492-496.

[25] P. Sethi and S. R. Sarangi, "Internet of things: architectures, protocols, and applications," Journal of Electrical and Computer Engineering, vol. 2017, 2017.

[26] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," IEEE communications surveys & tutorials, vol. 17, pp. 2347-2376, 2015.

[27]M. Marjani, F. Nasaruddin, A. Gani, A. Karim, I. A. T. Hashem, A. Siddiqa, et al., "Big IoT data analytics: architecture, opportunities, and open research challenges," ieee access, vol. 5, pp. 5247-5261, 2017.

[28] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. U. Khan, "The rise of "big data" on cloud computing: Review and open research issues," Information systems, vol. 47, pp. 98-115, 2015.

[29]F. Hussain and A. Al-Karkhi, "Big data and fog computing," in Internet of Things, Ed: Springer, 2017, pp. 27-44.

[30]R. Neware and U. Shrawankar, "Fog computing architecture, applications and security issues," International Journal of Fog Computing (IJFC), vol. 3, pp. 75-105, 2020.

[31]S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog computing: Platform and applications," in 2015 Third IEEE workshop on hot topics in web systems and technologies (HotWeb), 2015, pp. 73-78.

[32] R. K. Naha, S. Garg, D. Georgakopoulos, P. P. Jayaraman, L. Gao, Y. Xiang, et al., "Fog computing: Survey of trends, architectures, requirements, and research directions," IEEE access, vol. 6, pp. 47980-48009, 2018.

[33] F. C. Delicato, P. F. Pires, and T. Batista, Resource management for Internet of Things vol. 16: Springer, 2017.

[34] S. P. Singh, A. Nayyar, R. Kumar, and A. Sharma, "Fog computing: from architecture to edge computing and big data processing," The Journal of Supercomputing, vol. 75, pp. 2070-2105, 2019.

[35] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in Proceedings of the first edition of the MCC workshop on Mobile cloud computing, 2012, pp. 13-16.

[36] H. F. Atlam, R. J. Walters, and G. B. Wills, "Fog computing and the internet of things: A review," big data and cognitive computing, vol. 2, p. 10, 2018.

[37] R. Mahmud, R. Kotagiri, and R. Buyya, "Fog computing: A taxonomy, survey and future directions," in Internet of everything, ed: Springer, 2018, pp. 103-130.

[38] L. Gao, T. H. Luan, S. Yu, W. Zhou, and B. Liu, "FogRoute: DTN-based data dissemination model in fog computing," IEEE Internet of Things Journal, vol. 4, pp. 225-235, 2016.

[39] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," Journal of internet services and applications, vol. 1, pp. 7-18, 2010.

[40] P. Mell and T. Grance, "The NIST definition of cloud computing," 2011.

[41] A. Botta, W. De Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and internet of things: a survey," Future generation computer systems, vol. 56, pp. 684-700, 2016.

[42] Rao, B.P., et al. *Cloud computing for Internet of Things & sensing based applications*. In *2012 Sixth International Conference on Sensing Technology (ICST)*. 2012. IEEE.

[43]Yang, K. and X. Jia, *Data storage auditing service in cloud computing: challenges, methods, and opportunities.* World Wide Web, 2012. **15**(4): p. 409-428.

[44] Velte, A. T., Velte, T. J., Elsenpeter, R. C., & Elsenpeter. Cloud computing: a practical approach. 2010.

[45] Nguyen, Hai-Long, Yew-Kwong Woon, and Wee-Keong Ng. "A survey on data stream clustering and classification." Knowledge and information systems 45.3 vol .pp. 535-569, 2015.

[46] Elmangoush, A., Coskun, H., Wahle, S., & Magedanz, T. Design aspects for a reference M2M communication platform for Smart Cities. In *2013 9th International Conference on Innovations in Information Technology (IIT)* (pp. 204-209). IEEE, 2013.

[47] D. Blaauw, D. Sylvester, P. Dutta, Y. Lee, I. Lee, S. Bang, et al., "IoT design space challenges: Circuits and systems," in 2014 Symposium on VLSI technology (VLSI-technology): digest of technical papers, 2014, pp. 1-2.

