

ENHANCING DATA STREAM MINING IN WIRELESS SENSOR NETWORKS USING CLUSTERING ALGORITHMS

**By
Yassmeen Sanad Ahmad Alghamdi**

**A Thesis Submitted in Partial Fulfillment of the
Requirements for the Master Degree in Computer Science**

**Supervised By
Dr. Manal Abdulaziz Abdullah**

**FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY
KING ABDULAZIZ UNIVERSITY
JEDDAH – SAUDI ARABIA
Shaaban 1438 H – May 2017 G**

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

ENHANCING DATA STREAM MINING IN WIRELESS SENSOR NETWORKS USING CLUSTERING ALGORITHMS

**By
Yassmeen Sanad Ahmad Alghamdi**

**A Thesis Submitted in Partial Fulfillment of the
Requirements for the Master Degree in Computer Science**

**Supervised By
Dr. Manal Abdulaziz Abdullah**




**FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY
KING ABDULAZIZ UNIVERSITY
JEDDAH – SAUDI ARABIA
Shaaban 1438 H – May 2017 G**

ENHANCING DATA STREAM MINING IN WIRELESS SENSOR NETWORKS USING CLUSTERING ALGORITHMS

By
Yassmeen Sanad Ahmad Alghamdi

This thesis has been approved and accepted in partial fulfillment of the
requirements for the degree of Master of Computer Science

EXAMINATION COMMITTEE

	Name	Rank	Field	Signature
Advisor and Rapporteur	Dr Manal Abdullah	Associate Professor	Computer Networks	
External Examiner	Prof. Yasser Kadah	Professor	Electrical Engineering	
Internal Examiner	Dr Laila Nassef	Associate Professor	Computer Networks	

KING ABDULAZIZ UNIVERSITY
Shaaban 1438 H – May 2017 G

DEDICATION

**This work is dedicated to my loving parents, my husband and daughters. To my brothers and sister.
To my family and friends. To anyone contributed in the success of this research.
To a special friend who always was with me to continue my thesis journey.**

ACKNOWLEDGMENT

In the Name of Allah, the Most Merciful, the Most Compassionate, all praise be to Allah, the Lord of the Worlds, and prayers and peace be upon Mohamed, His servant and messenger. First and foremost, I must acknowledge my limitless thanks to Allah, the Ever-Magnificent, the Ever-Thankful, for His help and blessing. I am sure that this work would have never been completed without His guidance.

I am grateful to the people who worked hard with me from the beginning to the completion of this present research, particularly my supervisors, **Dr. Manal Abdulaziz** for her efforts, motivation, encouragement, and continuous support. It was an honor for me to work with her. I would also like to express my wholehearted thanks to my family (**my father, mother, husband, sister, brothers, daughters and my friends especially Azoof**) for the generous support they have provided me throughout my life and particularly throughout my Master's degree. Because of their unconditional love and prayers, I have had the chance to complete this thesis.

Yassmeen Alghamdi

ENHANCING DATA STREAM MINING IN WIRELESS SENSOR NETWORKS USING CLUSTERING ALGORITHMS

Yassmeen Sanad Alghamdi

ABSTRACT

The past few years have witnessed an increased interest in the potential use of Wireless Sensor Networks (WSNs) in a wide range of applications in the field of military surveillance, fire detection, habitat monitoring, industry, health monitoring and many more. WSNs consist of individual nodes that are able to interact with their environment by sensing and controlling physical parameters. Sensor nodes tend to generate a large amount of sequential small and tuple-oriented data that is considered as Data Streams. Data streams usually are huge data sets that arrive in an online fashion, flowing rapidly in a very high speed, where they are unlimited and there is no control on the arrival processing order. Due to sensor network limitations, some challenges are faced and urgently need to be solved. Such challenges include long lasting the WSN lifetime and reducing nodes energy consumption. Data mining could deal with the WSN challenges. Clustering is one of mining techniques and plays an important role in

organizing WSNs. It has proven its efficiency on network performance by extending network lifetime, saving energy of sensor nodes, reducing delay and delivering more data packets. This research develops an algorithm called the Density Grid-base Clustering algorithm (DeGiCA) that enhances the clustering mining technique in WSNs by combining density and grid techniques. The deployment density variation technique can find arbitrary shaped clusters while the grid technique is used to avoid clustering quality problems by discarding the boundary points of grids. DeGiCA helps to face the limitations found in WSNs that carry data streams. By using a MATLAB-based simulator, DeGiCA is compared with other clustering algorithms in WSNs that manipulate data streams. Fuzzy Clustering Means algorithm (FCM) and K-means algorithm are two selected algorithms used to be compared with DeGiCA performance metrics results. The simulation results indicate that the performance of DeGiCA outperforms K-Means in terms of network lifetime by 15%, energy consumption by 13% and packet delivery ratio by 40%. DeGiCA also outperforms FCM in terms of network lifetime by 17%, energy consumption by 11% and packet delivery ratio by 70%.

TABLE OF CONTENTS

DEDICATION	i
ACKNOWLEDGMENT	ii
ABSTRACT	iii
TABLE OF CONTENTS	v
LIST OF FIGURES	viii
LIST OF TABLES	xi
LIST OF SYMBOLS AND TERMINOLOGY	xii
Chapter I : Introduction	1
1.1 Introduction	1
1.2 Thesis Motivation.....	5
1.3 Problem Statement	5
1.4 Thesis Aim and Objectives	6
1.5 Research Methodology.....	6
1.6 Research Contribution.....	7
1.7 Thesis Organization	8
Chapter II : Literature Review	11
2.1 Introduction	11
2.2 Wireless Sensor Networks (WSNs)	13
2.3 Data Streams in WSNs.....	14
2.3.1 Traditional Data Mining and Data Stream Mining in WSNs.....	15
2.3.2 Data Stream Characteristics in WSNs.....	15
2.3.3 Algorithms of Data Streams.....	16
2.4 Overview on Data Mining in WSNs	17
2.4.1 Common Classes of Data Mining	17
2.4.2 Data Stream Mining in WSNs	18
2.4.2.1 Challenges of Data Stream Mining in WSNs	19
2.4.2.2 Taxonomy of Data Stream Mining Techniques for WSNs.....	20
2.4.2.3 Application Areas of WSNs Data Stream Mining	21
2.4.2.4 Implementation of WSNs Data Stream Mining	22
2.4.2.5 Limitations of Existing Data Stream Mining Techniques for WSNs ...	23
2.5 Data Stream Mining Clustering Techniques in WSNs	24
2.5.1 Components of Clustered WSNs	24

2.5.2 Hierarchical Clustering Structure in WSNs	25
2.5.3 WSNs Clustering Characteristics	27
2.5.4 Cluster Head and Member Node Properties in Clustering Technique	28
2.5.5 Designing Clustered WSNs.....	29
2.5.6 WSNs Clustering Parameters.....	30
2.5.7 Implementing Clustered WSNs.....	32
2.6 Classification of Clustering Protocols.....	33
2.6.1 Clustering Protocols in WSNs not involving Data Streams.....	33
2.6.1.1 Proactive and Reactive Clustering in WSNs.....	35
2.6.1.2 Clustering Algorithm Schemes in WSNs.....	36
2.6.2 Clustering Protocols involving Data Streams	39
2.6.3 Clustered WSNs involving Data Streams	43
2.6.3.1 Algorithms Based on FCM for Streaming Data in WSNs	45
2.6.4 Taxonomy of Clustering Protocols	47
2.7 Conclusion.....	49
Chapter III : Density Grid-Based Clustering Algorithm	51
3.1 Introduction	51
3.2 Overview on Density Grid-Based Clustering Algorithm (DeGiCA).....	53
3.2.1 General DeGiCA Assumptions	55
3.2.2 Algorithm Phases Description	56
3.3 DeGiCA Detailed Algorithm	60
3.3.1 Initialization Process of DeGiCA.....	60
3.3.1.1 Establishment Phase of DeGiCA	60
3.3.2 Rounds Process of DeGiCA.....	68
3.3.2.1 Data Transmission Phase of DeGiCA.....	70
3.3.2.2 CH-Election Phase of DeGiCA.....	72
3.4 Conclusion	72
Chapter IV : Experimental Results and Analysis	75
4.1 Introduction	75
4.2 DeGiCA Performance Metrics	76
4.3 DeGiCA System Requirements	77
4.3.1 DeGiCA Dataset Description	77
4.3.2 Experimental Setup Parameters	80
4.3.3 DeGiCA Simulator Requirements	81
4.4 DeGiCA Simulation Experimental Analysis	81
4.4.1 DeGiCA Expected Outcomes Description.....	82
4.4.2 Evaluating DeGiCA in terms of Network Lifetime and Energy Consumption.....	85
4.4.3 Evaluating DeGiCA in terms of Packet Delivery Ratio.....	92
4.4.4 DeGiCA Experimental Results to Determine Optimum g and σ	93
4.4.5 DeGiCA Scalability	95

4.5 Effect of DeGiCA Gridding on WSN Lifetime	96
4.6 Conclusion	100
Chapter V : Conclusions and Future Work	102
5.1 Research Summary.....	102
5.2 Conclusion of Results and Findings.....	105
5.2.1 DeGiCA Performance	106
5.2.2 Application Areas of DeGiCA	107
5.3 Thesis Future Work.....	108
LIST OF REFERENCES	110
APPENDICES	114
APPENDIX A : Fuzzy C-Means and K-Means Algorithms	115
A.1 Overview on K-means and Fuzzy C-means (FCM).....	116
A.1.1 K-means Clustering Algorithm	116
A.1.2 Fuzzy Clustering-Means Algorithm (FCM).....	117
A.2 Overview of FCM-Based Clustering Algorithms	120
A.2.1 Subtractive Fuzzy Cluster Means (SUBFCM).....	120
A.2.2 FCM of Particle Swarm Optimization (CAFPCPSO).....	121
APPENDIX B : Experimental Results Snapshots.....	122
PUBLISHED PAPERS	129

LIST OF FIGURES

Figure 1.1 Simple Clustered WSN Structure	3
Figure 2.1 Main Sensor Node Hardware Components	13
Figure 2.2 Clustered Sensor Networks Architecture.....	25
Figure 2.3 Data Communication in a Clustered Network.....	26
Figure 2.4 Proactive and Reactive Clustering Protocols.....	35
Figure 2.5 Clustering Algorithm Schemes in WSNs	39
Figure 2.6 Clustering Algorithms for Data Streams	44
Figure 2.7 Classification of FCM-Based Clustering Algorithms in WSNs.....	46
Figure 2.8 Classification of Clustering Algorithms in WSNs	47
Figure 3.1 Three Main Phases of DeGiCA Algorithm	54
Figure 3.2 Proposed DeGiCA General Flowchart	55
Figure 3.3 Techniques used in DeGiCA phases.....	56
Figure 3.4 Overview on DeGiCA Algorithm Steps	58
Figure 3.5 DeGiCA Algorithm Mechanism.....	59
Figure 3.6 Proposed DeGiCA Model Structure	62
Figure 3.7 Fill Grid Algorithm.....	63
Figure 3.8 Classify Dense Algorithm.....	64
Figure 3.9 Create Clusters Algorithm	64
Figure 3.10 Cluster Formation Process	65
Figure 3.11 Minimum High Dense Algorithm.....	65
Figure 3.12 Add Neighbors Algorithm	66
Figure 3.13 Initialization Process Pseudo Code.....	67
Figure 3.14 Initialization Process of DeGiCA Flowchart.....	68
Figure 3.15 Rounds Process Pseudo Code.....	69
Figure 3.16 DeGiCA Rounds Process Flowchart	70
Figure 3.17 DeGiCA Energy Consumption Function.....	71
Figure 4.1 Structure of Dataset Streaming Packet	80
Figure 4.2 Establishment Phase during Sensed area Gridding when $g = 150$ and $\sigma = 5$	83
Figure 4.3 Establishment phase when $g = 150$ and $\sigma = 5$ (a) Grid Classification (b) Corresponding Cluster Formation.....	83
Figure 4.4 Network Lifetime Graph when $g = 150$ and $\sigma = 5$	84

Figure 4.5 Energy Consumption for each Cluster when $g = 150$ and $\sigma = 5$ (a) for First Cluster (b) for Second Cluster	84
Figure 4.6 Energy Consumption for whole Network when $g = 150$ and $\sigma = 5$	85
Figure 4.7 Packet Delivery Ratio for each Cluster and for whole Network when $g = 150$ and $\sigma = 5$	85
Figure 4.8 Comparing DeGiCA, FCM and K-means when $g = 130$ and $\sigma = 5$ in terms of (a) Network Lifetime (b) Corresponding Energy Consumption.....	86
Figure 4.9 Comparing DeGiCA, FCM and K-means when $g = 155$ and $\sigma = 5$ in terms of (a) Network Lifetime (b) Corresponding Energy Consumption.....	87
Figure 4.10 Comparing DeGiCA, FCM and K-means when $g = 140$ and $\sigma = 6$ in terms of (a) Network Lifetime (b) Corresponding Energy Consumption.....	87
Figure 4.11 Number of Live Nodes when $g = 130$ and $\sigma = 5$	89
Figure 4.12 Average Network Lifetime for DeGiCA, FCM and K-means	89
Figure 4.13 Energy Consumption Percentage when $g = 130$ and $\sigma = 5$ in a Certain Time	91
Figure 4.14 Average Energy Consumption for DeGiCA, FCM and K-mean.....	91
Figure 4.15 Overall Packet Delivery Ratio when $g = 80$ and $\sigma = 3$	92
Figure 4.16 Average Delivered Packets for Competitors	93
Figure 4.17 Gridded WSN when (a) $g = 110$ (b) $g = 120$	96
Figure 4.18 Establishment phase when $g = 110$ at (a) Grid Classification and (b) Corresponding Cluster Formation.....	97
Figure 4.19 Establishment phase when $g = 120$ at (a) Grid Classification and (b) Corresponding Cluster Formation.....	97
Figure 4.20 Death of First Node when (a) $g = 110$ (b) $g = 120$	98
Figure 4.21 Number of Nodes to Die in a Certain Time when $g = 110, 120, 130, 140$	99
Figure A.1 Types of Clustering Based on Network Topology.....	117
Figure A.2 Pseudo Code of the Traditional FCM.....	119
Figure A.3 Traditional (Standard) FCM Flowchart.....	120
Figure B.1 DeGiCA, FCM and K-means Network Lifetimes when $g = 130$ and $\sigma = 5$	123
Figure B.2 DeGiCA, FCM and K-means Energy Consumption when $g = 130$ and $\sigma = 5$	123
Figure B.3 DeGiCA, FCM and K-Means Network Lifetimes when $g = 155$ and $\sigma = 5$	123
Figure B.4 DeGiCA, FCM and K-means Energy Consumption when $g = 155$ and $\sigma = 5$	1244
Figure B.5 DeGiCA, FCM and K-means Network Lifetimes when $g = 140$ and $\sigma = 6$	1244
Figure B.6 DeGiCA, FCM and K-means Energy Consumption when $g = 140$ and $\sigma = 6$	1244
Figure B.7 Number of Live Nodes when $g = 155$ and $\sigma = 5$ in a Certain Time...	1255

Figure B.8 Number of Live Nodes when $g = 140$ and $\sigma = 6$ in a Certain Time	1255
Figure B.9 Energy Consumption Percentage when $g = 155$ and $\sigma = 5$ in a Certain Time	1255
Figure B.10 Energy Consumption Percentage when $g = 140$ and $\sigma = 6$ in a Certain Time	1266
Figure B.11 Percentage of Overall Delivered Packets when $g = 110$ and $\sigma = 4$	1266
Figure B.12 Percentage of Overall Delivered Packets when $g = 140$ and $\sigma = 6$	1266
Figure B.13 Gridded WSN when $g = 130$	1277
Figure B.14 Gridded WSN when $g = 140$	1277
Figure B.15 Grid Classification when $g = 130$	1277
Figure B.16 Cluster Formation when $g = 130$	1277
Figure B.17 Grid Classification when $g = 140$	1277
Figure B.18 Cluster Formation when $g = 140$	1277
Figure B.19 Death of First Node when $g = 130$	128
Figure B.20 Death of First Node when $g = 140$	128

LIST OF TABLES

Table 2.1: Some Clustering Algorithms with Clustering Parameters.....	32
Table 2.2: Descendant of LEACH Protocol.....	37
Table 2.3: Classification of Clustering Algorithms	48
Table 4.1: Structure of Data Stream Packet.....	74
Table 4.2: Experimental Setup Parameters.....	80
Table 4.3: Machine Minimum Requirement for MATLAB R2008b.....	81
Table 4.4: Ten Best Simulation Experiments for DeGiCA.....	82
Table 4.5: Number of Live Nodes in DeGiCA, K-Means and FCM at a Certain Time	88
Table 4.6: Percentage of Consumed Energy Results in DeGiCA, K-Means and FCM in a Certain Time	90
Table 4.7: Grid Size and Threshold Simulation Experiment Selection to Determine their Optimum Values of g and σ	93
Table 4.8: Performance Metrics of Competitors for 5 best Selected Simulation Experiments to Determine the Optimum Values	94
Table 4.9: DeGiCA Scalability Based on Optimum Values at Certain Time	95
Table 4.10: Number of Nodes to Die in Networks when $g = 110, 120, 130, 140$	99

LIST OF SYMBOLS AND TERMINOLOGY

ACE	Algorithm for Cluster Establishment
APTEEN	Adaptive Periodic TEEN
BARC	Battery Aware Reliable Clustering
BS(s)	Base Station(s)
CACC	Clustering Algorithm based on Cell Combination
CFL	Clustering for Localization
CH(s)	Cluster Head(s)
COD	Clustering on Demand
CODE	Coordination-based data Dissemination mechanism
DCA	Distributed Clustering Algorithm
DEEC	Distributed Energy-Efficient Clustering algorithm
DeGiCA	Density Grid-based Clustering Algorithm
DISC	Distributed Single-pass Incremental Clustering
DMAC	Distributed and Mobility-Adaptive Clustering Algorithm
DWEHC	Distributed Weight-Based Energy-Efficient Hierarchical Clustering
EECS	Energy Efficient Clustering Scheme
EEHC	Energy Efficient Hierarchical Clustering
EEUC	Energy-efficient unequal clustering
FC	Fractal Clustering
FCM	Fuzzy Clustering Means algorithm
FLOC	Fast Local Clustering service
FoVs	Overlapped Field of View
GAF	Geographic Adaptive Fidelity protocol
HAS	Harmony Search Algorithms
HSRP	Hybrid-Structure Routing Protocol
KOCA	K-Hop Overlapping Clustering Algorithm
LCA	Linked Cluster Algorithm
LCA2	Linked Cluster Algorithm 2
LEACH	Low-energy Adaptive Clustering Hierarchy
LEACHC	Centralized-Low energy adaptive clustering hierarchy
MN(s)	Member Node(s)
MRPUC	Multi-hop routing protocol with unequal clustering
PEACH	Power-efficient and adaptive clustering hierarchy

PEGASIS	Power-Efficient GAthering in Sensor Information Systems
PEZCA	Power-Efficient Zoning Clustering Algorithm
PDCH	Pegasis Algorithm Improving Based on Double Cluster Head
PSO-C	Centralized-PSO
SEAD	Scalable Energy efficient Asynchronous Dissemination mechanism
SEP	Stable Election Protocol
SN(s)	Sensor Node(s)
TEEN	Threshold sensitive Energy Efficient sensor Network protocol
TL-LEACH	Two-Level Hierarchy LEACH
TTDD	Two-Tier Data Dissemination mechanism
VAP-E	Energy-Efficient Clustering -Virtual Area Partition
VoGC	Voting-on-Grid clustering
WCA	Weighted Clustering Algorithm
WSN(s)	Wireless Sensor Network(s)

Chapter I

Introduction

Chapter I

Introduction

1.1 Introduction

Today, new technologies that appeared recently such as science of computer, genetic engineering and the emerging field of nanotechnology, differ from technologies that preceded them. Telephone, automobile, television and air travel accelerated for a while, transformed the society, but then settled into a manageable rate of change. In other words, computer science, biotechnology and nanotechnology don't work that way, they are called self-accelerating, where they grow continuously and rapidly.

In recent years, a widespread use of Wireless Sensor Networks (WSNs) have been found in several real life applications. They are increasingly used and found in many fields such as environmental, industrial, military, and agriculture [1]. WSNs are a special kind of networks that have the ability to sense and process information. The main functionality of a WSN is to monitor a certain physical phenomenon across a geographic area.

A WSN refers to a large-scale ad hoc wireless network of hundreds or even thousands of tiny, independent built-in devices called **Sensor Nodes (SN)** scattered in a sensing area. Sensor nodes are effective tools for sensing and gathering data in a variety of

environments [2]. Each sensor node contains four basic components: sensing unit, processing unit, transducer, and energy source. They have various functions for monitoring a wide variety surrounding conditions and collecting high precision and speed data such as light, temperature, humidity, magnetic field, pressure, acoustic and voice-level information [3]. Unfortunately, sensor nodes are low-power transceivers and known to be “energy constrained with limited computation capacity”. They are limited in terms of power, processing, and memory (remove). Obviously, they have sensing circuitry to measure ambient conditions from a surrounding environment [4] and continuously report parameters and collect sensed signals from real-life applications that consumes node battery leading to consume network energy thus shorten its lifetime. WSNs depend hardly on their sensors that consume a large amount of battery energy. Unfortunately, the nature of WSNs make it very difficult to recharge sensor node batteries [1].

In some sensor network applications, sensor nodes tend to generate a large amount of sequential small and tuple-oriented data that is considered as Data Streams. Data streams usually are huge data sets that arrive in an online fashion, flowing rapidly in a very high speed, where they are unlimited and there is no control on the arrival processing order [5]. Synoptically, there are some differences between sensor streams and traditional streams. Sensor streams are only samples of the entire population, imprecise, noisy, and with a moderate size. While in traditional streams the entire population is used, data is exact, error-free, and huge in size.

Due to difficulties in recharging node battery in remote harsh environments or even hostile terrains, energy is considered to be a significant resource and an important objective design in WSNs. Energy consumption in WSNs can be categorized into two

parts, communication and computation. Most energy is exhausted through communication among nodes rather than sensing or computing. Most studies attempt to extend network lifetime, allowing scalability for large numbers of sensors, and supporting fault tolerance for battery consumption and broken nodes. Applying these traditional approaches reduces overall network lifetime [3]. WSNs suffer from several resource constraints, such as bandwidth, storage, processing and energy constrain. Therefore, WSNs algorithms should be accurately designed in terms of facing challenges [5].

The widespread deployment of WSNs and the need for data aggregation requires efficient organization network topology to reach load balancing, network lifetime extension and energy consumption reduction. Data mining techniques deal with such requirements by using a special grouping technique called **Clustering**. Clustering has proven to be an effective approach for organizing a network into a connected hierarchy. At any rate, clustering technique plays an important role in network organization and performance. Owing to a variety of advantages, clustering is becoming an active branch in WSN data mining. Figure 1.1 shows a simple structure of a clustered WSN.

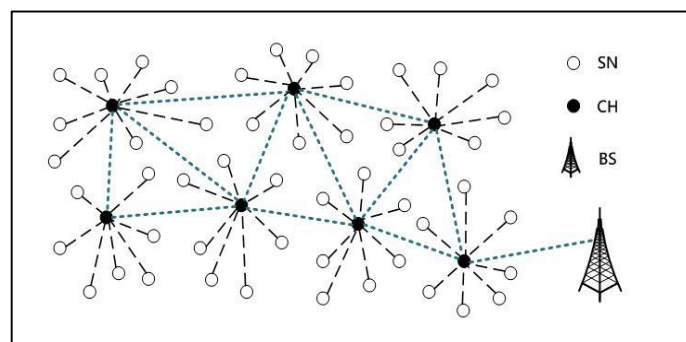


Figure 1.1 Simple Clustered WSN Structure [6]

As shown in figure 1.1, sensed streaming data is collected from surroundings by sensor nodes in a clustered WSN, upper level nodes with special characteristics called **Cluster Heads (CHs)** aggregate sensed data streams from sensor nodes to perform data processing and redundancy deletion. Each CH is considered to be leader on a group of sensor nodes called **Cluster**. After that, aggregated data is sent to another upper level with special observers called **Base Stations (BSs)** to be transferred later on to end users. A BS is relatively resourceful as compared to normal sensor nodes. It usually acts as a gateway to other networks. Transferring sensed data streams in a low-to-high level hierarchy through sensed medium exhausts network and consumes its energy [1].

In brief, clustering process starts when grouping similar sensor nodes in clusters. A node with specific parameters (i.e. residual energy, cluster ID, weight, etc.) or combination of parameters is selected to be a CH in a cluster. A CH is used to collect and aggregate data streams from other ordinary sensor nodes in its cluster. All sensor nodes within one cluster communicate and send data to their CH. CHs then aggregate data streams to be sent to the BS [4].

Due to WSNs restrictions listed above, sending a large amount of streaming data consumes nodes energy whenever transferred through wireless medium in a multi-hop data communication to reach the BS. For this reasons, it is aimed in this research to enhance data stream mining using clustering technique by developing a distributed clustering data stream algorithm called the Density Grid-base Clustering algorithm (DeGiCA) that enhances clustering mining technique in WSNs by combining density and grid techniques. By using a MATLAB-based simulator, DeGiCA is compared with other clustering algorithms in WSNs that manipulate with data streams. Simulation

results indicate that the developed DeGiCA outperforms its competitors in terms of network lifetime, energy consumption and packet delivery ratio.

1.2 Thesis Motivation

WSNs are playing a vital role in daily lives. Humans have depended on wired sensors for several years, from simple tasks such as temperature monitoring, to complex ones such as monitoring life-signs in hospital patients. Suddenly, WSNs appeared and provided unexpected applications, starting from military applications such as battlefield mapping and target surveillance, to creating context aware homes. In this case, sensors can monitor safety and provide automated services designed specially to individual users.

This thesis tried to narrow the performance gap of data stream mining in WSNs using density grid algorithms. Some existing clustering algorithms in WSNs don't provide high performance metrics. So, the DeGiCA is developed to enhance performance of some existing algorithms.

1.3 Problem Statement

With the urgent need of WSNs in several applications that process data streams, it is found that WSNs suffer from insufficient network lifetime that may impede workflow due to node battery energy consumption. Such applications require a long-lasting network lifetime. The most prominent challenges are how to long-last a WSN lifetime, How to reduce sensor nodes energy consumption, How to ensure delivering data stream packets from source to destination in this type of networks without failure.

To address such challenges, it is required to enhance data stream mining techniques in WSNs by using clustering technique. Data stream mining is concerned with extracting knowledge structures represented in models and patterns in non-stopping streams of information. Clustering plays an important role in organizing WSNs and affects network performance by extending network lifetime, saving energy of sensor nodes and ensuring delivery of data stream packets.

1.4 Thesis Aim and Objectives

Although WSNs are becoming a highly pursued research topic to be studied, evolving, increasingly used in various applications in data stream mining fields, WSNs still suffer from several important aspects it should characterize. Accordingly, the main research objectives of this thesis are:

1. The ambition to conserve sensor node energy during WSNs lifetime within the sensed area by using the efficient clustering data mining technique and consequently reducing their energy consumption as much as possible.
2. Applying clustering hierarchy that has proven its efficiency and effectiveness in prolonging WSNs lifetime while sensing streaming data.
3. Many clustering algorithms may fail while streaming data causing data loss. It is the aim to consider data streams during designing a clustered WSN. To ensure data stream packet delivery from source to final destination, clustering is used to benefit fault tolerance.

1.5 Research Methodology

To achieve the desired objectives, the thesis goes through the following steps:

1. Study related works in the field of WSNs, clustering and data streams. In terms of definitions, properties, structures and techniques welling to support background.
2. Review issues related to clustered WSNs schemes that stream data streams and study recently existing WSN clustering algorithms in terms of their advantages and disadvantages.
3. Developing Density Grid-based Clustering Model that needed several steps to be accomplished. Studying methods and combining techniques until putting together an overview of the proposed model.
4. Implementing DeGiCA and its competitors using MATLAB-based simulator. Take into consideration points of strengths and weaknesses, recording each and trying to solve weak points to achieve more enhancement.
5. Selecting two well-known WSN clustering algorithm that may address data stream: the standard Fuzzy Clustering-Means algorithm (FCM) and the well-known K-means. Both are considered to be DeGiCA competitors where they are used to evaluate the developed algorithm.
6. Testing and evaluating DeGiCA and comparing its performance metrics outcomes with its competitors' outcomes by using MATLAB-based simulator. Analyzing and discussing simulation experimental results and providing suggestions about future work.

1.6 Research Contribution

In recent researches, most WSNs clustering algorithms involving data streams heading towards a FCM-based clustering. FCM-based clustering algorithms are based on the

standard FCM. FCM is considered to be the most suitable WSN clustering algorithm that manipulates data streams due to its fuzzification nature. FCM-based clustering algorithms are developed to enhance FCM as discussed in chapter 2.

The DeGiCA has proven to be also a suitable environment for manipulating data streams in WSNs without being a FCM-based clustering algorithm. In fact, DeGiCA solves some problems found in FCM, as discussed later in chapter 3.

The research developed the DeGiCA that enhances clustering mining technique in WSNs by combining density and grid techniques. The density technique can find arbitrary shaped clusters while the grid technique is used to avoid clustering quality problems by discarding boundary nodes of grids. DeGiCA helps to face the limitations found in WSNs that involve data streams.

1.7 Thesis Organization

This thesis documentation is orderly organized and divided into five main chapters. The remaining of this thesis documentation is organized as follows:

Chapter 2 presents thorough literature review and previously related works. Specifically, it presents an overview on data mining and a survey of data clustering. It provides a discussion about WSNs, data streams and data mining. It provides a classification of WSN clustering algorithm described in details. Chapter 3 presents the proposed DeGiCA. It provides DeGiCA main parts and phases in detail. Each part of DeGiCA is explained by a pseudo code and flowchart. The dataset streaming data description is presented in chapter 4. It explains DeGiCA performance metrics,

experimental parameters and simulation analysis. Optimum values are obtained to scale the developed algorithm. Finally, chapter 5 presented thesis conclusion where comments on DeGiCA simulation results then suggested future work.

Chapter II

Literature Review

Chapter II

Literature Review

2.1 Introduction

In recent years, a widespread use of WSNs have been seen in various applications. A WSN is a special kind of ad-hoc networks that has the ability to sense and process information. They can be used in many fields such as environmental, industrial, military, and agriculture fields. WSNs contain tiny independent built-in devices called sensor nodes. Sensor nodes contain four basic components: sensing unit, processing unit, transducer, and energy source [7]. Sensor nodes are mainly used in data processing and continuously report parameters. Reports are transmitted by the sensor nodes and collected by special observers called Base Stations (BSs).

A WSN has several resource constraints, such as low computational power and limited energy source. WSNs depend hardly on their sensors that consumes a lot of battery energy. Unfortunately, the nature of WSNs makes it very difficult to recharge the sensor nodes batteries. Therefore, energy efficiency is an important objective design in WSNs [1] and their algorithms should be accurately designed based on energy saving.

