Summer 5-16-2018

Design of A Distributed Real-time E-Health Cyber Ecosystem with Collective Actions: Diagnosis, Dynamic Queueing, and Decision Making

Yanlin Zhou
University of Nebraska - Lincoln, yanlin.zhou@huskers.unl.edu

DESIGN OF A DISTRIBUTED REAL-TIME E-HEALTH CYBER ECOSYSTEM WITH COLLECTIVE ACTIONS: DIAGNOSIS, DYNAMIC QUEUEING, AND DECISION MAKING

by

Yanlin Zhou

A THESIS

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfilment of Requirements

For the Degree of Master of Science

Major: Electrical Engineering

Under the Supervision of Professor Qing Hui

Lincoln, Nebraska

May, 2018

DESIGN OF A DISTRIBUTED REAL-TIME E-HEALTH CYBER ECOSYSTEM WITH COLLECTIVE ACTIONS:
DIAGNOSIS, DYNAMIC QUEUEING, AND DECISION MAKING

Yanlin Zhou, M.S.
University of Nebraska, 2018

Adviser: Qing Hui

In this thesis, we develop a framework for E-health Cyber Ecosystems, and look into different involved actors. The three interested parties in the ecosystem including patients, doctors, and healthcare providers are discussed in 3 different phases. In Phase 1, machine-learning based modeling and simulation analysis is performed to remotely predict a patient's risk level of having heart diseases in real time. In Phase 2, an online dynamic queueing model is devised to pair doctors with patients having high risk levels (diagnosed in Phase 1) to confirm the risk, and provide help. In Phase 3, a decision making paradigm is proposed to help regional healthcare providers to logistically rearrange regional medical resources. Therefore, this thesis provides an end-to-end solution on: Health Risk Identification, Risk Level Confirmation, and Regional Health Alert Level Decision Support.

ACKNOWLEDGMENTS

First, I consider myself very fortunate to have worked under the guidance of Professor Qing Hui, Department of Electrical and Computer Engineering, University of Nebraska-Lincoln. I am greatly indebted to Professor Hui for both sharing with me his professional experience and academic knowledge. I feel short of words to express my deep appreciation for his immense help in guiding me through the research field. I feel very fortunate to have worked with Dr. Hui on control and optimization, and Dr. Hui has shown me the greatest enjoyment in Academia.

Another person who had a huge impact on my work is Professor Hamid Vakilzadian, Department of Electrical and Computer Engineering, University of Nebraska-Lincoln. I am extremely thankful to Dr. Vakilzadian for his inspiring guidance and encouraging discussion. I learned vastly from many lectures and discussions with Dr. Vakilzadian and I am honored to have had the opportunities to work with him. Dr. Vakilzadian has enlightened me with both academical suggestions and life wisdom. I am forever grateful for his kindness.

I gratefully acknowledge Professor Dietmar Moeller, Department of Applied Stochastics and Operations Research, Clausthal University of Technology. Dr. Moeller provided valuable guidance and inspiring support when helping me developing Phase 1 and Phase 2 of this thesis. I really enjoy his wisdom in this work. The topic for Phase 1 changed drastically during the brainstorming phase, his medical background provided a lot of valuable guidance.

I would also want to acknowledge Professor Ashok Samal, Department of Computer Science and Engineering, University of Nebraska-Lincoln for his invaluable help during my last semester. His suggestion and advices have been immense in

the course of the preparation of this thesis.

I would like to thank Dr. Andreas Deutschmann, Inst. of Air Transport and Airport Research (FW), DLR, Germany and Mr. Ryan Anderson for their endless support, guidance, and dedication when developing Phase 2. The work present in Phase 2 was done in collaboration with Ryan.

I would also want to thank Mr. Chen Peng for his supportive and insightful suggestions when developing Phase 3. The work presented in Phase 3 was done in collaboration with Chen. Chen provided me with endless support and guidance when preparing for publications, academic learning and life advices. I am grateful that I have the chance to be his lab member.

I am also thankful to Mr. Balaji Balasubramaniam for this great programming and researching skills when developing Phase 1. The work presented in Phase 1 was done in collaboration with Balaji, and I have learned a lot from him as well. I have always been amazed by his work ethics and punctual delivery manner.

I am indeed indebted to my father Yongping Zhou, my mother Fengzhen Jin, and my friends: Siyeong Ju, Jinwoo Kim, Jianghao Wang, Haipeng Zhang, Ling He, Pengfei Dong, Qishuai Liu, Fayrouz Isfoula and Mehdi Firouznia for their encouragement throughout my education.

As a research student at the Department of Electrical and Computer Engineering, University of Nebraska-Lincoln, I have received immense help from many faculty, staff, students and research scholars. I take this chance to thank them all.

Table of Contents

List of Figures viii

List of Tables x

1 Introduction 1

2 Background and Related Work 4

2.1 Health Risk Identification 5

2.2 Risk Level Confirmation 7

2.3 Regional Health Alert Level Decision Support 9

2.4 Contributions in a Glance 11

3 Health Risk Identification 13

3.1 Technical Overview 13

3.2 Technical Explanation 13

3.2.1 Dataset 15

3.2.2 Preliminary analysis 17

3.2.3 Attribute selection 18

3.2.3.1 Supervised 19

3.2.3.2 Non-Supervised 21

3.2.4 Classifier Comparison with Cross Validation 25

3.2.5 Neural networks . . . 26
3.3 Results discussion . . . 27
3.4 Chapter Summary . . . 29

4 Risk Level Confirmation **31**
4.1 Overview . . . 31
4.2 Technical Details . . . 32
4.3 Material and Methods . . . 33
4.3.1 General Framework . . . 33
4.3.1.1 General MATLAB Framework . . . 33
4.3.1.2 SIMULINK Flow Chart . . . 33
4.3.1.3 Simplified MATLAB Framework . . . 36
4.3.2 Flow Logic . . . 36
4.3.3 Service Time Distribution . . . 39
4.3.4 Arrival Simulation . . . 39
4.3.4.1 Arrival Data and Shape . . . 39
4.3.4.2 Multimodal Arrival Distribution . . . 41
4.3.4.3 Poisson Arrival Distribution . . . 42
4.3.4.4 Normal Arrival Distribution . . . 42
4.3.4.5 Scenario Analysis . . . 43
4.4 Results and Discussion . . . 44
4.4.1 Time Spent in System . . . 44
4.4.2 Number of Servers Per Passenger . . . 45
4.4.3 Server Utilization Per Passenger . . . 46
4.4.4 Time Spent in Queue . . . 47
4.4.5 End User Application . . . 49

4.5 Chapter Summary 50

5 Regional Health Alert Level Decision Support **61**

5.1 Technical Overview 62

5.2 Technical Explanation 63

5.3 Preliminaries 64

5.3.1 Notation and Preliminaries 64

5.3.2 Self-Organized Criticality 65

5.3.3 Boltzmann Machine 66

5.3.4 Drift-Diffusion Model (DDM) 68

5.3.5 Generalized Exponential Integrate-and-Fire Model 69

5.4 Decision Making Dynamics 71

5.4.1 Behaviors 73

5.4.2 Sampling with EIF 74

5.5 Collective Behavior 78

5.5.1 Coupling and Connectivity 78

5.5.2 Mean Field Analysis 79

5.6 Convergence Analysis 80

5.6.1 Global Criticality from Local Dynamics 81

5.6.2 Absorbing States 82

5.7 Semistability 85

5.7.1 Lyapunov Theory for Semistability 86

5.8 Chapter Summary 92

6 Conclusion **98**

Bibliography **99**

List of Figures

3.1 Definition of Precision and Recall [64] 22
3.2 Correlation Matrix 23
3.3 PCA Based Top 40 Attributes 24
3.4 One Hidden Layer Neural Network Configuration 27
3.5 Three Hidden Layer Neural Network Configuration 28

4.1 A Queueing Model with SimEvent 35
4.2 A Flow Chart of the Simplified Logic 37
4.3 The Histogram of Real Data [45] 40
4.4 A Histfit for the Real Data 41
4.5 A Multimodal Distribution 42
4.6 A Multimodal Distribution PDF 43
4.7 A Multimodal Distribution Histogram 44
4.8 A Poisson Distribution PDF 45
4.9 A Poisson Distribution Histogram 46
4.10 A Normal Distribution PDF 47
4.11 A Normal Distribution Histogram 48
4.12 Time Spent in System Histogram – Poisson 49
4.13 Time Spent in System Histogram – Normal 50
4.14 Time Spent in System Histogram – MultiModal 51

4.15 Number of Active Servers – Poisson 52
4.16 Number of Active Servers – Normal 53
4.17 Number of Active Servers – MultiModal 54
4.18 Server Utilization – Poisson 55
4.19 Server Utilization – Normal 56
4.20 Server Utilization – MultiModal 57
4.21 Time Spent in Queue Histogram – Poisson 58
4.22 Time Spent in Queue Histogram – Normal 59
4.23 Time Spent in Queue Histogram – MultiModal 60

5.1 SODM model shows SOC property 65
5.2 Voltage Activated Current Gating Variable 94
5.3 Power-law Probability Distribution 95
5.4 In this Figure, as the firing probability increases with the membrane potential, σ converges to 1 with proper connectivity constraints. Cases with different numbers of active and connected neighbors are shown in a network system with $N = 10$. 96
5.5 In this figure, we remove the constraint that the cumulation of the local branching ratio in each iteration caps at 1. It is clear that as the number of connected neighbors increases, the network system enters the active/loading phase first, and then evolves to the dissipation/absorbing phase. The system clearly shows SOC dynamics, that is $\sigma = 1$ at both minimum and maximum connectivity. 97

List of Tables

3.1 PRELIMINARY ANALYSIS 18
3.2 TOP 20 FEATURE ATTRIBUTES 20
3.3 ENTROPY BASED PRELIMINARY CLASSIFICATION RESULTS 21
3.4 PCA BASED PRELIMINARY CLASSIFICATION RESULTS 24
3.5 CROSS VALIDATION ON HEART DISEASE DETECTION RESULTS . . 25
3.6 NEURAL NETWROK BASED CLASSIFICATION RESULTS 26
3.7 CONFUSION MATRIX FOR DECISION TREE RESULTS 27

Chapter 1

Introduction

The E-health system is receiving growing interest from healthcare ecosystem participants as to its immense help in real-time life-critical issues, but a comprehensive solution is yet to exist. This work develops a framework that emphasizes on three most important parties of an *E-health Cyber Ecosystem*: patients, doctors, and healthcare providers. This model coordinates three most important phases of an ecosystem: health diagnosis, dynamic online queueing for risk confirmation, and decision making on regional healthcare alert level and provides a complete end-to-end solution.