[48] S. Zahoor and R. N. Mir, "Resource management in pervasive Internet of Things: A survey," Journal of King Saud University-Computer and Information Sciences, vol. 33, pp. 921-935, 2021.

[49] F. Cao, M. Estert, W. Qian, and A. Zhou, "Density-based clustering over an evolving data stream with noise," in Proceedings of the 2006 SIAM international conference on data mining, 2006, pp. 328-339.

[50] S. Ding, F. Wu, J. Qian, H. Jia, and F. Jin, "Research on data stream clustering algorithms," Artificial Intelligence Review, vol. 43, pp. 593-600, 2015.

[51] M. Aazam and E.-N. Huh, "Fog computing and smart gateway based communication for cloud of things," in 2014 International conference on future internet of things and cloud, 2014, pp. 464-470.

[52] Garcia Lopez P, Montresor A, Epema D, Datta A, Higashino T, Iamnitchi A, Barcellos M, Felber P, Riviere E,Edge-centric computing: vision and challenges. ACM SIGCOMM Comput Commun Rev 45(5):37–42, 2015.

[53] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," Journal of internet services and applications, vol. 1, pp. 7-18, 2010.

[54] S. A. Abdulzahra, A. K. M. Al-Qurabat, and A. K. Idrees, "Data reduction based on compression technique for big data in IoT," in 2020 international conference on emerging smart computing and informatics (ESCI), 2020, pp. 103-108.

[55]A. K. M. Al-Qurabat, C. Abou Jaoude, and A. K. Idrees, "Two tier data reduction technique for reducing data transmission in IoT sensors," in 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC), 2019, pp. 168-173.

[56] Madden S: Intel Berkeley research lab data, Intel corporation, 2004 [2004-06-08]. http://berkeley. Intel -research.net/labdata.html, (2003).

[57] K. Hossain and S. Roy, "A data compression and storage optimization framework for iot sensor data in cloud storage," in 2018 21st International Conference of Computer and Information Technology (ICCIT), 2018, pp. 1-6.

[58] T. Moulahi, S. El Khediri, R. U. Khan, and S. Zidi, "A fog computing data reduce level to enhance the cloud of things performance," International Journal of Communication Systems, vol. 34, p. e4812, 2021.

[59] L. M. Abualigah, A. T. Khader, and E. S. Hanandeh, "A hybrid strategy for krill herd algorithm with harmony search algorithm to improve

the data clustering," Intelligent Decision Technologies, vol. 12, pp. 3-14, 2018.

[60] A. K. Idrees and A. K. M. Al-Qurabat, "Energy-efficient data transmission and aggregation protocol in periodic sensor networks based fog computing," Journal of Network and Systems Management, vol. 29, pp. 1-24, 2021.

[61] V. L. Miguéis and A. S. Camanho, "Mining customer loyalty card programs: The improvement of service levels enabled by innovative segmentation and promotions design," in International Conference on Exploring Services Science, 2011, pp. 83-97.

[62] A. Fahad, N. Alshatri, Z. Tari, A. Alamri, I. Khalil, A. Y. Zomaya, et al., "A survey of clustering algorithms for big data: Taxonomy and empirical analysis," IEEE transactions on emerging topics in computing, vol. 2, pp. 267-279, 2014.

[63] H.-L. Nguyen, Y.-K. Woon, and W.-K. Ng, "A survey on data stream clustering and classification," Knowledge and information systems, vol. 45, pp. 535-569, 2015.

[64] R. Alhussaini, A. K. Idrees, and M. A. Salman, "Data transmission protocol for reducing the energy consumption in wireless sensor networks," in International conference on new trends in information and communications technology applications, 2018, pp. 35-49.

[65] K. Hammouda and F. Karray, "A comparative study of data clustering techniques," University of Waterloo, Ontario, Canada, vol. 1, 2000.