In some sensor network applications, data that WSNs process usually contain a large amount of datasets that flow rapidly in a very high speed and arrive in an online fashion. Data are unlimited and there is no control on the arrival order of the elements being processed. Such data are called Data Streams [5, 7]. In general, there are some differences between sensor streams and traditional streams. Sensor streams are only samples of the entire population, imprecise, noisy, and with a moderate size. While in traditional streams the entire population is used, the data is exact, error-free, and huge in size [5].

The widespread deployment of WSNs and the need for aggregating data streams from them requires an efficient organization of network topology to reach load balancing and network lifetime extension. This can be done by using mining techniques. **Clustering** is a mining technique that has proven to be an effective approach in WSNs to solve the problem of network lifetime, energy consumption, data aggregation, load balancing, scalability [8, 9], delay and delivering data packets. It organizes WSNs into a connected hierarchy. Generally, there are two categories of networks in WSNs, flat networks and clustered (i.e. hierarchical) networks [10]. At any rate, clustering phenomenon plays an important role in network organization, and also affects the network performance. Owing to a variety of advantages, clustering is an active branch in mining WSNs data. In a clustered WSN, the network is divided into groups called clusters, each cluster has a leader elected from the sensor nodes called Cluster Head (CH) and other Member Nodes (MNs). Data streams are aggregated from the MNs by their CH inside a cluster. Then it is transmitted from CHs to the BS. Transmitting data streams in the wireless medium by a multi-hop communication to reach the BS resumes the energy of sensor nodes leading to shorten the network lifetime.

This chapter briefly provides some important concepts in WSNs, data streams, data stream mining, simulation, simulators, and clustering algorithms. The chapter is organized as follows: section 2.2 presents an overview on Wireless Sensor Networks (WSNs). Section 2.3 presents data streams in WSNs. Section 2.4 is an overview on data mining in WSNs. Section 2.5 presents data streams mining clustering technique in WSNs. Section 2.6 is a general classification for clustering protocols. Lastly, section 2.7 is the conclusion.

2.2 Wireless Sensor Networks (WSNs)

WSNs are the key to gather information needed by smart environments, whether in buildings, utilities, homes, shipboard, transportation systems automation, or elsewhere [11]. A WSN is a special kind of ad-hoc networks that have the ability to sense and process information. They can be used in many fields such as environmental, industrial, military, and agriculture fields [7]. In harsh regions, running wires or cabling is usually impractical. A WSN is required to be fast and easy to install and maintain [11]. WSNs contain tiny independent built-in devices **called sensor nodes**. Sensor nodes contain five basic components as shown in figure 2.1 [7].

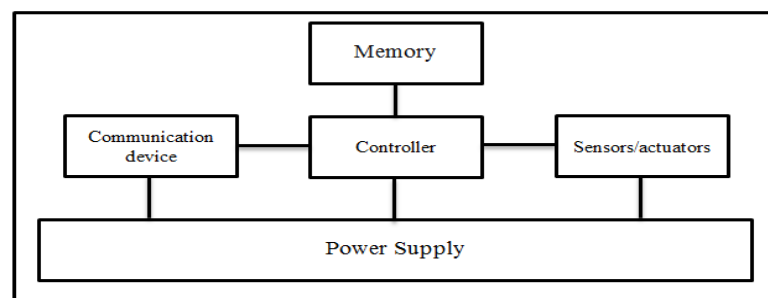


Figure 2.1 Main Sensor Node Hardware Components [7]

Nodes contain sensors, processor, memory and wireless transceivers often are small and have only very limited computational power and communication bandwidth [12]. There are two main types of applications for sensor networks: monitoring and actuating applications. In monitoring applications, the nodes only processes the data. In actuating applications, the nodes can interfere in the monitored environment [13]. In WSNs, the network lifetime is an important issue when building any algorithm and it can be defined as the time until the first/last node in the network depletes its energy, or it can be defined as the time until a node is disconnected from the BS [14].

2.3 Data Streams in WSNs

In many emerging applications, huge data streams are monitored in a network environment. For example, large sensor networks are widely used in wildlife monitoring, road traffic monitoring, and environment surveillance. Each sensor generates a data stream where new data entries keep arriving in a continuous manner. In order to aggregate and analyze streaming data under monitoring, it is often required to transmit data streams over the network [12].

WSNs are energy constrained, and the extension of their lifetime is one of the most important issues during the design. WSNs process a specific type of data called data streams. Nodes tend to generate the data streams in a large amount of sequential small and tuple-oriented form. Data streams usually are huge datasets that arrive in an online fashion, flowing rapidly in a very high speed, where they are unlimited and there is no control on the arrival processing order [5, 7]. However, there is a difference between

sensor streams and traditional streams. Sensor streams arrive continuously, thus clustering algorithms have to perform in a single scan.

An important characteristic of data streams is that they can be mined in a distributed fashion. Individual processors may have limited processing and memory. In sensor networks, it may be desirable to mine data streams with limited processing and memory [15].

2.3.1 Traditional Data Mining and Data Stream Mining in WSNs

Traditional data mining differs from WSN data stream mining. The traditional data mining is centralized, computationally expensive, and focuses on disk-resident transactional data. It collects data at the central site which is not bounded by computational resources. On the other hand, a WSN data streams flow continuously in systems with varying update rates. It is impossible to store the entire data stream or to scan it through multiple times, due to its high speed, huge amount and high storage cost [16]. Specifically, data stream clustering analysis causes some challenges for traditional clustering algorithms. First, data can only be examined in one pass. Second, viewing a data stream as a long vector of data is not enough in many applications [17].

2.3.2 Data Stream Characteristics in WSNs

Data stream has different characteristics of data collection compared to the traditional database model. First, when data stream arrives, it isn't easy to be controlled by arrival order. Second, data streams are continuously generated as time progresses. Third, data

streams are dynamic. Additionally, data stream must be read and processed based on the arrival order. The order of data cannot be changed to improve the results [18]. Based on the data stream characteristics, the processing of data stream requires first, each data element must be examined once, because it is impossible to keep the entire stream in the main memory. Second, each data element in data streams should be processed as soon as possible. Third, the memory usage for mining data streams should be limited even though new data elements are continuously generated. Finally, the results generated by the online algorithms should be immediately available upon user request [18].

2.3.3 Algorithms of Data Streams

There are several data stream algorithms found in literature to handle the data streams through various environment [15]. The following two techniques provide brief descriptions on the concept of clustering in data streams and mining streams in sensor networks that are focused on the research scope study. Clustering is a widely studied problem in data mining. Due to the one-pass constraint on the dataset, it is difficult to adapt arbitrary clustering algorithms to data streams. In the context of data streams, it is better to determine clusters in specific user defined horizons rather than on the entire dataset. The micro-clustering technique determines clusters over the entire dataset. Many applications of micro-clustering which can perform effective summarization based analysis of the dataset [15].

On the other hand, it has become possible to track large amounts of data in a distributed fashion with the use of sensor technology. The large amounts of data collected by the

sensor nodes makes the problem of monitoring. Due to the resource limitations in sensor networks, when a given network have millions of sensor nodes, it becomes very expensive to localize all data at a given global node for analysis. This is a point of view on storage and communication [15].

2.4 Overview of Data Mining in WSNs

A widespread use of data mining is found in several application fields and various environments in the last decades. Data mining is defined as the computational process of discovering patterns in large datasets involving methods at the intersection of artificial intelligence (AI), machine learning, statistics and database systems [19]. Large amount of databases in various areas have been generated from the development of information technology (IT). The research in databases and IT has given the importance to store and manipulate such data for further decision making. Data mining is a process of extracting information and patterns from huge data [20] or even rapidly flowing data such as data streams.

2.4.1 Common Classes of Data Mining

There are six common classes of tasks for data mining [21]:

- Anomaly detection: (outlier/change/deviation detection) unusual data records identification or data errors that require further investigation.

- Association rule learning: (dependency modelling) searches for relationships between variables. For example, in market basket analysis, a supermarket might gather data on customer purchasing habits by using this rule.
- Clustering: it is the task of discovering groups of data that are similar to each other.
- Classification: it is the task of generalizing known structure to apply to new data.
- Regression: it allows to find a function which models the data with the least error.
- Summarization: it provides a compact representation of the dataset, including visualization and report generation.

2.4.2 Data Stream Mining in WSNs

Today many organizations have a lot of large databases that grow without limit at a rate of several million records per day. Mining these continuous data streams brings new challenges. Knowledge discovery systems are constrained by three main limited resources: time, memory and sample size. In contrast, in many data mining applications, the bottleneck is time and memory, not samples [22].

Managing and processing data streams in WSNs has become a topic of research in several fields of data mining. The main purpose of deploying the WSNs is to make the real-time decision which has been proved to be challenging due to many resource constrained computing, communicating capacities, and huge volume of data streams generated by WSNs. This challenge helps the research community to find data mining techniques dealing with extracting knowledge from large continuous arriving data streams from WSNs. Traditional data mining techniques are not suitable for mining

data streams in WSNs, due to the nature of sensor streaming data, their characteristics, and network limitations [16].

2.4.2.1 Challenges of Data Stream Mining in WSNs

Conventional data mining techniques when handling data streams in WSNs are challenging for the following reasons, [16]:

- 1- Resource constraint. The sensor nodes are resource constraints in terms of power, memory, communication bandwidth, and computational power.
- 2- High data rate and huge data size. The nature of streaming data in WSNs is having a very high speed. They flow rapidly and continuously. In many fields, data streams arrive faster than they could be mined. The challenge for data mining techniques is how to manipulate with the continuous, rapid, and changing data streams and also how to incorporate user interaction during high-speed data stream arrival.
- 3- Online data stream distributed mining. In WSNs, data stream is geographically distributed, inputs arrive continuously and so far newer data may change the results based on older ones. Most data mining techniques that analyze data in an offline manner do not meet the requirement of handling distributed stream data. So, how to process distributed streaming data online is a big challenge.
- 4- Modeling changes of mining results over time. Data-generating phenomenon is changing over time. So the extracted model should be updated continuously. Due to the continuity of data streams, capturing the change of mining results is more important than mining the results.

- 5- Data transformation. Sensor nodes are limited in terms of bandwidth. So, transforming original data over the network is not easy.
- 6- Dynamic network topology. Sensor networks are deployed in harsh, uncertain, heterogenic, and dynamic environments. Sensor nodes may move among different locations at any point over time. This can increase the complexity of designing an appropriate mining technique.

2.4.2.2 Taxonomy of Data Stream Mining Techniques for WSNs

There are three main classification levels in data stream mining techniques for WSNs. The highest-level [16] of classification is based upon the general data mining classes used, such as frequent pattern mining, sequential pattern mining, clustering, and classification. For the frequent pattern mining and sequential pattern mining, they have adopted the traditional frequent mining techniques to find the association among large WSNs data. For the clustering, it adopted the K-mean, hierarchical, and data correlation-based. While in the classification, the traditional classification techniques were adopted. Such techniques are decision trees, rule-based, nearest neighbor, and support vector machines.

The second level [16] of classification is based on the ability to process data streams in a centralized or distributed manner. Since WSNs nodes has limited resource, the approach meant for distributed processing requires one-pass algorithms to complete a part of data mining locally, and then gather the results. The distributed approaches are used to increase the WSNs lifetime, and can extract a large number of data streams from the environment.

The third level [16] of classification is selected based on how to face a specific problem. In WSNs, research has been focused on two aspects: performance and application. Mainly, sensor nodes are constrained in some resources, so, algorithm that address such constrains are needed to maximize the WSNs performance. On the other hand, a WSNs application requires data accuracy, fault tolerance, event prediction, scalability and robustness.

2.4.2.3 Application Areas of WSNs Data Stream Mining

The following points are examples of real-world WSNs applications using data stream mining techniques [16]:

1. In the environmental monitoring, sensors are deployed in an unattended region to monitor the natural environment. Data mining techniques can identify when and where an event may occur and gives an alarm whenever detection.
2. For the health monitoring, patients are equipped with small sensors on multiple different positions of their body to monitor their health or behavior. Data mining technique can identify the abnormal behavior and help to take effective action.
3. Sensors in object tracking are embedded in moving targets to track them in real-time. Data mining techniques help to find the location of targets and to make tracking more accurate.
4. WSNs are usually deployed in harsh environments. Sensor nodes are resource constrained especially in terms of power. Data mining techniques help to identify the faulty or dead nodes.

5. In data analysis, data mining techniques help to discover data patterns in a sensor network for a certain application.
6. In real-time monitoring, data mining techniques help to identify certain patterns and predict future events, which make real-time response and action feasible.

2.4.2.4 Implementation of WSNs Data Stream Mining

Three main techniques are used for data stream mining implementation in WSNs [16]:

- 1- Evaluation method. Analytical modeling (very complex), simulation (the most popular and effective), and real deployment (difficult) are the most commonly used techniques to analyze the performance of data stream mining technique in WSNs.
- 2- Data source. The dataset used to experimentally validate the proposed technique. Two types of dataset are used, synthetic and real. Most techniques use simulation on synthetic dataset to validate results.
- 3- Optimization objective. WSNs are constrained in different resources such as network size, communication overhead, energy efficiency, memory requirements, node mobility, and so on. Data stream mining techniques should consider those constraints but mostly they cannot efficiently cover all the performance metrics. The large variations in the performance metrics makes it difficult to present a comprehensive evaluation.

2.4.2.5 Limitations of Existing Data Stream Mining Techniques for WSNs

Existing data stream mining algorithms have many limitations. Some of these limitations are: [16]

1. Most techniques do not take into account the heterogeneous streaming data and assume that sensor streaming data is homogenous.
2. Most techniques have a high computational complexity and a reduced accuracy due to considering only spatial, temporal or spatiotemporal correlations among sensor streaming data of neighboring nodes and ignoring the attribute dependency among sensor nodes.
3. The techniques that consider spatial correlation suffer from the choice of appropriate neighborhood range, while techniques which consider temporal correlation suffer from the choice of sliding window size.
4. Most techniques use centralized approach where data streams are transmitted to the BS for identifying certain patterns. The centralized approach causes communication overhead and more delay.
5. Although a lot of simulators are available and play an important role for developing and testing new techniques, simulation results may not be accurate.
6. The techniques evaluated by analytical modeling use certain simplification and assumption to evaluate the performance of proposed technique. This may lead to inaccurate results with limited confidence.
7. Most techniques assume stationary sensor nodes and do not consider node mobility. Applying these techniques for mobile networks or networks with dynamic changed topology would be challenging.

8. Frequent pattern mining approaches suffer from the choice of proper and flexible support and confidence threshold. Clustering techniques suffer from the choice of an appropriate parameter of cluster width, and computing the distance between data instances in heterogeneous data is computationally expensive. Classification-based techniques require some prior knowledge to classify the incoming data stream.

2.5 Data Stream Mining Clustering Techniques in WSNs

Clustering technique plays an important role in affecting a network performance and organization. There are several key limitations in WSNs clustering schemes must consider, such as [23], energy of nodes, network lifetime, abilities of nodes and application dependency, where an optimal clustering algorithm should be able to adapt to a variety of application requirements. **Data Clustering** is grouping similar objects. In clustering, it is easy to identify dense and sparse regions and also discover overall distributed patterns and correlations among data attributes [20]. In order to support data aggregation through efficient network organization, nodes can be partitioned into a number of small groups called **Clusters**. Each cluster has a coordinator called Cluster Head (CH), and a number of Member Nodes (MNs) [1].

2.5.1 Components of Clustered WSNs

Clustering in the organizational structure of WSNs has five important components shown in figure 2.2, and listed as the following:

- Sensor nodes, which are considered the core component of any WSN. They can take on multiple roles, such as, sensing, storing data, routing, and processing.
- Clusters, which are considered the organizational unit for WSNs. The natures of these networks require to be broken down into clusters to simplify tasks.
- Cluster heads, which are used as an organization leaders of a cluster. They are required to organize activities in the cluster. Such activities include data aggregation and organizing the communication schedule of a cluster.
- The BS, which is used at the upper level of the hierarchical WSN. It gives the communication link between the sensor network and the end-user.
- End User, the queries in a sensor network are generated by the end user due to the wide-range of applications found in the sensor networks [8, 14].

2.5.2 Hierarchical Clustering Structure in WSNs

In a hierarchical network structure, each cluster has a CH that performs special tasks (i.e. fusion and aggregation) and several common sensor nodes as members [24].

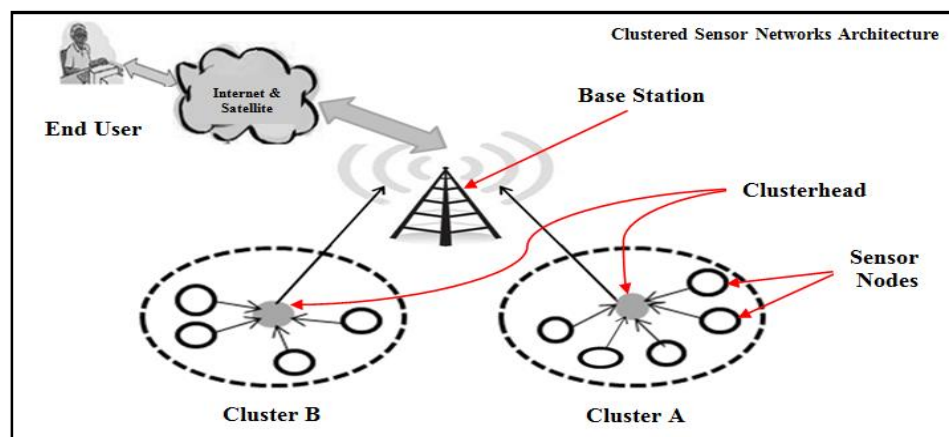


Figure 2.2 Clustered Sensor Networks Architecture [8] [21]

The cluster formation leads to a three-level hierarchy where the BS is the highest level, CHs form the middle level and sensor nodes form the lower level. The sensor nodes periodically transmit their data to their corresponding CHs. CHs aggregate streaming data packets and then transmit them to the BS. This can be directly or through the intermediate communication with other CHs. CHs spend a lot of energy more than other sensor nodes due to sending the aggregated data all the time to higher distances. A common solution to balance the energy consumption among all nodes in the network, is to periodically re-elect new CHs (rotating the CH role among all nodes over time) in each cluster. Figure 2.3 shows an example of a hierarchical data communication in a clustered network (assuming single-hop intra-cluster communication and multi-hop inter-cluster communication) [24]. The BS is the point of data processing for the data received from sensor nodes, and where data is accessed by the end user. The BS is considered to be fixed and at a far distance from the sensor nodes. CHs act as gateways between the sensor nodes and the BS. In some way, any CH is a sink for its nodes, and the BS is the sink for the CHs.

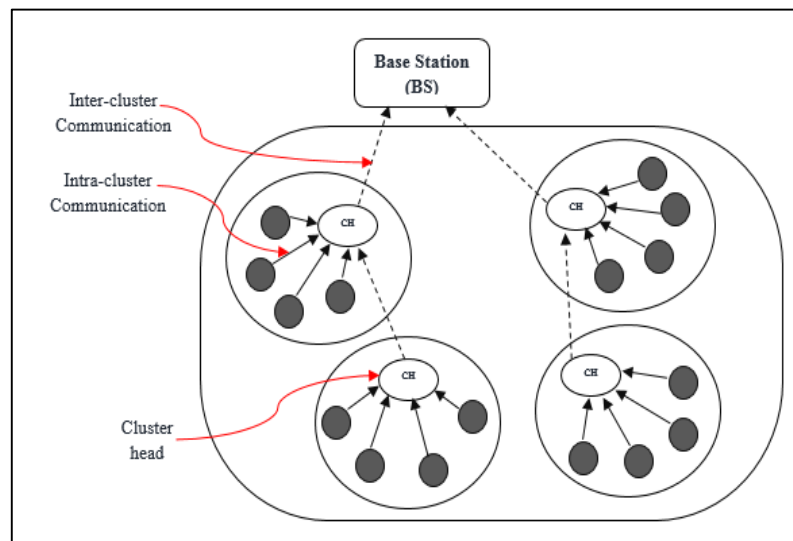


Figure 2.3 Data Communication in a Clustered Network [18]

2.5.3 WSNs Clustering Characteristics

In WSNs, several clustering methods could be applied separately or combined to reach the best network organization, depending mainly on the application being used. Clustering methods are: the partitioning method, the hierarchical (i.e. divisive) methods, the density based methods, the grid-based methods and the model-based methods [20]. The DeGiCA is a combination of three methods (i.e. hierarchical, density and grid-based methods). The clustering methods share the following clustering characteristics: [1]

1. Rotating the role of CHs, it is important to rotate the role of CH between nodes not to overload some nodes with more duties than others.
2. Node duty cycle, allowing sensors to sleep when not active, is a main factor that extends the battery lifetime. Node's duty cycle can be done in one of two ways, depending on the type of application. First, non-CH nodes are allowed to sleep when they are not communicating or not sensing. Second, an application needs sensors to monitor the field for unexpected events continuously.
3. Optimal cluster size, most clustering protocols assume a fixed cluster transmission range that results in similar cluster sizes. However, this results in unequal CHs load distribution.
4. Node synchronization, when sensor nodes are synchronized, distributed clustering protocols achieve their best performance. Node synchronization ensures that clustering process starts at the same time in the network.

2.5.4 Cluster Head and Member Node Properties in Clustering Technique

Clustered nodes in WSNs has some properties include:

- 1- **Connectivity:** there are two types of communication. **Single-hop** communication is a direct communication between sensor nodes and the BS. It is simple and easy to implement. **Multi-hop** communication is usually more complex where it uses some kind of intermediate nodes when transmitting data packets from a source to a BS. The DeGiCA uses a multi-hop communication in a hierarchical structure.
- 2- **Mobility:** Node mobility increases the complexity of any WSN data mining technique. The majority of techniques assume that sensor nodes are static, only few techniques consider the mobility of nodes.
- 3- **Node Role:** Nodes can take one of the following roles:
 - **Regular Sensor.** These nodes have limited resources, they are used to sense the environment and send the sensed data to the BS.
 - **Cluster Head (CH).** This can be a regular sensor node, or can be a node with rich resources. In centralized approaches, CHs are regular sensor nodes. In distributed approaches, besides responding for cluster formation, CHs perform aggregation/fusion of the collected data.
 - **Relay.** A node that acts as medium to transmit data packets from one node to another.
- 4- **Node Task:** In centralized approach, nodes are used to sense the environment being monitored and send the sensed data to the BS. In distributed approaches, nodes can make computations, and take actions based on detected events [16].

2.5.5 Designing Clustered WSNs

There are several key attributes designers must carefully consider when designing clustered WSNs [23]:

1. Cost of clustering. Although clustering plays a vital role in organizing sensor networks, there are some resources, such as communication and processing that are required in the creation and maintenance of clustering.
2. Selecting CHs. Clustering concept provides lots of benefits for WSNs. However, when designing a particular application, designers must carefully examine the formation of clusters in the network. This prerequisite may have an impact on how CHs are selected in a certain application.
3. Real time operation. Some applications such as habitat monitoring, simply receiving data is enough for analysis, meaning that delay is not an important issue. Other applications such as military tracking, real-time data acquisition becomes much more vital.
4. Synchronization. One of the primary limitations in WSNs is the limited energy capacity of nodes. Slotted transmission schemes, allow nodes to regularly schedule sleep intervals to minimize energy consumption. Such schemes require synchronization mechanisms to setup and maintain the transmission schedule.
5. Data aggregation. The ability for data aggregation in WSNs is one of its advantages. In crowded networks, there are many nodes sensing similar data. Data aggregation allows distinguishing between sensed data and useful data.
6. Repair mechanisms. WSNs are often vulnerable to node mobility, node death and interference that can result in link failure. So, it is important to surround the

mechanisms of link recovery and reliable data communication in clustering schemes.

7. Quality of Service (QoS): QoS has requirements in WSNs. Many of these requirements are application dependent (i.e. acceptable delay and packet loss tolerance), it is important to look at these metrics when choosing a clustering scheme.

2.5.6 WSNs Clustering Parameters

Clustering has many parameters, listed as follows [24, 25]:

1. Number of clusters (cluster count). The number of clusters is a critical parameter with regard to the efficiency of clustering algorithm.
2. Intra-cluster communication. The communication between a sensor and its designated CH is assumed to be a one-hop communication. However, multi-hop communication is required when the communication range of CHs is limited.
3. Nodes and CH mobility. When CHs or nodes are assumed to be mobile, the cluster membership for each node should dynamically change and clusters need to be continuously maintained. On the other hand, static CHs tend to yield stable clusters and facilitate intra-cluster and inter-cluster network management.
4. Nodes types and roles. In heterogeneous networks, CHs are able to have more computation and communication resources than others. In homogeneous networks, all nodes have the same capabilities and some are designated as CHs.
5. Cluster formation methodology. Clustering mostly is performed in a distributed manner without coordination. In few earlier approaches, a centralized (or hybrid

- especially when CHs are in rich resources) approach uses one or more coordinator nodes to partition the whole network off-line and control the cluster membership.
6. Cluster-Head selection. CHs can be pre-assigned in heterogeneous environments. In most cases, CHs are selected from the deployed set of nodes. They can be chosen randomly or based on other criteria such as the residual energy, connectivity etc.
 7. Algorithm complexity. The clustering algorithm complexity can be constant or dependent on the number of CHs and/or sensors.
 8. Adaptability. A clustering algorithm is said to be adaptive when the number of clusters changes and the node's membership evolves overtime. Otherwise, it is considered fixed when sensors do not switch among clusters and the number of clusters is not changed.
 9. Number of levels. The concept of a multi-level cluster hierarchy provides better energy distribution and total energy consumption instead of using only one cluster level.

Table 2.1, shows some clustering parameters for some clustering algorithms. Many clustering protocols shown in table 2.1 are considered to be distributed. They could randomly select their CHs but has a limited node mobility such as the Low-energy Adaptive Clustering Hierarchy (LEACH) [24, 25] and Two-Level Hierarchy LEACH (TL-LEACH) [25, 26]. The Centralized-Low energy adaptive clustering hierarchy (LEACHC) [24, 25] selects CHs randomly and has a limited node mobility but is considered to be a centralized clustering protocol. Linked Cluster Algorithm (LCA) [24] is a distributed clustering protocol with a possible node mobility and an ID-based selection for CHs. Some distributed protocols are based on the highest energy for CH selection with no node mobility such as the Energy Efficient Clustering Scheme

(EECS) [24, 25]. On the other hand, some are depending on the connectivity in CH selection with possible node mobility such as the Algorithm for Cluster Establishment (ACE) [24]. The distributed Weighted Clustering Algorithm (WCA) [24] uses a weight-based CH selection with node mobility. GROUP [24] is a hybrid clustering protocol with no node mobility.

TABLE 2.1 Some Clustering Algorithms with Clustering Parameters

Name of Algorithm	CH Selection	Node Mobility	Clustering Methodology	In cluster topology	Multiple Levels
LCA	ID-based	Possible	Distributed	1-hop	No
LEACH	Random	Limited	Distributed	1-hop	No
HEED	Random	Limited	Distributed	1-hop	No
TL-LEACH	Random	Limited	Distributed	1-hop	Yes
GROUP	Proximity	No	Hybrid	k-hop	No
EECS	Energy	No	Distributed	1-hop	No
WCA	Weight-based	Yes	Distributed	1-hop	No
ACE	Connectivity	Possible	Distributed	k-hop	No
LEACHC	Random	Limited	Centralized	1-hop	No

2.5.7 Implementing Clustered WSNs

The most popular and effective method to build and evaluate WSNs is the Evaluation Method as mentioned in section 2.4.2.4, by using a simulation tool. In simulation, the behavior of a network can be modeled by calculating the interaction between different network components using mathematical formulas. A simulation can be used together with different applications and services in order to observe end-to-end or point-to-point performance in networks [27]. Network simulators are useful in allowing network designers to test new networking protocols or to change existing protocols in a controlled and reproducible manner. Network simulators model the real world networks. However, network simulators are not perfect. They can't perfectly model all the details of the network. If well modeled, they are close enough to give researchers meaningful insights about networks under test and show operation changes [27]. Some

examples of commercial simulators are the OPNET, and QualNet. While open source simulators are NS2, NS3, OMNeT++, SSFNet, and J-Sim [27].

The developed DeGiCA is implemented and evaluated by using a MATLAB-based simulator. MATLAB [28] (matrix laboratory) is a multi-paradigm numerical computing environment and fourth-generation programming language. It allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, C#, Java, Fortran and Python.

2.6 Classification of Clustering Protocols

Clustering protocols could be generally classified into three main types in this research scope. First, WSNs clustering algorithms. Second, clustering algorithms involving data stream. Third, WSNs clustering algorithms involving data stream. The following sections describe in details the main three classifications.

2.6.1 Clustering Protocols in WSNs not involving Data Streams

This section presents the first classification of clustering algorithms. Based on network structure, protocols found in WSNs can be divided into two categories: protocols for either flat networks or hierarchical networks. In a flat network topology, all nodes perform the same tasks and have the same functionalities. Data transmission is performed hop by hop using some form of flooding. Clustering protocols in flat WSNs include Flooding and Gossiping, Sensor Protocols for Information via Negotiation

(SPIN), Directed Diffusion (DD), Rumor, Greedy Perimeter Stateless Routing (GPSR), Trajectory Based Forwarding (TBF), Energy-Aware Routing (EAR), Gradient-Based Routing (GBR), Sequential Assignment Routing (SAR). These clustering protocols are effective in small-scale networks. However, flat WSNs protocols are undesirable in large-scale networks because resources are limited, but all sensor nodes generate more data processing and bandwidth usage [10].

On the other hand, in a hierarchical topology, nodes perform different tasks and are organized into lots of clusters according to specific requirements or metrics. Generally, each cluster has a CH and number of MNs. In general, CHs have the highest energy in the clusters to perform data processing and information transmission, while nodes with low energy act as MNs and perform the task of information sensing [10].

Clustering protocols in a hierarchical WSN topology include Low-energy Adaptive Clustering Hierarchy (LEACH), Hybrid Energy-Efficient Distributed clustering (HEED), Distributed Weight-based Energy-efficient Hierarchical Clustering protocol (DWEHC), Position-based Aggregator Node Election protocol (PANEL), Two-Level Hierarchy LEACH (TL-LEACH), Unequal Clustering Size (UCS) model, Energy Efficient Clustering Scheme (EECS), Energy-Efficient Uneven Clustering (EEUC) algorithm, Algorithm for Cluster Establishment (ACE), Base-Station Controlled Dynamic Clustering Protocol (BCDCP), Power-Efficient Gathering in Sensor Information Systems (PEGASIS), Threshold sensitive Energy Efficient sensor Network protocol (TEEN), The Adaptive Threshold sensitive Energy Efficient sensor Network protocol (APTEEN), Two-Tier Data Dissemination (TTDD), Concentric Clustering Scheme (CCS), Hierarchical Geographic Multicast Routing (HGMR), and etc. Clustering technique is an active branch in hierarchical WSNs due to many

advantages, such as more scalability, data aggregation/fusion, less load, less energy consumption, more robustness [10]. Clustering WSNs protocols not manipulating with data streams has two types of clustering, proactive and reactive clustering.