The Phase 1 is described as **Health Risk Identification** which focuses on health surveillance instantaneous awareness. The threshold problem of real-time disease diagnosis is to assess the risk automatically with an accurate model. A machine learning approach is adopted here to analyze a heart disease dataset provided by University of California, Irvine and classify the risk level (from 0 to 4) of a patient having heart disease. The model developed in this thesis reports an average of 10% increased prediction accuracy compared to other work. The identified risk level information offers decision support for both Phase 2 and Phase 3.

As the risk level increases, the Phase 2 is named **Risk Level Confirmation** provided by online doctors and caregivers in the health cyber ecosystem. Since the

misclassified errors by machine-learning techniques do exist in the classification model of Phase 1, vital parameters of patients with higher levels, for example level 2 and 3, should be checked remotely by the online doctors/caregivers. Note that a level 4 triggers an ambulance request immediately. Due to the nature of unbalanced patient/physician ratio ($>> 1$), it is usually the case that a patient's data cannot be viewed immediately if his/her risk level outbreaks. To address the challenge of providing an online queueing model and minimizing the waiting time, a dynamic queueing model is proposed with three common arrival patterns for online customer arrival rates: Poisson, Normal, and Multimodal distribution. The model achieves satisfactory upper bound and steady server utilization rate for all three cases.

The most imperative feature required by all healthcare providers is the **Decision Making Support** guided by **Regional Health Alert Level Analysis**. It is usually the case that the amount of medical resource supply such as: drugs, blood bags, equipment or even doctors, does not match the regional need. The ultimate decision made by a regional healthcare provider can be considered as a two-alternative forced choice: either requests from or dispatches to other regions, the medical resources. A novel collective decision making scheme using Spiking Neural Network (SNN) is proposed here to analyze the overall regional health alert level. In this model, each patient is considered as a single neuron in a SNN, where the optimal match between alert levels and resources is described by self-organized criticality behavior, while super-critical and sub-critical represents unbalanced patient-to-resource ratios. Both analytically and experimentally, the Boltzmann-Machine guided absorbing region provides a decision support for healthcare providers.

Therefore, this thesis successfully addresses three important research objectives

in E-health Cyber Ecosystem, including: Health Risk Identification, Risk Level Confirmation, and Regional Health Alert Level Decision Support. Assumptions and justifications will be made throughout this work due to the limited and confidential datasets.

Chapter 2

Background and Related Work

In the Introduction chapter, a framework comprises of three different phases is introduced in a very general view. This chapter gives a detailed explanation of: how the problem is defined; what are the related work; and how does this thesis patch all the missing links. The technical details will be covered in later corresponding chapters.

Cardiovascular disease (CVD) and the accompanying sequelae including angina, arrhythmias, and heart failure have become the leading cause of death in United States and the leading cause of death worldwide. In 2016, CVD is responsible for approximately 800,000 death in United States [48]. It is only more heart breaking to realize that one out of three deaths is caused by CVD, and about an average of one person dies from CVD every 40 seconds. Currently, more than 90 million Americans are diagnosed with CVD and their lives could be in danger if not treated properly and timely.

This work provides an end-to-end solution to: monitor patients' vital signs; predict the risk of having heart disease remotely and automatically; queue with doctors; reconfirm the high risk readings; and eventually provide decision support for all involved parties in ecosystem – patients, doctors, and regional healthcare providers.

2.1 Health Risk Identification

Heart disease has been a major threat to public health for decades. It is shown in the work of Eikendal *et al.* [21] as well as other reports [31, 23, 68, 46] that the most imperative action needed to prevent the death from heat disease is to perform the routine measurement on the vital signs, understand the warning sign and symptoms of a heart disease, and assess the risk level of the patients. Although there exits multiple companies that are currently working on wearable devices that track the vital signs, very few work has been done in modeling and simulation of online medical diagnosis to accurately describe the relationship between vital signs and risk level, as well as the sophisticated follow up service provided by doctors and healthcare providers.

The existing research methods focus largely on the relation between cardiovascular risk factors and health condition in an extended time frame. For instance, Fiorini et al [23] have proposed a personalized medical system that supports senior residences presenting chronic disease with reminders, which applies cyber-physical systems through a hybrid robot-cloud approach. Zhang et al [68] have focused on user-centric assistive cloud and big data that gathers information of patients and make these vital signs available for both users and physicians. Other work on related topics have studied either on monitoring [31] or data collection [54]. There, we have identified a gap in knowledge between the state-of-the-art research and the actual need of a monitoring system, that is to make decisions accordingly to analyze the severity of patients' condition and initiate different actions such as pill reminders, ambulance requests, *etc*.

While these mentioned researches are all interesting and insightful, the current solutions only provide basic level of diagnosis, straight forward "If this, then

that" (IFTTT) softwares, or models that focus on monitoring and data collection mainly. However, in a E-health Cyber Ecosystem, models should make great use of automation nature of cyber systems and run real-time diagnosis, physician support and regional logistic management to surpass the traditional healthcare system. Therefore, the goal of the model in Phase 1 is to accurately describe the relationship between vital signs and risk level of having a heart disease, using machine-learning techniques.

The work in Phase 1 would leverage the existing wearable vital sign collection products (i.e. Cova necklace) to save the nursing costs as well as to provide a preliminary medical decision support. In contrast to the existing straight forward heart disease anomaly detection in boundary approaches [68], this work compares several complex classifiers with machine-learning techniques, and in a degree provides decision support for all the ecosystem participants: patients themselves, doctors, caregivers, and healthcare providers. Note that, this risk level information is also used by Phase 2 to arrange the patients' priority in the queues and by Phase 3 to analyze the regional health alert level.

Technically, the goal of the Phase 1 in this thesis, to be specific, is to determine the risk level of a patient having heart disease based on six machine learning approaches – baseline classifier, Bayesian classifier, decision tree, nearest neighbor, support vector machine, and neural network. A heart disease medical dataset, collected from real patients, published by the University of California, Irvine is used. However, this dataset contains 76 attributes and it is important to select the most contributing attributes to speed up the diagnosis process and to improve the accuracy. Therefore, both supervised (using information gain, which is a measure of information) and non-supervised methods (using principal component analysis, which is a dimension reduction technique) are used in attribute selection step.

Since all the data are labeled with five risk levels, from 0 to 4, it is convenient to evaluate the accuracy of the proposed techniques. Note that, although this thesis only focuses on the automated heart disease diagnosis, this framework can always be expanded and transferred to work on other deadly diseases that require routine vital sign measurements, such as: Chronic Lower Respiratory Disease, Stroke, Diabetes, Pneumonia and Kidney Disease [48].

2.2 Risk Level Confirmation

The machine-learning based model in Phase 1 gives accurate diagnosis of patients' risk levels, but the performance is not perfectly accurate (*i.e.* accuracy $< 100\%$). Nevertheless, this work has achieved very high precision and recall rates compared to other work, which will be discussed in a later chapter. Due to the classification errors in this model, the vital parameters of patients who are identified with high risk levels should be checked again online by trained caregivers or real doctors. Note that this assumption would not result in massive amount of online confirmation requests for doctors, since a large portion of patients usually carry risk levels of 0 and 1.

The above described process is the Phase 2 of the whole thesis: online dynamic queueing analysis. As explained earlier, the doctors/caregivers are one of three most important parties in the E-health cyber ecosystems. It should be pointed out that this differs a little from the conventional queueing models in hospitals because this framework mainly deals with the interactions in cyber space. That being said, the online queueing model here is used to assign a doctor to check on patients' real-time vital signs, and reconfirm the risk level. To eliminate the ambiguity and for simplicity purpose, we call the distribution of reconfirmation

requests to be: patient arrival distribution.

In this work, risk levels of 0 and 1 will not be sent to doctors, and only patients that are diagnosed with risk levels of 2 and 3 would be queued with a doctor. Further, we assume that a level 3 risk should receive the higher privilege in the queueing model.

Due to the nature of the imbalanced ratio of patients/doctors, which is usually ($>> 1$), and the highly dynamic process of the cyber arrival rate of confirmation requests for an online doctor's end, a dynamic queueing model is needed to mitigate the problem and to ensure that each patient's data will be checked on time. Also, the extreme cases such as bulk arrival and zero request can happen so a dynamic queueing model is needed to efficiently use the manpower of doctor resource. To be specific, the utilization ratio, a term to express the stability of a queue, is kept at a satisfactory level by adding or subtracting the number of online doctors.

Studies [30, 65] have shown that scheduling systems can be described by models of fluid dynamics, which opens a new approach to model states and processes of e-health cyber ecosystems in a mathematical and physical sense, considering sequences of processes as laminar, turbulent or chaotic flows. However, unlike a traditional queueing model for a hospital that all the processes are well scheduled (usually described by laminar sequences of processes), an online queueing model is unpredictable (thus usually turbulent and chaotic). To author's best knowledge, there is no published work on this topic about using the idea of turbulent flow to model a dynamic queueing system, and therefore, this work is the first of its kind.

Moreover, let's go over some state-of-the-art queueing models. A multiserver that randomly opens an extra server is proposed in the work of Brunell and Wittevrongel [12]. A detailed analysis is given to illustrate the system performance

with two servers instead of one in random periods. A simulation model for generating flow of passengers is developed in [39] by Artur and Tomasz. They described this queueing process as an optimization problem that the required number of stations is calculated based on the maximum customer queue size. However, the different dynamics of passengers' arrival distribution are not discussed thus suggests the missing knowledge.

Therefore, the goal of the second phase is to close the gap among the state-of-the-art planning schemes and bring the top down to the focus of a dynamic queueing model describing the cyber queueing process. The expected performance is to guarantee a reliable, punctual and efficient reconfirmation for high risk patients through a robust management of doctors' duty table, and later generalize from the bottom up to apply the model to other processes as well.

2.3 Regional Health Alert Level Decision Support

The last party in this e-health cyber ecosystem framework is the healthcare providers. Traditional healthcare providers can refer to the following individuals: Hospitals, Nursing Homes, Home health agencies, Dialysis facilities, Inpatient Rehabilitation Facilities, Hospice Agencies, *etc*. However, this work focuses on a higher point of view and provide decision support for regional healthcare organizations. The National Association of The Strategic Regional Healthcare Organizations (SRHO) is an example of this regional healthcare providers on which this framework studies.

Despite of the increasing emergence of these collaborative organizations, SRHO and other big groups are currently only looking to achieve economies goals to take role in population health initiatives. This is due to the lack of an E-health

Cyber Ecosystem [53] that provides members with shared data, shared resource, and regional status and needs. However, in the work of Plaza *et al*. as of January 2018, there is yet to exist a distributed decision support model in this field[53].

