

Index

Acknowledgement.....	V
Abstract.....	VI
Contents.....	XIII
Abbreviations.....	XVI
List of Figures.....	XVII
List of Tables.....	XIX
Chapter 1 Introduction.....	01
1.1 Internet of Things (IoT) in Health Care.....	01
1.2 Cloud Computing in Healthcare.....	03
1.3 Integration of IoT and Cloud Computing	05
1.4 Fog Computing in Health care.....	06
1.4.1 Overview of Fog Computing.....	06
1.4.2 Fog Computing in Healthcare domain.....	07
1.5 Motivation.....	09
1.6 Aims and Objectives.....	09
1.7 Proposed methodology.....	10
1.8 Overview of Research.....	12
1.9 Thesis Organization.....	20
Chapter 2 Literature Review	21
2.1 Health Care in Fog Computing.....	21
2.1.1 Research Gap.....	23
2.2 Optimization in Fog Computing.....	23
2.2.1 Research Gap.....	27
2.3 Need of Lightweight encryption and multi-level Fog scenario.....	27
2.3.1 Research Gap.....	33
2.4 Problem definition.....	33
2.5 Problem Statement.....	35
Chapter 3 Implementing and Evaluating Health-as-a-Service (HaaS) in Fog Computing.....	36
3.1 Why ECG analysis is very crucial?.....	37
3.2 What is Health-as-a-Service (Haas) and why it is better supported in Fog Computing?.....	38
3.3 Reading ECG signal using Einthoven's Triangle.....	39
3.4 Pre-processing to De-noise the ECG Signal.....	40
3.5 AD8232 ECG Sensor IC.....	41
3.6 Decision Making in ECG Signal as Normal or Abnormal.....	42
3.6.1 Recognizing Real Time ECG Anomalies Using Arduino,AD8232 and Java.....	45

3.6.2 ECG Heartbeat Arrhythmia Classification Using Time-Series Augmented Signals and Deep Learning Approach.....	46
3.6.3 ECG Image Classification using Deep Learning Approach.....	47
3.7 System Architecture.....	49
3.8 QoS Parameters.....	50
3.9 Different time stamps and defining the QoS parameters.....	53
3.10 Experimental Results.....	53
3.10.1 Transmission Delay.....	56
3.10.2 Computation Delay.....	58
3.10.3 Response Time.....	59
3.10.4 Standard Deviation.....	60
Concluding Remarks.....	61

Chapter 4 Distributed Computing in Fog Computing.....63

4.1 Distributed Computing and Dispy.....	63
4.1.1 Why dispy?.....	64
4.2 Raspberry Pi and R-Pi cluster.....	64
4.3 Computing Job description in dispy.....	65
4.4 Implementing and Analyzing ECG waves in dispy.....	66
4.4.1 Analysis of ECG Waves using Dispy API.....	67
4.4.2 Test Bed Configurations.....	68
4.4.3 Results.....	69
4.5 Improving Pattern matching performance in Genome sequences using Run Length Encoding in Distributed Raspberry Pi Clustering Environment.....	70
4.5.1 Problems with the traditional Genome Pattern matching.....	70
4.5.2 Results and Implementation discussions.....	70
Concluding Remark.....	71

Chapter 5 OptiFog: Optimization of Heterogeneous Fog Computing.....73

5.1 Understanding the Working Environment.....	75
5.1.1 The need for Heterogeneous Configuration.....	75
5.1.2 Hardware Configurations of the cluster nodes.....	76
5.1.3 Continuous Load on cluster nodes.....	77
5.2 Processing ECG signal using a Dispy manner.....	77
5.3 System Symbols, Functions and Performance indicators and their implications.....	81
5.3.1 System Functions.....	81
5.3.2 System parameters.....	82
5.4 Finding different optimization parameters and using them in the system to understand its effect on computation.....	82
5.4.1 Applying Response time in the proposed System.....	83
5.4.2 ECG signal processing by CPU Usage, Available memory and Number of free cores.....	85
5.4.3 ECG signal Processing based on CPU Usage.....	86
5.4.4 ECG signal Processing based on available main memory.....	88
5.4.5 ECG signal Processing based number of cores.....	90
5.5 OptiFog algorithm.....	92

5.5.1 Insights to the OptiFog Algorithm.....	92
5.6 Testing OptiFog algorithm.....	96
5.6.1 Test scenario 1 Speedup.....	96
5.6.2 Test scenario 2.....	96
5.6.3 Test scenario 3.....	99
Concluding Remark.....	100
Chapter 6 Lightweight Multi-level authentication scheme for Multi-level IoT-Fog Architecture.....	102
6.1 Traditional Security Approaches.....	104
6.2 Lightweight Encryption:.....	105
6.3 Research Gap.....	106
6.4 Proposed System Architecture and System Designing.....	107
6.4.1 System Architecture.....	108
6.5 Designing the secured scheme.....	112
6.5.1 Opcodes.....	112
6.5.2 System Functions and Parameters.....	113
6.5.3 H1 Security Scheme:.....	115
6.5.4 V1 Security Scheme:.....	116
6.5.5 Higher level security schemes.....	116
6.5.6 String to BigInteger and BigInteger to String conversion.....	117
6.6 Implementation results.....	118
6.7 Advantages of the Proposed Algorithm.....	121
6.8 Security against different vulnerable attacks.....	122
Concluding Remark.....	123
Chapter 7 Conclusion.....	124
References.....	126