2.6.1.1 Proactive and Reactive Clustering in WSNs

Proactive clustering algorithms are based on assuming that sensors always have data to send, for this reason, they should all be considered during cluster formation process. On the other hand, reactive algorithms take the advantage of user queries for the sensed data or of specific triggering events occur in WSNs. Namely, nodes may react immediately to sudden hard changes in the value of a sensed attribute. Reactive approach is useful for time-critical applications, but not suited for applications where data retrieval is required on a regular basis [24]. Figure 2.4, shows some examples of “proactive” and “reactive” clustering protocols.

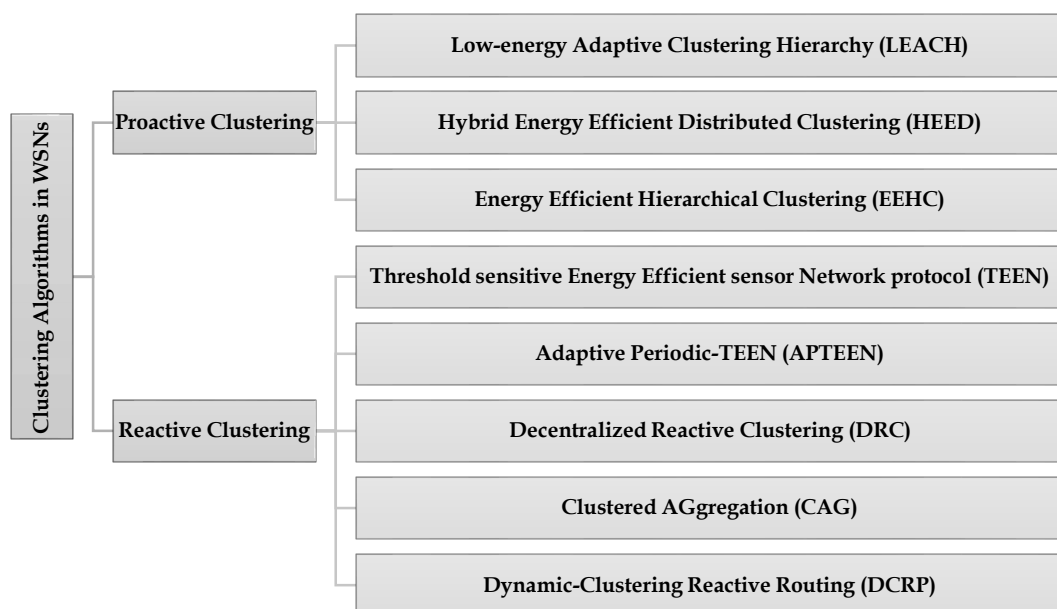
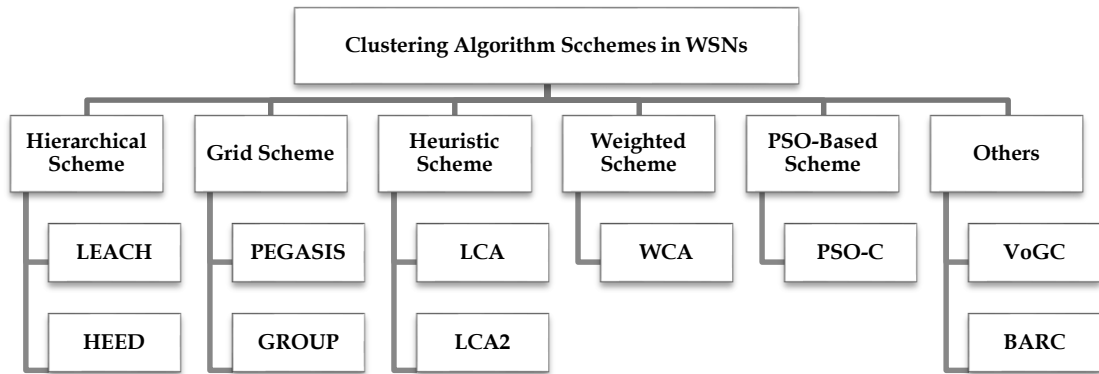


Figure 2.4 Proactive and Reactive Clustering Protocols

2.6.1.2 Clustering Algorithm Schemes in WSNs

Clustering WSNs algorithms could be considered under specific schemes, such schemes are, hierarchical scheme, grid scheme, heuristic scheme, weighted scheme, PSO-Based scheme and other schemes. Each scheme is described in brief [14, 23, 29]. Figure 2.5 summarizes the clustering schemes in WSNs with examples of each clustering scheme.



1- Hierarchical scheme. The cluster formation in the hierarchical scheme leads to a two-level hierarchy where CHs form the higher level and sensor nodes form the lower level. LEACH, Low-Energy Adaptive Clustering Hierarchy (LEACH) is one of the first major improvements on traditional clustering hierarchical approaches in WSNs. It provides a balancing of energy usage by rotating CHs randomly. Data-fusion in LEACH can be used to reduce the amount of data transmission. The decision of whether a node becomes a CH is made dynamically at each interval. [17, 21]. LEACH is not suitable when CHs are far from the BS. Therefore, a large number of algorithms have been proposed to improve LEACH, such as PEGASIS, TEEN, APTEEN, MECH, LEACH-C, EEPSC [8]. To reduce inter-cluster and intra-cluster collisions, LEACH uses a TDMA or a CDMA MAC. The energy

consumption to transfer gathered information from nodes to the BS depends on the number of CHs. So, it can be reduced by organizing nodes in clusters [29]. Table 2.2 gives some examples of the LEACH descendants.

TABLE 2.2 Descendant of LEACH Protocol

Descendant of LEACH	Name
LEACH	Low energy adaptive clustering hierarchy
LEACH-C	Centralized-Low energy adaptive clustering hierarchy
LEACH-B	Balanced-Low energy adaptive clustering hierarchy
LEACH-ET	Energy threshold-Low energy adaptive clustering hierarchy
TL-LEACH	Three Layer-Low energy adaptive clustering hierarchy
Armor-LEACH	Advance LEACH routing protocol for micro-sensor networks
O-LEACH	Optical-Low energy adaptive clustering hierarchy
MR-LEACH	Multi-hop hop routing-Low energy adaptive clustering hierarchy
LEACH-D	Low energy adaptive clustering hierarchy-D

HEED, Hybrid Energy-Efficient Distributed Clustering, is a multi-hop hierarchical clustering algorithm in WSNs. It focuses on efficient clustering by proper selection of CHs based on the physical distance between nodes. HEED reduces energy consumption during the CH selection phase and minimizes the network overhead. In HEED, CH selection is determined based on two important parameters. First, the residual energy of each node. Second, intra-cluster communication, to determine which cluster to join. This is usually used when a given node falls within the range of more than one CH [14, 23].

2- Grid scheme. This scheme divides the sensed area into equal size of cells called grids. Sensor nodes are pointed inside grids. Grid scheme has proven to be an effective technique to enhance WSNs performance. PEGASIS, Power-Efficient GATHERing in Sensor Information Systems, is a data-gathering algorithm where energy savings can result from nodes not directly forming clusters. If nodes form a chain from a source to a BS, only one node in any given transmission time-frame is transmitting. The average

transmission range required by a node to relay information can be much less than in LEACH [14, 23]. Another grid-based clustering algorithm is called GROUP. In GROUP, one sink is called a primary sink, it dynamically and randomly builds the cluster grid. Each new CH then selects more CHs along the grid until all CHs have been selected. The selections are based on the residual energy of nodes near the corners of the grid [14, 23].

3- Heuristic scheme. It is an algorithm that has one or both of its goals during solving a problem. First, finding an algorithm with reasonable run-time. Second, finding an optimal solution. Linked Cluster Algorithm (LCA), is one of the very first developed heuristic clustering algorithms. It is developed for wired sensors, but later implemented in WSNs. In LCA, each node is assigned to a unique ID and can become a CH in two ways. Firstly, if a node has the highest ID number in a set including all neighbor nodes and the node itself. Secondly, assuming none of its neighbors are CHs, then it becomes a CH [14, 23]. Linked Cluster Algorithm 2 (LCA2) is proposed to eliminate the election of an unnecessary number of CHs found in LCA. LCA2 introduced the concept of covered and non-covered nodes. A node is covered when one of its neighbors is a CH [14, 23].

4- Weighted scheme. In this scheme, CH election depends on weights. Weighted Clustering Algorithm (WCA), is a non-periodic procedure to CH election, invoked when every time a reconstruction of networks topology is unavoidable. WCA finds a long-lasting architecture during the first CH election. When a sensor loses connection with its CH, the election procedure is invoked to find a new clustering topology. This is an important feature in power saving. WCA is based on a combination of metrics such as node degree, transmission power, mobility and residual energy. WCA is fully

distributed where nodes in the mobile network share the same responsibility acting as CHs [14, 23].

5- PSO-Based scheme. Centralized-Particle Swarm Optimization (PSO-C). Nodes having energy above the average energy resource are elected as CHs. Simulation results show that PSO outperforms to LEACH and LEACH-C in term of network lifetime and throughput [29].

6- Other clustering schemes. Voting-on-Grid Clustering (VoGC), is a combination of voting method and clustering algorithm, developing new clustering schemes for WSNs secure localization. VoGC is used instead of traditional clustering to reduce the computational cost. This scheme can provide good localization accuracy and identify a high degree of malicious beacon signals [29]. A new mathematical battery model for implementation in WSNs called Battery Aware Reliable Clustering (BARC), is used in clustering algorithm. It improves WSNs performance over other clustering algorithms due to using Z-MAC and CH rotation according to battery recovery schemes. BARC enhances network lifetime greatly compared to other clustering algorithms [29].

2.6.2 Clustering Protocols involving Data Streams

This section presents the second classification of clustering algorithms. It presents some common clustering algorithms involving data streams not applied on WSNs. In 2006, Feng Cao proposed the DenStream algorithm for clustering dynamic data stream [30]. It is an effective and efficient method that can discover clusters of arbitrary shape in data streams, but it is insensitive to noise [31]. DenStream extends the micro cluster concept, and introduces the outlier and potential micro clusters to distinguish between

real data and outliers. New data records are added to existing potential micro clusters, which increases the radius of micro clusters [32]. Heng Zhu Wei proposed a density and space clustering algorithm called CluStream [30]. CluStream is a data-stream clustering algorithm based on k-means that is inefficient to find clusters of arbitrary shapes and cannot handle outliers. Further, they require to know k parameter and user-specified time window [17]. DenStream and CluStream are not able to reveal clusters of arbitrary shape effectively and cannot distinguish clusters which have different levels of density [30].

K-means algorithm is used in the offline phase of some algorithms such as Clustream. It is a divide and conquer schemes that partition data streams into segments and discover clusters in data streams. K-means has a number of limitations. First, it aims at identifying spherical clusters but is incapable of revealing clusters of arbitrary shapes. Second, it is unable to detect noise and outliers. Third, the algorithm requires multiple scans of data, making it not directly applicable to a large data stream volume [17, 30]. STREAM and CluStream are two well-known extensions of K-means on data streams [32]. Many recent data stream clustering algorithms are based on CluStream's two-phase framework. Wang et al. [17] proposed an improved offline component using an incomplete partitioning strategy. An extensions of this component including clustering multiple data streams, parallel data streams, distributed data streams and applications of data stream mining.

LOCALSEARCH, STREAM, DenStream and CluStream are clustering algorithms involving data streams. They ignore grid border problems. Data streams come with a large number in chronological order, and makes original grids no longer adapt to new data mapping, so a large number of data is likely to fall on grids borders. But if the

data is simply discarded, it affects the clustering quality. If grids are updated in time, cost is greatly increased and the clustering efficiency is affected greatly [30].

D-Stream is a density grid-based real-time stream data clustering algorithm where data points are mapped to the corresponding grids and grids are clustered based on their density. D-Stream clustering quality depends on the granularity of the lowest grid structure level. This may reduce the clusters accuracy despite the technique processing time speed [32]. The algorithm uses an online component which maps each input data record into a grid. It also has an offline component which computes the grid density and clusters the grids based on their density. It adopts a density decaying technique to capture the dynamic changes of a data stream [17].

MR-Stream is an algorithm that can cluster data streams at multiple resolutions [33]. It partitions the data space in cells and a data structure tree which keeps the space partitioning. MR-Stream increases the clustering performance by determining the exact time to generate clusters [32]. FlockStream is a density-based clustering algorithm based on a bio-inspired model. It uses the flocking model, where independent micro-cluster agents form clusters together. FlockStream merges online and offline phases where agents form clusters at any time. It can get clustering results without performing offline clustering. DenStream, MR-Stream, D-Stream and FlockStream are density-based clustering algorithms evolving data streams. They can affectively detect arbitrary shape clusters and handle noise, but their quality decrease when using clusters with variant densities. In fact, D-Stream, MR-Stream and ExCC are grid-based clustering algorithms over data streams [32].

LOCALSEARCH algorithm uses dividing and conquering to partition data streams into segments, and discovers clustering of data streams in finite space, by using the K-means algorithm [30]. Later on, STREAM algorithm was proposed by O'Callaghan which is based on the LOCALSEARCH. It puts equal weights to outdated and recent data and cannot capture evolving characteristics of data stream [30].

Incremental DBSCAN is an incremental method for data warehouse applications. It can only handle a relatively stable environment but it can't deal with limited memory and fast changing streams. The LOCALSEARCH is a subroutine performed every time when a new chunk arrives to generate cluster centers of the chunk. HPStream introduces the concept of projected cluster to data streams. It cannot be used to discover clusters of arbitrary shapes in data streams [31].

A framework to dynamically cluster multiple evolving data streams called Clustering on Demand (COD) was proposed [34]. It produces a summary hierarchy of data statistics in the online phase, whereas clustering is performed in the offline phase [34]. It summarizes data streams using the Discrete Fourier Transform (DFT) technique. Then it applies a K-means algorithm to cluster the summarized data streams [34]. An Online Divisive-Agglomerative Clustering (ODAC) algorithm was also proposed to incrementally construct a tree-like hierarchy of clusters using a top-down strategy. The previous techniques assumes that all data streams are gathered at a centralized site before they are processed [34].

Many density-based clustering algorithms for multi density datasets are not suitable for data stream environments. First, they need two-pass of data and this condition is impossible for data streams where they arrive continuously and need to be performed

in a single scan. GMDBSCAN and ISDBSCAN use a two-pass data. Second, some multi density clustering algorithms need the whole data. Third, other algorithms have a high execution time which makes them not applicable for data streams. DSCLU is a density-based clustering for data stream in multi density environments [32].

A DD-Stream, is framework for density-based clustering stream data. It adopts a density decaying technique to capture the evolving data stream and extracts the boundary points of grids by using a DCQ-means algorithm. It is used to resolve the problem of evolving automatic clustering of real-time data streams, it can find arbitrary shaped clusters with noise and also avoid the clustering quality problems caused by discarding the boundary points of grids. The DD-Stream has better scalability in processing large-scale and high dimensional stream data as well [30].

2.6.3 Clustered WSNs involving Data Streams

This section presents the third classification of clustering algorithms. It is divided into two categories: algorithms based on FCM algorithm and algorithms for multimedia streaming data. The developed DeGiCA belongs to research study scope of algorithms presented in this section.

Fuzzy C-Means or Fuzzy Clustering Means (FCM), is the most widely used algorithm in the field of data mining clustering technique when involving data streams in WSNs. Most clustering algorithms are descendant from FCMs when solving data stream problems in WSNs. FCM requires prior information of how many clusters C to partition the data space into. The number of clusters C within the WSN datasets is

unknown previously [35]. Figure 2.6 shows some algorithms based on K-means and Fuzzy C-Means.

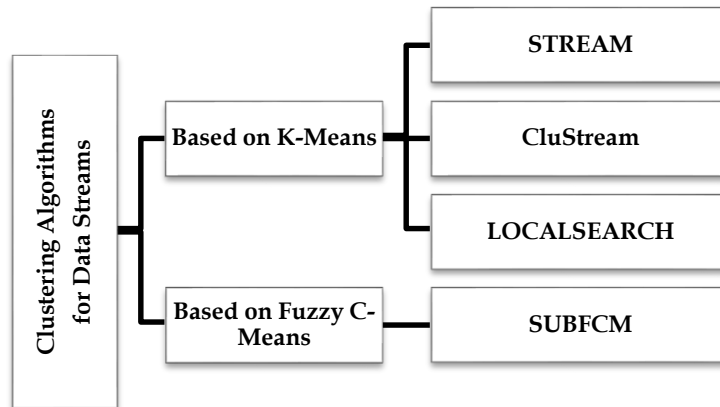


Figure 2.6 Clustering Algorithms for Data Streams

An algorithm based on FCM, a distributed WSN data stream clustering algorithm called SUBFCM (Subtractive Fuzzy Cluster Means) is proposed to minimize sensor nodes energy consumption and extend the network lifetime in WSNs involving data streams. The SUBFCM focuses on data stream clustering problem. Simulations show that the energy efficient algorithm SUBFCM can achieve WSN data stream clustering with significantly less energy than the FCM and K-means algorithms do. SUBFCM reduces the total data transmission required without significantly affecting vital information in data streams [35]. SUBFCM is a result of blending a subtractive clustering algorithm with the FCM.

For algorithms based on multimedia streaming data, in Wireless Multimedia Sensor Networks (WMSNs), multimedia clustering protocols use the quality of service (QoS) parameters [36]. The requirements of QoS differ according to different types of multimedia applications. QoS has several metrics such as delay, bandwidth, reliability, jitter [37] and packet loss [36]. Many multimedia applications are time critical, they need to be reported with a limited time. The multimedia sensors have the ability to

capture video, image, audio and scalar sensor data. Then deliver the multimedia content through sensors network [37]. A clustering algorithm for wireless multimedia sensor networks has been proposed based on Overlapped Field of View (FoV) areas. This algorithm aims to find the intersection polygon and computing the overlapped areas to establish clusters and determine cluster membership. FoVs prolongs network lifetime and saves energy [29].

2.6.3.1 Algorithms Based on FCM for Streaming Data in WSNs

This section provides an overview on FCM clustering algorithm and algorithms based on it. It includes the reasons behind choosing the standard Fuzzy Clustering Means algorithm (FCM) used in clustered WSN environment holding data streams. Generally, clustering algorithms in WSNs involving data streams are classified into distributed, centralized and hybrid clustering algorithms [38]. In distributed clustering, any node can choose itself as a CH or join an already existed cluster on its own initiative, independent of other nodes. Distributed clustering techniques are classified into four sub types based on cluster formation criteria and parameters used for CH election. Sub-types are based on either identity, neighborhood information, probabilistic or iterative [38]. In centralized techniques, global network information are required to provide BSs the abilities to control the whole network. CHs are selected by the BS in this approach. Hybrid techniques are a combination of centralized and distributed approaches. In a hybrid environment, distributed schemes are responsible for coordinating between CHs, while centralized schemes are used to build individual clusters [38]. Based on the previous network topologies, clustering algorithms must include a technique to compute similarities or distance between vectors. Distance is

considered to be the most natural method for numerical data similarity measurement. Lower values indicate more similarities. The most common distance metrics are the Euclidean distance and Manhattan distance. However, the distance metric does not work properly with non-numerical data. Fuzzy C-Means (FCM) and K-means use the Euclidean distance [39] and considered to be the most suitable algorithm for WSNs involving streaming data due to its soft clustering nature. Figure 2.7, shows some FCM-based algorithms.

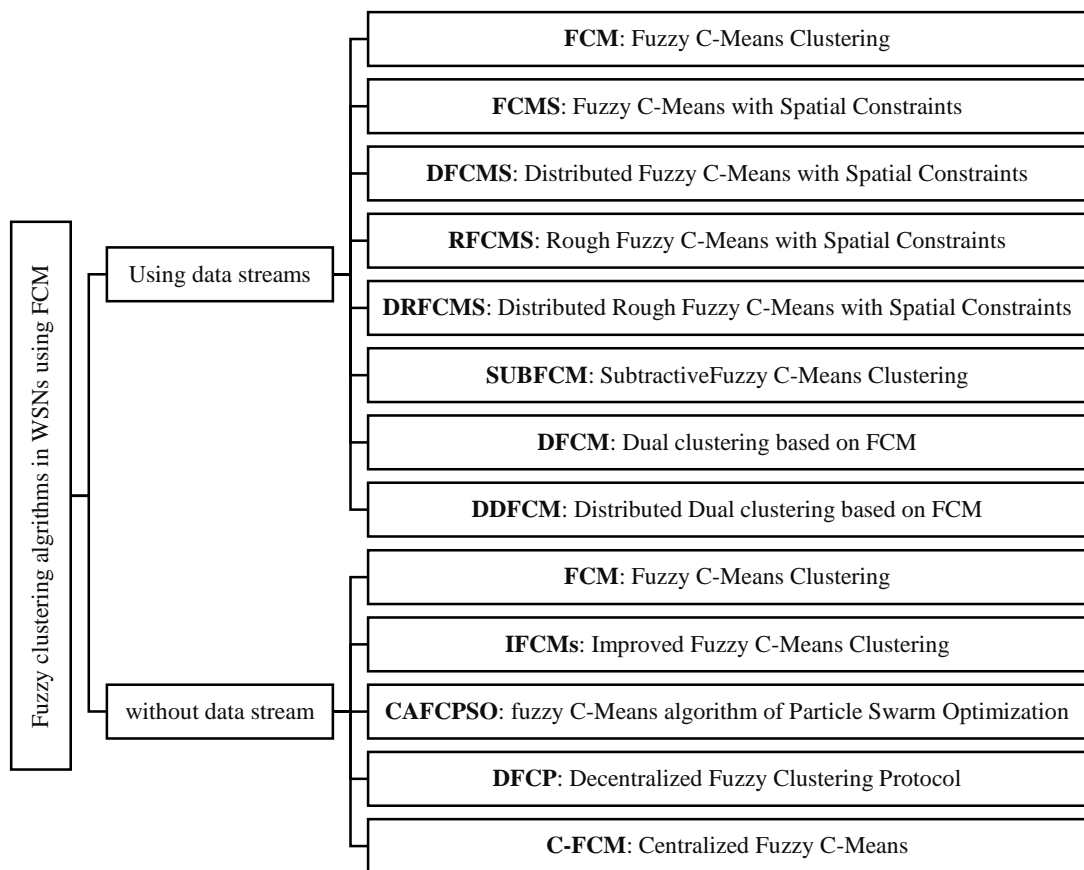


Figure 2.7 Classification of FCM-Based Clustering Algorithms in WSNs

Most algorithms listed in figure 2.7 are FCM-based clustering algorithms. They introduce the subtractive clustering algorithm (SUBFCM) to predetermine number of clusters and their cluster centers [Appendix A.2.1]. As known, FCM requires a pre-

knowledge of number of clusters and each cluster center, since the output rules depend strongly on the initial values [40, 41]. Cluster formation at the subtractive clustering assumes each node is a potential cluster center. Then a calculation is done to measure possibilities that each node would define a cluster center.

2.6.4 Taxonomy of Clustering Protocols

As mentioned previously, clustering protocols are classified in our research to three main types. First, clustering algorithms in WSNs (without data streams). Second, clustering protocols for streaming data [42]. Third, clustered WSNs for data streams. Figure 2.8 shows the clustering algorithm classification with examples of each class.

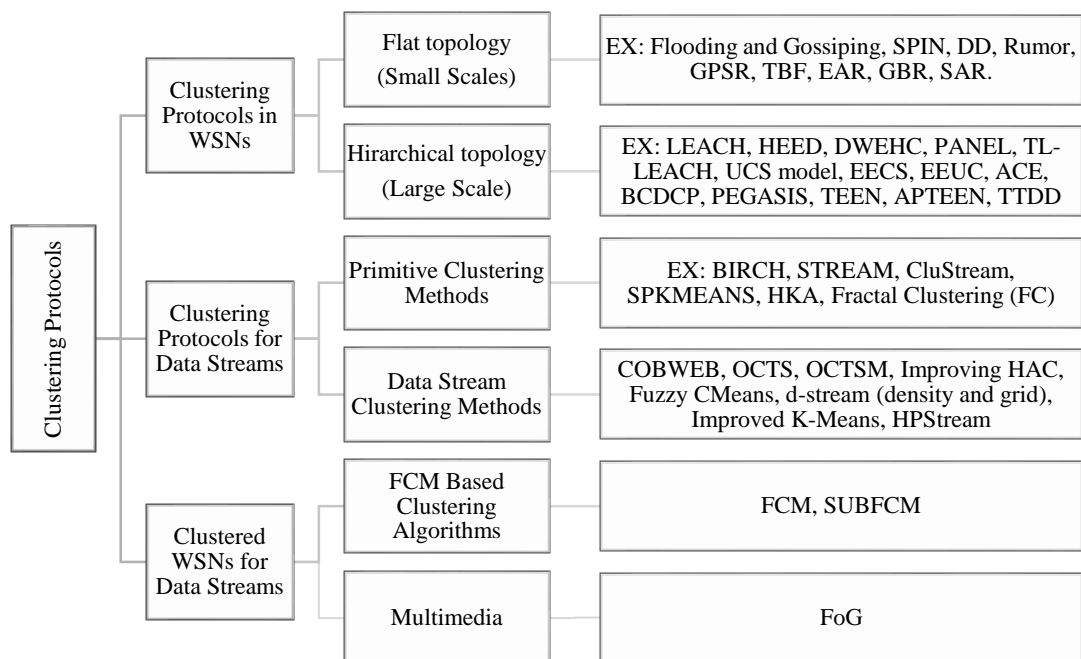


Figure 2.8 Classification of Clustering Algorithms in WSNs

Moreover, table 2.3 provide several clustering algorithms according to the classification of clustering algorithms.

TABLE 2.3 Classification of Clustering Algorithms

Clustering Classification	Classes	Examples
Clustered WSNs for Data Stream	WSNs	Subtractive Fuzzy Cluster Means (SUBFCM)
	WMSNs	Overlapped Field of View (FoVs)
Data Stream Clustering Algorithms	Primitive Clustering Methods	BIRCH
		STREAM
		CluStream
		SPKMEANS
		HKA
		Fractal Clustering (FC)
	Data Stream Clustering Methods	COBWEB
		OCTS
		OCTSM
		Improving HAC
		Fuzzy C-Means
		d-stream (density and grid)
		Improved K-Means
		HPStream
	Density-Based Clustering Algorithms	DenStream
		DSCCLU
		FlockStream
		CluStream
	Density-Grid Based Clustering Algorithms	D-Stream
		DD-Stream
		MR-Stream
Grid Based Clustering Algorithms	Excc	
Other	Distributed Single-pass Incremental Clustering (DISC)	
	Online Divisive-Agglomerative Clustering (ODAC)	
	Clustering on Demand (COD)	
Clustering Algorithms in WSNs	LEACH	Low-energy Adaptive Clustering Hierarchy
	TL-LEACH	Two-Level Hierarchy LEACH
	EECS	Energy Efficient Clustering Scheme
	HEED	Hybrid Energy Efficient Distributed Clustering
	EEUC	Energy-efficient unequal clustering
	EEHC	Energy Efficient Hierarchical Clustering
	MRPUC	Multihop routing protocol with unequal clustering
	PEACH	Power-efficient and adaptive clustering hierarchy
	HSRP	Hybrid-Structure Routing Protocol
	TTDD	Two-Tier Data Dissemination mechanism
	SEAD	Scalable Energy efficient Asynchronous Dissemination mechanism
	HSRP	Hybrid-Structure Routing Protocol
	CODE	Coordination-based data Dissemination mechanism
	TEEN	Threshold sensitive Energy Efficient sensor Network protocol
	APTEEN	Adaptive Periodic TEEN
	DMAC	Distributed and Mobility-Adaptive Clustering Algorithm
	DISC	Distributed Single-pass Incremental Clustering

TABLE 2.3 Classification of Clustering Algorithms (Cont.)

CACC	Clustering Algorithm based on Cell Combination
VAP-E	Energy-Efficient Clustering -Virtual Area Partition
CFL	Clustering for Localization
FoVs	Overlapped Field of View
KOCA	K-Hop Overlapping Clustering Algorithm
PEZCA	Power-Efficient Zoning Clustering Algorithm
VoGC	Voting-on-Grid clustering
BARC	Battery Aware Reliable Clustering
SEP	Stable Election Protocol
DEEC	Distributed Energy-Efficient Clustering algorithm
LCA	Linked Cluster Algorithm
LEACHC	Centralized-Low energy adaptive clustering hierarchy
LCA2	Linked Cluster Algorithm 2
WCA	Weighted Clustering Algorithm
PEGASIS	Power-Efficient GATHERing in Sensor Information Systems
GAF	Geographic Adaptive Fidelity protocol [43]
HAS	Harmony Search Algorithms
DCA	Distributed Clustering Algorithm
PDCH	Pegasis Algorithm Improving Based on Double Cluster Head
DWEHC	Distributed Weight-Based Energy-Efficient Hierarchical Clustering
FLOC	Fast Local Clustering service
ACE	Algorithm for Cluster Establishment
PSO-C	Centralized-PSO

2.7 Conclusion

The past few years have witnessed increased interest in the use of WSNs in a wide range of applications and it has become a hot research area in the field of data mining. This chapter focused on the most important concepts of WSNs, data stream mining, and data streams and clustering algorithms. It provided an overview comparison between clustering algorithms in WSNs. A classification for clustering algorithms was given and based on the research study background. They were classified into three main types. Clustering protocols in WSNs (without data streams), clustering protocols for data streams and clustered WSNs for streaming data. The chapter focused in details on WSNs clustering algorithms involving streaming data that are called FCM-based clustering algorithms that are similar to the research study scope of the developed DeGiCA.

Chapter III

Density Grid-Based Clustering Algorithm

Chapter III

Density Grid-Based Clustering Algorithm

3.1 Introduction

WSNs generate massive data streams with spatial and sensor measurements information [44]. A WSN consists of a powerful BS that serves as sensed streaming data final destination. Passing sensory streaming data to the BS requires energy. As mentioned previously, a WSN suffers from some constraints such as energy, memory and computational capabilities. One challenge is power consumption during data stream transmission. Sensor nodes should be energy efficient. Energy efficiency affects the entire WSN lifetime. Therefore, in order to ensure the WSN's operational longevity, energy consumption is a critical consideration during designing WSNs algorithms. Furthermore, since sensor nodes are in difficult-to-reach locations, replacing batteries is impractical [45].

A WSNs can benefit a great deal from stream mining clustering algorithms in terms of energy saving. However, to achieve better energy conservation, data stream mining has to be performed in a distributed manner, due to their resource constraints [39].

Clustering algorithms are designed to achieve load-distribution among CHs, energy saving, high connectivity, and fault tolerance. In WSNs, clustering provides resource utilization and it minimizes energy consumption by reducing the number of nodes that take part in long distance transmission [44]. Clustered WSNs algorithms running streaming data are usually partitioned in two main steps, cluster formation step and data transmission step [46]. But specifically, the cluster based operation of clustered WSN algorithms consist of rounds. Rounds involve cluster formation, CH selection, and data transmission to the BS.