Let's look at an example when this decision support will be needed. Say we have a regional healthcare organization in Omaha, Nebraska and another regional healthcare organization in Kansas City, Missouri. Assume that sadly in this ecosystems, there are 50 patients diagnosed with high risk levels during April in Omaha, and sadly for the same month the number is 700 for Kansas City. However, assume that the medical resources in Kansas City in April cannot afford to potentially treat such a large number of patients. Let's also assume that both regional healthcare organizations can provide the same amount in drugs, blood bags, equipment and cardiologists in town that can only offer proper treatment for up to 600 people. In this case, since the risk level of the whole Kansas City Region is higher than Omaha Region, it is wise to dispatch the medical resources from Omaha to Kansas City to prepare for the potential treatment need in May.

However, due to the lack of the description of E-health Cyber Ecosystem, as well as the scarce attention in this field, the distributed decision math model is currently non-existing. Therefore, the goal of the third phase in this thesis is to provide an ultimate decision support for healthcare organizations to make a two-alternative choice: either requests from or dispatches to other regions, the medical resources. Note that this framework is a free-response model, so when decisions are not given, it means that a neural decision is implied. This framework should better prepare healthcare providers to succeed in population health care delivery.

2.4 Contributions in a Glance

With the above background introduction and literature review, here we list some most important contributions of this thesis.

For Phase 1: Health Risk Identification via Machine Learning

1. Attribute selection using supervised (information gain and entropy) and non-supervised methods (principal component analysis) for assessing the heart risk;

2. A decision-making model for different level of risks and it is reported with six different classifier models.

For Phase 2: Online Queueing for Risk Level Confirmation

1. A dynamic queueing model for E-health Cyber Ecosystem that changes the number of servers automatically;

2. Identify the characteristic number that can be used to correlate the laminar and turbulent flow processes;

3. Test Poisson, Normal, and Multimodal patient arrival distributions as inputs to this model.

For Phase 3: Decision Making on Regional Healthcare Alert Level

1. To the best of the authors' knowledge, this thesis is the first published work on modeling decision-making processes with the SOC property;

2. A collective decision making model, i.e., EDM, is proposed to implement the DDM methodology on EIF spiking neurons;

3. A probability inference scheme on EIF sampling is proposed, which extends an existing leaky integrate-and-fire sampling method;

4. Mean field analysis of the connectivity of EDM is given, which exhibits global criticality;

5. A detailed analysis is given to reveal SOC behavior of the EDM model under criticality conditions.

Chapter 3

Health Risk Identification

Phase 1: Health Risk Identification via Machine Learning

3.1 Technical Overview

This project aims to provide modeling and simulation analysis to perform a routine check on vital signs of patients and automatically make appropriate decisions. More specifically, we envisage to model medical readings (such as blood pressure and respiratory rate that can be directly retrieved by devices like Cova necklace, while blood sugar can be measured with the help of a blood sugar measuring app) in real time. Then, the model performs an analysis to assess the risk of a patient having heart diseases using machine learning approach. The risk level information will be prioritized in Phase 2 and physicians will be notified if necessary. Also, a detailed analysis can be performed by cloud computing system and provide decision support for Phase 3.

3.2 Technical Explanation

This chapter proposes a novel modeling and simulation environment to assess the risk of a patient having heart diseases. This model comprises three parts, real-time

data collection, both online and offline data analysis, and real-time risk level prediction. The data collection is performed by routinely check on vital signs of patients, which can be done by commercial products such as Cova necklace. Then, the pre-recorded information (i.e., age, sex, smoke, *etc*.) are compared against a pre-trained machine learning model. Finally, the current condition of the patient is classified into 5 groups ranging from no-risk to high-risk.

As mentioned in a previous chapter, there are 2 contributions of this work, first in attribute selection process and the second in comparing the different machine learning approaches. In attribute selection step, both supervised (using information gain, which is a measure of information) and non-supervised methods (using principal component analysis, which is a dimensionality reduction technique) are introduced to filter out the least contributing attributes for assessing the presence of heart disease. Secondly, six different machine-learning techniques – baseline classifier, Bayesian classifier, decision tree, nearest neighbor, support vector machine, and neural network are compared. Overall, the developed model achieved an average of 10% increased prediction accuracy compared to other work; and the decision tree and support vector machine methods are the most accurate ones.

Here we have the following assumption for data-collection step:

Assumption 1. *The patients' vital parameters can be collected remotely through non-medical purpose devices (i.e., Apple Watch), Medical portable devices (i.e., Cova Necklace), and Implantable medical devices (i.e., implantable cardioverter-defibrillator). All the data are collected in real-time.*

3.2.1 Dataset

The Hungarian medical dataset provided by the University of California, Irvine (available at the following link: *http://archive.ics.uci.edu/ml/datasets/heart+Disease*) is used throughout Phase 1. This database contains 76 attributes and 294 patients' real data in total, but all published experiments refer to using smaller subsets, *i.e.*, around 14 in all publications [2].

The class dirtibution of Hungarian dataset is as follow:

1. Class 0 $\rightarrow$ 188
2. Class 1 $\rightarrow$ 37
3. Class 2 $\rightarrow$ 26
4. Class 3 $\rightarrow$ 28
5. Class 4 $\rightarrow$ 15

Here, we are more interested in the presence of chest pain type so we focusing on predicting the attribute 9 in the dataset: chest pain type. The chest pain types of the dataset are as follows:

1. Value 1: typical angina 11 instances
2. Value 2: atypical angina 106 instances
3. Value 3: non-angina pain 54 instances
4. Value 4: asymptomatic 123 instances

Note that the data is not very evenly distributed, and we will discuss how does this affect our results in Section 3.3.

In fact, Hungarian Dataset have attracted many data Mining researchers as it is publicly available since 1990s. For example, Aha and Kibler [2] proposed an instance-based algorithm and achieved 77% and 74.8% of accuracy with their own modified models NTgrowth and C4.5 techniques. Detrano *et al.* [19] fully examined the dataset and concluded that patients with chest pain and transitional diseases are the higher risk subjects. Gennari *et al.* [25] developed a clustering algorithm and achieved an higher accuracy of 78.9%. Edmonds [20] devised structured distributed learning algorithm to make the prediction and compared with the global evolutionary computation approach. Interestingly, performance of the above work depends largely on the attribute selection, which encourages the first step of machine-learning in Phase 1.

Recently, researchers have started to apply computational intelligence methods to Hungarian dataset, including machine learning and statistical data mining techniques [50]. Note that the following papers have all followed the rule of using 14 carefully picked attributes in the work of Aha [2]. These 14 attributes are: age, sex, pain type, resting blood pressure, serum cholestoral, fasting blood sugar, resting electrocardiographic results, maximum heart rate achieved, exercise induced angina, ST depression induced by exercise relative to rest, the slope of the peak exercise ST segment, number of major vessels (0-3) colored by flourosopy, heart rate level and diagnosis of heart disease (angiographic disease status). Interested readers should refer to [2] for further information. Nahar *et al.* closely worked with medical experts and developed judgment-based-feature selection process. They proposed to used medical knowledge driven learning to achieve the good results. Moreover, popular algorithms such as Naive Bayes, SMO, IBK, AdaBoostM1, J48, PART have been applied on these 14 attributes. Dangi *et al.* [17] envisaged a range for the expected statues of the heart disease ranging from 0 to 4 and provided

different probability distribution for each. Other often used machine learning techniques are KNN (kth Nearest Neighbors), Naive Bayes, logistic regression, Neural network, SVM (Support Vector Machine), GBM (Generalized Boosted Regression Model), LDA (Linear Discriminant Analysis), PCA (Principle Component Analysis), Multinomial Regression, Random Forest, *etc*. The average accuracy of the mentioned work are between 74% and 79%. In this work, we report the 10% increased accuracy.

3.2.2 Preliminary analysis

The preliminary data pre-processing is performed to examine the complexity of the data, as well as to determine which classifier works the best for this dataset. This is needed because different machine learning algorithms have entirely different performance on different datasets. In this step, five different classifiers are applied:

1. **Baseline classifier** that provides a baseline performance benchmark for other classifiers. This method simply outputs the majority class.

2. **Bayesian classifier** is a probabilistic model that the classification of variables are probabilistically related to each other.

3. **Decision tree** that assumes tree-structured branched decisions for classification.

4. **Nearest neighbor** that compares the features of neighbors and predicts the output.

5. **Support vector machine** that draws a separation line in hyperplane and maximize the projection margin while minor outliers can be ignored.

Note that these are commonly used classifiers, interested readers can refer to popular massive open online courses such as the one provided by Google: (https://developers.google.com/machine-learning/crash-course/ml-intro).

Note that, the 50%|50% rule is used where half of the dataset is used for training and the other half for testing. The results are shown in Table 3.1, performed by using WEKA - a machine learning software [26]. In fact, WEKA is used in a large amount of publications [50]. It is clearly shown that the data is almost evenly separated with 42.9% correctly classified patients in baseline benchmark. Also, both decision tree and support vector machine techniques perform much better than the rest, thus we keep using these two methods for the very next section.

Table 3.1: PRELIMINARY ANALYSIS

Classifier name	Correctly classified (%)	Incorrectly classified (%)	Precision (%)	Recall (%)
Baseline classifier	42.9	57.1	18.4	42.9
Bayesian classifier	72.1	27.9	70.6	72.1
Decision tree	87.1	13.0	85.7	87.1
Nearest neighbor	64.6	35.4	70.0	64.6
Support Vector	81.0	19.0	77.4	81.0

3.2.3 Attribute selection

After the preliminary analysis, the very next step is similar to the publications mentioned in Section 3.2.1. However, here we choose our own attributes instead of using the commonly used 14 attributes as in [2]. The main two purposes of this step is to: 1) Speed up the classification process; and 2) Remove the less contributing attributes.

Since the author has no medical domain expertise or active collaborations with cardiologists, here we use supervised and non-supervised attribute selection.

3.2.3.1 Supervised

Here, the Information gain and Entropy [17] techniques are used, because we know the output labels for each instance and can rank the importance. The output class is provided as a part of the dataset and is divided into 5 categories: from 0 (no risk) to 4 (high risk).

The the entropy of dataset is calculated by the mentioned technique using Equation 3.1. We firstly remove two useless attributes: name and ID, and determine how much of each of the rest attribute contributes to the classification.

The Entropy is given in Equation 3.1.

$$-\sum_{i=1}^{n} P(x_i) \log_b P(x_i) \tag{3.1}$$

where n is the number of total attributes, 74; P is the probability distribution of a certain feature attribute that is calculated over all the n attributes.

The second technique is called the information gain which ranks the rest 74 attributes according to their importance while at classification. Thus we have the following equation:

$$IG^i = E_d^i - E_a^i \tag{3.2}$$

where IG stands for information gain, E_d means Entropy of dataset and E_a means entropy with attribute.

Here, we list out all top 20 attributes in Table 3.2 using information gain technique.