[66] J. Han, M. Kamber, and J. Pei, “Data mining concepts and techniques third edition,” Morgan Kaufmann Ser. Data Manag. Syst., pp. 83–124, 2011.

[67] R. S. Sidhu, S. Khullar, P. S. Sandhu, R. Bedi, and K. Kaur, "A subtractive clustering based approach for early prediction of fault proneness in software modules," International Journal of Computer and Systems Engineering, vol. 4, pp. 1165-1169, 2010.

[68] K. M. Bataineh, M. Naji, and M. Saqer, "A Comparison Study between Various Fuzzy Clustering Algorithms," Jordan Journal of Mechanical & Industrial Engineering, vol. 5, 2011.

[69] S. M. Berneti, "Design of fuzzy subtractive clustering model using particle swarm optimization for the permeability prediction of the reservoir," International Journal of Computer Applications, vol. 29, pp. 33-37, 2011.

[70] J. Rantala and H. Koivisto, "Optimised subtractive clustering for neuro-fuzzy models," in 3rd WSEAS international conference on fuzzy sets and fuzzy systems, 2002.

[71] S. L. Chiu, "Fuzzy model identification based on cluster estimation," Journal of Intelligent & fuzzy systems, vol. 2, pp. 267-278, 1994.
[72] L. T. Ngo and B. H. Pham, "A type-2 fuzzy subtractive clustering algorithm," in Mechanical Engineering and Technology, Ed: Springer, 2012, pp.395-402.

[73] https://www.kaggle.com/datasets/muthuj7/weather-dataset.

[74] Z. Lubis, P. Sihombing, and H. Mawengkang, "Optimization of K Value at the K-NN algorithm in clustering using the expectation maximization algorithm," in IOP Conference Series: Materials Science and Engineering, 2020, p. 012133.

[75] M. El-Tarabily, R. Abdel-Kader, M. Marie, and G. Abdel-Azeem, "A PSO-based subtractive data clustering algorithm," International Journal of Research in Computer Science, vol. 3, pp. 1-9, 2013.

[76] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in Proceedings of the 33rd annual Hawaii international conference on system sciences, 2000, p. 10 pp. vol. 2.

[77]D.E. Goldberg, “Genetic Algorithms in Search, Optimization and Machine Learning”, Addison-Wesley, New York, 1989.

[78] M. Tabassum and K. Mathew, "A genetic algorithm analysis towards optimization solutions," International Journal of Digital Information and Wireless Communications (IJDIWC), vol. 4, pp. 124-142, 2014.

[79] D. T. Larose, "An introduction to data mining," Traduction et adaptation de Thierry Vallaud, 2005.

[80] A. Mukhopadhyay and U. Maulik, "Towards improving fuzzy clustering using support vector machine: Application to gene expression data," Pattern Recognition, vol. 42, pp. 2744-2763, 2009.

[81] S. L. Chiu, "Fuzzy model identification based on cluster estimation," Journal of Intelligent & fuzzy systems, vol. 2, pp. 267-278, 1994.

[82] A. Halder, S. Pramanik, and A. Kar, "Dynamic image segmentation using fuzzy c-means based genetic algorithm," International Journal of Computer Applications, vol. 28, pp. 15-20, 2011.

[83] A. Ghosh, N. S. Mishra, and S. Ghosh, "Fuzzy clustering algorithms for unsupervised change detection in remote sensing images," Information Sciences, vol. 181, pp. 699-715, 2011.

[84] Z. Lianjiang, Q. Shouning, and D. Tao, "Adaptive fuzzy clustering based on genetic algorithm," in 2010 2nd International Conference on Advanced Computer Control, 2010, pp. 79-82.

[85] Y. Liu and Y. Zhang, "Optimizing parameters of fuzzy c-means clustering algorithm," in Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007), 2007, pp. 633-638.

[86] D. L. Davies and D. W. Bouldin, "A cluster separation measure," IEEE transactions on pattern analysis and machine intelligence, pp. 224-227, 1979.

[87] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," Journal of computational and applied mathematics, vol. 20, pp. 53-65, 1987.