Grid-based clustered WSNs, are type of networks where a sensed area is partitioned into a finite number of equal sized cells called grids. Grid-based clustering scheme has been proven to have a fast processing time compared to other types of clustering algorithm schemes due to clustering operations are performed on the grid cells instead of the whole dataset stream [9].

This chapter proposes a grid-based clustering algorithm for WSNs model called Density Grid-based Clustering Algorithm (DeGiCA), a distributed clustering algorithm proposed due to its suitability for the suggested environment. The DeGiCA is a clustering algorithm based on combining a density technique and a grid technique. Beside the advantages of the clustering technique listed above, the density technique can find arbitrary shaped clusters with noise while the grid technique is used to avoid clustering quality problems by discarding the boundary points of grids. This powerful combination of techniques decrease the algorithm computational time, reduce energy consumption and thus extend the network lifetime resulting the desirable simulation results. To reach the aim of this research, DeGiCA must combine the limited dataset streams as fast as possible, to ensure that a processor can take on next set of streams.

This chapter presents the DeGiCA proposed model. It is organized as follows: section 2 presents an overview on the DeGiCA. Section 3 presents the general DeGiCA algorithm. In section 4, the conclusion of this chapter is given.

3.2 Overview on Density Grid-Based Clustering Algorithm (DeGiCA)

The proposed Density Grid-based Clustering Algorithm (DeGiCA) is a clustering algorithm that forms clusters based on the density of each grid in a gridded WSN. The proposed scheme is done by dividing a sensor network area into equal size of grids. Grids then are classified into three main density classes, high density grids, low density grids and empty grids. The density classification for each grid is done by comparing its nodes number with a specific value σ called threshold. By using the DeGiCA, grids close to each other are combined after finding their density to form clusters. Empty grids are used as delimiters to reduce the algorithm execution time.

In general, the DeGiCA goes through three main phases as shown in figure 3.1. First, the establishment phase that creates the gridded sensor network. It classifies each grid based on its density and combine close grids to form an arbitrary shaped cluster. After this phase, DeGiCA selects initially a CH for each cluster based on the nearest distance to the BS. The establishment phase is done once. The second phase is called the data transmission phase, it is responsible for transmitting sensed streaming data from source nodes to the final destination at the BS through CHs. CHs aggregate sensed streaming data and remove redundancy. The third phase is the CH-Election phase, it selects new CHs based on nodes residual energy. Data transmission phase and CH-Election phase are repeated in rounds until the end of network lifetime.

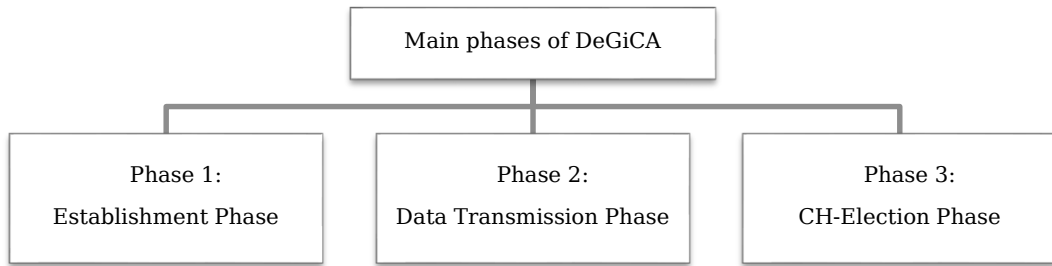


Figure 3.1 Three Main Phases of DeGiCA Algorithm

Figure 3.2 presents the general flowchart of the proposed DeGiCA. As shown in the flowchart, DeGiCA starts its first process called the initialization process which includes firstly the establishment phase. At the establishment phase, a sensed area with $(M \times M)$ square unit is created permanently and the BS is located at the center of the sensed area. The sensed area is divided into equal size of grids G with n sensor nodes are randomly scattered in the gridded WSN. Clusters are formed based on the density of grids. When the establishment phase ends, the initial process selects CHs based on nodes having the nearest distance to the BS. The initial process is done only once during the WSN lifetime.

After this process, the DeGiCA moves to the rounds process and the sensed area is ready to flow streaming data. At the rounds process, the data transmission phase and the CH-Election phases are rotated until the network lifetime ends. For each round, CHs are elected inside their clusters based on nodes having the highest residual energy among cluster nodes. After each round completion, processed streaming data packets are sent to the BS.

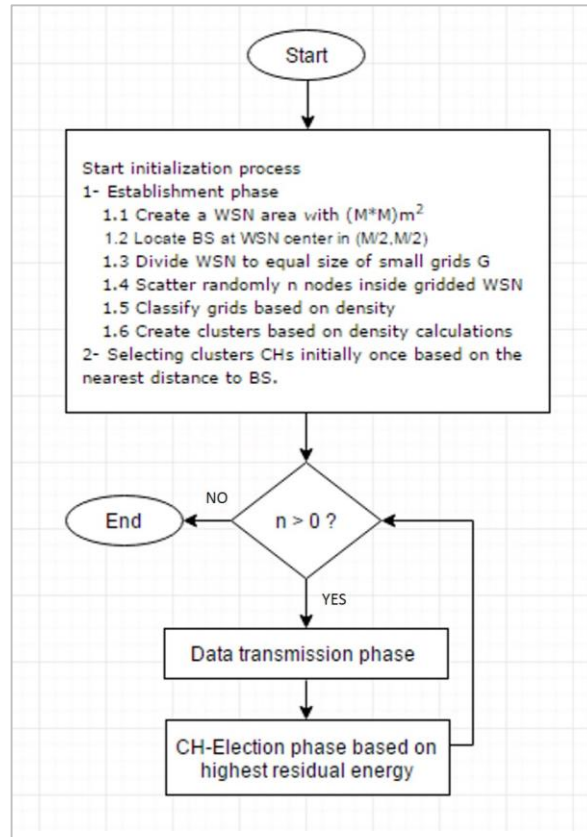


Figure 3.2 Proposed DeGiCA General Flowchart

3.2.1 General DeGiCA Assumptions

For the sake of clarity, some assumptions are made about the proposed DeGiCA network model. These assumptions are listed as follows:

- 1- Network area size or sensed area is equal to $(M \times M)$ square unit (i.e. m^2). The area exist between coordinates $(0,0)$ to $(1000,1000)$.
- 2- Network is created at the establishment phase once during the whole network lifetime at the beginning of DeGiCA.
- 3- The sensed area is gridded once establishing the network outside rounds process.
 - 1- Grids G has number of grids G_{num} is equal to $(\frac{M}{g})^2$, where $g \in X$, and $g \in Y$.

- 2- Number of nodes in the network is n .
- 3- Nodes are assumed to be immobile, homogenous and energy constrained.
- 4- BS is immobile and located in the middle of the sensed area at $(\frac{M}{2}, \frac{M}{2})$. It is assumed to be a central BS.
- 5- All nodes have the same initial residual energy measured in joules.
- 6- Data stream packets are formed in a standard research purpose form as shown later in figure 4.1.

3.2.2 Algorithm Phases Description

DeGiCA phases are partitioned to several stages with different techniques for each stage. Figure 3.3 shows the techniques used for each phase in the proposed DeGiCA.

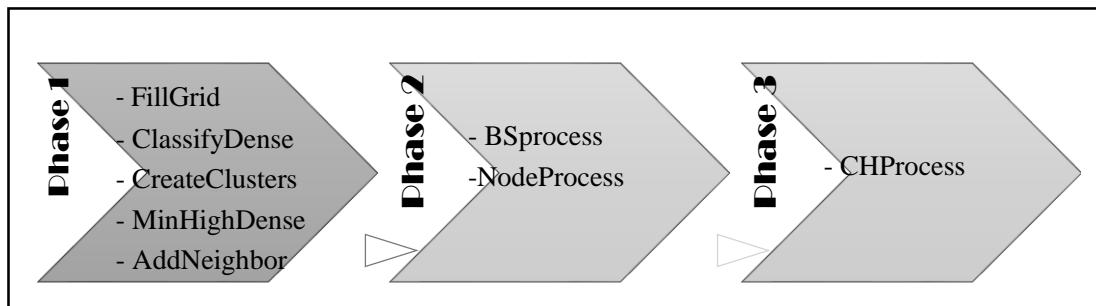


Figure 3.3 Techniques used in DeGiCA phases

In brief, DeGiCA is divided into two main processes, initialization process that is done once at the beginning of a WSN lifetime and rounds process that is repeated rotationally until the end of a WSN lifetime.

The initialization process has two main steps. Firstly, the establishment phased and secondly, CH initial election based on the nearest distance to the BS. The establishment phase has many techniques used during sensed area creation. First of all,

a WSN area is created permanently with a size equal to $(M * M) \text{ m}^2$ and a BS positioned at the center $(\frac{M}{2}, \frac{M}{2})$. The area is divided into small equal size of grids and n nodes are scattered randomly inside grids. This step is applied by a **FillGrid** method. To form the network clusters, grids are classified based on their density into either a high dense grid (H), or a low dense grid (L), or an empty grid cell (E). Grid density classification is done by using a method called **ClassifyDense**. It classifies grids by comparing number of nodes in a specific grid with a threshold σ . Additionally, any empty grid cell on network boarder or within sensed area are discarded. After classifying all grids in the network, clusters are created by using three methods, **CreateClusters**, **MinHighDense** and **AddNeighbor**. After completing the establishment phase, the initialization process selects initially a CH for each cluster based on the nearest distance to the BS. It also determines centers \mathbf{v} of each cluster, to be used in FCM and K-means cluster formation (i.e. for comparison purposes). The network after initialization process is ready to receive and process streaming data packets.

The second main part of DeGiCA is the rounds process. It is repeated rotationally between the data transmission phase and the CH-Election phase. Data transmission phase is responsible for transmitting sensed streaming data from source nodes to the final destination at the BS through CHs. This is done by methods called **BSprocess** and **NodeProcess**. The third phase is the CH-Election phase. It is done by **CHProcess**, where it selects CHs based on nodes having highest residual energy among cluster nodes. It also makes CHs aggregate sensed streaming data and remove redundancy. This discussion is provided in more details on section 3.3.

Figure 3.4 shows orderly steps used to build the proposed DeGiCA algorithm. It shows in brief techniques used during the WSN lifetime.

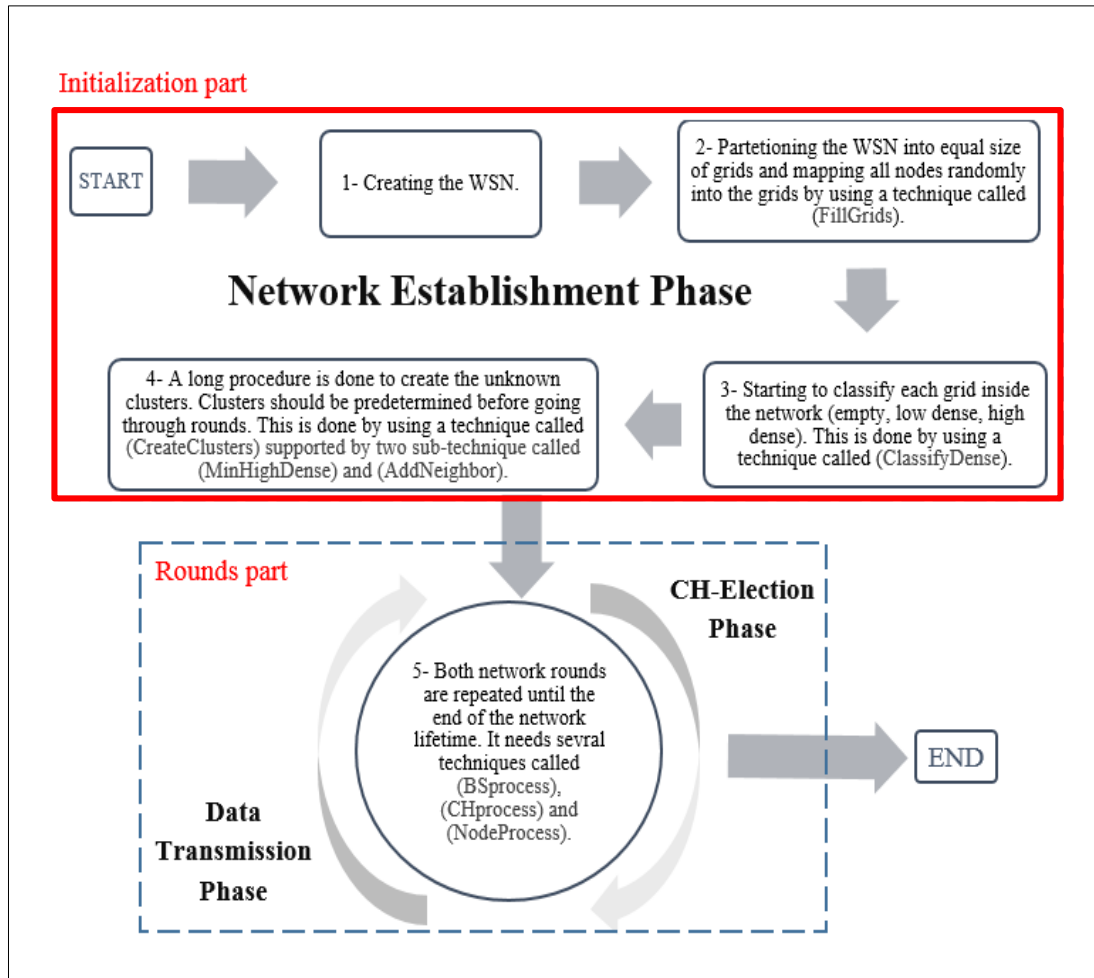


Figure 3.4 Overview on DeGiCA Algorithm Steps

To completely represent DeGiCA more, figure 3.5 shows DeGiCA phases and methods. It has two main parts and each part has its phases. Phases are further partitioned to stages and each stage uses its own techniques.

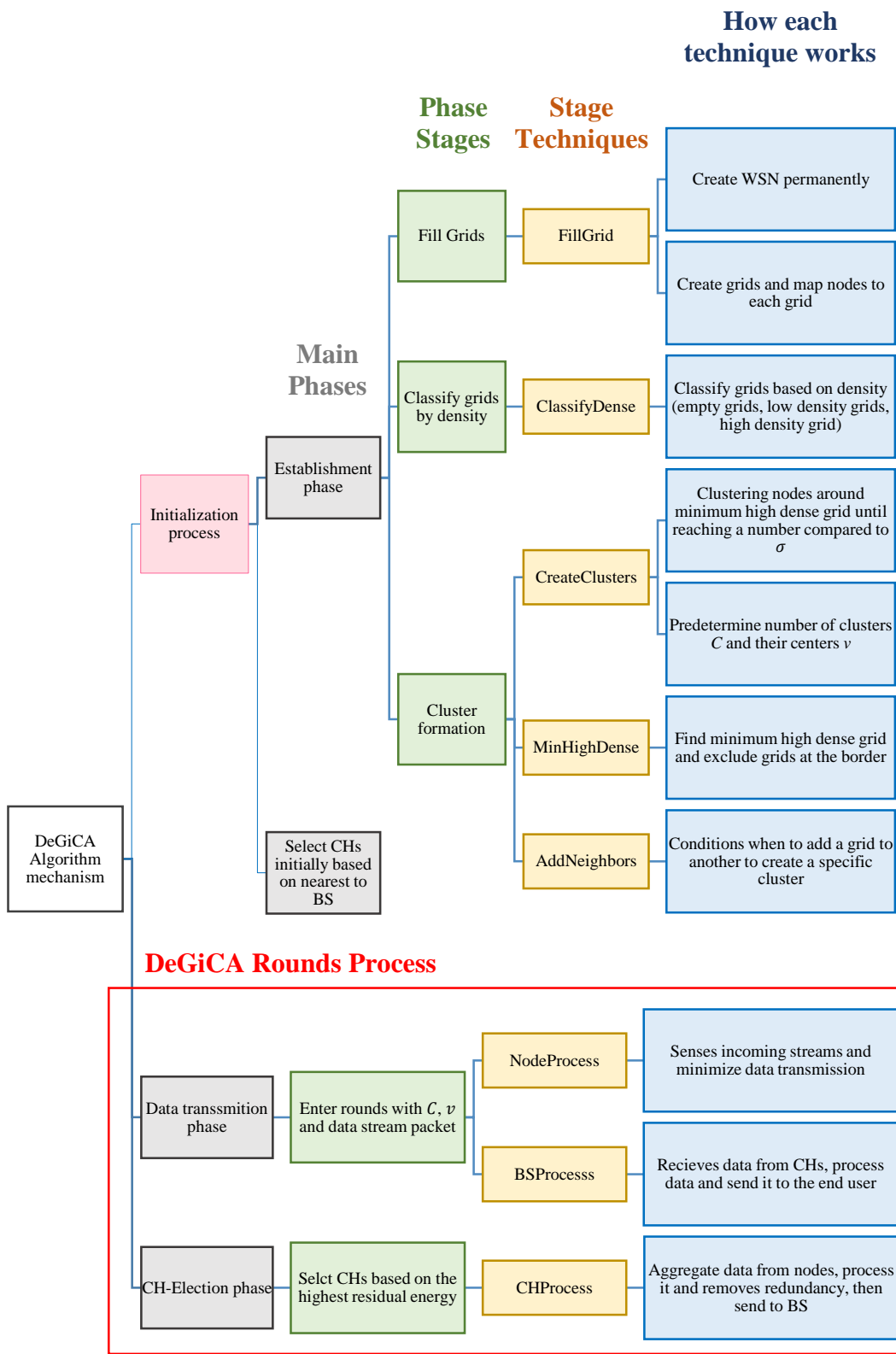


Figure 3.5 DeGiCA Algorithm Mechanism

3.3 DeGiCA Detailed Algorithm

The proposed DeGiCA is a clustering algorithm that enhances WSNs mining streaming data. It enhances the traditional Fuzzy C-Means (FCM) algorithm [Appendix A.1.2], by solving its two problems:

- 1- FCM requires two input data that have to be known a priori and predetermined specifically. These inputs are (number of clusters C , and center of each cluster ν).
- 2- Finding some nodes belong to more than one cluster. So, it is important to disjoint those points.

3.3.1 Initialization Process of DeGiCA

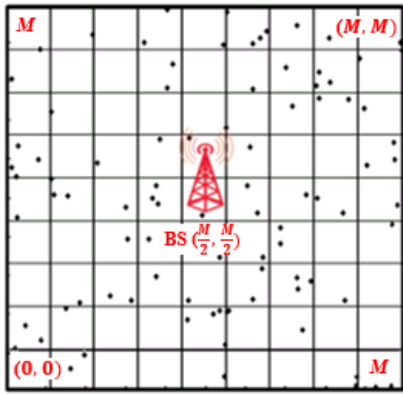
The initialization process is the most important part that distinguishes DeGiCA from other WSNs clustering data stream mining algorithms. It enhances WSNs data stream mining by a powerful combination of clustering, grid and density techniques. The initialization process has two steps. First, the establishment phase. Second, the initial CHs selection of each cluster based on the nearest distance to the BS before moving to the rounds process.

3.3.1.1 Establishment Phase of DeGiCA

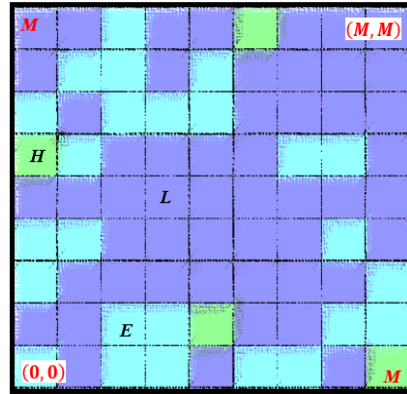
The establishment phase goes through three sequential processing steps: *setup process*, *gridding network process* and *cluster formation process*. At the *setup process*, a WSN experimental area is created permanently with size equal to $(M * M) m^2$ and a BS positioned at the center of the sensed area $(\frac{M}{2}, \frac{M}{2})$. During *gridding network process*,

the experimental sensed area is divided to small equal size of grid G , and n nodes are scattered randomly inside grids, all nodes have equal initial energy, see figure 3.6 (a). Grids are classified based on their density into either a high dense grid (H), or a low dense grid (L), or an empty grid cell (E), figure 3.6 (b). Each grid is classified by calculating its nodes number and comparing it with a threshold σ . Experimental sensed area borders are eliminated and empty grids are discarded.

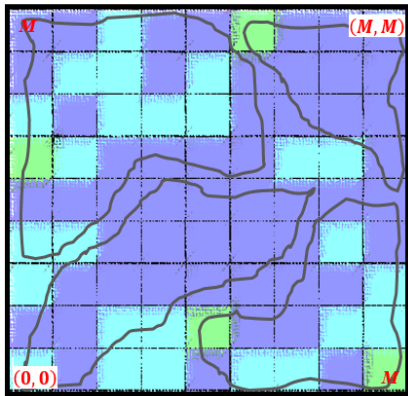
At the *cluster formation process*, figure 3.6 (c) and 3.6 (d), close grids are combined with each other to form one specific cluster by assuming that any two adjacent high dense grids are joined in a cluster, any two adjacent high dense grid and low dense grid are also joined in a cluster, and any two adjacent low dense grids are being an outlier for a cluster as discussed lately. Figure 3.6 (e), presents a final DeGiCA WSN structure.



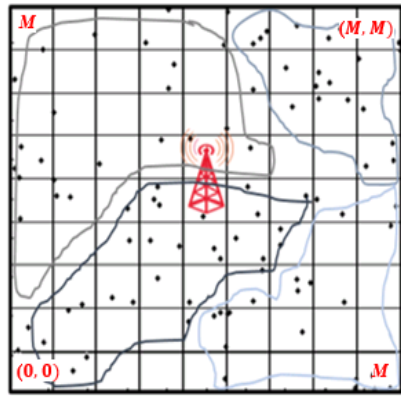
(a) Gridded WSN with a Single BS



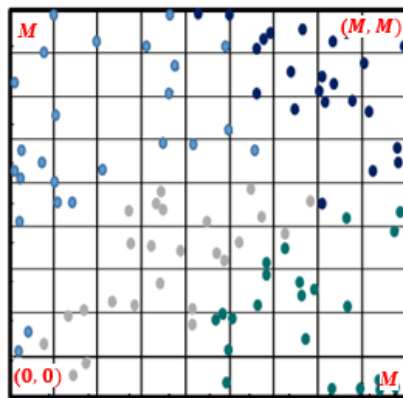
(b) Classifying Grids Based on Density



(c) Cluster Formation Process



(d) Creating Clusters



(e) Final DeGiCA Structure

Figure 3.6 Proposed DeGiCA Model Structure

In step representation, the following subsections provide more details about the DeGiCA processes:

Setup Process

Step 1. As assumed at section 3.2.1, a WSN experimental area is created permanently with size equal to $(M * M) \text{ m}^2$ and a BS positioned at the center of a sensed area $(\frac{M}{2}, \frac{M}{2})$.

Gridding Network Process

Step 2. Partitioning a network sensed area permanently once during the whole network lifetime into equal size of cells called grids \mathbf{G} to find later their effect on the proposed DeGiCA. After that, a number of n nodes are randomly mapped inside grids. This step is done by using a method called **FillGrid** algorithm shown in figure 3.7.

```

1. Input: static data
2. Output: Grid matrix with data points
3. Gridx=ceil(Node x dimension /width)
4. Gridy=ceil(Node y dimension /length)
5. Grid Counter(grid x dimension ,grid y dimension )= Grid Counter (grid x
dimension, grid y dimension )+ 1

```

Figure 3.7 Fill Grid Algorithm

Step 3. Finding number of clusters \mathbf{C} and their centers \mathbf{v} by applying a special density grid-based technique. It is briefly obtained by following the steps:

- a) Empty grid (E) is used as outlier. Each non-empty grid in the network is classified to either high dense grid (H), low dense grid (L). Grid classification procedure is done by comparing the number of nodes in each grid with a threshold σ . The threshold is obtained from calculating a Standard Deviation \mathbf{SD} of network nodes number n . The threshold is calculated by equation (3.1).

$$\sigma = SD \times 2 \quad (3.1)$$

$$\text{Where } SD = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}}$$

and n is the total number of nodes in a WSN is, x is a single node in the network, and \bar{x} is the mean of network nodes.

This step can be applied using the **ClassifyDense** algorithm presented in figure 3.8.

```

1. Input: Grid Matrix
2. Output: Grid type Matrix
3. Standard Deviation = Standard Deviation Function (Grid matrix);
4. threshold= Standard Deviation *2;
for every grid in the network field
6. if (there is no node in the Grid)
7. Grid type is 'empty';
8. else if (the number of nodes in the grid is less than threshold)
9. Grid type is 'Low';
10. else if (the number of nodes in the grid is more than or equal than
threshold)
11. Grid type is 'High';
12. endif
13. endfor

```

Figure 3.8 Classify Dense Algorithm

Cluster Formation Process

b) Creating clusters, where nodes are clustered around a *minimum high dense grid* until a certain number of nodes is reached to the cluster threshold σ , this is done by using a **CreateClusters** algorithm shown in figure 3.9.

```

Input: Grids in the field
Output: Clusters
%determine first high
2. for every grid not in the border of the network field
3. if (the grid type is dense)
4. number of High grid incremented ;
5. minimum = hold the x and y dimensions for the minimum high dense grid;
6. endif
7. endfor
8. minimum high dense grid x and y dimensions = findMinHigh(ClusterMatrix,width,length);
9. %Cluster around the minimum high
10. for (every High dense grids in center sub matrix)
11. if (Neighbour not in cluster && Neighbour Type is "High " && Cluster value less than or equal Cluster threshold)
Add Neighbour to cluster;
go to the next Neighbour( Grid);
elseif(Neighbour not in cluster && Neighbour Type is "Low" && Cluster value less than or equal Cluster threshold)
Add Neighbour to cluster;
go to the next Neighbour(Grid);
endif
endif
endif

```

Figure 3.9 Create Clusters Algorithm

In more details, figure 3.10 shows how to create clusters based on their grids density. Nodes are clustered around a selected *minimum high dense* grid as shown in figure 3.11 until reaching a specific number of nodes as cluster threshold. Assuming that two adjacent high dense grids (H) are joined in a cluster, two adjacent high dense grid (H) and low dense grid (L) are also joined in a cluster, and two adjacent low dense grids (L) are being an outlier for a cluster.

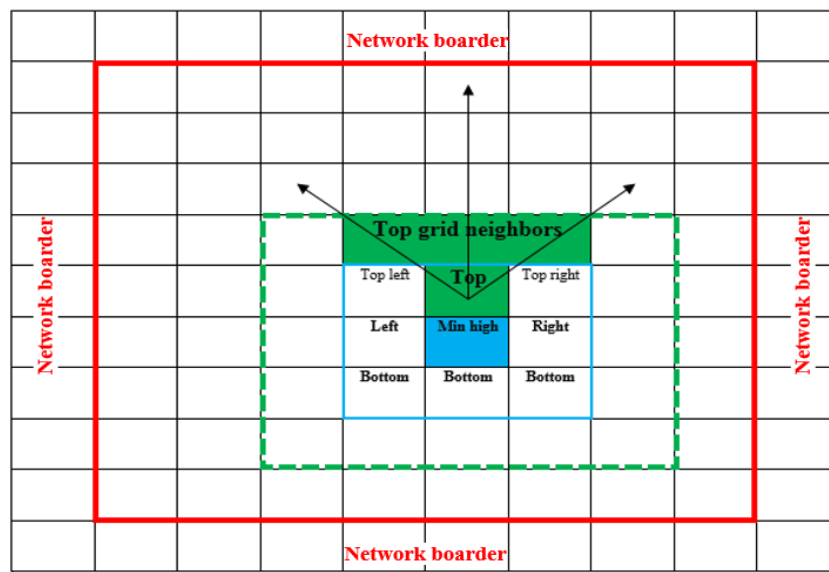


Figure 3.10 Cluster Formation Process

```

1. Input: Cluster Matrix , width , high
2. Output: minimum high dense grid x and y dimensions
3. for (every grid in center sub matrix)
4. if (Grid type is 'High' && it is not in a cluster && grid nodes< minimum )
5. minimum = The Grid x and y dimension ;
6. endif
7. endfor

```

Figure 3.11 Minimum High Dense Algorithm

For every next minimum high dense grid, if it is not included in any cluster before, the algorithm **AddNeighbors** shown in figure 3.12 is used to find its cluster.

```

Input: Cluster Matrix
Output: Add Grid to Cluster
1. % add Neighbour of the Neighbour function
2. go to the next Neighbour ( Grid) {
3. while end not reached
4. if (Neighbour not in cluster &&Neighbour Type is "High " &&next
Neighbour Type is "High" && Cluster value less than or equal Cluster
threshold)
5. Add Neighbour to cluster ;
6. go to the next Neighbour(Grid);
7. elseif(Neighbour not in cluster &&Neighbour Type is "High " && next
Neighbour Type is "low" &&Cluster value less than or equal Cluster
threshold)
8. Add Neighbour to cluster;
9. go to the next Neighbour(Grid);
10. elseif (Neighbour not in cluster && Neighbour Type is "Low "
&&next Neighbour Type is "High" && Cluster value less than or
equal Cluster threshold)
11. Add Neighbour r to cluster
12. go to the next Neighbour(Grid);
13. elseif(Neighbour not in cluster &&Neighbour Type is "Low "
&& next Neighbour Type is "low" &&Cluster value less than or
equal Cluster threshold)
14. Add Neighbour to cluster;
15. break;
16. endif
17. endwhile
18. } end of go to the next Neighbour function

```

Figure 3.12 Add Neighbors Algorithm

- c) After forming all clusters, CHs in each cluster are initially selected based on the nearest distance to the BS.
- d) Finding the center of each cluster ν , for comparison purpose.

Step 4. Extracting nodes found on network boundaries, figure 3.10, and eliminating noise (i.e. empty grids).

DeGiCA is now ready to run data stream packets and move to the rounds process. Using number of clusters C and their centers ν obtained from the previous steps are used for comparison purpose. Figure 3.13 presents the initialization process pseudo code of DeGiCA. The complete process of DeGiCA initialization is provided in a flowchart shown in figure 3.14.

Pseudo code of DeGiCA Initialization Process

Inputs: n, G, M, σ

Outputs: C, v

Initialize $n = 100, G_{num} = 0, M = 1000, \sigma = 0$.
Create WSN permanently with $M * M$ square unit.