Now let us look at the experiment with top ranked 5, 10, 20 attributes. We keep using Decision tree and Support vector machine. The results are shown in Table 3.3. Note that precision and recall are two common terms used in Machine

Table 3.2: TOP 20 FEATURE ATTRIBUTES

Rank	Attribute name	Description
1	chol	serum cholestoral in mg/dl
2	painexer	provoked by exertion
3	relrest	relieved after rest
4	thalach	maximum heart rate achieved
5	thalrest	resting heart rate
6	age	age in years
7	tpeakbps	peak exercise blood pressure (first of 2 parts)
8	thaltime	time when ST* measure depression was noted
9	num	diagnosis of heart disease (angiographic status) – Value 0: <50% diameter narrowing – Value 1: >50% diameter narrowing
10	ekgday	day of exercise ECG reading
11	thaldur	duration of exercise test in minutes
12	exang	exercise induced angina
13	cday	day of cardiac cath
14	rldv5e	height at peak exercise
15	tpeakbpd	peak exercise blood pressure (second of 2 parts)
16	trestbps	resting blood pressure (in mm Hg)
17	slope	the slope of the peak exercise ST* segment – Value 1: upsloping, – Value 2: flat, – Value 3: downsloping
18	oldpeak	ST* depression induced by exercise relative to rest
19	trestbpd	resting blood pressure
20	rldv5	height at rest
*The ST segment represents the isoelectric period when the ventricles are in between depolarization and repolarization.		

Learning to evaluate the performance of the classification [56]. Precision is given as: true positive/ (true positive+false positive) while recall is: true positive/(true positive + false negative). A relation picture is given in Figure 3.1.

Interested reader may refer to [56] for further information. Here we give the definition of Precision and Recall with regard to the theme. Precision measures the fraction of classified heart disease instances that are correct, Recall measures the fraction of actual heart disease instances which are correctly predicted.

Table 3.3: ENTROPY BASED PRELIMINARY CLASSIFICATION RESULTS

	Decision tree (%)		Support Vector Machine (%)	
NO. of features	PRECISION	RECALL	PRECISION	RECALL
Top 5	86.2	88.1	84.5	85.0
Top 10	86.2	88.1	83.8	85.0
Top 20	86.3	87.8	78.3	80.6

As shown in Table 3.3, we only observe slight improvement for decision tree when less attributes are selected. However, there are 6% and 5% increase in rate for precision and recall, respectively, when the number of attributes is reduced from 20 to 5.

3.2.3.2 Non-Supervised

Besides supervised approaches, we should also check non-supervised methods since the labeled data may provide a bias for attribute selection. Therefore, we should identify the top contributing attributes without revealing the output class.

Let us first start by dimensional reduction method to better analysis the correlation between different input attributes. Here we consider a famous tool in model reduction, principel component analysis (PCA) [51]. PCA is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly

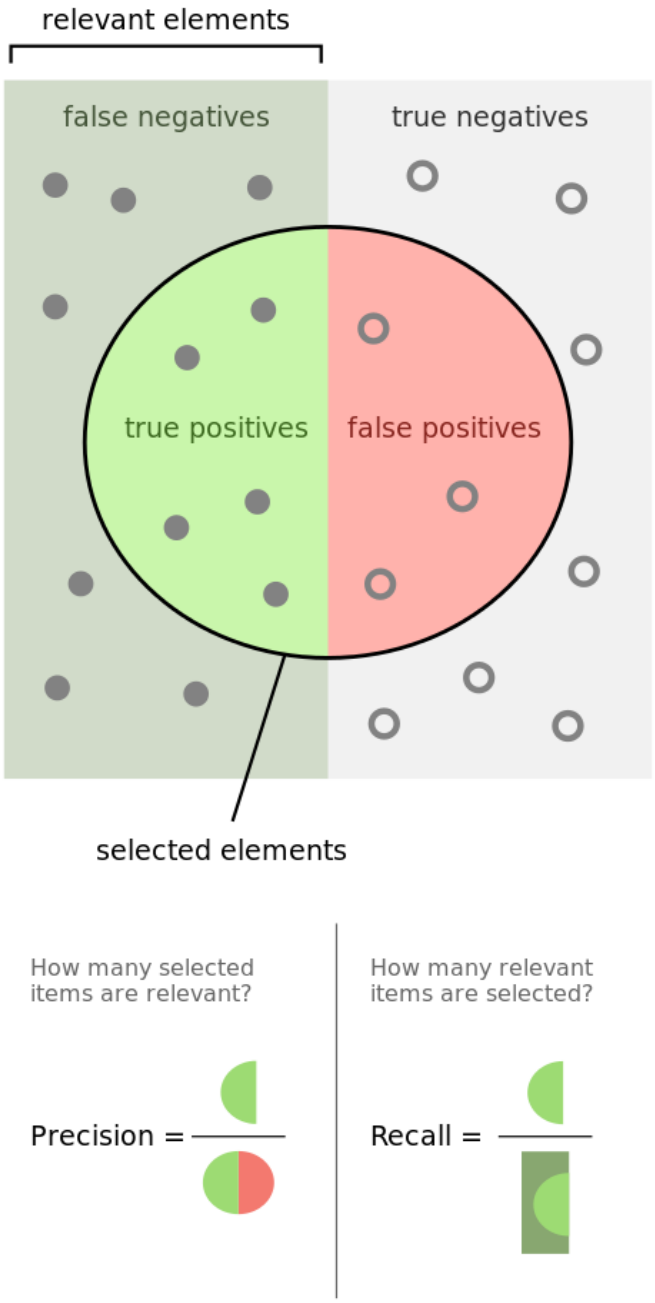


Figure 3.1: Definition of Precision and Recall [64]

correlated variables into a set of values of linearly uncorrelated variables called principal components.

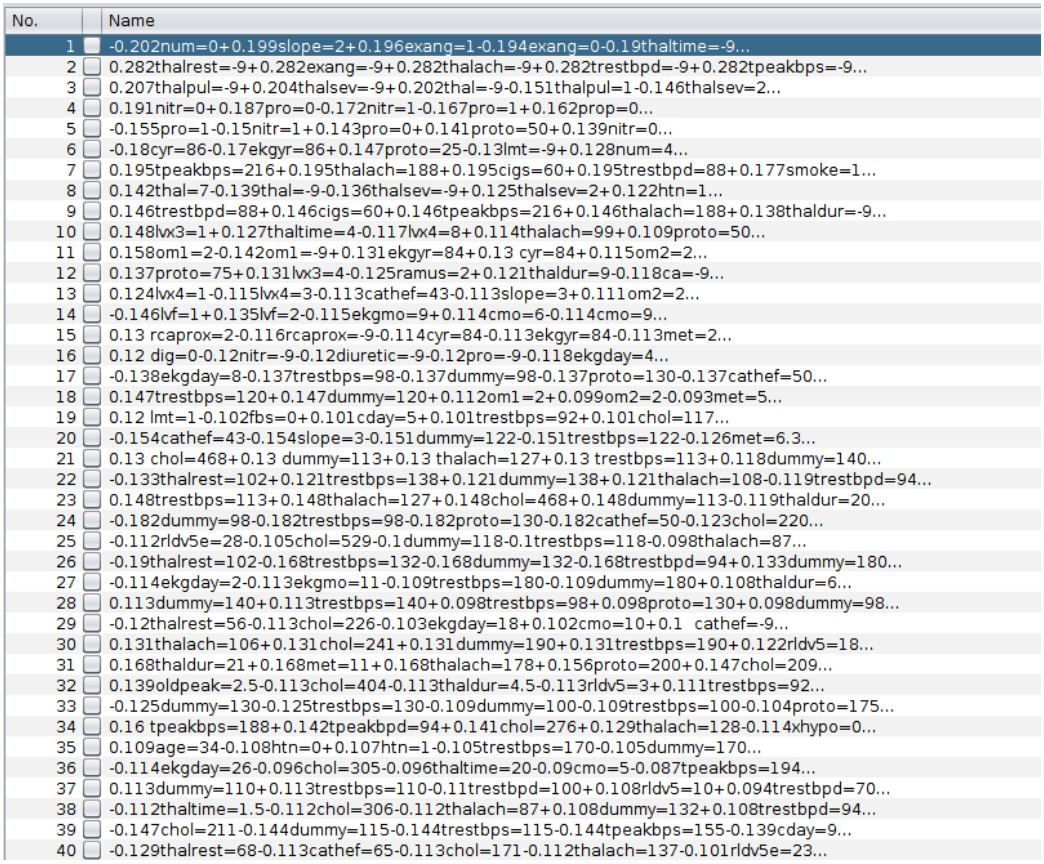
The first step of performing PCA is to calculate the correlation matrix (or covariance matrix), a matrix used to examine the dependence between multiple variables at the same time. A portion of the correlation matrix is shown in Figure 3.2, where the columns vs. rows represent attributes' correlation with each other and the diagonal is 1 (attribute to itself). Note that due to space constraints here we only look at a portion of the whole graph. Nevertheless, this does not affect our final conclusion in this section.

As we can clearly see, all non-diagonal elements in the correlation matrix are small negative values. The possible correlation values are in $[-1, +1]$, where -1 means no correlation at all and +1 means highly correlated. It is clearly seen that all of the values in our correlation matrix are negative and thus weakly correlated. This further suggests that the attributes are almost independent of each other.

Figure 3.2: Correlation Matrix

To further confirm that the attributes are almost independent to each other, we perform the classification on the top 40 attributes selected by PCA. In this case, the

dimension is reduced from 74 to 40. The listed 40 feature attributes are shown in Figure 3.3, and the results of the new classification is shown in Table 3.4.