[88] J. Chen, Z. Qin, and J. Jia, "A weighted mean subtractive clustering algorithm," Information Technology Journal, vol. 7, pp. 356-360, 2008.

[89] L.-H. Yang, F.-F. Ye, J. Liu, Y.-M. Wang, and H. Hu, "An improved fuzzy rule-based system using evidential reasoning and subtractive clustering for environmental investment prediction," Fuzzy sets and systems, vol. 421, pp. 44-61, 2021.

[90] L. M. Candanedo and V. Feldheim, "Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models," Energy and Buildings, vol. 112, pp. 28-39, 2016.

الخلاصة

لقد تم إنشاء قدر كبير من البيانات الضخمة نتيجة الاستخدام الواسع النطاق لإنترنت الأشياء وزيادة الأدوات المتصلة بالإنترنت.حيث يعد تدفق مثل هذه البيانات إلى مراكز البيانات السحابية للمعالجة والتحليل والتخزين هو الإجراء المعتاد ، ويتميز هذا التدفق بالتكرار.حيث يؤدي التكرار إلى ضياع موارد الحوسبة على مستويات مختلفة.

يشير مصطلح البيانات المتدفقة الى سلسله مرتبه من نقاط البيانات ذات ابعاد متعددة .والتي يمكن قراءتها مرة واحدة فقط بسبب إمكانيات المعالجة المحدودة مثل الوقت والذاكرة .غالبًا ما تتم معالجة تدفقات البيانات في كتل يُشار إليها بالنوافذ ذات الحجم المحدد المعروف باسم حجم النافذة.ان تقدير حجم النافذة يؤثرعلى الموارد المطلوبة لتنفيذ الخوارزمية وأدائها الذي يقاس بالدقة ونسبة التقليل.

ان هذه الرسالة تبحث في نهج فعال لإدارة الموارد للأغراض التجريبية حيث يتم جمع البيانات من أجهزة إنترنت الأشياء المختلفة في طبقة الحوسبة الضبابية واستنادًا إلى تقنية تقليل البيانات العنقودية التي يتم تنفيذها على عقدة ضبابية.تم استخدام الخوارزمية الجينية (GA) لتحسين حجم النافذة لعملية التقليل. الجمع بين GA مع خوارزمية التجميع الطرحي (FSC) كدالة صلاحية هي الطريقة المقترحة لاختيار حجم النافذة الفعال. حيث تظهر النتائج التجريبية تأثير حجم النافذة على الدقة ونسبة التقليل لكل نوع من بيانات إنترنت الأشياء. ثم يتخذ حجم النافذة الأمثل القرار بشأن تحديد الطبقة لتنفيذ كل عملية (طبقة إنترنت الأشياء ،او طبقة الضباب ، او طبقة السحاب).

تحديد الطبقة المناسبة عن طريق حساب الموارد المطلوبة لحجم النافذة الأمثل مثل (الذاكرة ، والتخزين ، والطاقة ، وعرض النطاق الترددي) . باستخدام حجم النافذة الأمثل في الطبقة المختارة ، يتم إجراء التقليل عبر الإنترنت وإرسال النتائج إلى السحابة. باستخدام مجموعة بيانات مختلفة ، تُظهر النتائج التي تم الحصول عليها من هذه الرسالة الى ان نسبة التقليل مرتفعة (99٪) ودقة عالية (90٪) باستخدام متوسط مربع الخط(MSE) كمقاييس للتقييم.

وزارة التعليم العالي والبحث العلمي
جامعة بابل
كلية العلوم للبنات
قسم علوم الحاسوب

أدارة الموارد الكفؤه في بيئة أنترنيت ألاشياء بالأعتماد على الحوسبة الضبابية

رسالة مقدمة إلى مجلس كلية العلوم للبنات جامعة بابل وهي جزء من متطلبات نيل درجة الماجستير في العلوم/علوم الحاسوب

مقدمه من قبل

رواء ناظم سعيد

بإشراف

أ.م.د .محمد عبيد مهدي

أ.م.د .مهدي عبد سلمان

1444هـ

2022م