1. Locate BS at center $(M/2, M/2)$.
2. Create grids G .
3. Map nodes n into $G_j \forall j > 0$.
4. Put $i = n$
5. While ($i > 0$) {
6. Calculate $SD = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}} \forall n \in \{1, 2, \dots, 100\}; i = i - 1;$ end loop
7. $\sigma = SD * 2$.
8. Calculate $j = G_{num}, \forall G \in (0, 0) - (M, M)$.
9. While ($j > 0$) \ \ \ Classify grids based on density (empty, low, high).
10. if (no nodes in G_j), then $G_j =$ "empty";
11. else if (number of nodes in $G_j < \sigma$), then $G_j =$ "low".
12. else if (number of nodes in $G_j \geq \sigma$), then $G_j =$ "high".
13. $j = j + 1$ }. \ \ \ End while loop.
14. Find minimum high dense grid
15. Exclude grids at network border.
16. Clustering nodes around minimum high dense grid until reaching a number compared to σ .
17. Examining eight neighbors of minimum high grid.
 - a. If (neighbor grid "high" || "low") && (clustered previously)! && (σ not reached), then G is included.
 - b. Repeat (a) until satisfying the following conditions:
 - i. if (neighbor is "high") && any of its adjacent neighbors is "high", then both are included in the cluster.
 - ii. else if (neighbor is "high" && any of its adjacent neighbors is "low"), then both are included in the cluster.
 - iii. else if (neighbor is "low" && its neighbor is "high"), then both are included in the cluster.
 - iv. else if (neighbor is "low" && its neighbor is "low"), then both are included and this forms cluster boarder.
18. Repeat (a) and (b) for every next minimum high dense grid if it is not included in any cluster before.
19. Calculate $j =$ number of $G, \forall G \in (0, 0) - (M, M)$.
20. For ($j = 0; j > 0; j + +$)
21. {Select CH_{BS} of each cluster based on nearest to the BS.}
22. Output C and v .

Figure 3.13 Initialization Process Pseudo Code

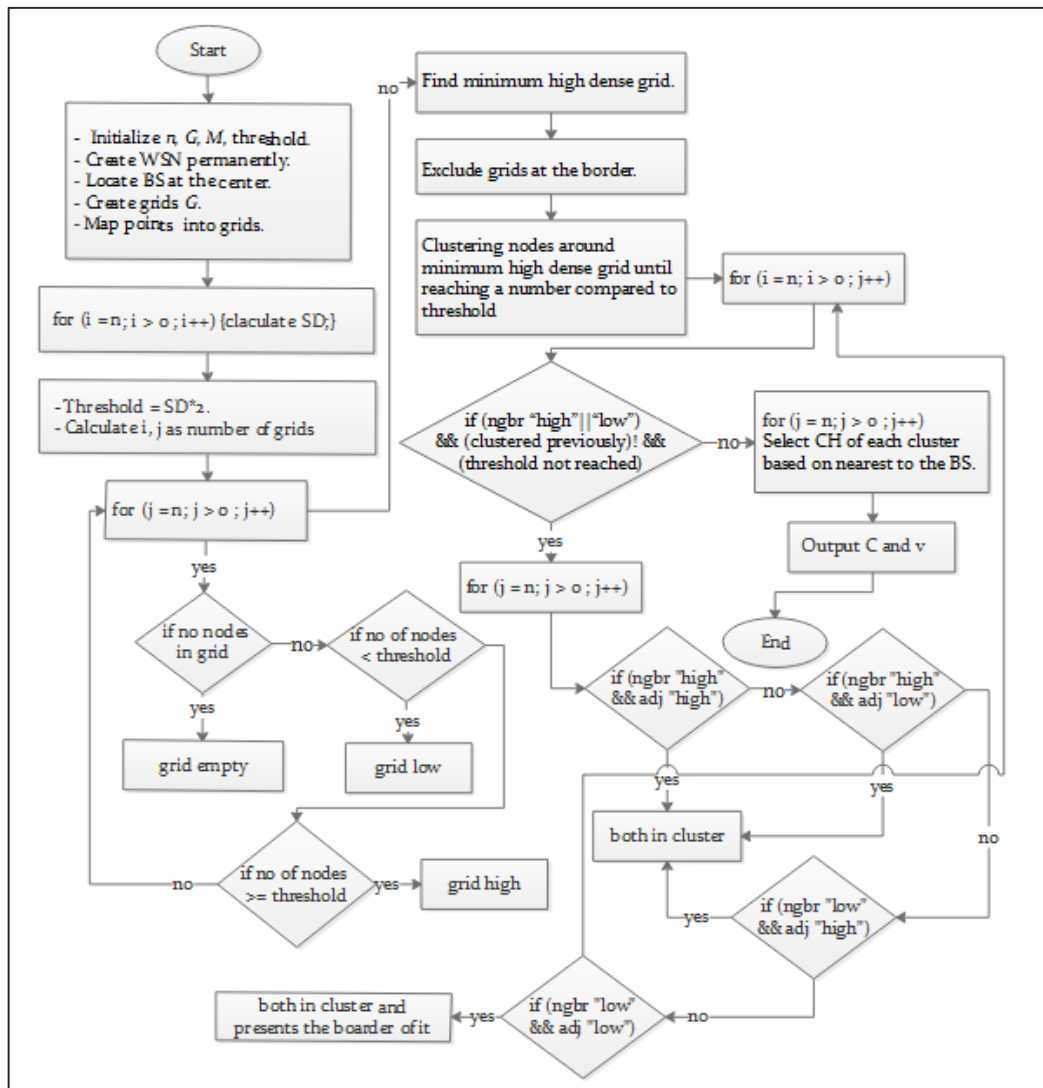


Figure 3.14 Initialization Process of DeGiCA Flowchart

3.3.2 Rounds Process of DeGiCA

After establishing a WSN infrastructure as mentioned in section 3.3.1, DeGiCA is ready to flow streaming data packets and begin network rounds process. Each round is divided into two phases, data transmission phase and CH-Election phase. During first round, CH of each cluster is chosen previously based on its shortest distance to the BS. Then each round selects the next round CHs at the CH-Election phase before

starting the next round. Both phases are repeated rotationally at round process until the end of network's lifetime. Processing data streams is done only at the rounds process. Figure 3.15 describes the DeGiCA rounds process pseudo code. Rounds process runs a data stream packet DS_{packet} to result data D aggregated by CHs from all sensor node n . CHs send data D to the BS, then lately to the end user. Additionally, for more clarity, figure 3.16 explains the rounds process flowchart.

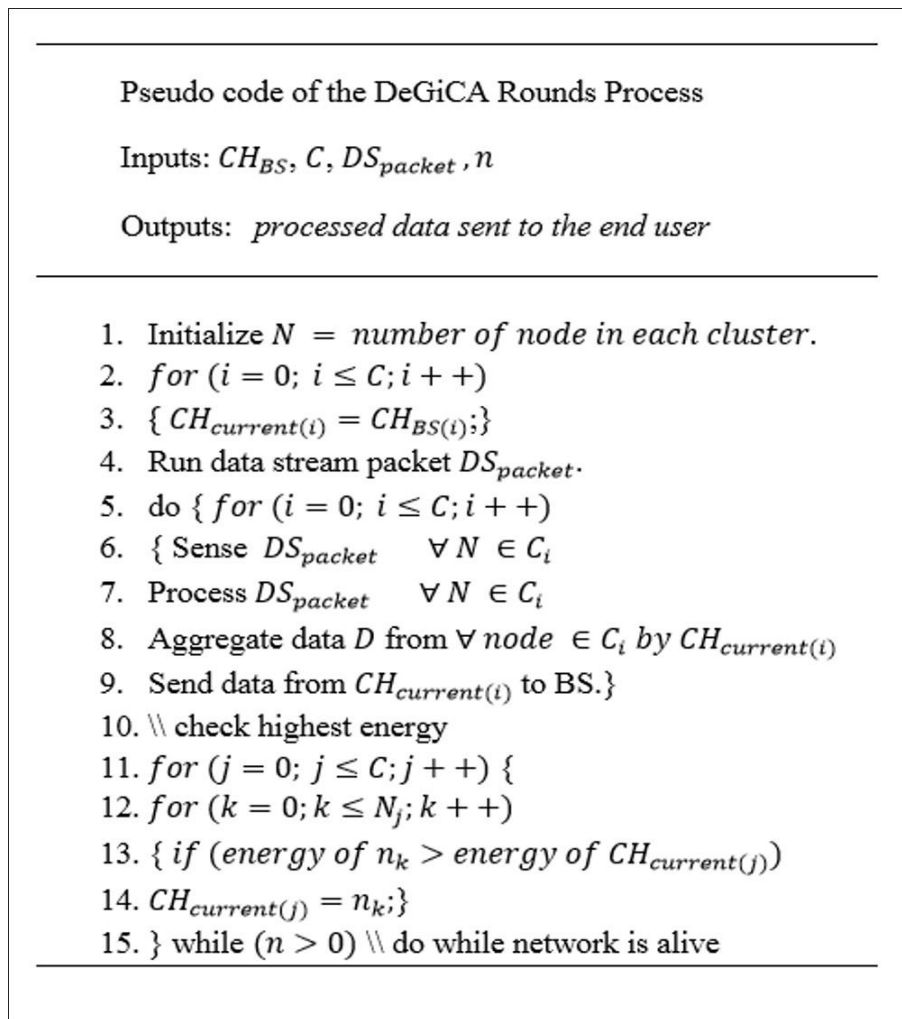


Figure 3.15 Rounds Process Pseudo Code

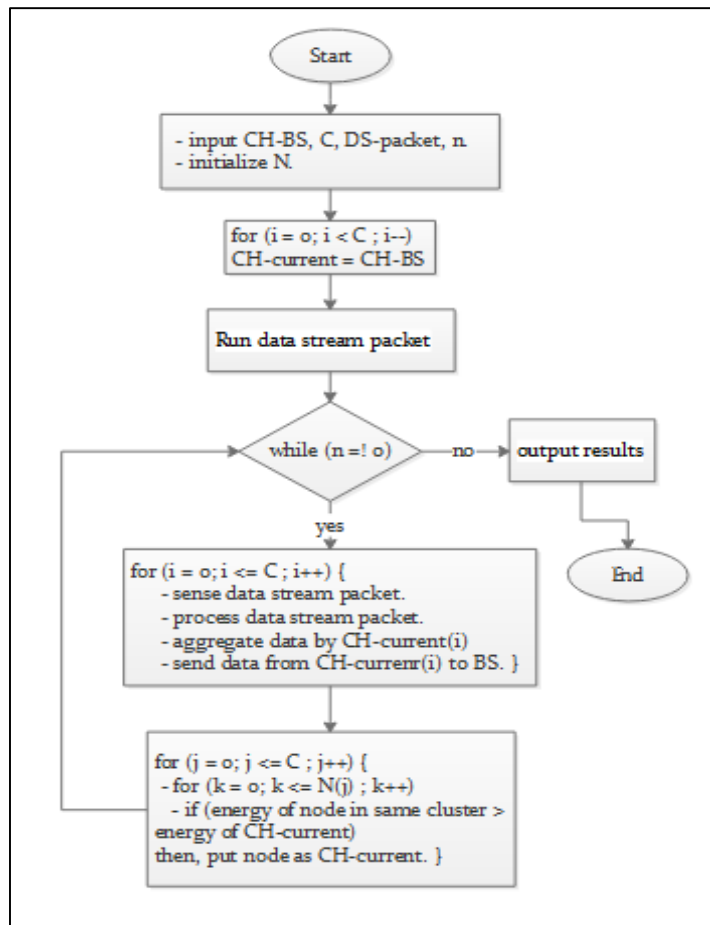


Figure 3.16 DeGiCA Rounds Process Flowchart

3.3.2.1 Data Transmission Phase of DeGiCA

Clustering technique used in DeGiCA leads to a two-level hierarchy where CHs form the higher level and nodes form the lower level. Nodes periodically transmit streaming data to their corresponding CHs. CHs aggregate, process streaming data packets, eliminate their redundancy and transmit them to the BS. The BS is the point of data processing for data received from nodes, and where data is accessed by the end user. The BS is considered to be at the center of the sensed area to be approximately near to all clusters. CHs act as gateways between sensor nodes and the BS.

Once all nodes in a cluster receive a join message, and a transmission schedule is initialized at the BS, sensor nodes start to perform data sensing and transmission to CHs. Once CHs receive all data, they perform data aggregation and processing. The resultant data are sent from CHs to the BS. This communication is a multi-hop communication in a hierarchical WSN topology. A multi-hop communication reduces the amount of information being transferred, hence reducing energy consumption. This phase is applied by using a method called **NodeProcess** that senses incoming streams and minimizes data transmission and **BSprocess** that process data and send it to the end user. In the proposed DeGiCA, three types of messages are used to perform the whole communication process. These messages are:

- **Advertise message (ADV)**; in case any node own a new data to be shared with the nodes, it should firstly employ ADV message in order to advertise that it has some data to share.
- **Request message (REQ)**; after sharing the data, any node accepts this data should take a response to ADV message by sending REQ message as indication that it wants to be a recipient for actual data.
- **Data message (DATA)**; includes the actual data to be shared by the node that initiate the communication process by ADV message.

During data transmission, an amount of nodes' energy is consumed, to calculate energy consumption of each node individually inside a cluster or even calculating the energy consumption for the whole network, the following function in figure 3.17 is used.

```

ClusterEnergy(j,x) = Cda(j,3);    %node j over x rounds
if(Cda(j,3) > .0001)
    Cda(j,2) = Cda(j,2) + 1;
    Cda(j,4) = Cda(j,4) + Cdist(j) * randn * 5e - 1 + 1;
end

```

Figure 3.17 DeGiCA Energy Consumption Function

At the end of each round, the energy consumption function has the ability to calculate the residual energy of each node inside a cluster and calculate the energy consumption of each cluster to outcome the overall network energy consumption. Calculating consumed energy of each node is a required process to obtain their residual energy. Node residual energy is used during CH election between cluster nodes.

3.3.2.2 CH-Election Phase of DeGiCA

CHs spend a lot more of energy than other normal sensor nodes do, due to sending the aggregated data to a higher distances (i.e. BS). A common solution to balance energy consumption among all nodes, is to periodically re-elect new CHs (rotating the CH role among all nodes over time) in each cluster. For each cluster, CH-Election is done by electing the node has the highest residual energy among all nodes including the current CH. This enhances energy consumption thus extends network lifetime. CHs election process for the upcoming rounds is done locally in each cluster. The current CHs (that are elected from the previous round) calculate the energy level of all alive nodes in its cluster. The competition between candidate nodes to be a CH is done by a method called **CHProcess** that also aggregates, processes streaming data, eliminates redundancy and sends data to the BS.

3.4 Conclusion

Density-based clustering can detect arbitrary shape clusters, handle outliers and do not need the number of clusters in advance. In this chapter, an overview of the proposed Density Grid-based Clustering Algorithm (DeGiCA) model was given. The chapter

provided in details the phases of DeGiCA: establishment phase, data transmission phase and CH-Election phase.

The DeGiCA is based on the concept of density of each grid to create the clusters in a gridded WSN. The idea of the proposed scheme was based on dividing a sensor network area into equal size of grids. Meanwhile, grids are classified to three main classifications, high dense, low dense and empty grids. Grids then are combined to form clusters and empty ones are used as delimiters. Advance nodes are elected to become CHs. CHs are initially chosen based on the shortest distance to BS. Then CHs are elected based on highest residual energy in the remaining rounds.

Chapter IV

Experimental Results and Analysis

Chapter IV

Experimental Results and Analysis

4.1 Introduction

Clustering has proven to be an effective approach for organizing WSNs into connected hierarchy. Clustering technique, density technique and grid, have proven their efficiency to reach a high network performance. Besides clustering advantages, density can find arbitrary shaped clusters with noise and network gridding avoids clustering quality problems by discarding boundary points of grids. The combination of techniques enhances clustering performance as being presented later in this chapter.

This chapter provides an evaluation of the developed Density Grid-based Clustering Algorithm (DeGiCA) to indicate its efficiency among other stream mining clustering algorithms to enhance WSNs performance. Moreover, the chapter discusses several comparisons between the two competitors mentioned previously (i.e. FCM and K-means), their final simulation results and performance metrics. To achieve fair evaluation of DeGiCA, both standard FCM and K-means in such comparisons and evaluations have been modified to stream data and run same dataset streaming packets. For a research purpose, a standard dataset streaming packet form is used with the three competitors to show their final experimental results.

The procedure used to obtain the final experimental results for three competitors is by running DeGiCA first to form its clusters, then gain number of clusters C and their centers v . After that, DeGiCA competitors are run each individually using C and v . DeGiCA and its competitors form three different individual WSNs with same number of clusters. A comparison function is then used to compare between performance metrics results of the three competitors. Comparisons between competitors are done for three main metrics, in terms of overall network lifetime, overall energy consumption for entire network, and lastly, overall packet delivery. The rest of this chapter is organized as follows: section 2 represents DeGiCA performance metrics. Section 3 presents DeGiCA system requirements that are needed for implementation. Section 4 presents DeGiCA simulation experimental analysis. Section 5 presents the effect of gridding on a WSN lifetime. Finally, section 6 presents the chapter conclusion.

4.2 DeGiCA Performance Metrics

This section provides the most important definitions and concepts related to required research work. Clarifying definitions and concepts simplifies some difficulties faced during simulation result analysis. The following points are performance metrics definitions specializes for DeGiCA and its competitors:

- 1- Network Lifetime.** The lifetime of a WSN can be defined as the time elapsed until the last node dies, or a fraction of nodes dies. Network Lifetime is usually measured in seconds. For scientific research, it is assumed to initially charge nodes with very small energies, equivalent to 1 *joule*, so that nodes die faster to get

results in seconds. Common network lifetime definition is the time until the first/last node in the network depletes its energy and the time until a node is disconnected from the BS [47].

$$\text{Network Lifetime} = \text{Number of round} = \frac{\text{total network energy}}{\text{total energy consumed in each round}}$$

- 2- **Energy Consumption.** It is a measure of amount at which energy is dissipated by sensor nodes in a WSN within a specific period of time. Energy consumption is measured in joules.
- 3- **Packet Delivery Ratio.** It is the ratio of total number of delivered packets successfully received by the BS to the number of packets sent by all sensor nodes in the network. It is expressed in percentage. Packet delivery ratio depends on number of lost packets in physical and MAC layers. Packet loss may occur only when nodes start to die, but in this case, CHs are responsible for transmitting packets. So, all packets are received by the BS in DeGiCA.

4.3 DeGiCA System Requirements

This sections presents main requirements to implement DeGiCA. It provides dataset description, simulation experimental parameters and minimum simulator requirements.

4.3.1 DeGiCA Dataset Description

Dataset streaming packet used in this research is a standard form used for scientific research purposes, where used simulator (i.e. based on MATLAB) doesn't generate

randomly a desirable dataset streaming packet. Dataset packet is generated randomly from distributed nodes. Packet's form is suitable for different clustering schemes and algorithms. Dataset streaming packet is assumed to be non-uniform. If uniformed, cluster formation is considered to be uniformed and the clustering algorithm research study becomes inefficient.

Each node in the experimental sensed area has a specific data structure consists of the following:

- 1- Nodes are immobile. So, each node has **coordinates** (x, y) that defines its location inside the sensed area.
- 2- **Node energy** that represents a node current energy is holding. Initially, all nodes have an energy = 1 *joule*. This value is actually used for some scientific research studies. Using a value greater than 1 Joules during research experiments causes long execution time and may effect on the used machine usability.
- 3- **Energy losses factor** at each frame transmission.
- 4- **Status** of the node. Nodes could be **active** during sensing and transmitting otherwise they remain **sleep**.
- 5- **Index** of the node.

The dataset streaming packet is designed in an appropriate manner to fit the requested network. It is built in a simple form used to communicate between nodes. Generally, dataset streaming packets flow in a very high speed. At streaming packet arrival, time is recorded whenever an event occurs. Needless, data streams are not saved. They are read and processed then released immediately. Saving such data leads to the process

of manipulating with big data in warehouses which is not considered in this research scope.

Data streams differs from static data. **Static data** are unchangeable and not a real-time data. **Data streams** are all about *real-time* data where data are collected from various sensors. Data stream packet used in the developed DeGiCA is a 126 byte/message, it has the following parameters as shown in table 4.1:

TABLE 4.1 Structure of Data Stream Packet

Parameter	Size	Parameter	Size
Delimiter	1 byte	Source Address	8 bytes
Length	2 bytes	Destination Address	8 bytes
Network ID	2 bytes	Options Flag	1 byte
PAN ID	2 bytes	Payload Data	100 byte
Node Type	1 byte	Checksum	1 byte

Table 4.1 shows the structure of dataset streaming packet used by this research for DeGiCA and its competitors. Each parameter in dataset stream packet is described as:

- **Delimiter.** The starting of the data stream packet frame indicator.
- **Length.** Length of the data stream packet frame.
- **Network ID.** Refers to the overall network.
- **PAN ID.** Refers to cluster ID.
- **Node type.** Indicates the leaf node, head of cluster ... etc.
- **Source address.** A 64-bit address of the node for the sent data.
- **Destination address.** A 64-bit address of the node that is receiving the data.
- **Options flag.** It provides the network management options.
- **Payload data.** The message or streaming data.
- **Checksum.** The data validation check sum.

Figure 4.1 presents the dataset stream packet structure used in DeGiCA and its competitors.

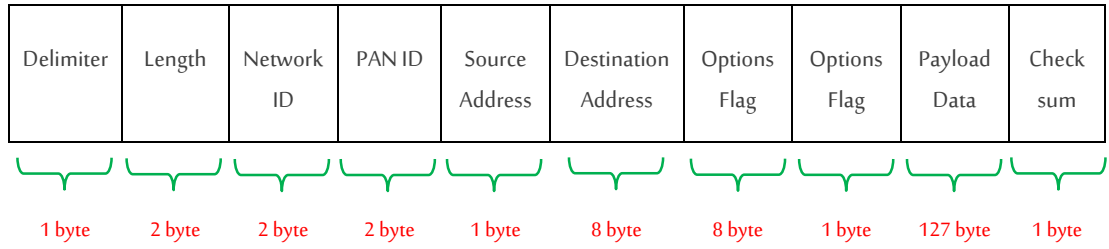


Figure 4.1 Structure of Dataset Streaming Packet

Figure 4.1 shows each dataset streaming packet component size in byte. This structure is used in DeGiCA simulation experiments. Due to scientific research purpose, data in the sample is considered to be digital, so it is presented without payload data.

4.3.2 Experimental Setup Parameters

Before evaluating DeGiCA, experimental setup parameters used to implement the developed algorithm and its competitors are clarified in this section. Table 4.2 presents experimental setup parameters and corresponding values used in evaluation and scalability experiments.

TABLE 4.2 Experimental Setup Parameters

Experimental Parameters	Experimental Parameters Values
Sensed area network size	1000m X 1000m
Initial energy for each node	1 J/node
Number of nodes n	100, 200 node
E_{elec} (transmit/receive energy)	100 pJ/bit
Data message / Data packet size	126 byte/message
Control packet	26 byte
Node Communication	IEEE 802.15.4 Industrial Wireless Sensor Networks
BS location	At the centre of all networks; its coordinates are (500, 500)
Threshold σ	4, 5, 6
Grid size g	80, 110, 120, 130, 140, 150, 155 m

4.3.3 DeGiCA Simulator Requirements

In order to evaluate the developed DeGiCA, the experimental setup parameters are used to implement three different simulations (i.e. DeGiCA and its competitors) using the famous MATLAB software version R2008b. In this research experiments, MATLAB is used in a machine with Windows 7 Service Pack 1 with 1TB disk space, 64-bit operating system, Intel ® Core™ i7 processor and 8GB RAM. Table 4.3 shows machine minimum requirements to install this version of MATLAB.

TABLE 4.3 Machine Minimum Requirement for MATLAB R2008b

Operating System	Processors	Disk Space	RAM
1- Windows XP (Service Pack 1 or 2) 2- Windows Server 2003 x64 (Service Pack 1 or 2, R2) 3- Windows Vista	1- Intel Pentium (Pentium 4 and above) 2- Intel Celeron 3- Intel Xeon 4- Intel Core 5- AMD64	510 MB (MATLAB only)	1024 MB (2048 recommended)

The reason behind using this version of MATLAB is due to its ability to handle operations and procedures of DeGiCA and its techniques with no need for more advanced libraries as seen in MATLAB 2016 that requires long time to process and start running. Competitors stream the same dataset streaming packet in an environment with 100 node scattered randomly at an equivalent sensed area space size.

4.4 DeGiCA Simulation Experimental Analysis

Several simulation experiments are done on DeGiCA and its competitors. Many experiments had preferable DeGiCA performance metrics outcomes compared to its competitors. Ten best performance metrics simulation experiments are chosen to be

presented in this research. Obviously, setup parameters are used at the initialization process. Grid size g and threshold σ are parameters that are being tested until finding their optimum values. Table 4.4 presents ten best experiments to evaluate DeGiCA and obtain optimum g and σ .

TABLE 4.4 Ten Best Simulation Experiments for DeGiCA

Threshold σ	Grid Size g	Clusters C
3	80	6
4	110	4
4	120	4
4	130	5
4	140	6
5	130	2
5	140	3
5	150	3
5	155	5
6	140	2

This section is the most important part of thesis research study. It provides a detailed discussion about DeGiCA simulation experimental results. First, it provides a description about all presented graphs from simulator. Second, it evaluates DeGiCA by comparing its final outcomes with its competitor's outcomes. Third, it provides a study on the optimum network grid size and threshold for a DeGiCA WSN. Fourth, based on the obtained optimum grid size and threshold, DeGiCA is scaled.

4.4.1 DeGiCA Expected Outcomes Description

Representing outcomes and visualizing results in graphs is preferred to achieve simplicity and realization. DeGiCA has several expected results requires clarification to measure its performance metrics. So, the following description provides a selected DeGiCA experiment results, assuming its experimental area size equivalent to $(1000 \times 1000) m^2$ with $n = 100$. Each node has a random specific position with

initial energy = 1 joule. Figure 4.2 presents this experiment at the establishment phase during grid process when $g = 150$ and threshold $\sigma = 5$.

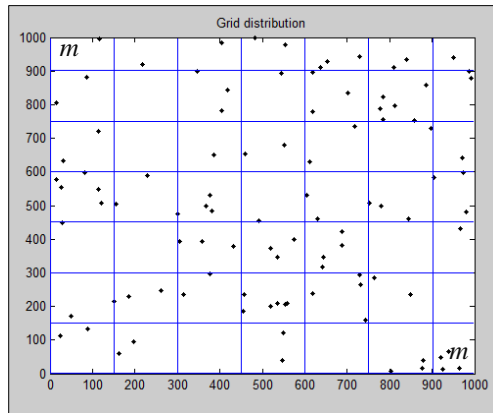


Figure 4.2 Establishment Phase during Sensed area Gridding when $g = 150$ and $\sigma = 5$

Grids are classified based on their density, then nodes are clustered based on cluster formation process in DeGiCA. Figure 4.3 (a) shows grid classification based on density, green cells for high dense grids, dark blue for low dense while light blue cells are empty grids. Figure 4.3 (b) shows corresponding clusters. Each cluster has a specific color and a clear CH. After cluster formation process and accomplishing network initialization steps, rounds process starts, and results are obtained.

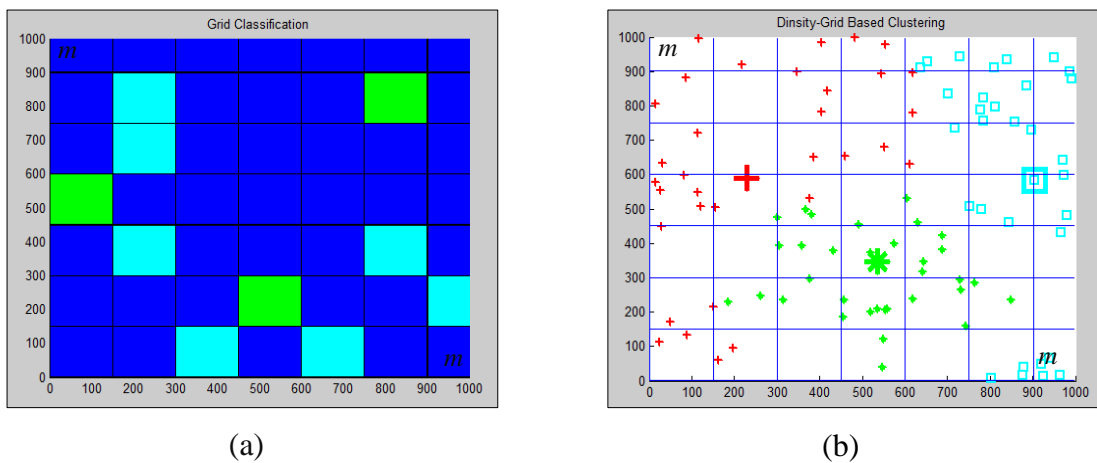


Figure 4.3 Establishment phase when $g = 150$ and $\sigma = 5$ (a) Grid Classification (b) Corresponding Cluster Formation

To measure DeGiCA performance metrics, a scheme of its network lifetime shows the relationship between time and number of a live nodes, as shown in figure 4.4.

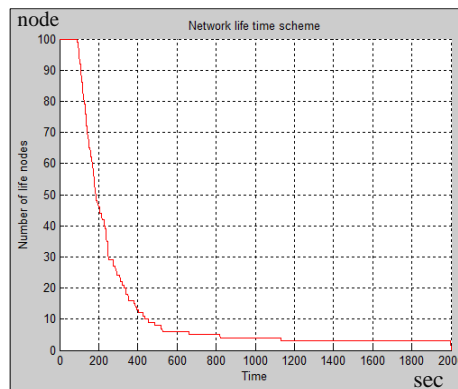


Figure 4.4 Network Lifetime Graph when $g = 150$ and $\sigma = 5$

In addition, a scheme calculates energy consumption for each cluster is represented. It shows the relationship between time and energy. Figure 4.5 presents energy consumption for each cluster in the network, (a) for the first cluster and (b) for the second cluster.

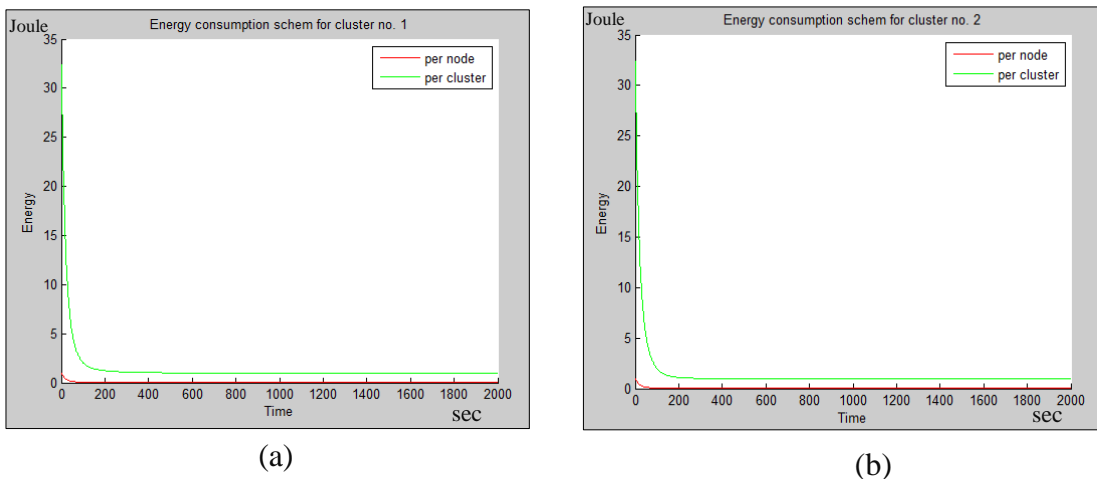


Figure 4.5 Energy Consumption for each Cluster when $g = 150$ and $\sigma = 5$ (a) for First Cluster (b) for Second Cluster

Energy consumption scheme for the whole network shows the relationship between time and energy as shown in figure 4.6.