No.	Name
1	-0.202num=0+0.199slope=2+0.196exang=1-0.194exang=0-0.19thaltime=-9...
2	0.282thalrest=-9+0.282exang=-9+0.282thalach=-9+0.282trestbpd=-9+0.282tpeakbps=-9...
3	0.207thalpul=-9+0.204thalsev=-9+0.202thal=-9-0.151thalpul=1-0.146thalsev=2...
4	0.191nitr=0+0.187pro=0-0.172nitr=1-0.167pro=1+0.162prop=0...
5	-0.155pro=1-0.15nitr=1+0.143pro=0+0.141proto=50+0.139nitr=0...
6	-0.18cyr=86-0.17ekgyr=86+0.147proto=25-0.13lmt=-9+0.128num=4...
7	0.195tpeakbps=216+0.195thalach=188+0.195cigs=60+0.195trestbpd=88+0.177smoke=1...
8	0.142thal=7-0.139thal=-9-0.136thalsev=-9+0.125thalsev=2+0.122htn=1...
9	0.146trestbpd=88+0.146cigs=60+0.146tpeakbps=216+0.146thalach=188+0.138thaldur=-9...
10	0.148lvx3=1+0.127thaltime=4-0.117lvx4=8+0.114thalach=99+0.109proto=50...
11	0.158om1=2-0.142om1=-9+0.131ekgyr=84+0.13 cyr=84+0.115om2=2...
12	0.137proto=75+0.131lvx3=4-0.125ramus=2+0.121thaldur=9-0.118ca=-9...
13	0.124lvx4=1-0.115lvx4=3-0.113cathef=43-0.113slope=3+0.111om2=2...
14	-0.146lvf=1+0.135lvf=2-0.115ekgmo=9+0.114cmo=6-0.114cmo=9...
15	0.13 rcaprox=2-0.116rcaprox=-9-0.114cyr=84-0.113ekgyr=84-0.113met=2...
16	0.12 dig=0-0.12nitr=-9-0.12diuretic=-9-0.12pro=-9-0.118ekgday=4...
17	-0.138ekgday=8-0.137trestbps=98-0.137dummy=98-0.137proto=130-0.137cathef=50...
18	0.147trestbps=120+0.147dummy=120+0.112om1=2+0.099om2=2-0.093met=5...
19	0.12 lmt=1-0.102fbs=0+0.101cday=5+0.101trestbps=92+0.101chol=117...
20	-0.154cathef=43-0.154slope=3-0.151dummy=122-0.151trestbps=122-0.126met=6.3...
21	0.13 chol=468+0.13 dummy=113+0.13 thalach=127+0.13 trestbps=113+0.118dummy=140...
22	-0.133thalrest=102+0.121trestbps=138+0.121dummy=138+0.121thalach=108-0.119trestbpd=94...
23	0.148trestbps=113+0.148thalach=127+0.148chol=468+0.148dummy=113-0.119thaldur=20...
24	-0.182dummy=98-0.182trestbps=98-0.182proto=130-0.182cathef=50-0.123chol=220...
25	-0.112rldv5e=28-0.105chol=529-0.1dummy=118-0.1trestbps=118-0.098thalach=87...
26	-0.19thalrest=102-0.168trestbps=132-0.168dummy=132-0.168trestbpd=94+0.133dummy=180...
27	-0.114ekgday=2-0.113ekgmo=11-0.109trestbps=180-0.109dummy=180+0.108thaldur=6...
28	0.113dummy=140+0.113trestbps=140+0.098trestbps=98+0.098proto=130+0.098dummy=98...
29	-0.12thalrest=56-0.113chol=226-0.103ekgday=18+0.102cmo=10+0.1 cathef=-9...
30	0.131thalach=106+0.131chol=241+0.131dummy=190+0.131trestbps=190+0.122rldv5=18...
31	0.168thaldur=21+0.168met=11+0.168thalach=178+0.156proto=200+0.147chol=209...
32	0.139oldpeak=2.5-0.113chol=404-0.113thaldur=4.5-0.113rldv5=3+0.111trestbps=92...
33	-0.125dummy=130-0.125trestbps=130-0.109dummy=100-0.109trestbps=100-0.104proto=175...
34	0.16 tpeakbps=188+0.142tpeakbpd=94+0.141chol=276+0.129thalach=128-0.114xhypo=0...
35	0.109age=34-0.108htn=0+0.107htn=1-0.105trestbps=170-0.105dummy=170...
36	-0.114ekgday=26-0.096chol=305-0.096thaltime=20-0.09cmo=5-0.087tpeakbps=194...
37	0.113dummy=110+0.113trestbps=110-0.11trestbpd=100+0.108rldv5=10+0.094trestbpd=70...
38	-0.112thaltime=1.5-0.112chol=306-0.112thalach=87+0.108dummy=132+0.108trestbpd=94...
39	-0.147chol=211-0.144dummy=115-0.144trestbps=115-0.144tpeakbps=155-0.139cday=9...
40	-0.129thalrest=68-0.113cathef=65-0.113chol=171-0.112thalach=137-0.101rldv5e=23...

Figure 3.3: PCA Based Top 40 Attributes

Table 3.4: PCA BASED PRELIMINARY CLASSIFICATION RESULTS

Classifier name	Correctly Classified (%)	Incorrectly Classified (%)	Prec (%)	Recall (%)
Baseline classifier	41.84	58.16	17.50	41.8
Bayesian classifier	50.61	49.32	55.40	50.70
Decision tree	51.02	48.98	49.90	51.00
Nearest neighbor	48.29	51.70	50.70	48.30
Support Vector	65.65	34.35	54.20	65.60

As illustrated in Table 3.4, the accuracy is very low for all techniques, with the highest accuracy of 65.65% using SVM. Therefore, we have experimentally proved

that the supervised methods are better approaches for this dataset.

3.2.4 Classifier Comparison with Cross Validation

With the above mentioned results, we have determined to use supervised approaches. Now we perform a rigorous training on the dataset. To be specific, the cross validation is used.

Here we adopt 10-fold cross-validation. This is a trick that divides dataset into 10 equal subsets, 9 of which are used for training and the rest part is used for testing. Therefore, the training-testing ratio is $9:1$. This step is iterated 10 times until all 10 portions are tested on and we get a a final averaged accuracy rate. Note that, each iteration starts with no prior knowledge, and these 10 tests are independent.

The results of 10-fold cross validation for all 5 supervised techniques are shown in Table 3.5. Again, we observe that decision tree and support vector machine achieve higher precision and recall rates. This further strengthen our conclusion in Section 3.5 that decision tree and support vector machine are good techniques to use with this dataset.

Table 3.5: CROSS VALIDATION ON HEART DISEASE DETECTION RESULTS

Classifier name	Correctly classified (%)	Incorrectly classified (%)	Prec (%)	Recall (%)
Bayesian classifier	84.01	15.98	81.90	84.0
Decision tree	88.10	11.90	86.20	88.10
Nearest neighbor	82.99	17.07	82.50	83.00
Support Vector	85.00	14.97	83.80	85.00

3.2.5 Neural networks

We discuss neural network here in a separate section because it is significantly different from presented 5 classifiers. Here we feed the top 20 attributes retrieved in Table 3.2 as input to the neural network with different layers and different configurations. When constructing a neural network, the most important parameters are number of hidden layers, number of neurons present in the hidden layer, epochs and batch size. Note that epoch means the iteration times, and batch size means the number of data points that are passed at a time to train the model.

Now let us start by using the default setup in WEKA, and modify each parameter until the classification performance drops. The setup and parameters are shown in Table 3.6. Also, the neurons are adjusted to use sigmoid activation fucntion for smoothness.

Table 3.6: NEURAL NETWROK BASED CLASSIFICATION RESULTS

Layers	Neurons	Epoch	Batch Size	Correctly Classified	Incorrectly Classified	Prec (%)	Recall (%)
1	200	200	100	85.03(%)	14.97(%)	82.10	85.00
1	40	200	100	79.59(%)	20.41(%)	76.13	79.60
3	100-50-10	200	100	73.81(%)	26.19(%)	71.00	73.80
1	200	500	100	84.35(%)	15.64(%)	81.70	84.44
1	200	100	10	85.03(%)	14.97(%)	82.40	85.00

The default configuration and three-layer configurations are both presented in Figure 3.4 and 3.5. Due to the complex structure and high dimensionality, the training length are all at least greater than an hour. Further, the results are similar to supervised methods.

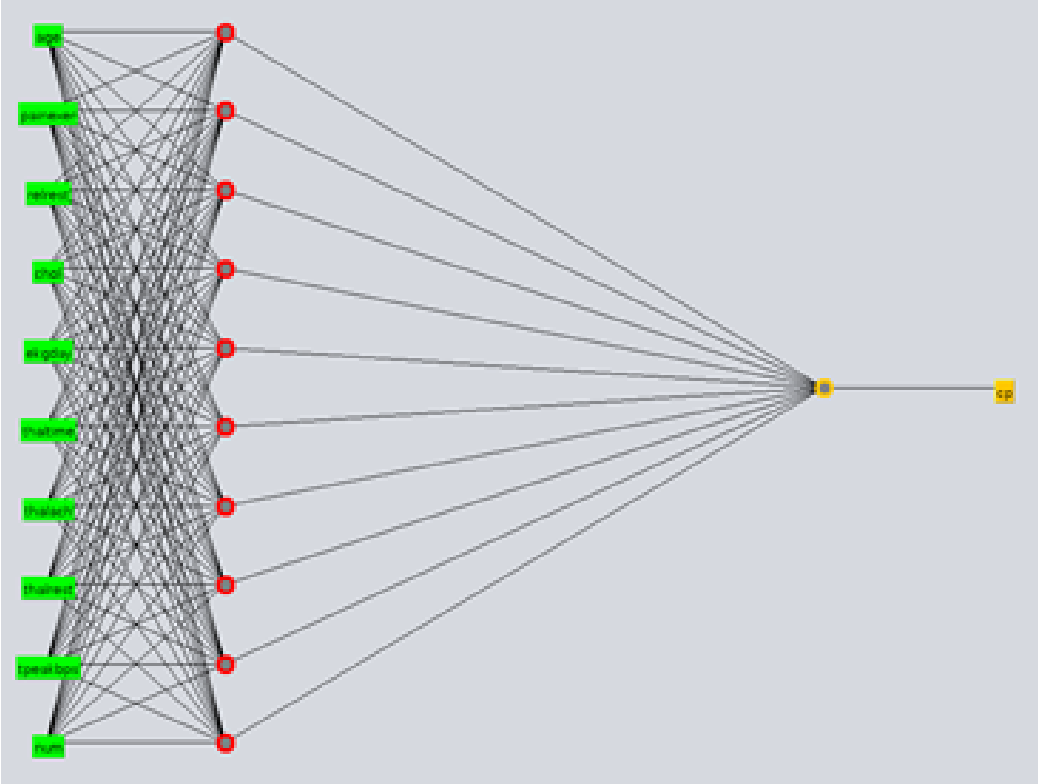


Figure 3.4: One Hidden Layer Neural Network Configuration

3.3 Results discussion

Here, we have concluded that the decision tree classifier achieves the best precision and recall: 86.20% and 88.10%, respectively. Now we examine the confusion matrix of the decision tree, aka, the matrix to represent the mis-classified results.