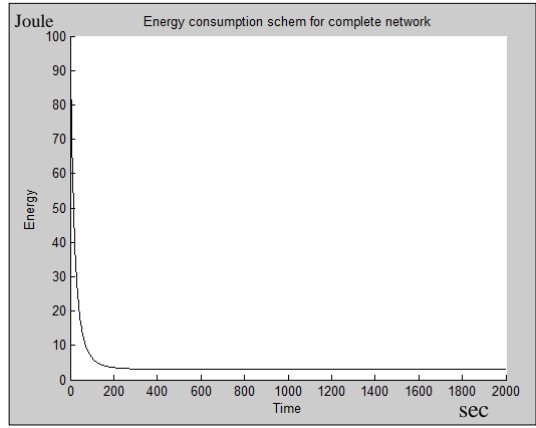


Figure 4.6 Energy Consumption for whole Network when $g = 150$ and $\sigma = 5$

To measure DeGiCA selected experiment percentage of packet delivery ratio, a scheme for total network packets delivered is shown in figure 4.7.

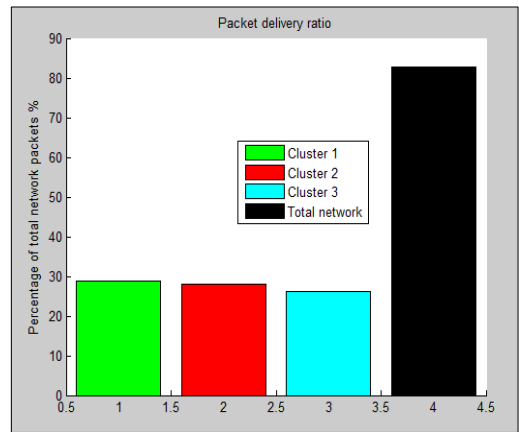


Figure 4.7 Packet Delivery Ratio for each Cluster and for whole Network when $g = 150$ and $\sigma = 5$

4.4.2 Evaluating DeGiCA in terms of Network Lifetime and Energy Consumption

Three best selected simulation experiments are chosen to compare DeGiCA and its competitors between their final performance metrics result in terms of network lifetime and energy consumption. First experiment is when $g = 130$ and $\sigma = 5$. Second experiment is when $g = 155$ and $\sigma = 5$. The last experiment is when $g = 140$

and $\sigma = 6$. All competitors in their experiments are streaming the same dataset stream packet with size 126 byte/message in a $(1000 \times 1000) m^2$ sensed area, with $n = 100$ nodes scattered randomly each with an initial energy equal to 1 joule. Figure 4.8 presents a comparison between DeGiCA, FCM and K-means when $g = 130$ and $\sigma = 5$ in terms of (a) network lifetime, (b) their energy consumptions. Other experimental results comparing DeGiCA, FCM and K-Means when $g = 130$ and $\sigma = 5$ are shown in Appendix B (figure B.1, figure B.2).

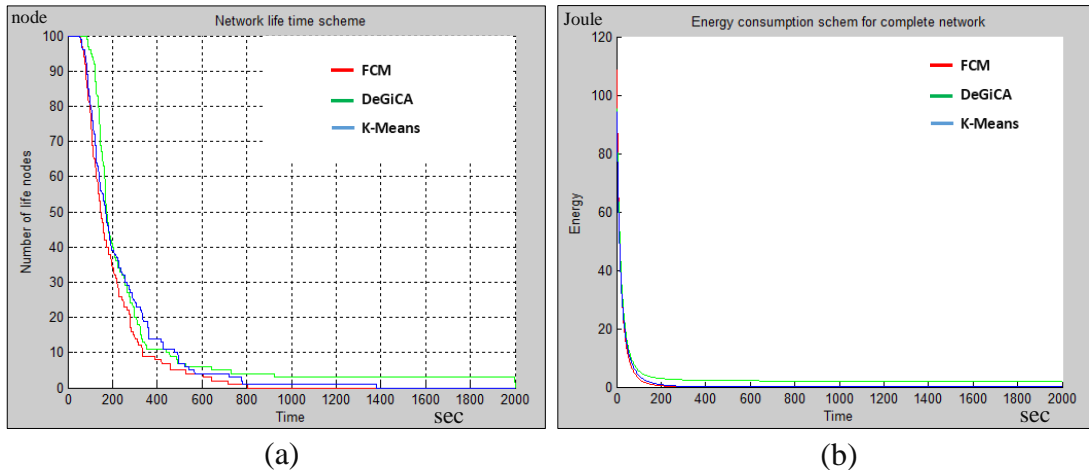


Figure 4.8 Comparing DeGiCA, FCM and K-means when $g = 130$ and $\sigma = 5$ in terms of (a) Network Lifetime (b) Corresponding Energy Consumption

From the figure, we noticed that number of alive nodes are dropped sharply at 600 sec. DeGiCA is better than FCM and K-Means. The energy consumption are dropped sharply at 100 sec. DeGiCA has little performance than its competitors.

For the second experiment, figure 4.9 presents a comparison between DeGiCA, FCM and K-means when $g = 155$ and $\sigma = 5$ in terms of (a) network lifetime, (b) their energy consumptions. Other experimental results comparing DeGiCA, FCM and K-Means with $g = 155$ and $\sigma = 5$ are shown in Appendix B (figure B.3, figure B.4).

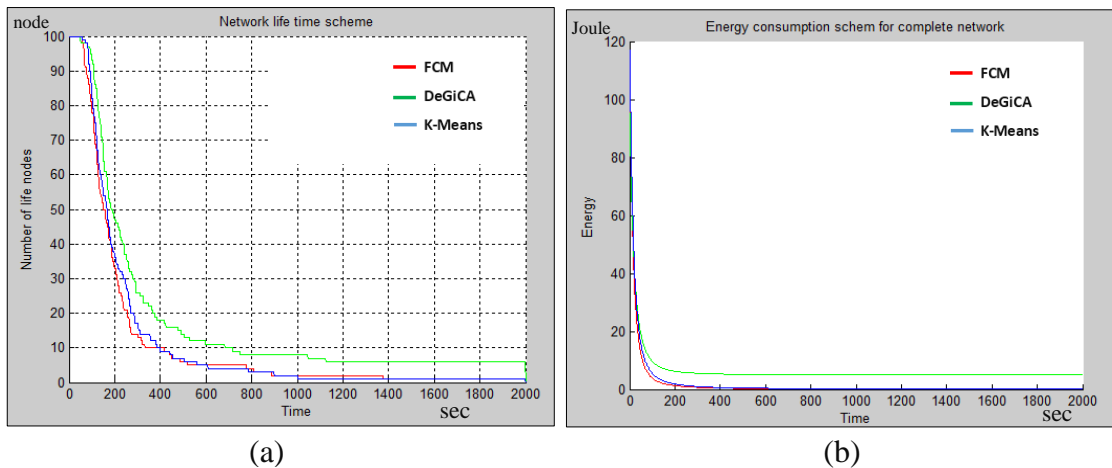


Figure 4.9 Comparing DeGiCA, FCM and K-means when $g = 155$ and $\sigma = 5$ in terms of (a) Network Lifetime (b) Corresponding Energy Consumption

From the figure, we noticed that number of alive nodes are dropped sharply at 500 sec. DeGiCA is better than FCM and K-Means. The energy consumption are dropped sharply at 100 sec. DeGiCA has little performance than its competitors.

Lastly, the third simulation experiment is shown in figure 4.10 presents a comparison between DeGiCA, FCM and K-means when $g = 140$ and $\sigma = 6$ in terms of (a) network lifetime, (b) their energy consumptions. Other experimental results comparing DeGiCA, FCM and K-Means when $g = 140$ and $\sigma = 6$ are shown in Appendix B (figure B.5, figure B.6).

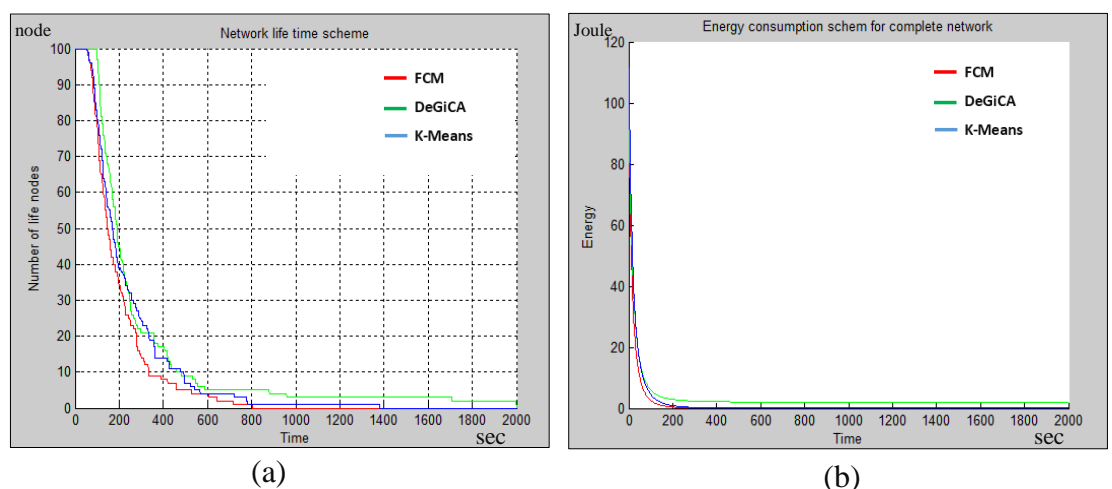


Figure 4.10 Comparing DeGiCA, FCM and K-means when $g = 140$ and $\sigma = 6$ in terms of (a) Network Lifetime (b) Corresponding Energy Consumption

From the figure, we noticed that number of alive nodes are dropped sharply at 400 sec. DeGiCA is better than FCM and K-Means. The energy consumption are dropped sharply at 100 sec. DeGiCA has little performance than its competitors.

Briefly, as shown in the simulated experimental result graphs, it is clear that the developed DeGiCA has the longest network lifetime compared to its competitors. Table 4.5 provides examples for some live nodes in a certain running time during network lifetime of DeGiCA, K-means and FCM. First Node to Die is referred to FND.

TABLE 4.5 Number of Live Nodes in DeGiCA, K-means and FCM at a Certain Time

	Time	DeGiCA	K-means	FCM
$g = 130$ $\sigma = 5$	FND	95	50	60
	100	96	83	80
	300	24	25	14
	500	8	9	5
	1000	4	2	1
	1500	4	1	1
	2000	4	1	1
$g = 155$ $\sigma = 5$	FND	50	50	50
	100	90	77	73
	300	29	17	13
	500	14	7	5
	1000	8	3	3
	1500	6	2	2
	2000	6	2	2
$g = 140$ $\sigma = 6$	FND	100	40	50
	100	100	80	79
	300	23	25	15
	500	19	17	6
	1000	4	2	1
	1500	4	1	0
	2000	3	1	0

Table 4.5 is expressed by using charts to present live nodes in DeGiCA and its competitors. By applying Rounds for X-axis and number of live nodes for Y-axis,

figures 4.11 presents number of live nodes when $g = 130$ and $\sigma = 5$. Other experimental results presents number of live nodes for DeGiCA and its competitors when $g = 155$ and $\sigma = 5$ and when $g = 140$ and $\sigma = 6$ as shown in Appendix B (figure B.7, figure B.8). K-means usually has the first node to die while DeGiCA has the last one to die. So, it seems that DeGiCA has the longest network lifetime than both competitors where FCM has the shortest lifetime.

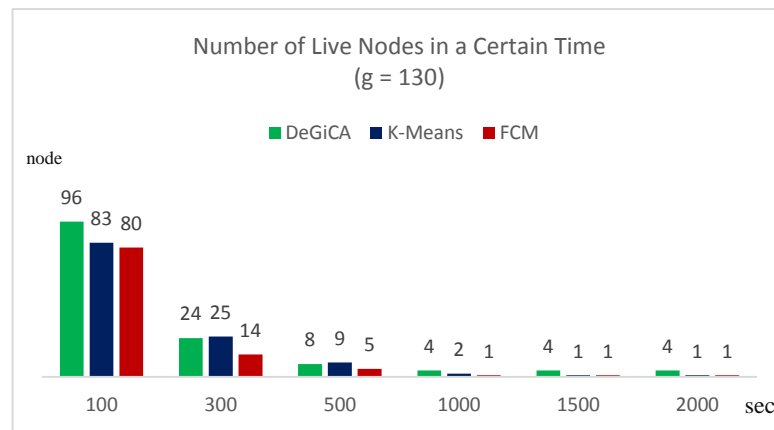


Figure 4.11 Number of Live Nodes when $g = 130$ and $\sigma = 5$

In average, for the simulated experimental result comparisons shown above, DeGiCA has shown to have the longest network lifetime compared to both FCM and K-means. Network lifetime is enhanced in DeGiCA compared to its competitors. This can be presented in figure 4.12, where X-axis presents the average network lifetime.

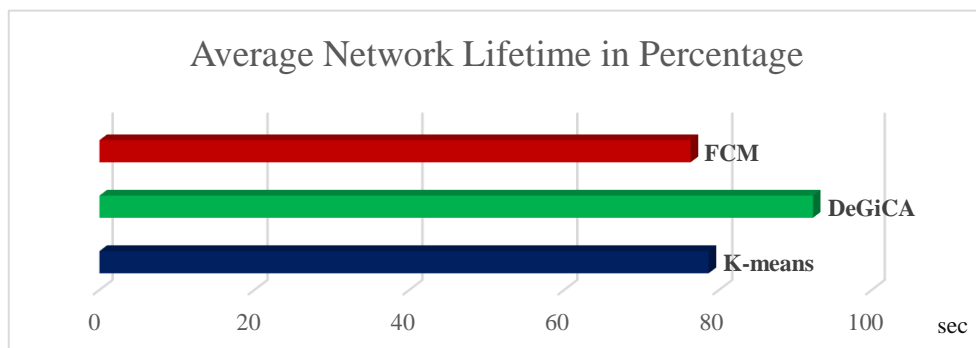


Figure 4.12 Average Network Lifetime for DeGiCA, FCM and K-means

For energy consumption analysis shown in figure 4.8 (b), figure 4.9 (b) and figure 4.10 (b), the developed DeGiCA consumes less energy than its competitors. Table 4.6 provides some approximated consumed energy results for DeGiCA, K-means and FCM for the selected simulation experiments when $g = 130, 155, 140$ respectively.

TABLE 4.6 Percentage of Consumed Energy Results in DeGiCA, K-Means and FCM in a Certain Time

	Time	Percentage of Remaining Energy in Joule		
		DeGiCA	K-means	FCM
$g = 130$ $\sigma = 5$	10	67	75	60
	25	33	35	28
	50	15	16	10
	75	8	8	7
	100	8	7	4
	200	6	2	1
	300	4	2	1
	450	3	1	1
	500	3	1	1
$g = 155$ $\sigma = 5$	10	65	80	80
	25	35	38	36
	50	19	17	11
	75	15	9	5
	100	11	8	5
	200	10	8	3
	300	9	7	2
	450	7	5	2
	500	6	4	1
$g = 140$ $\sigma = 6$	10	65	60	69
	25	35	29	30
	50	17	12	13
	75	9	6	4
	100	9	4	3
	200	6	3	2
	300	4	2	2
	450	4	2	1
	500	4	2	1

To effectively express table 4.6, Figures 4.13 presents energy consumption percentage when $g = 130$ and $\sigma = 5$ at a certain time, Other experimental results presents energy consumption percentage for DeGiCA and its competitors when $g = 155$ and $\sigma = 5$ and when $g = 140$ and $\sigma = 6$ as shown in Appendix B (figure B.9, figure B.10). It seems that the developed DeGiCA consumes less energy than both competitors.

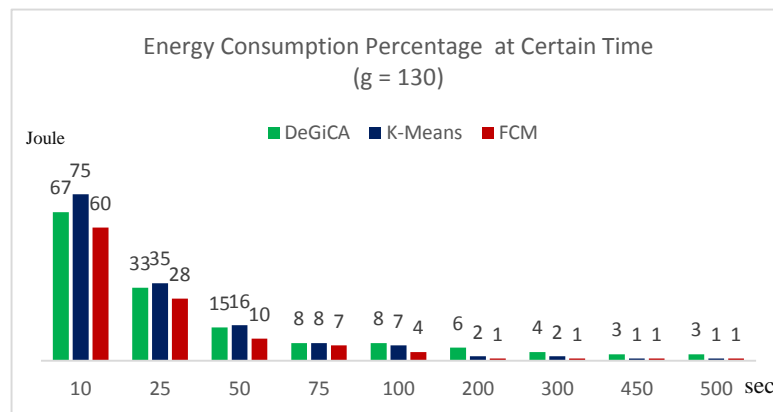


Figure 4.13 Energy Consumption Percentage when $g = 130$ and $\sigma = 5$ in a Certain Time

Obveiuosly, DeGiCA resumes less energy than its competitors do. As a matter of fact, it extends network lifetime but uses only its sufficient amount of energy. Figure 4.14 shows that average energy consumption in DeGiCA and competitors.

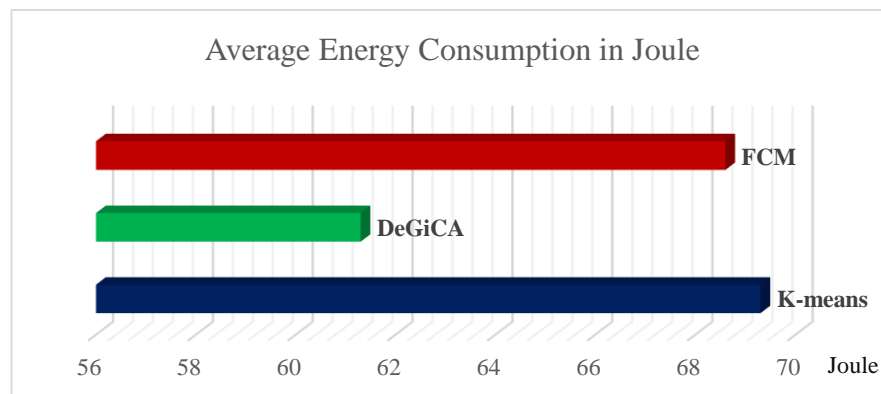


Figure 4.14 Average Energy Consumption for DeGiCA, FCM and K-mean

4.4.3 Evaluating DeGiCA in terms of Packet Delivery Ratio

Three best simulation experiments are chosen to perform a comparison between DeGiCA and its competitors and finally evaluating the developed algorithm in terms of packet delivery ratio. Competitors are streaming same dataset stream packet having a size equivalent to 126 byte/message in a $(1000 \times 1000) m^2$ sensed area, with $n = 100$ nodes scattered randomly each with an initial energy equal to 1 joule. The first experiment is when $g = 80$ and $\sigma = 3$. Second experiment is when $g = 110$ and $\sigma = 4$. The last experiment is when $g = 140$ and $\sigma = 6$.

The following figures show delivered packets ratio for the competitors. Figure 4.15 presents a chart when $g = 80$ and $\sigma = 3$, other experimental results charts when $g = 110$ and $\sigma = 4$ and when $g = 140$ and $\sigma = 6$ as shown in Appendix B (figure B.11, figure B.12).

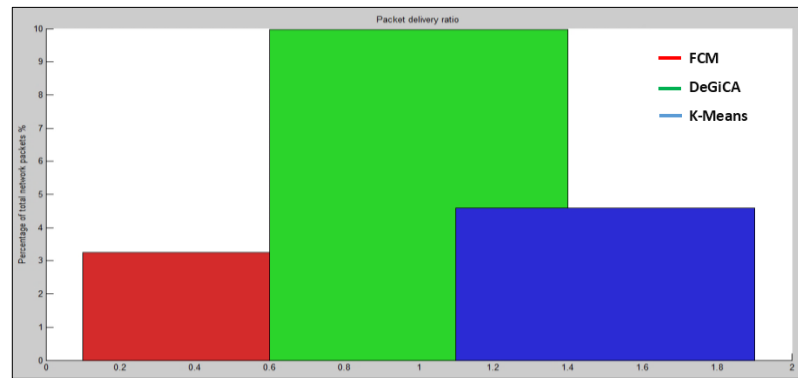


Figure 4.15 Overall Packet Delivery Ratio when $g = 80$ and $\sigma = 3$

In average, the developed DeGiCA has proven strongly to deliver a high ratio of data packets compared to FCM and K-Means. It almost receives all data packets at the final destination. The delivery ratio is enhanced in DeGiCA compared to its competitors, this can be shown in figure 4.16.

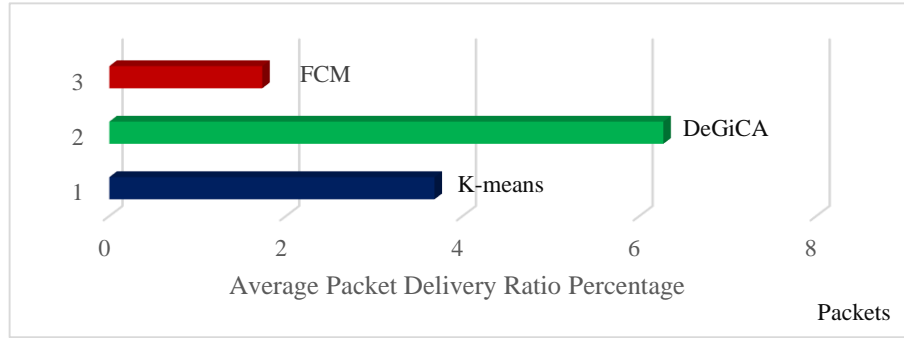


Figure 4.16 Average Delivered Packets for Competitors

4.4.4 DeGiCA Experimental Results to Determine Optimum g and σ

This section provides strategy used to determine the optimum value of grid size g and threshold σ used at clustering formation process in DeGiCA. Once determining optimum values, DeGiCA starts scalability testing found in the next section.

It's required to specify the parameters related to each other in DeGiCA and find their relationship. DeGiCA has three main parameters related to each other: number of clusters C , grid size g and threshold σ . Number of clusters is obtained from the algorithm, so it is an output based on g and σ . To determine the optimum value of g and σ , several simulation experiments were selected to find the best performance metrics based on keeping g fixed while σ is a variable and vice versa. Table 4.7 presents selected experiments with best performance metrics results.

TABLE 4.7 Grid Size and Threshold Simulation Experiment Selection to Determine their Optimum Values of g and σ

σ is fixed and g is a variable			g is fixed and σ is a variable		
σ	g	C	g	σ	C
4	120	4	140	4	6
	130	5		5	3
	140	6		6	2

As a matter of fact, and from several simulation experimental results, whenever grid size increases, number of clusters decreases resulting an **inverse relationship**, but in this section it seems to be not. The reason behind that is, the best selected values of g are almost close to each other and all selected five experiments shown in the previous table have a high enhancement of network performance metrics. Table 4.8, presents performance metrics results of five selected simulation experiments to determine the optimum value of g and σ .

TABLE 4.8 Performance Metrics of Competitors for 5 best selected Simulation Experiments to Determine the Optimum Values

	DeGiCA			FCM			K-means		
	Lifetime	Energy	Packets	Lifetime	Energy	Packets	Lifetime	Energy	Packets
$g = 120$ $\sigma = 4$	5%	8%	14%	2%	2%	11%	0%	0%	12%
$g = 130$ $\sigma = 4$	7%	9%	17%	1%	1%	10%	6%	5%	13%
$g = 140$ $\sigma = 4$	7%	9%	17.5%	1%	1%	9%	5%	6%	13%
$g = 140$ $\sigma = 5$	4%	5%	16%	2%	2%	9%	3%	1%	13%
$g = 140$ $\sigma = 6$	2%	3%	14%	1%	2%	10%	1%	1%	12%

As clearly presented in table 4.8, it is found that the optimum value of g and σ is when $g = 140$ and $\sigma = 4$, where network lifetime, energy consumption and packet delivery ratio are enhanced perfectly compared to other selected g and σ . In table 4.8, percentage of remaining nodes is used to measure network lifetime in a certain time, percent of remaining energy in the network is used to measure its energy consumption in a certain time and finally percent of delivered packets is used to measure packet delivery ratio in a certain time

4.4.5 DeGiCA Scalability

To measure DeGiCA scalability, optimum values of both grid size g and threshold σ are used in DeGiCA scalability test. DeGiCA has the ability to be scaled based on applying the developed algorithm on two different environments running the same dataset stream, then comparing the resulted performance metrics with its competitor's results. DeGiCA is scaled using the following simulation experiments with $g = 140$ and $\sigma = 4$:

Table 4.9, presents simulation experimental results of DeGiCA scalability compared to its competitors using optimum values when $g = 140$ and $\sigma = 4$.

TABLE 4.9 DeGiCA Scalability Based on Optimum Values at Certain Time

	DeGiCA		FCM		K-means	
	$n = 100$	$n = 200$	$n = 100$	$n = 200$	$n = 100$	$n = 200$
Network Lifetime	7%	6%	1%	1%	5%	1%
Energy Consumption	9%	14%	1%	16%	6%	17%
Packet Delivery Ratio	17.5%	19%	9%	15%	13%	16%

In table 4.9, percentage of remaining nodes in the network is used to measure network lifetime in a certain time, percent of remaining energy in the network is used to measure its energy consumption in a certain time while percent of delivered packets is used to measure packet delivery ratio in a certain time. Methodology followed to scale DeGiCA is by firstly running the algorithm in $(1000 \times 1000) m^2$ sensed area, with $n = 100$ node then running it in $(1000 \times 1000) m^2$ sensed area, with $n = 200$ node, It seems that DeGiCA enhances performance metrics compared to its

competitors but it provide less enhancement than when $n = 100$ node. Still, DeGiCA is assumed to be scalable.

4.5 Effect of DeGiCA Gridding on WSN Lifetime

This section provides analysis and discussion about the effect of grid technique on network lifetime. To discussions the effect of grid size g on DeGiCA network lifetime performance metrics, four different simulation experiments are chosen based on grid size (i.e. $g = 110, 120, 130, 140$). Each experiment streams the same dataset streaming packet with size 126 byte/message in a $(1000 \times 1000) m^2$ sensed area, with $n = 100$ nodes scattered randomly each with an initial energy equal to 1 joule. Threshold σ is set to 4 for all experiments resulting in 4 gridded networks with 4, 4, 5 and 6 clusters consequently. The following figures represent four gridded WSNs where grid size g is predetermined to divide both X - axis and Y - axis resulting network with equal size of cells. Figure 4.17 presents a gridded network when (a) $g = 110$, (b) $g = 120$, other experimental results presents gridded networks when $g = 130$ and when $g = 140$ as shown in Appendix B (figure B.13, figure B.14).

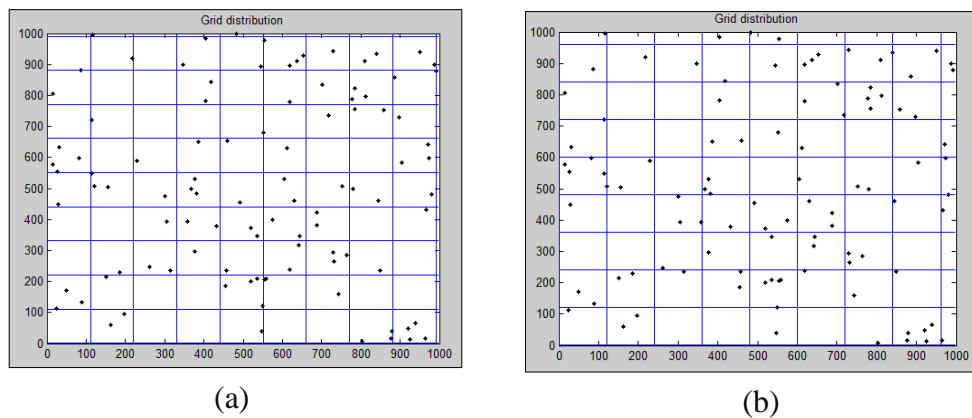


Figure 4.17 Gridded WSN when (a) $g = 110$ (b) $g = 120$

After gridding the networks by DeGiCA, density and grid techniques found in the establishment phase in turn take role to form clusters based on density classification, resulting clustering formation process as shown in the following figures. Figure 4.18 presents establishment phase when $g = 110$ at (a) grid classification and (b) corresponding cluster formation resulting 4 clusters. Figure 4.19 presents establishment phase when $g = 120$ at (a) grid classification and (b) corresponding cluster formation resulting 4 clusters. Other experimental results grid classification and cluster formation when $g = 130$ and when $g = 140$ as shown in Appendix B (figure B.15, figure B.16, figure B.17, and figure B.18). Each cluster is colored in a specific color and the node with biggest size is considered to be a CH in its cluster.