Table 3.7: CONFUSION MATRIX FOR DECISION TREE RESULTS

1	2	3	4	← Classified as
0	10	0	1	1: typical angina
0	102	0	4	2: atypical angina
0	18	34	2	3: non-angina pain
0	0	0	123	4: asymptomatic

As shown in Table 3.7, we observe that typical angina (value 1) are misclassified to either atypical angina (value 2) or asymptotic (value 4), which causes most of

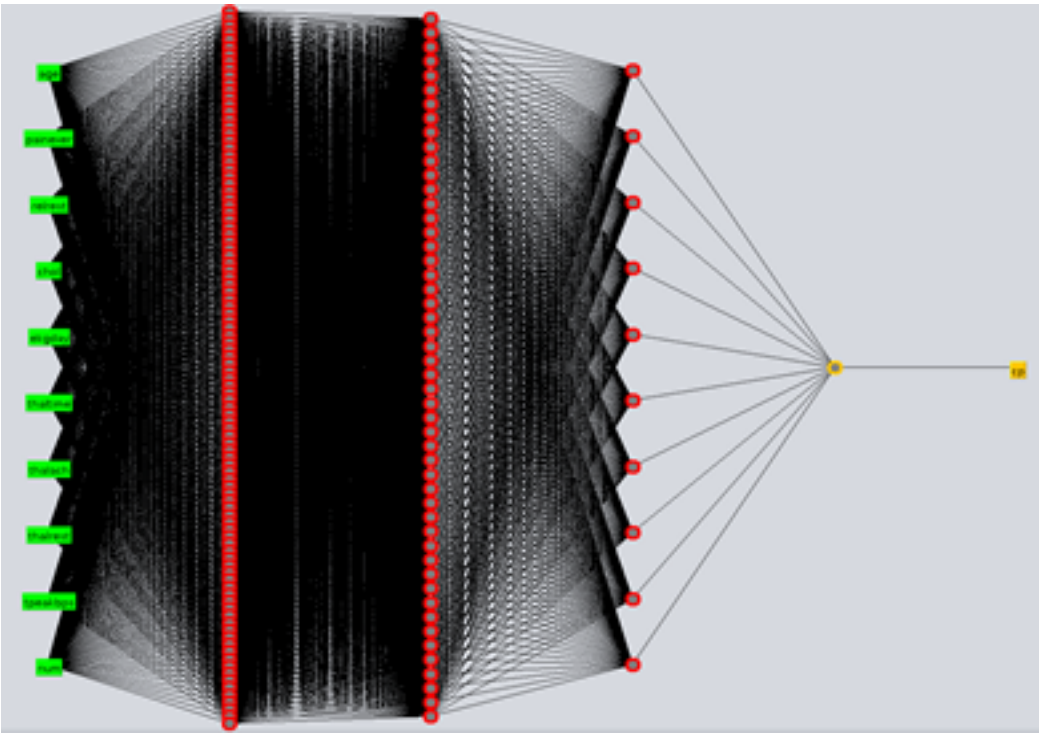


Figure 3.5: Three Hidden Layer Neural Network Configuration

our errors. Similarly, non-angina pain (Value 3) has poor prediction rate as well. However, in other two categories, we have achieved almost perfect results.

Moreover, let us analyze the reason for mis-classification for typical angina (Value 1) and non-angina (Value 3). We hypothesized that misclassification may be resulted by the wild age group ranges. In the dataset, we see ages ranging from 28 to 66, and the misclassified 11 instances for Value 1 do not belong to a specific age group: 30, 34, 35, 43, 43, 46, 47, 54, 55, 57 and 62. Similarly, the age group for misclassified Value 3 vary between 36 to 60. This is left as a future extension of our work to connect with real cardiologists to figure out the reason behind the mis-classification.

Nevertheless, we examine the maximum heart rate to find out the reason for mis-classification. We observe that the mis-classified instances have maximum

heart rate range from 98 to 185 for Value 1, and from 120 to 188 for Value 3. Therefore, we conclude that the mis-classified attributes does not confine to a certain range in any attribute.

However, we observe that both Value 1 and Value 3 have significantly less instances compared to almost perfectly predicted Value 2 and Value 4. This opens up a potential solution for us to improve the accuracy, that is to increase the data size and train with more instances. The ideal scenario would be having almost equal number of all instances in the dataset.

3.4 Chapter Summary

In this chapter, we have developed a machine learning classifier model to analyze the risk level of patients having heart disease. The data is assumed to be collectible by wearable devices and we apply the well-known heart disease dataset from UCI. One advantage of using the dataset is that the expected output label is known and provided, so we can determine the precision and recall of our model easily. In fact, we have proved that the attribute selection with supervised methods that make use of the output labels are more accurate than the non-supervised approaches that do not. In addition, PCA showed that attributes are almost independent to each other. We report our final highest accuracy to be 88.1% with 10-fold cross validation with decision tree, followed by 85.0% using Support Vector Machine.

However, due to the lack of the expertise in this area. Future work is needed to find out the reason behind the misclassified classes. Also, we need more data to train our model since current dataset is not evenly distributed, with almost $10:1$ unbalanced ratio. This can be done by collecting data from real patients or create synthetic data with the help of the cardiologists.

Due to the existence of the mis-classified patients, especially Value 1 and Value 3, we need to ask online doctors and caregivers to look at patients' data manually in E-health Cyber Ecosystem. This step would help patients to confirm the risk levels and enable online doctors/caregivers to provide medical suggestions. This leads to Phase 2 in next chapter.

Chapter 4

Risk Level Confirmation

Phase 2: Online Queueing for Risk Level Confirmation

4.1 Overview

Since there exists risk level prediction errors in Phase 1, in this chapter we queue high risk patients with online doctors/caregivers to confirm the risk level and provide remote care. Here we devise a queueing model that arranges the matching between patients and online doctors/caregivers. Further, we analyze the model behavior under different arrival distributions: Poisson, Normal and Multimodal distributions. Poisson and Normal distributions are commonly considered in most publications, and the reason for choosing Multimodal distribution will be discussed in Section 4.3.4.

Among all the mentioned work in Chapter 2, only normal arrival distribution and Poisson arrival distribution are considered. However, other than bulk arrival, we can expect to see multiple peaks of arrival in a day. For example, assume that patients' risk level may arise during lunch time due to the digestion of sugary food. Then it is possible to see multi-modal distribution for doctors' end because a cyber space serves the entire United States that spans over multiple time zones.

As a result, this assumption would lead to multiple peak arrivals during the noon.

And most importantly, to reduce the wait time and increase the queueing capacity, the idea of turbulence flow is used because it carries more energy than that it laminar flows which is commonly used for well-scheduled systems.

4.2 Technical Details

With the queueing models mentioned above, we realized that a robust queueing model is necessary before increasing the capacity handling by applying the turbulent flow idea. To be specific, the characteristic numbers have to be determined that represent the transition between laminar and turbulent flows as well as turbulent and chaotic flows. Instead of applying Reynolds Number and Lambda 2 [36], two major techniques used in fluid dynamic analysis, a flexible model is developed based on the systems server utilization, and thus a reflection of systems chaotic level.

In the literature review, many dynamic models were created by changing the number of servers within a fixed time, based on patients flow. However, the service rate has not been considered. For instance, different doctors can have different speeds in recognizing risks, which results in different service rates as well. This work explores server utilization as the optimization factor. This model does not only focus on the standard measurement of how often do patients arrive into the cyber queue and how quickly they are served, but also aims to develop a method to simulate arrival time that covers three different distributions: Poisson, Normal and Multimodal distributions.

This model is developed based on the publicly available online customer arrival distribution dataset. We also test the model with simulated arrival distribution data

and the model is ready for use for real-time data. Due to the data confidentiality, no active collaboration has been established with medical institutions/organizations for the time of writing this portion. However, the follow up work can be expected from other PhD candidates in the lab.

Most importantly, it has been successfully identified that the server utility rate can be used as an impacting correlation factor for the Lyapunov characteristic number.

4.3 Material and Methods

4.3.1 General Framework

4.3.1.1 General MATLAB Framework

In this phase, the author have mainly developed the model with MATLAB and Simulink software. The MATLAB project has been divided into 3 hierarchical levels: 1) The basic class definitions that contain all the information of patients, doctors, and caregivers 2) The middle level functions that reads or generates arrival data and service data for use 3) Top layer script that accommodates the whole queueing process and calls all subroutines such as public/secret key verification, latency test, list online doctors, *etc*.

4.3.1.2 SIMULINK Flow Chart

SIMULINK is mainly used as a schematic and graphical representation for the whole queueing process. As shown in Figure 4.1, the patient arrival distribution data is read from the MATLAB function, and is stamped before going into the queue switch. There are three discussed distributions: Poisson, Normal, and Multimodal. Each arriving patients will be paired with either a doctor or a

caregiver, based on his/her complexity of medical background. This is determined by a switching algorithm. The switching algorithm is based on the historical medical data of patients, such as history of having heart attack within last 3/6/12 months.

In this case, we assume that the doctors and caregivers provide two performance levels of diagnosis reconfirmation, that the doctors will be mainly responsible for patients with complex medical history, and caregivers should focus on reading data of the patients with simple medical background. Also, if a caregiver cannot make a confident decision about a patient, this data will be forwarded to a doctor. This setup of doctors and caregivers is parallel with the current service provider situation in most traditional hospitals, and represents two different service qualities.

Moreover, the behavior of two servers: doctors and caregivers are very similar. When patients are in the different cyber queues, the availability of online caregivers and doctors will be checked by external MATLAB functions. The queueing block has its own flags to check if the system is ready and if the doctors are online. The flags are checked by external MATLAB functions and once done, the messages will be sent to indicate success or failure. When doctors and caregivers are ready, the multi-server queues of both doctors and caregivers start to operate properly. In addtion, both queueing blocks have priority sub-queues to give priority for level 3 patients, since their risk of having a heart attack is higher. Therefore, priority queues give higher priority to patients with Level 3 risk, and gives lower priority to level 2 patients. Furthermore, a input switch is added before priority queue for doctors' end. This input switch takes both inputs from doctors' general queue and the output from caregivers. This feature enables caregivers to forward complex data to doctors if s/he cannot confirm the risk level.

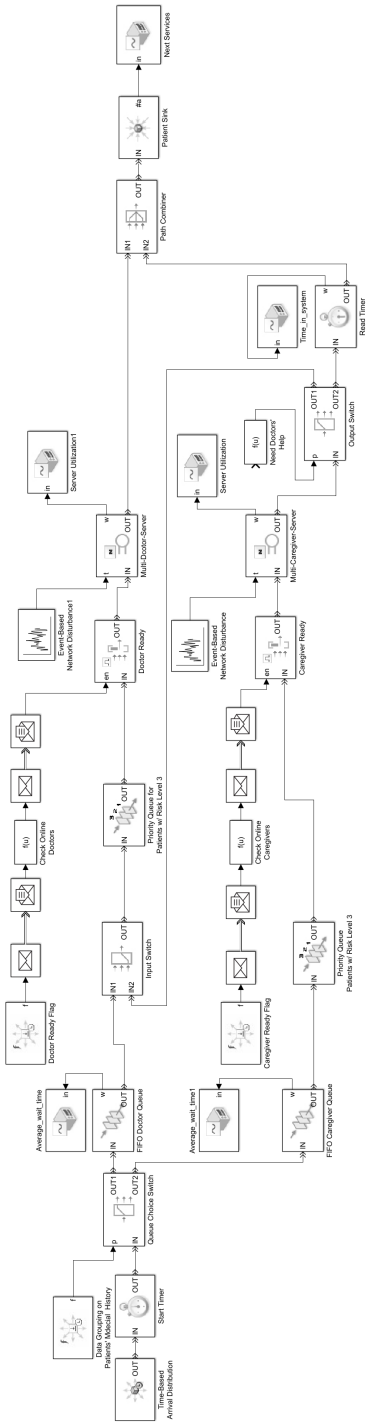


Figure 4.1: A Queueing Model with SimEvent

The server block for doctors and caregivers are similar as well, as they both take in the patients' data from priority queue and receive event-based network disturbances. For instance, the network package drop, network latency, gateway verification and other issues can lead to unexpected delays. These are modeled as event-based entity generator feeding into the server. Each server is currently considered as Markovian arrival rate and Markovian service rate: $M/M/N$, where N is the number of servers. Also, each of their utilization rates can be checked separately. In the end, all patients will take another time stamp when leaving the check-in queue. By adding more timers, the patients' total time in system, waiting time and service time become available to us.

4.3.1.3 Simplified MATLAB Framework

After generating a complete MATLAB general model and SIMULINK model, the actual dynamic behavior became urgent for author to examine. Therefore, another simplified online model has been developed for validation purpose by removing all unrelated subroutines, such as checking network connections. Then the updated model objective becomes the following:

- Generate patients' arrival time for the model;
- Change the number of servers dynamically to optimize the server utilization rate;
- Can be used to apply to other disease models (*i.e.* Diabetes).

4.3.2 Flow Logic

The overall flow logic is therefore simplified, as shown in Figure 4.2. Since there are no medical queueing data publicly available online and most researches assume

Poisson arrival distribution [53], the simulated data is used here and will be discussed in detail later. The simulated arrival data for all the patients with level 2 or level 3 risks are written into a csv file first. In csv file, each row represents a patient and has information about the patient: age, gender, heart rate, risk level, *etc*. These information are critical for doctors' diagnosis. With different levels of risk, the incoming patients will be lined up with respect to their priority. Also note that, each patient has been classified with a binary variable into either complex or simple medical background. This information can be updated each time after a doctor or a caregiver finishes examining the patient's record. Likewise, the priority level is also a binary number to differentiate level 2 or level 3 risk.

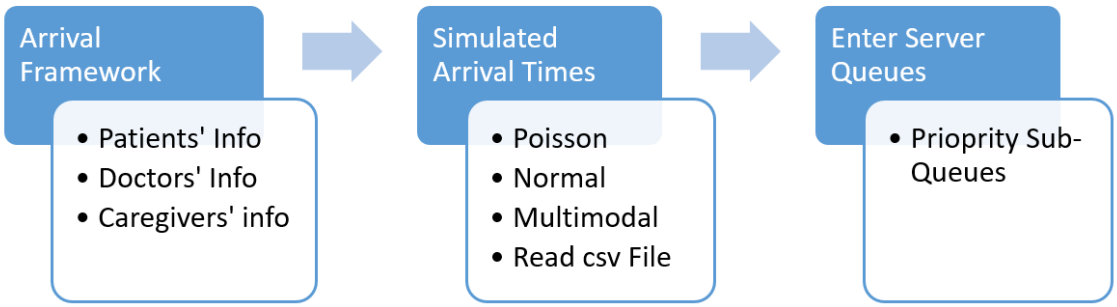


Figure 4.2: A Flow Chart of the Simplified Logic

With the generated input file for MATLAB model, the arrival distribution data is then fed into the server queues, once they become ready. For simplicity purpose, here we assume that each server has a maximum capacity of 3 servers and the queue size is not limited. Server 1 will calculate the time of patients spent in the process and will assign an end service time value for each passenger. This value will serve as the arrival time for next patient where the same process will occur. This continues until the end of Station 3. To be specific, all of the patient arrival data is fed into the station and it iterates for each patient based on arrival time.

The logic for this model is that the current patient i cannot start their service time until either server becomes available. In the case that there are n servers present, the program will check if the patient n spaces ahead $i - n$ has ended their service time.

Once a patient starts the service, the patient is fed into the next server (if the current server is full) or stays in the first place in the corresponding service queue if all servers are full. The total time in the system is calculated from the difference between the initial arrival time and the time they leave the system.

Remark 1. *One advantage of this setup is that, patients do not need to commit to any specific doctors/caregivers, and can be serviced if any doctor/caregiver becomes available.*

Further, this model is dynamic because it adjusts the number of servers per station based on the stability of the system. Here, the stability of a queue is determined by server utilization which is defined in Equation: $\rho = \gamma/\mu$, where ρ is the server utilization, γ is the average patient arrival rate, and μ is the average service rate. If $\rho > 1$, the average arrival rate is greater than the average service rate and the system must adjust by adding a server. If $\rho < 1$, the service rate is greater and the system can work less by removing a server since it is currently stable. The model uses a WHILE loop to iterate for each patient in the dataset. At the end of each loop, it checks for the value of ρ and adjust the number of servers if needed.

Remark 2. *This queueing model is dynamic because it adjusts the number of servers based on the stability of the system. The server utilization rate is then our Lyapunov Characteristic Number.*

4.3.3 Service Time Distribution

The service time for each patient is calculated by using distributions simulated based on medical background and severity level by the program subroutines, which greatly differ for each doctor or caregiver. Again, here we assume for Markovian service rate, thus we have $M/M/N$ queues. In some cases, a passengers' data will likely to cost more time for a doctor to read, if his/her medical background is complex or severity level is high.

4.3.4 Arrival Simulation

As suggested in [47, 53, 67], researchers commonly focus on Poisson distributed arrival rate, normal distributed arrival rate, or the modified versions whose centers are slightly shifted. Here, we randomly pick an online customer queueing statistics and compare both Poisson and normal distributions to mimic the function shape.

4.3.4.1 Arrival Data and Shape

Here, we start by looking at given datasets from Mic [45] and Sayarshad [63] both of which have distributions with multiple peaks. Mic's data is more intuitive as shown in Figure 4.3, so it is our goal of simulation. It is assumed that the arrivals are independent, therefore the distribution is exponential. In order to achieve the shape of the given data, both Poisson distribution and normal distribution are simulated to mimic the data.

In Figure 4.3, it is clear that multiple peaks can be identified. After applying the *histfit* under statistic and machine learning toolbox, a corresponding normal distribution of histogram is shown in Figure 4.4.

Although peaks and rush hours can be easily identified, the distribution func-

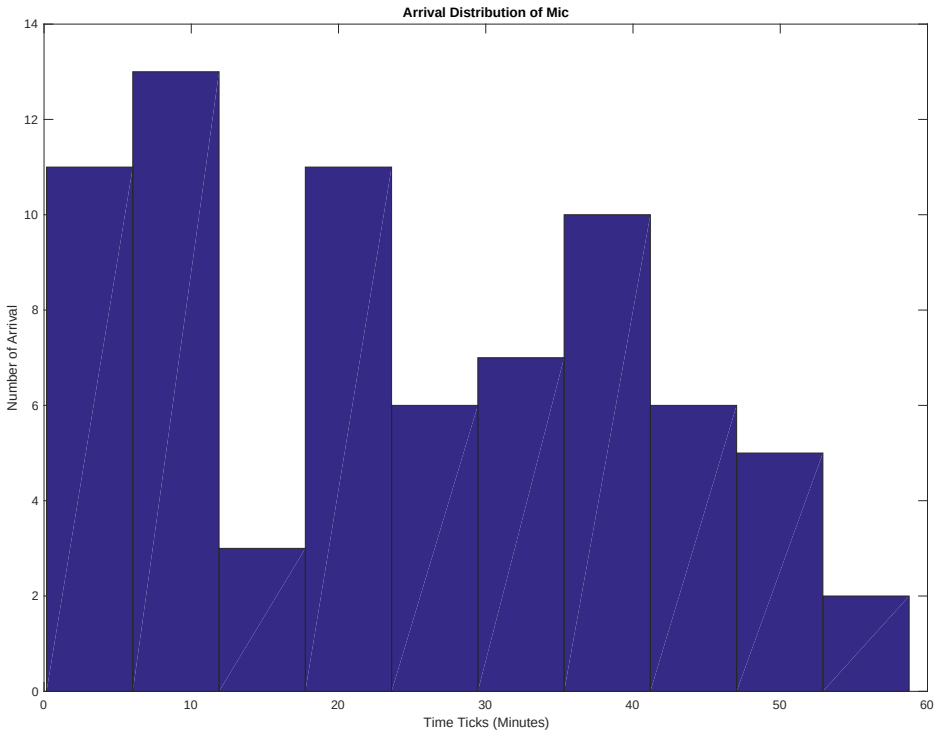


Figure 4.3: The Histogram of Real Data [45]

tion is discrete, therefore lacks the capability of representing the continuous arrival situations in real world. Also, in order to examine the potential of transferring the well-scheduled laminar process to a more energy-carrying turbulent process, a continuous system is needed for fluid dynamic analysis.

With multiple attempts using different lambda values in Poisson distribution, or normal distribution with different means and variances, neither seems to capture the muti-peak behavior thus lacks the indication for system to dynamically realize the queueing load. Although the normal distribution cannot be used to generate multiple peaks, and the Poisson distribution does not span over a lot of domain of arrival time, they are still used to test the behavior of the online model.

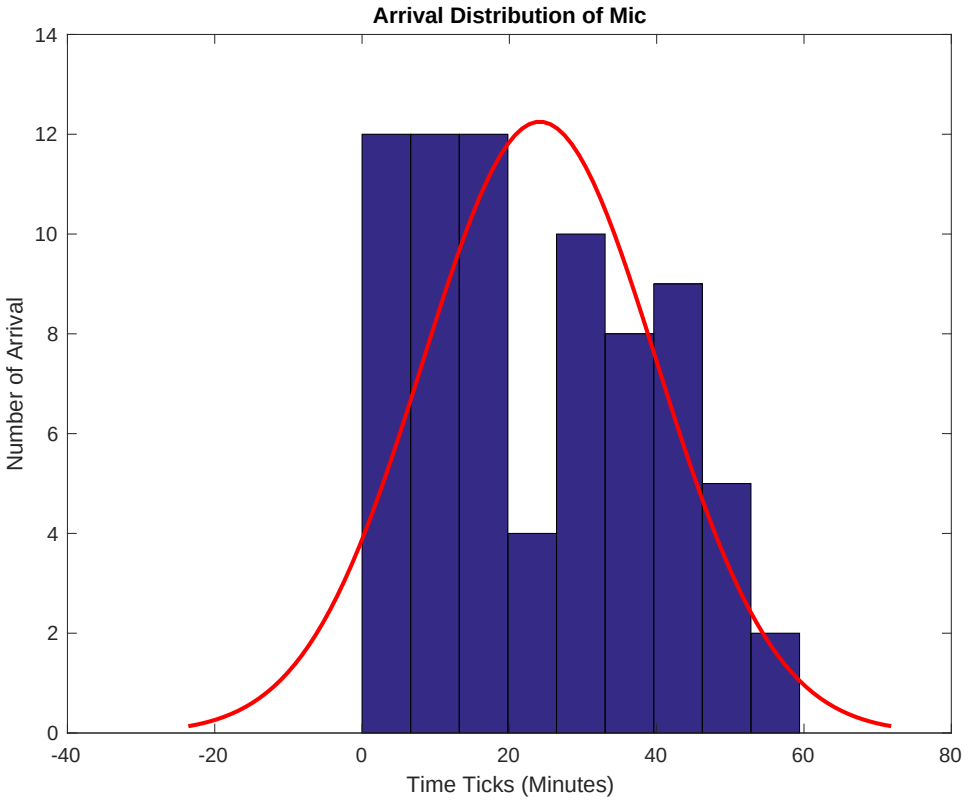


Figure 4.4: A Histfit for the Real Data

4.3.4.2 Multimodal Arrival Distribution

In order to have a multi-peak property of the arrival distribution, author has adopted the Multimodal Gaussian function to generate the distribution with more than two peaks, as shown in Figure 4.5.