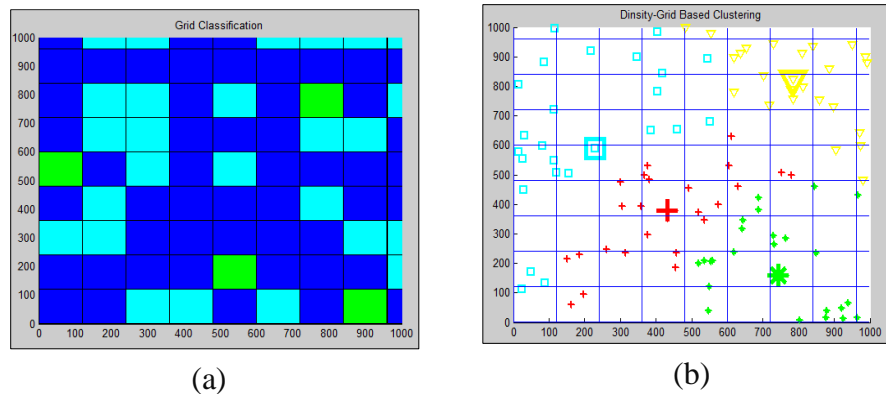


Figure 4.18 Establishment phase when $g = 110$ at (a) Grid Classification and (b) Corresponding Cluster Formation

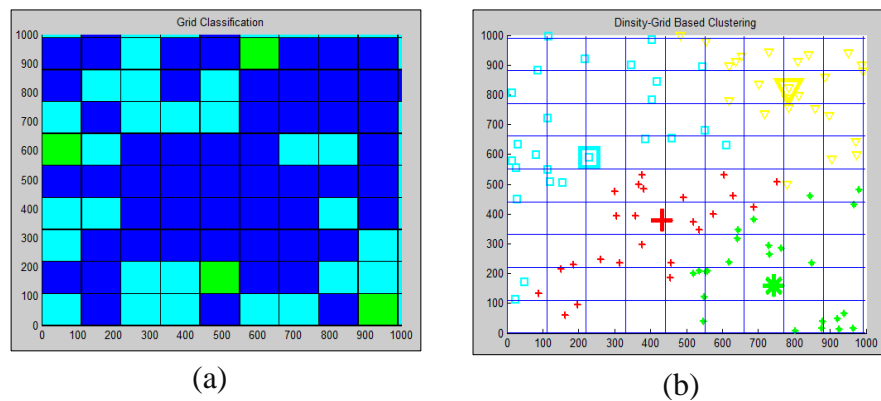
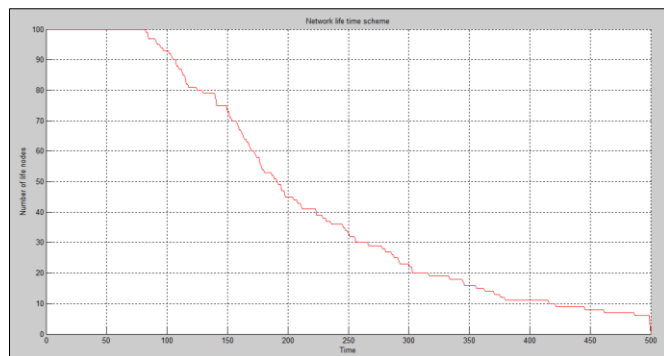
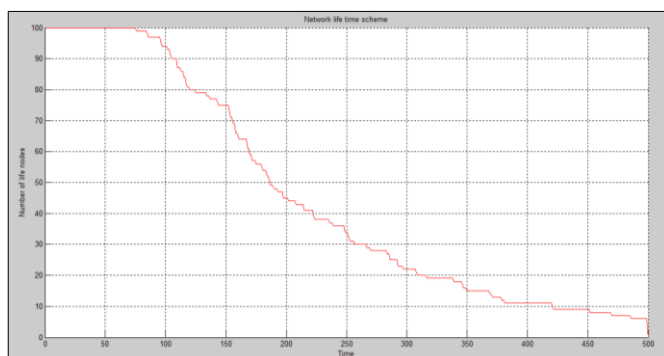


Figure 4.19 Establishment phase when $g = 120$ at (a) Grid Classification and (b) Corresponding Cluster Formation

Whenever clusters are formed, network is ready to stream data packets and starts processing rounds until the end of network lifetime. Round process goes through two main phases, data transmission phase and CH-Election phase. As discussed, the aim of DeGiCA is to extend a WSN lifetime, reduce its energy consumption and reach a high percent of delivered data packets compared to similar algorithms (i.e. FCM and K-Means) where the technique of density-gridding plays a main role to achieve those goals. The purpose of this section is to find the effect of cluster size g found in DeGiCA on a WSN lifetime. Figure 4.20 (a) presents the death of first node when $g = 110$, while (b) presents the death of first node when $g = 120$, other experimental results presents the death of first node when $g = 130$ and when $g = 140$ as shown in Appendix B (figure B.19, figure B.20).



(a)



(b)

Figure 4.20 Death of First Node when (a) $g = 110$ (b) $g = 120$

As shown from the previous figures, whenever cluster size g becomes larger, the death of first node becomes earlier. In addition, whenever the grid size g is increasing, the network lifetime extended. Table 4.10 shows the time of first node to die and number of nodes in a certain time.

TABLE 4.10 Number of Nodes to Die in Networks when $g = 110, 120, 130, 140$

Grid Size	death of the first node in s	Number of live node in a certain time in (sec)								
		150	300	500	800	1200	1800	6000	60000	90000
110	80 s	74	23	7	7	4	4	4	4	4
120	75 s	74	23	7	7	4	4	4	4	4
130	70 s	72	23	12	9	8	8	8	5	5
140	45 s	70	24	12	10	8	8	8	6	6

Figure 4.21 represents table 4.10, as clearly shown, network lifetime is extended whenever grid size g is increased. Unfortunately, a gridded network starts to have its first node to die before gridded networks having smaller grid size. For the selected experimental sensed area, the best grid size g is when ($80 \leq g \leq 150$) for a threshold between ($3 \leq \sigma \leq 6$) resulting a number of clusters between ($2 \leq C \leq 6$). Gridding is an effective tool used to enhance the mining clustering technique in WSNs flowing streaming data in terms of network lifetime. There has to be an agreement when choosing both grid size g and threshold σ to result an appropriate number of clusters C .

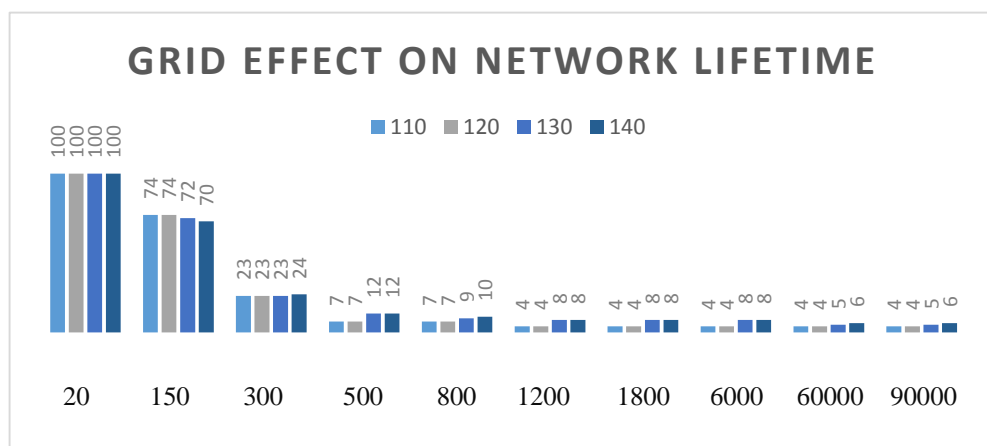


Figure 4.21 Number of Nodes to Die in a Certain Time when $g = 110, 120, 130, 140$

4.6 Conclusion

The developed DeGiCA is a built-from-scratch algorithm that is based on the density grid-based technique in clustered WSNs. This combination has proven its efficiency in reaching high network performance. It can find arbitrary shaped clusters with noise by applying the density technique and avoid clustering quality problems by discarding boundary points of a grid.

Based on the evaluation analysis, DeGiCA enhances clustering miming technique in WSNs streaming data streams in terms of network lifetime, energy consumption and packet delivery ration compared to its competitors resulting simulated performance metrics. By determining DeGiCA optimum parameter values, DeGiCA has proven strongly its ability to be scaled compared to its competitors.

The chapter presented also several points mentioned, as dataset stream packet description, displaying the data stream packet in some simulated experiments, the effect of gridding on a WSN running data streams.

Chapter V

Conclusions and Future Work

Chapter V

Conclusions and Future Work

5.1 Research Summary

A widespread use of WSNs have been found in several real life applications. They can be used in many fields such as environmental, industrial, military, and agriculture fields. A WSN suffers from several resource constraints, such as high computational power and limited energy source that affect its lifetime. WSNs depend hardly on their sensors that consumes a lot of battery. Unfortunately, the nature of WSNs makes it very difficult to recharge the sensor nodes batteries. Therefore, energy efficiency is an important objective design in WSN algorithms.

In some sensor network applications, data streams are processed by WSNs, they usually contain a large amount of datasets that flow rapidly in a very high speed and arrive in an online fashion. Data are unlimited and there is no control on the arrival order of the elements being processed. Thus, flowing streaming data consumes a large amount of energy that reduces its lifetime.

To solve WSN challenges listed above and achieve thesis objectives, this research developed an algorithm called Density Grid-base Clustering Algorithm (DeGiCA) that

enhances clustering mining technique in WSNs by developing a powerful combining between clustering technique and both density and grid techniques. Clustering algorithms are designed to achieve load-distribution among CHs, energy saving, high connectivity, and fault tolerance that guarantee a safe data stream transmission through the sensed media. In WSNs, clustering provides resource utilization and minimizes energy consumption by reducing number of nodes that take part in long distance transmission. Density technique can find arbitrary shaped clusters with noise, while grid technique is used to avoid clustering quality problems by discarding boundary points of grids. The density-grid technique combination enhances clustering mining technique during cluster formation process by eliminating unused spaces (i.e. empty spaces without sensor nodes) in a WSN area. The powerful combination enhances the WSN performance by focusing on used spaces and reducing data transmission, thus, consuming less battery of sensor nodes, saving overall network energy, extending network lifetime, that in turn guarantee data stream packet delivery.

The procedure used to build the developed DeGiCA is done in two main processes: *initialization process* and *rounds process*. The initialization process is divided into establishment phase, and CHs initial selection based on a node in each cluster having the nearest distance to the BS. The rounds process has the data transmission phase and CH-Election phase iterated rotationally through several rounds until the end of network lifetime. Initialization process is done once and permanently at the network deployment before running data streams and starting network lifetime during rounds process.

This contribution focuses mainly on the *initialization process* that distinguishes DeGiCA from other WSN clustering data stream mining algorithms. It enhance WSNs

data stream mining by the powerful combination of techniques. In brief, at the initialization process, a WSN is divided into equal size of grids (i.e. grid size is determined previously). Nodes then are distributed randomly inside grids. Each grid is classified based on its density to either high dense grid, low dense grid or empty grid. Density classification depends on comparing number of nodes in each grid with a special threshold. A grid having possibility to be a cluster center is selected and considered to be an active grid. Another possible grid is then selected to expand the active grid and so on, until all network nodes are included within their specific clusters. After this process, number of clusters is determined and available for other use. The final step in the *initialization process* is CHs initial selection based on nodes in each cluster having nearest distance to the BS, this is done separately for each cluster. After *initialization process* completion, DeGiCA moves to its *round process*, where CH election is done at each round depending on a node having the highest residual energy among all nodes in its cluster. At each round, data are streamed and phases starts its processes.

The DeGiCA helps to face limitations found in WSNs that stream data streams. By using a simulator implemented by MATLAB R2008b, compared to other clustering algorithms in WSNs that stream data streams (i.e. standard FCM and well-known K-means), simulation results conclude that the developed DeGiCA enhances the performance of a WSN by prolonging its network lifetime, reducing energy consumption, decreasing delay and providing better packet delivery ratio.

5.2 Conclusion of Results and Findings

Final obtained DeGiCA performance metrics results were compared with its competitor's results (i.e. FCM and K-means) with same system parameters and circumstances to evaluate the developed algorithm. DeGiCA and its competitors have run the same dataset streaming packet. DeGiCA is more complex than its commentators, due to the need to get number of clusters after cluster formation process; for comparison purposes. During simulation experiments, DeGiCA is run first to create clusters, thus, results number of clusters \mathcal{C} and their centers \mathbf{v} . \mathcal{C} and \mathbf{v} are used as inputs in both DeGiCA competitors. Then, both competitors are run individually to get their performance metrics results.

After running DeGiCA and its competitors, three different individual networks with same number of clusters are created, each with its own results of performance metrics. A comparison function is then used to compare between performance metrics results of the three competitors. Comparison is done between three main metrics, in terms of overall network lifetime, overall energy consumption for entire network, and lastly packet delivery ratio.

Some preliminary experiments are conducted to decide the optimum grid size \mathbf{g} , and optimum threshold σ . These optimum values give the best performance of the three algorithms in terms of energy consumption, network lifetime and packet delivery ratio. These values are then used to test DeGiCA evaluation and scalability.

5.2.1 DeGiCA Performance

Several simulation experiments were discussed and analyzed in chapter 4. This section presents the results obtained from the simulation analysis as follows:

- 1- For the selected experimental sensed area, it is found that the best grid size g is when ($80 \leq g \leq 150$) for a threshold between ($3 \leq \sigma \leq 6$) resulting a number of clusters ($2 \leq C \leq 6$). Gridding is an effective tool used to enhance the mining clustering technique in WSNs flowing streaming data in terms of network lifetime, energy consumption and packet delivery ratio. There has to be an agreement when choosing both grid size g and threshold σ to result an appropriate number of clusters C . Choosing a threshold ($3 \geq \sigma \geq 6$) in DeGiCA system parameters and circumstances results number of clusters ($2 > C \geq 8$) that may not be desired in some applications. Still even if number of obtained cluster are undesirable, DeGiCA outperforms its competitors.
- 2- After comparing DeGiCA with its competitor's performance metrics, it is found that DeGiCA enhances the well-known K-means and the standard FCM as well. The following presents the DeGiCA enhancement in average percent for each performance metrics in details:
 - a- The DeGiCA extends network lifetime by about 15% more than K-mean and by about 17% more than FCM. DeGiCA processes small grids, were all operations are performed on grid cells rather than processing the whole sensed area space and exhaustion the network as found in K-means and FCM.
 - b- The DeGiCA reduces energy consumption by about 13% less than K-means and by about 11% less than FCM.

- c- The DeGiCA enhances packet delivery ratio, it delivers more packets by about 40% than K-mean and by about 70% than FCM. The high enhancement in packet delivery ratio is due to that DeGiCA is built especially for WSNs environment involving data stream packets that guarantees data packet delivery through this media, while both competitors are applied in WSNs and could stream datastreams
- 3- By increasing number on nodes $n = 100$ to $n = 200$ and using the optimum values of grid size when $g = 140$ and threshold $\sigma = 4$, a sensed area equivalent to $(1000 \times 1000) m^2$ existing between coordinators $(0,0)$ to $(1000, 1000)$, it is found that the developed DeGiCA has the ability to be scalable in terms of network lifetime, energy consumption and packet delivery ratio. Simulation results prove that the performance of DeGiCA outperforms K-Means in terms of network lifetime by 16%, energy consumption by 18% and packet delivery ratio by 16%. DeGiCA also outperforms FCM in terms of network lifetime by 16%, energy consumption by 12% and packet delivery ratio by 22%.

5.2.2 Application Areas of DeGiCA

The combination of density and grid in streaming WSN algorithms is rare and developing a density grid-based algorithm (i.e. DeGiCA) in this field is a good initiative. Since the obtained experimental results proved DeGiCA efficiency, it can be applied in real life applications. Several application areas could apply DeGiCA to achieve better results in extending network lifetime, reducing energy consumption thus less node batteries damage, and better data packet delivery ratio. Applying DeGiCA

requires hardware issues that was missed in this research (i.e. sensors and different requirement) to create a network. The following are some examples of real-world application areas in WSNs that can apply DeGiCA:

- 1- DeGiCA can be applied during environmental monitoring such as smart buildings, or harsh environments as forests.
- 2- DeGiCA can be used in health monitoring, where patients can be equipped with small sensors to monitor their health or behavior such as monitoring patient's heart beating, breathing, etc.
- 3- DeGiCA can be used in physical uses like tracking a specific object movements, information or even a device. Such as tracking moon light or weather humidity.
- 4- DeGiCA may be used for discovering data patterns in a sensor network for a certain application, where it can be used for data pattern extracting.
- 5- DeGiCA could be used in monitoring road traffics, where it has the ability to read high speed data streams when an event of several devices occur, it may record speeds, number of devices, etc.

5.3 Thesis Future Work

As future work, some suggestions can extend this work such as:

- 1- Applying hard clustering on the developed DeGiCA rather than the fuzzy soft clustering. Form the simulation results obtained at the final comparisons, it is found that K-means algorithm could provide better results than FCM algorithm in terms of network lifetime and packet delivery ratio.

- 2- Deeply applying scalability on the developed DeGiCA by configuring, testing, and modifying the algorithm to get better results. Especially when applying it on heavy, high dense and large scale WSNs.
- 3- Node mobility and BS mobility are considered to be an important point of study and a hot research area to be applied on the DeGiCA.
- 4- Provide a mathematical model to describe propagation model, traffic model, energy model and optimize formation of clusters. Willing to implement a more intelligent DeGiCA.
- 5- Complexity analysis of developed algorithm to show it is applicable to resource constraint WSNs by specifying MAC clocks and evaluating end to end delay.
- 6- Considering transmission range and synchronization between nodes.
- 7- Considering changing BS location and duty cycle variation of nodes.

LIST OF REFERENCES

- [1] O. Younis, M. Krunz, and S. Ramasubramanian, "Node clustering in wireless sensor networks: recent developments and deployment challenges," *Network, IEEE*, vol. 20, pp. 20-25, 2006.
- [2] S. Bandyopadhyay and E. J. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, 2003, pp. 1713-1723.
- [3] J.-L. Chen, M.-C. Chen, P.-Y. Huang, and Y.-C. Chang, "Cluster-grid structure routing protocol for sensor mobility management," in *Sarnoff Symposium, 2007 IEEE*, 2007, pp. 1-5.
- [4] S. Ali and S. A. Madani, "Distributed grid based robust clustering protocol for mobile sensor networks," *Int. Arab J. Inf. Technol.*, vol. 8, pp. 414-421, 2011.
- [5] A. L. de Aquino, C. M. Figueiredo, and E. F. Nakamura, "Data Stream Algorithms For Processing of Wireless Sensor Network Application Data."
- [6] Yang, Cheng-Lung, et al. "A security mechanism for clustered wireless sensor networks based on elliptic curve cryptography." *Proceedings of IEEE SMC–eNewsletter*,(33). Retrieved from http://www.my-smc.org/news/back/2010_12/main_article3.html (2010).
- [7] A. L. de Aquino, C. M. S. Figueiredo, E. F. Nakamura, L. S. Buriol, A. Loureiro, A. O. Fernandes, *et al.*, "A sampling data stream algorithm for wireless sensor networks," in *Communications, 2007. ICC'07. IEEE International Conference on*, 2007, pp. 3207-3212.
- [8] O. Boyinbode, H. Le, A. Mbogho, M. Takizawa, and R. Poliah, "A Survey on Clustering Algorithms for Wireless Sensor Networks," in *2010 13th International Conference on Network-Based Information Systems*, 2010, pp. 358-364.
- [9] M. Abdullah, H. N. Eldin, T. Al-Moshadak, R. Alshaik, and I. Al-Anesi, "Density Grid-Based Clustering for Wireless Sensors Networks," *Procedia Computer Science*, vol. 65, pp. 35-47, 2015.
- [10] X. Liu, "A survey on clustering routing protocols in wireless sensor networks," *sensors*, vol. 12, pp. 11113-11153, 2012.
- [11] F. L. Lewis, "Wireless sensor networks," *Smart environments: technologies, protocols, and applications*, pp. 11-46, 2004.

- [12] E. Soroush, K. Wu, and J. Pei, "Fast and quality-guaranteed data streaming in resource-constrained sensor networks," in *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, 2008, pp. 391-400.
- [13] A. L. De Aquino, C. Figueiredo, E. F. Nakamura, L. S. Buriol, A. A. Loureiro, A. O. Fernandes, *et al.*, "Data stream based algorithms for wireless sensor network applications," in *Advanced Information Networking and Applications, 2007. AINA'07. 21st International Conference on*, 2007, pp. 869-876.
- [14] R. Mitra and D. Nandy, "A survey on clustering techniques for wireless sensor network," *International Journal of Research in Computer Science*, vol. 2, p. 51, 2012.
- [15] C. C. Aggarwal, *Data streams: models and algorithms* vol. 31: Springer Science & Business Media, 2007.
- [16] A. Mahmood, K. Shi, S. Khatoon, and M. Xiao, "Data mining techniques for wireless sensor networks: A survey," *International Journal of Distributed Sensor Networks*, vol. 2013, 2013.
- [17] Y. Chen and L. Tu, "Density-based clustering for real-time stream data," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2007, pp. 133-142.
- [18] L. Su, H.-y. Liu, and Z.-H. Song, "A new classification algorithm for data stream," *International Journal of Modern Education and Computer Science (IJMECS)*, vol. 3, p. 32, 2011.
- [19] S. Chakrabarti, M. Ester, U. Fayyad, J. Gehrke, J. Han, S. Morishita, *et al.*, "Data mining curriculum: A proposal (Version 1.0)," Intensive Working Group of ACM SIGKDD Curriculum Committee, 2006.
- [20] M. Bharati and M. Ramageri, "Data mining techniques and applications," 2010.
- [21] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI magazine*, vol. 17, p. 37, 1996.
- [22] P. Domingos and G. Hulten, "Mining High-Speed Data Streams," 2000.
- [23] D. Dechene, A. El Jardali, M. Luccini, and A. Sauer, "A Survey of Clustering Algorithms for Wireless Sensor Networks."
- [24] B. Mamalis, D. Gavalas, C. Konstantopoulos, and G. Pantziou, "Clustering in wireless sensor networks," *RFID and Sensor Networks: Architectures, Protocols, Security and Integrations*, Y. Zhang, LT Yang, J. Chen, *eds*, pp. 324-353, 2009.
- [25] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer communications*, vol. 30, pp. 2826-2841, 2007.

- [26] Y. Alghamdi and M. Abdullah, "Classification for data stream clustering protocols in wireless sensor networks," in *Communication, Management and Information Technology: Proceedings of the International Conference on Communication, Management and Information Technology (Iccmit 2016)*, 2016, pp. 671-680.
- [27] J. Pan and R. Jain, "A survey of network simulation tools: Current status and future developments," *Email: jp10@cse.wustl.edu*, 2008.
- [28] Goering, Richard (4 October 2004). "Matlab edges closer to electronic design automation world". EE Times.
- [29] V. Kumar, S. Jain, and S. Tiwari, "Energy efficient clustering algorithms in wireless sensor networks: A survey," 2011.
- [30] C. Jia, C. Tan, and A. Yong, "A grid and density-based clustering algorithm for processing data stream," in *Genetic and Evolutionary Computing, 2008. WGEC'08. Second International Conference on*, 2008, pp. 517-521.
- [31] F. Cao, M. Ester, W. Qian, and A. Zhou, "Density-Based Clustering over an Evolving Data Stream with Noise," in *SDM*, 2006, pp. 328-339.
- [32] A. Amini, H. Saboohi, and T. Y. Wah, "A multi density-based clustering algorithm for data stream with noise," in *Data Mining Workshops (ICDMW), 2013 IEEE 13th International Conference on*, 2013, pp. 1105-1112.
- [33] L. Wan, W. K. Ng, X. H. Dang, P. S. Yu, and K. Zhang, "Density-based clustering of data streams at multiple resolutions," *ACM Transactions on Knowledge discovery from Data (TKDD)*, vol. 3, p. 14, 2009.
- [34] J. Yin and M. M. Gaber, "Clustering distributed time series in sensor networks," in *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, 2008, pp. 678-687.
- [35] H. Sabit, A. Al-Anbuky, and H. Gholam-Hosseini, "Distributed WSN data stream mining based on fuzzy clustering," in *Ubiquitous, Autonomic and Trusted Computing, 2009. UIC-ATC'09. Symposia and Workshops on*, 2009, pp. 395-400.
- [36] J. R. Diaz, J. Lloret, J. M. Jimenez, and J. J. Rodrigues, "A QoS-based wireless multimedia sensor cluster protocol," *International Journal of Distributed Sensor Networks*, vol. 2014, 2014.
- [37] M. Abazeed, N. Faisal, S. Zubair, and A. Ali, "Routing protocols for wireless multimedia sensor network: a survey," *Journal of Sensors*, vol. 2013, 2013.
- [38] D. A. S. Juned M. Khan, "PERFORMANCE COMPARISON OF FCM AND K-MEAN CLUSTERING TECHNIQUE FOR WIRELESS SENSOR NETWORK IN TERMS OF COMMUNICATION OVERHEAD," *Global Journal of Advanced Engineering Technologies and Science*, pp. 26 - 29, May, 2014.

- [39] H. Sabit and A. Al-Anbuky, "Multivariate spatial condition mapping using subtractive fuzzy cluster means," *Sensors*, vol. 14, pp. 18960-18981, 2014.
- [40] J. Huang and J. Zhang, "Distributed Dual Cluster Algorithm based on FCM for Sensor Streams," *Advances in Information Sciences and Service Sciences*, vol. 3, pp. 201-209, 2011.
- [41] J. Huang and J. Zhang, "Fuzzy C-means clustering algorithm with spatial constraints for distributed WSN data stream," *International Journal of Advancements in Computing Technology*, vol. 3, pp. 165-175, 2011.
- [42] M. Khalilian and N. Mustapha, "Data stream clustering: Challenges and issues," *arXiv preprint arXiv:1006.5261*, 2010.
- [43] R. G. Akl and U. Sawant, "Grid-based coordinated routing in wireless sensor networks," 2007.
- [44] J. Huang and J. Zhang, "Rough Set Based Generalized Fuzzy C-Means clustering Algorithm with Spatial Constraints for Distributed WSN Data Stream," *Journal of Convergence Information Technology*, vol. 6, 2011.
- [45] O. M. Alia, "A decentralized fuzzy C-means-based energy-efficient routing protocol for wireless sensor networks," *The Scientific World Journal*, vol. 2014, 2014.
- [46] S. Zhong, G. Wang, X. Leng, X. Wang, L. Xue, and Y. Gu, "A Low Energy Consumption Clustering Routing Protocol Based on K-Means," *Journal of Software Engineering and Applications*, vol. 5, p. 1013, 2012.
- [47] Malik, A. S., & Qureshi, S. A. (2016). Analyzing the factors affecting network lifetime for cluster-based wireless sensor networks. *Pakistan Journal of Engineering and Applied Sciences*.
- [48] A. Thangavelu and A. Pathak, "Clustering Techniques to Analyze Communication Overhead in Wireless Sensor Network," *Editorial Committees*, p. 75.
- [49] M. Keerthi and D. B. S. Babu, "An Improved FCM's Clustering Protocol for Wireless Sensor Networks," in *International Conference on Information Technology, Electronics and Communications (ICITEC-2012)*, 2012, pp. 140-143.
- [50] L. Qin, "Research on Fuzzy Clustering Algorithm in Wireless Sensor Network," *5th International Conference on Education, Management, Information and Medicine (EMIM 2015) - Published by Atlantis Press*, pp. 503 - 506, 2015.

APPENDICES

APPENDIX A

Fuzzy C-Means and K-Means Algorithms

APPENDIX A

A.1 Overview on K-means and Fuzzy C-means (FCM)

This section provides a brief overview on two main clustering algorithms that are used in this research as DeGiCA competitors, the K-Means and FCM.

A.1.1 K-means Clustering Algorithm

The simplest algorithm that solve a well-known clustering problem is called the K-means clustering algorithm. K-means has an efficient CH selection method to maximize energy efficiency of a WSN. K-means is based on finding a CH that minimizes the sum of Euclidean distances between CH and nodes [38, 48]. It reduces communication overhead, energy consumption and extends network lifetime. It is used to partition a sensed area into K clusters. The procedure follows a simple way to classify a given dataset through a certain number of clusters fixed a priori [48]. In K-means, there is a distance threshold called R for calculating distance between CH and BS. If their distance is less than R , they use a single-hop transmission, otherwise, they use a multiple-hops transmission [46]. There is also an energy threshold called E for all CHs. If CH energy is less than E , then CH broadcasts a quit message to all nodes inside the cluster. Hence, other nodes which have higher residual energy are elected to become CHs [46]. Nodes near boundary region in K-means are affected since, the degree of belongingness is described in terms of either zero or one. For this reason, K-means clustering is called hard clustering. Edge nodes may have the same degree of belongingness to more than one clusters. In K-means, there is an optimal cluster

formation. Nodes are assigned to a cluster based on the degree of belongingness when network area deployment. Degree of belongingness needs to be computed in each round for every node inside a cluster [49]. Obviously, the major limitation of K-means clustering algorithm is predetermining parameter k [39].

A.1.2 Fuzzy Clustering-Means Algorithm (FCM)

Fuzzy C-Means algorithm (FCM) is considered to be good solutions to improve network lifetime. FCM was developed by Dunn and later improved by Bezdek [39, 45]. It is used in cluster analysis, image processing, pattern recognition, and so on. In WSNs, FCM assigns each node to a cluster with a degree of membership [45]. As mentioned previously in K-means, data is divided into distinct clusters, where each node belongs to exactly one cluster, this is called hard clustering. In fuzzy clustering, nodes are allowed to belong to several clusters at the same time. It is done with different degrees of membership. In many cases, fuzzy clustering is more natural than hard clustering. In soft clustering, nodes on boundaries between several clusters are not forced to fully belong to one cluster, but rather assigned to a membership degrees between 0 and 1 that indicates partial membership [38, 39, 48] . Figure A.1, shows types of clustering based on nodes membership degree.

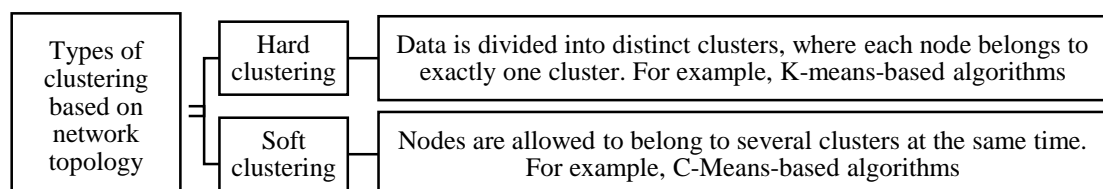


Figure A.1 Types of Clustering Based on Network Topology

In real applications, there is no sharp boundary between neighboring clusters. This makes soft fuzzy clustering the most suitable algorithm for data streams. The most common fuzzy clustering algorithm is FCM, a fuzzification of K-means [39]. In FCM, the degree of belongingness is given within the range between [0, 1]. Each sensor nodes computes its degree of belongingness in terms of Euclidean distance [39, 49]. The Euclidean distance is used to compute distance between sensor nodes and CH as shown in equation (2.1).

The symbol “ $\| \cdot \|$ ” denotes the Euclidean distance, it defines distance between two points $p(p_1, p_2, \dots, p_m)$ and $q(q_1, q_2, \dots, q_m)$. p and q are two points in Euclidean m -space, then the distance d from p to q , or from q to p is given by

$$d(p, q) = d(q, p) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \dots + (q_m - p_m)^2} \quad (2.1)$$

Generally, FCM algorithms have a function called objective function. It is used to shorten the distance between sensor nodes to CH [49]. Specifically, FCM is an iterative clustering method [39] that generates an optimal c partition by minimizing weighted within group sum of squared error objective function J_{FCM} as shown in equation (2.2).

$$J_{FCM} = \sum_{i=1}^n \sum_{j=1}^c (u_{ij})^m d^2(x_i, v_j), \quad (2.2)$$

Where $X = \{x_1, x_2 \dots x_N\} \subseteq R^O$ is a dataset, n is number of dataset, $2 \leq c < N$ is number of clusters, u_{ij} is degree of membership of x_i in the i^{th} cluster, m is the weighting exponent on each fuzzy membership, v_j is center of cluster, $d^2(x_i, v_j)$ is a distance measure between x_i and v_j . This is shown in equation (2.3)

$$c_j^{(b+1)} = \frac{\sum_{i=1}^n u_{ij} x_i}{\sum_{i=1}^n u_{ij}} \quad 1 \leq j \leq C \quad (2.3)$$

giving, Figure A.2 provides the traditional FCM pseudo code [39].

Algorithm of the Traditional FCM		
Inputs:	{ $c_1, c_2, c_3 \dots c_C$ }	{ $u_1, u_2, u_3 \dots u_i$ }
Outputs:	$u_i, i = 1, 2, 3, \dots, C$	$U_{ij}, i = 1, 2, 3, \dots, C, j = 1, 2, 3, \dots, n$
Initialize:	ε, θ, A	

1. For, $l = 1, 2, 3, \dots$ Repeat
2. Compute cluster centers (prototypes):
3.
$$v_i = \frac{\sum_{i=1}^n (v_{ci})^\theta u_i}{\sum_{i=1}^n (v_{ci})^\theta}; \quad 1 \leq i \leq c$$
4. Compute distances
5. $d_{c1A}^2 = (u_i - v_i)^T A (u_i - v_i), 1 \leq c \leq C, 1 \leq i \leq n$
6. Update the partition matrix:
7. For $1 \leq i \leq n$
8. If $d_{ciA} > 0$ for all $c = 1, 2, \dots, C$
9.
$$v_{ic} = \frac{1}{\sum_{j=1}^C (d_{ciA} / d_{cjA})^{\frac{2}{\theta-1}}}$$
10. Else
11. $v_{ci} = 0$ if $d_{ci} > 0$ and $v_{ci} \in [0, 1] \sum_{i=1}^C v_{ci} = 1$
12. Until $\|v_{ci}^l - v_{ci}^{(l-1)}\| < \varepsilon$

Figure A.2 Pseudo Code of the Traditional FCM [39]

Similar to K-means algorithm, a main drawback of FCM is the predetermination of clusters number within the data space [39]. Moreover, computing the distance between each sensor nodes to other CHs causing time-consuming, thus effecting the execution time and efficiency of clustering process [49]. Figure A.3 presents the traditional FCM flowchart [45].

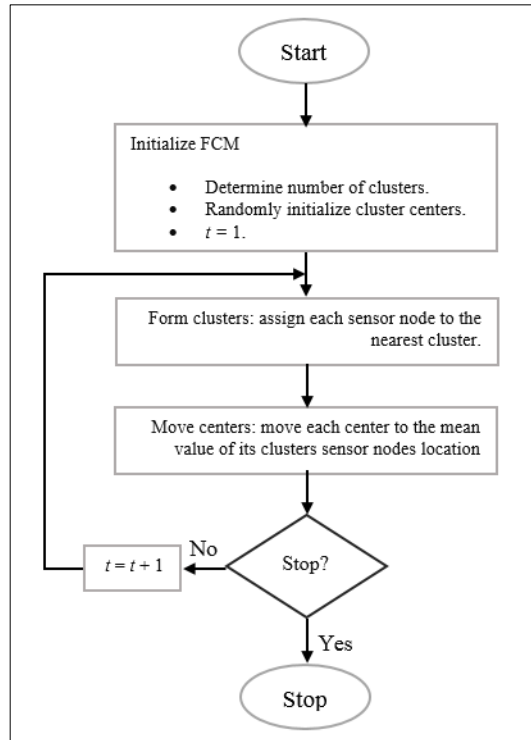


Figure A.3 Traditional (Standard) FCM Flowchart [45]

A.2 Overview of FCM-Based Clustering Algorithms

This section provides some FCM-based clustering algorithms in WSNa that streams data streams packets

A.2.1 Subtractive Fuzzy Cluster Means (SUBFCM)

In some applications where number of clusters in a dataset must be previously known, FCM algorithm cannot be used directly. In WSN clustering, number of clusters has to be determined from the datasets. Hence, the subtractive clustering and FCM algorithms are combined to generate an algorithm that determines number of clusters.

This combination is called Subtractive Fuzzy Cluster Means (SUBFCM). The purpose of using SUBFCM in distributed clustering of WSN data stream, is to reduce the total data transmission [41]. The subtractive clustering is an extension of a method called mountain clustering method proposed by R.Yager [40]. SUBFCM goes through the following steps:

- 1- Selecting nodes with highest potential to be the first cluster center.
- 2- Removing all nodes near the first cluster center (as determined by radii), in order to determine the next cluster and its center location.
- 3- Repeating until all nodes is within radii of a cluster center. After that, number of clusters centers is taken [40].

A.2.2 Fuzzy C-Mean Clustering of Particle Swarm Optimization (CAFCPSO)

A clustering algorithm based on FCM of Particle Swarm Optimization (CAFCPSO) has been proposed. The CAFCPSO is a combination between the particle swarm optimization algorithm and the Fuzzy C-Means clustering algorithm [50].

APPENDIX B

Experimental Results Snapshots

Appendix B

This section presents the simulation experimental graphs mentioned in chapter 4.

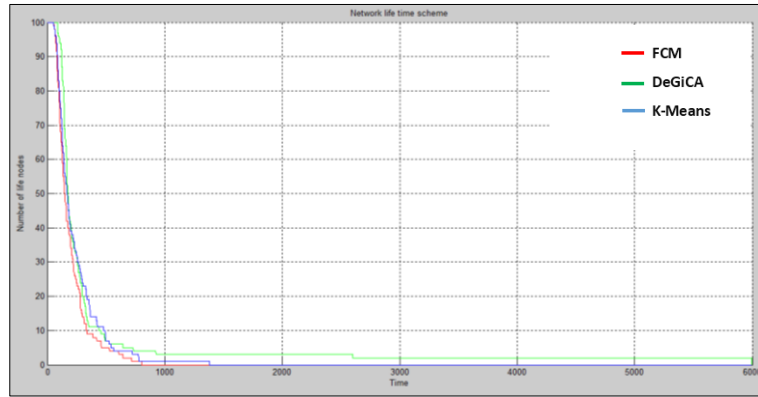


Figure B.1 DeGiCA, FCM and K-means Network Lifetimes when $g = 130$ and $\sigma = 5$

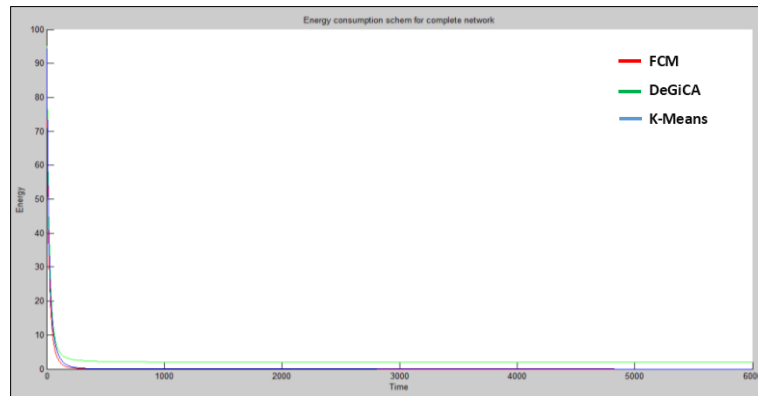


Figure B.2 DeGiCA, FCM and K-means Energy Consumption when $g = 130$ and $\sigma = 5$

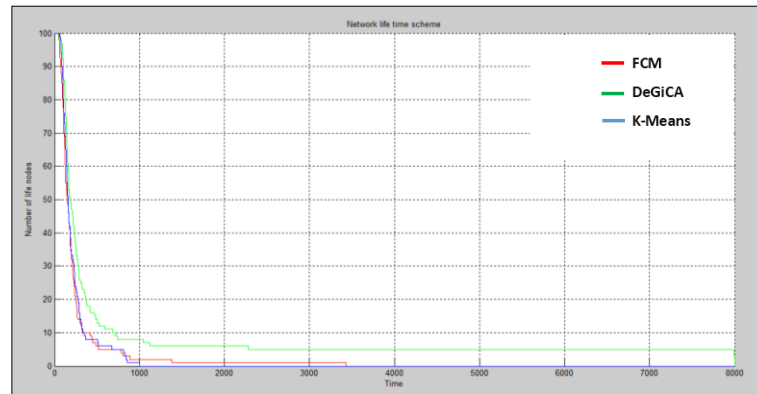


Figure B.3 DeGiCA, FCM and K-Means Network Lifetimes when $g = 155$ and $\sigma = 5$

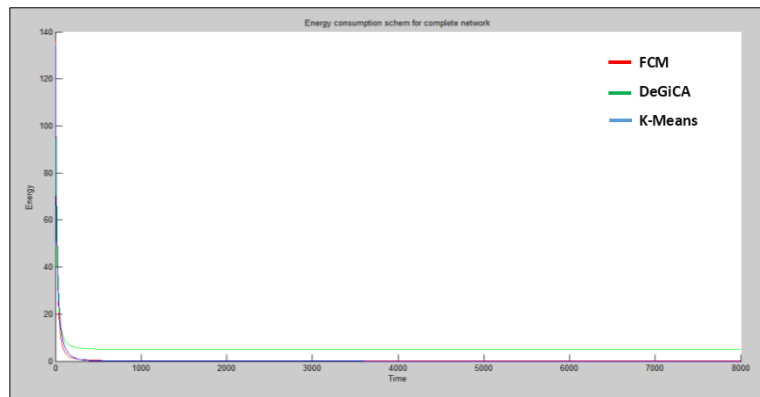


Figure B.4 DeGiCA, FCM and K-means Energy Consumption when $g = 155$ and $\sigma = 5$

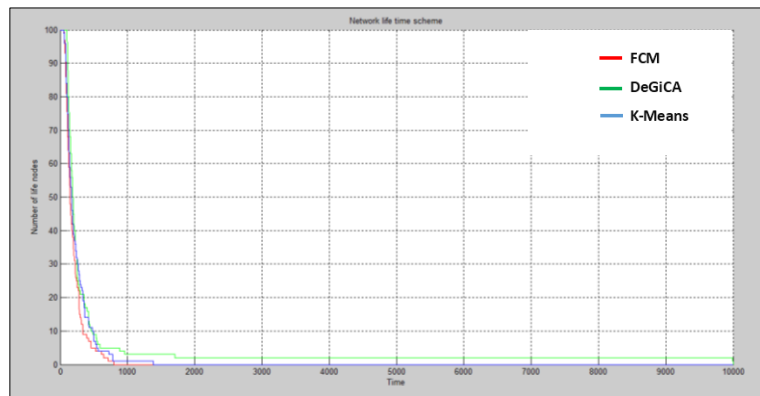


Figure B.5 DeGiCA, FCM and K-means Network Lifetimes when $g = 140$ and $\sigma = 6$

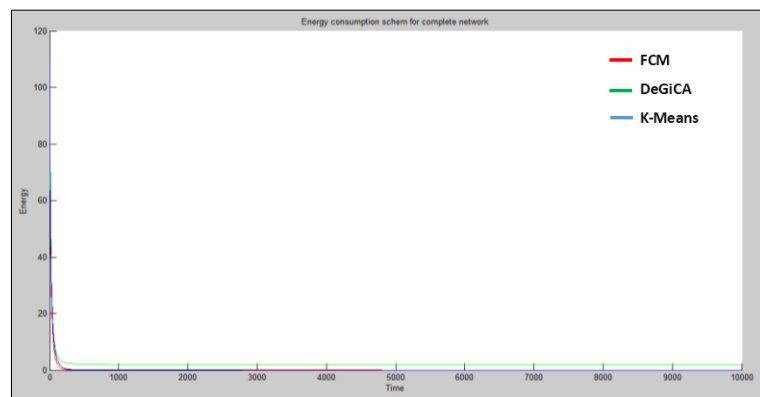


Figure B.6 DeGiCA, FCM and K-means Energy Consumption when $g = 140$ and $\sigma = 6$

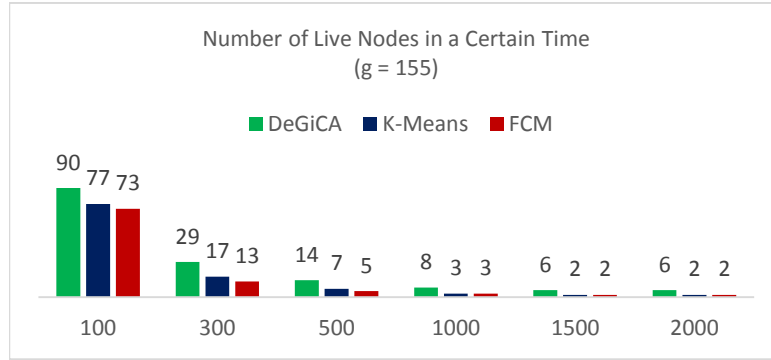


Figure B.7 Number of Live Nodes when $g = 155$ and $\sigma = 5$ in a Certain Time

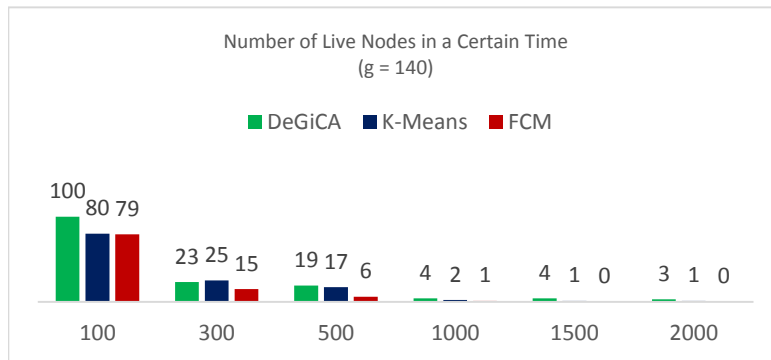


Figure B.8 Number of Live Nodes when $g = 140$ and $\sigma = 6$ in a Certain Time

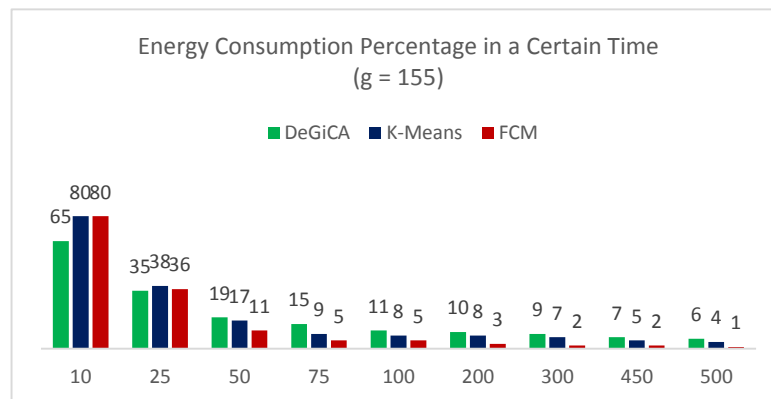


Figure B.9 Energy Consumption Percentage when $g = 155$ and $\sigma = 5$ in a Certain Time

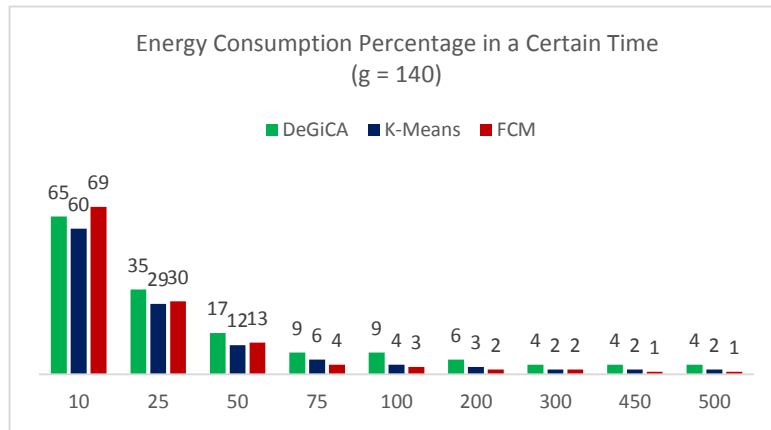


Figure B.10 Energy Consumption Percentage when $g = 140$ and $\sigma = 6$ in a Certain Time

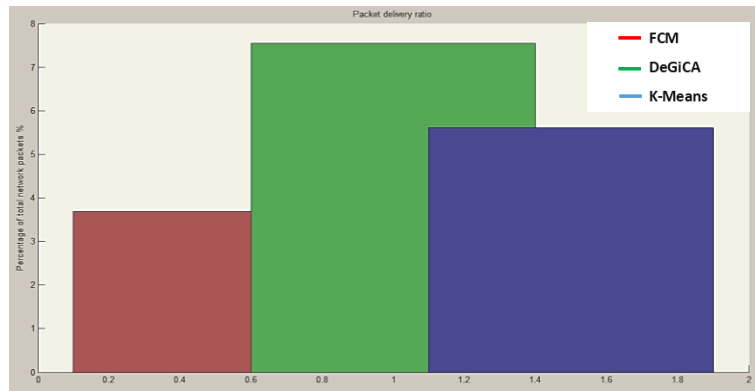


Figure B.11 Percentage of Overall Delivered Packets when $g = 110$ and $\sigma = 4$

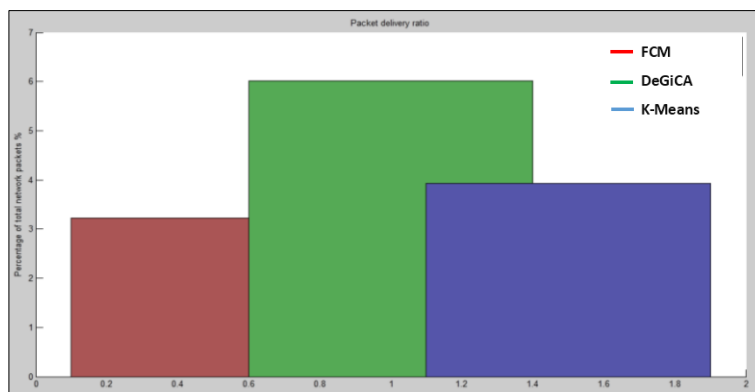


Figure B.12 Percentage of Overall Delivered Packets when $g = 140$ and $\sigma = 6$

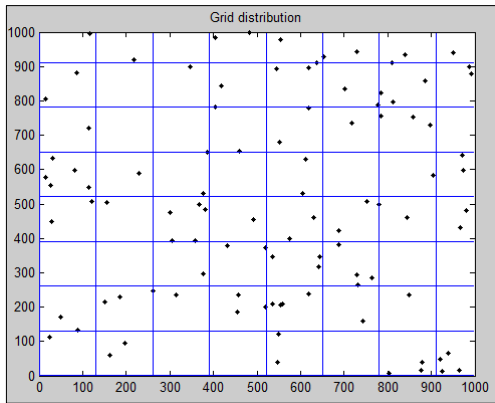


Figure B.13 Gridded WSN when $g = 130$

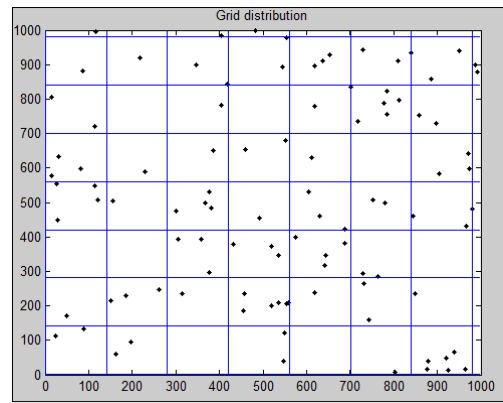


Figure B.14 Gridded WSN when $g = 140$

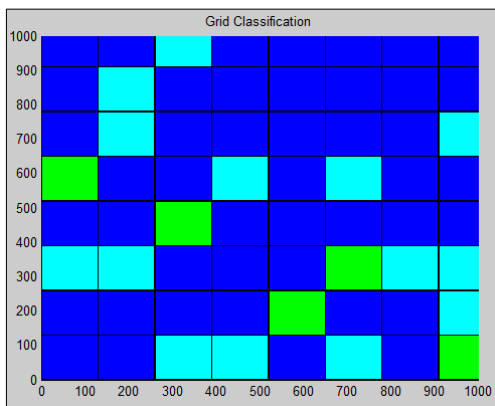


Figure B.15 Grid Classification when $g = 130$

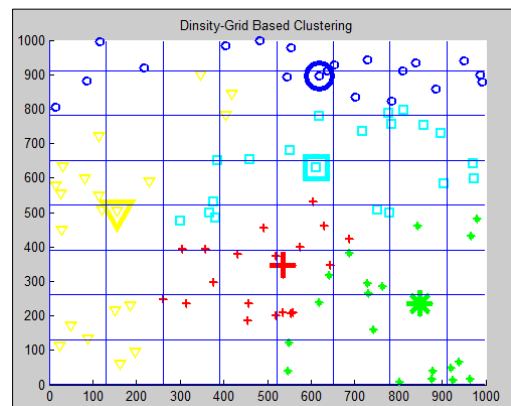


Figure B.16 Cluster Formation when $g = 130$

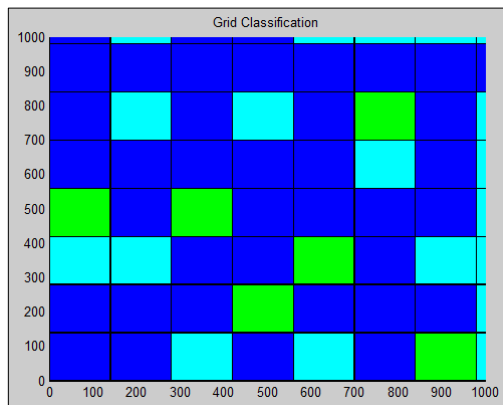


Figure B.17 Grid Classification when $g = 140$

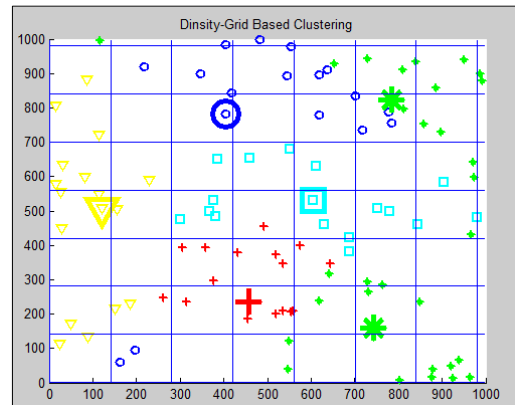


Figure B.18 Cluster Formation when $g = 140$

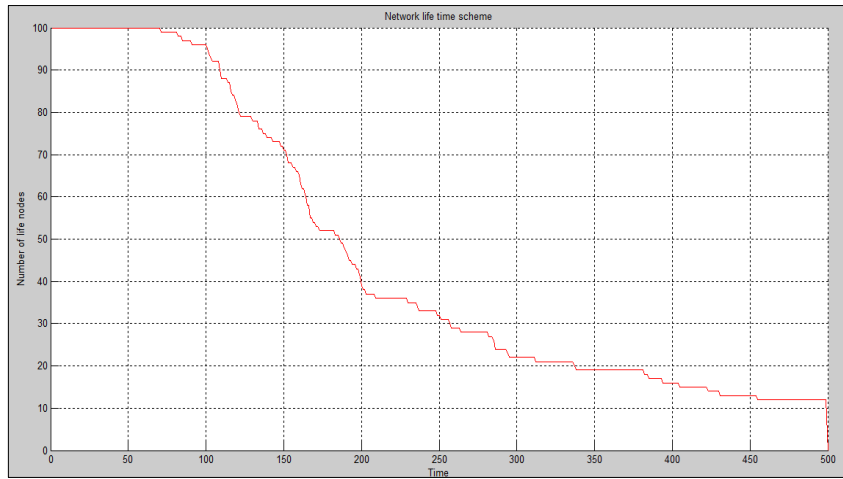


Figure B.19 Death of First Node when $g = 130$

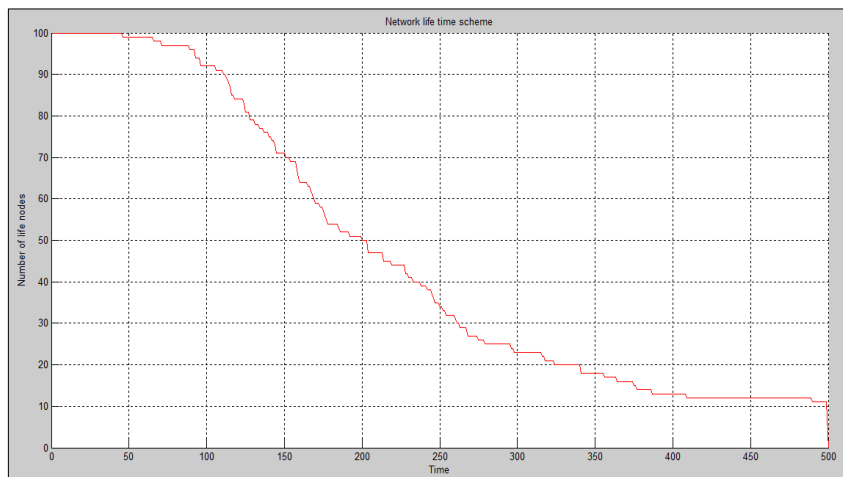


Figure B.20 Death of First Node when $g = 140$

PUBLISHED PAPERS

1. Y. Alghamdi and M. Abdullah, "Classification for data stream clustering protocols in wireless sensor networks," in Communication, Management and Information Technology: Proceedings of the International Conference on Communication, Management and Information Technology (Iccmit 2016), 2016, pp. 671-680.
2. Manal Abdullah and Yassmeen Alghamdi "Streaming Data Classification in Clustered WSNs "Book Chapter in the Book:
Shaping the Future of ICT: Trends in Information Technology, Communications Engineering, and Management, Taylor & Francis Publisher. Accepted and under editing.
3. Yassmeen Alghamdi and Manal Abdullah “Enhancing Fuzzy C-Means by Grid-Density Clustering for Distributed WSN Data Stream” in International Conference on Innovative Research in Engineering and Science (IRES 2017) – Accepted.
4. Yasmen Alghamdi and Manal Abdullah "Improving K-Means Algorithm by Grid-Density Clustering for Distributed Data Stream WSNs" JSN (Journal of Sensor Networks) ISSN 2160-3545 (print), 2160-3537 (online) New World Publishing, under review

5. Yasmen Alghamdi and Manal Abdullah " DeGiCA: Density Grid-based Clustering Algorithms for Distributed Data Stream WSNs" Journal of Computer Science and Technology, Under Review

تعزير تنقيب البيانات المتدفقة في شبكات الاستشعار اللاسلكية باستخدام خوارزميات التجميع

إعداد
ياسمين سند أحمد الغامدي

بحث مقدم لنيل درجة الماجستير في علوم الحاسبات

إشراف
د. منال عبدالعزيز عبدالله

كلية الحاسبات وتقنية المعلومات
جامعة الملك عبدالعزيز
جدة - المملكة العربية السعودية
شعبان 1438 هـ - مايو 2017م

تعزيز تنقيب البيانات المتدفقة في شبكات الاستشعار اللاسلكية باستخدام خوارزميات التجميع

ياسمين سند الغامدي

المستخلص

شهدت السنوات القليلة الماضية اهتماماً متزايداً في استخدام شبكات الاستشعار اللاسلكية المختصرة بـ (WSNs) في مجموعة واسعة من التطبيقات ، حيث أصبح هذا النوع من الشبكات مجالاً ساخناً للبحث ، وبسبب التقدم والنمو في مجال تكنولوجيا الاتصالات اللاسلكية ، فقد أصبحت شبكات الاستشعار اللاسلكية مرغوبة على نحو متزايد للعديد من التطبيقات في مجالات مختلفة ، كمجال المراقبة العسكرية ، والكشف عن الحرائق ، ومراقبة المساكن الطبيعية ، وفي الصناعة ، والمراقبة الصحية وغيرها الكثير .

تتكون شبكات الاستشعار اللاسلكية من عقد استشعار فردية صغيرة قادرة على التفاعل مع بيئاتها بواسطة ميزة الاستشعار عن بعد والسيطرة على البارامترات المادية ، لكن هذه العقد

الصغيرة تعاني من قصور مصادر الطاقة الكائنة بداخلها مما قد يؤدي إلى تقصير عمر شبكة الاستشعار اللاسلكية بشكل عام .

تميل هذه الشبكات إلى توليد كمية كبيرة من البيانات الصغيرة المتتابعة والنابعة من عدة عقد استشعار تسمى بالبيانات المتدفقة ، تتدفق البيانات المتدفقة بسرعة عالية جداً عبر الإنترنت ، وتتميز الحزمة منها بأنها ضخمة وغير محدودة وقد يتم فقد السيطرة على معالجتها حسب ترتيب وصولها .

وبسبب القيود السابقة الذكر لشبكات الاستشعار اللاسلكية ، بالإضافة إلى سرعة وصول البيانات المتدفقة وضخامة حجمها ، توجد هناك حاجة ماسة لحل التحديات التي قد تعيق عمل شبكات الاستشعار اللاسلكية ، وتكمن هذه التحديات في محاولة إطالة عمر شبكة الاستشعار اللاسلكية ، كما تكمن في محاولة خفض استهلاك طاقة عقد الاستشعار ، وأخيراً تكمن في الحد من التأخير الذي يحدث نتيجة لبعض العمليات ولحركة البيانات داخل هذا النوع من الشبكات . تقنية التنقيب عن البيانات هي تقنية يمكن أن تستخدم لمواجهة ومعالجة التحديات المذكورة بشبكات الاستشعار اللاسلكية ، وتشمل هذه التقنية عدة أنواع منها التجميع ، والتصنيف ، والعد المتكرر ، وتحليل السلاسل الزمنية وغيرها .

كما أثبتت الأبحاث أن خوارزميات التجميع تلعب دوراً هاماً في تنظيم شبكات الاستشعار اللاسلكية وحل التحديات المذكورة سابقاً . بناء عليه سيقوم البحث بتطوير ومحاكاة خوارزمية باستخدام إحدى تقنيات التجميع وتسمى هذه الخوارزمية بخوارزمية التجميع الكثافي الشبكي المختصرة — (DeGiCA) . هذه الخوارزمية تعزز من أداء تقنية التجميع في شبكات الاستشعار اللاسلكية وذلك من خلال الجمع بين ثلاث تقنيات هي : تقنية التجميع وتقنية الكثافة وتقنية الشبكة .

تتميز تقنية الكثافة باستطاعتها استخلاص الأشكال التقديرية للتجميع ، أما تقنية الشبكة فبإمكانها إزالة العقد من جوانب الشبكة والتي وجودها قد يقلل من جودة التجميع ، وبذلك تكون خوارزمية الـ (DeGiCA) قادرة على مواجهة التحديات أثناء نقلها للبيانات المتدفقة .

وباستخدام برنامج النمذجة والمحاكاة المعروف بالـ (MATLAB) ، وبالمقارنة مع خوارزميات أخرى تعالج نفس التحديات المذكورة أعلاه بشبكات الاستشعار اللاسلكية ، تمت مقارنة نتائج خوارزمية (DeGiCA) بنتائج خوارزميتين أخريتين تعرفان باسم (K-means) و (FCM) ، كلتا الخوارزميتين تستخدم تقنية التجميع في شبكات الاستشعار اللاسلكية والتي تنقل البيانات المتدفقة .

على وجه الخصوص ، تتفوق خوارزمية الـ (DeGiCA) على خوارزمية الـ (K-Means) من حيث إطالة عمر الشبكة بنسبة 15% ، ونسبة 13% من حيث الحفاظ على الطاقة ، ونسبة 40% من حيث وصول حزم البيانات المتدفقة. من جهة أخرى فإن خوارزمية الـ (DeGiCA) تتفوق على خوارزمية الـ (FCM) من حيث إطالة عمر الشبكة بنسبة 17% ، ونسبة 11% من حيث الحفاظ على الطاقة ، ونسبة 70% من حيث وصول حزم البيانات المتدفقة.