In this simulation, the number of the patients is set to 100, the means of arrival time (in minutes) are set to 10, 30, 50, and the variances for these three peaks are 2, 3, 7, respectively. Therefore, the function generates random variables from a mixture of 3 Gaussian distribution. Although not perfectly matching all the spanning and height of the given distribution, the major peaks have been identified and ready to be used for testing the system behavior.

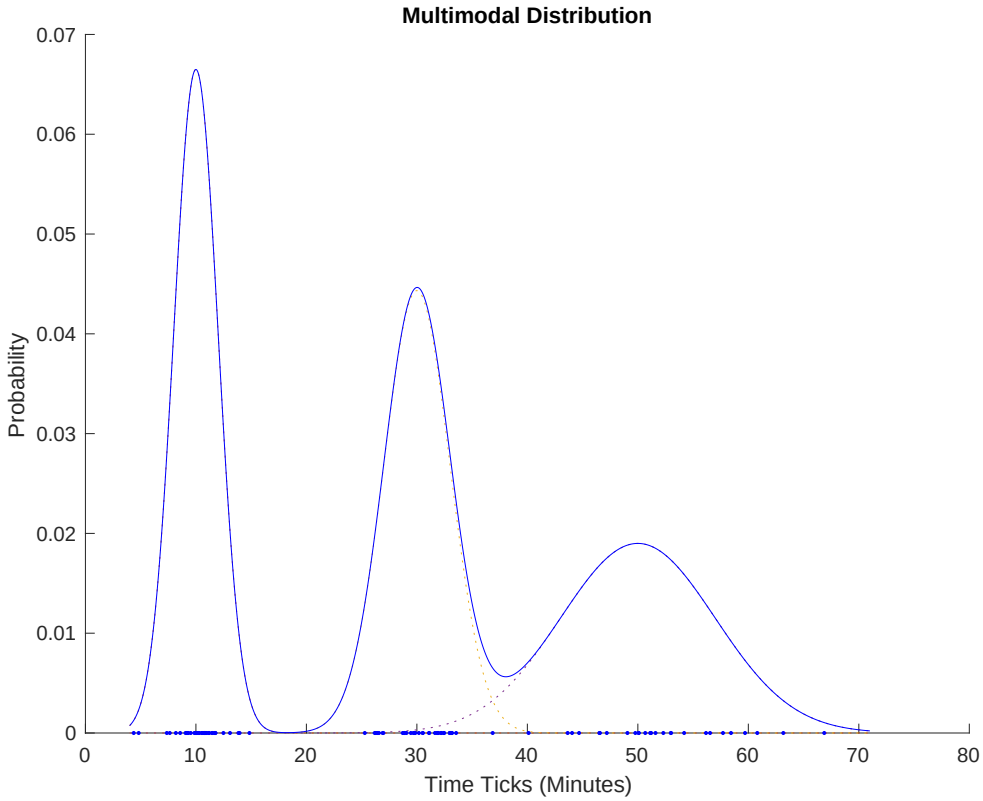


Figure 4.5: A Multimodal Distribution

4.3.4.3 Poisson Arrival Distribution

Since the range is very dense (within an hour), it would be a good course to determine if a continuously busy period would drive our model unstable. The distribution is given in Figure 4.8 and Figure 4.9. Note that the number of the patients is set to 100 and the lambda value is set to 40.

4.3.4.4 Normal Arrival Distribution

The data of normal distribution is generated as well in Figure 4.10 and Figure 4.11. Note that the number of the patients is set to 100, the mean of the distribution is set to 40, and the standard deviation is set to 15.

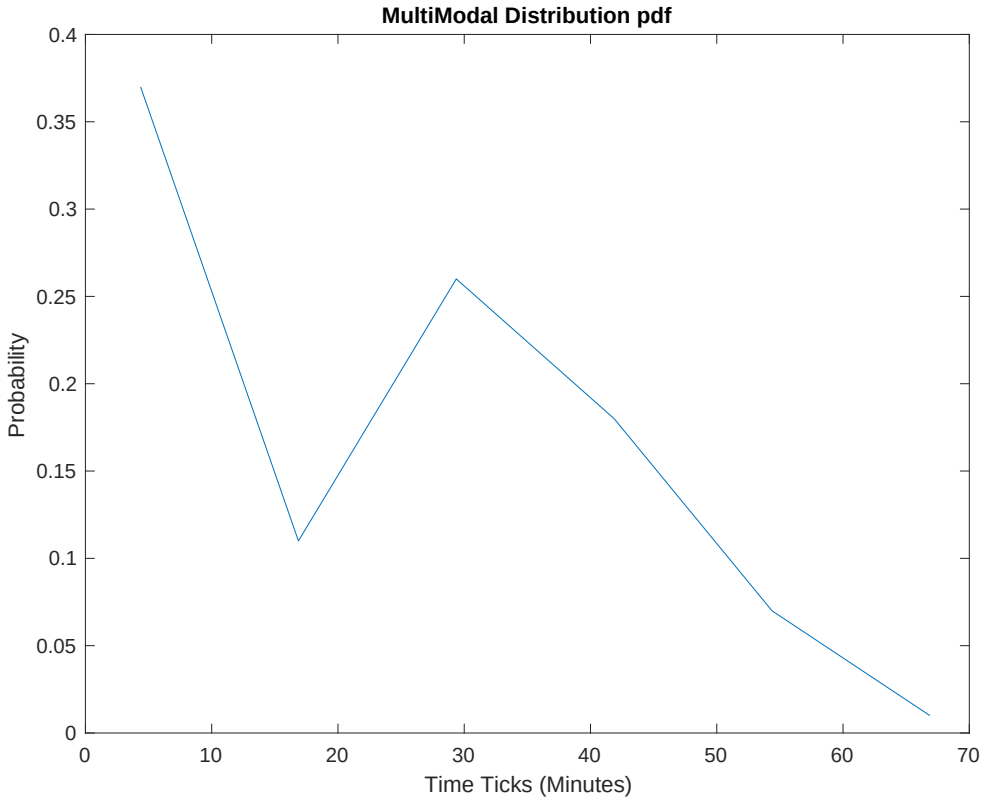


Figure 4.6: A Multimodal Distribution PDF

4.3.4.5 Scenario Analysis

With the simulated arrival distributions shown in previous part, we can see that different scenarios can be achieved using different simulation algorithms. To test the response of the model to various types of arrival data, the different arrival distribution simulations were input into the model as different scenarios, aka, simulations for the Multimodal, Poisson, and Normal distributions created in the previous section. The datasets generated by each scenario will be used as inputs into the program and compared to understand the system response. The results does not only show how the system time for each patient changes with difference arrival distributions, but also how the dynamic nature of the servers in the model

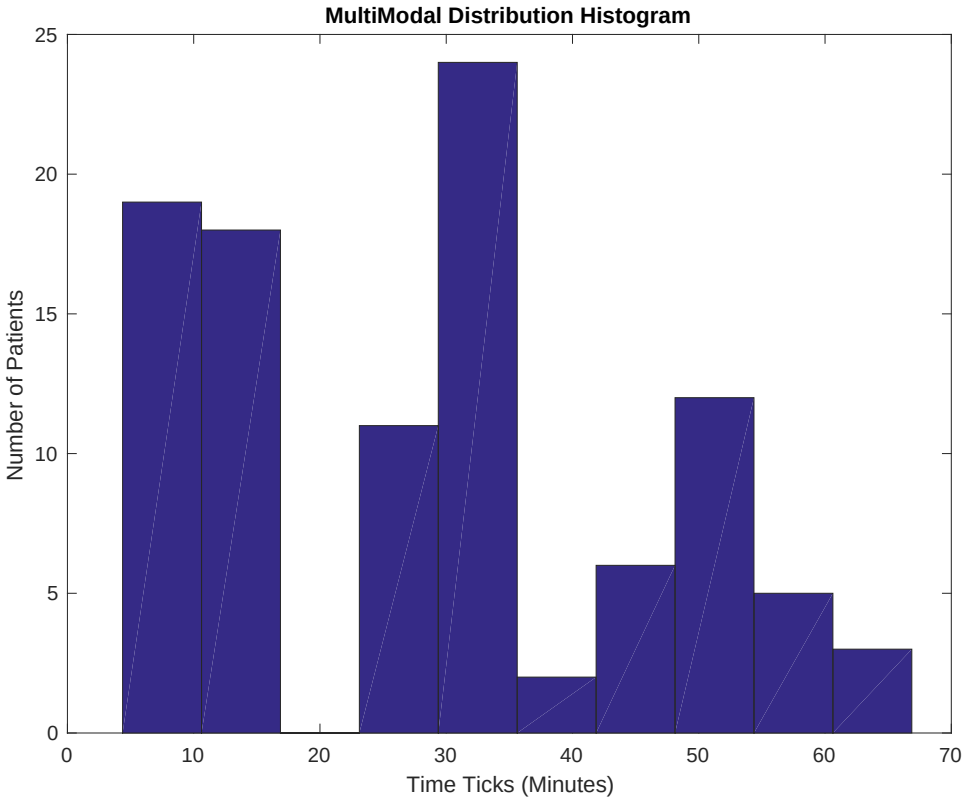


Figure 4.7: A Multimodal Distribution Histogram

can impact the outputs.

4.4 Results and Discussion

4.4.1 Time Spent in System

The total time spent in system for all 100 patients are shown in Figure 4.12, 4.13, 4.14. There are occasional long time spent in system which are either because they gave up their priority to high risk patients, or modeled as randomized uncertainty due to network disturbance, i.e., network latency, package drop, etc. Due to the dynamical increase and decrease of the server number, this model stays at a

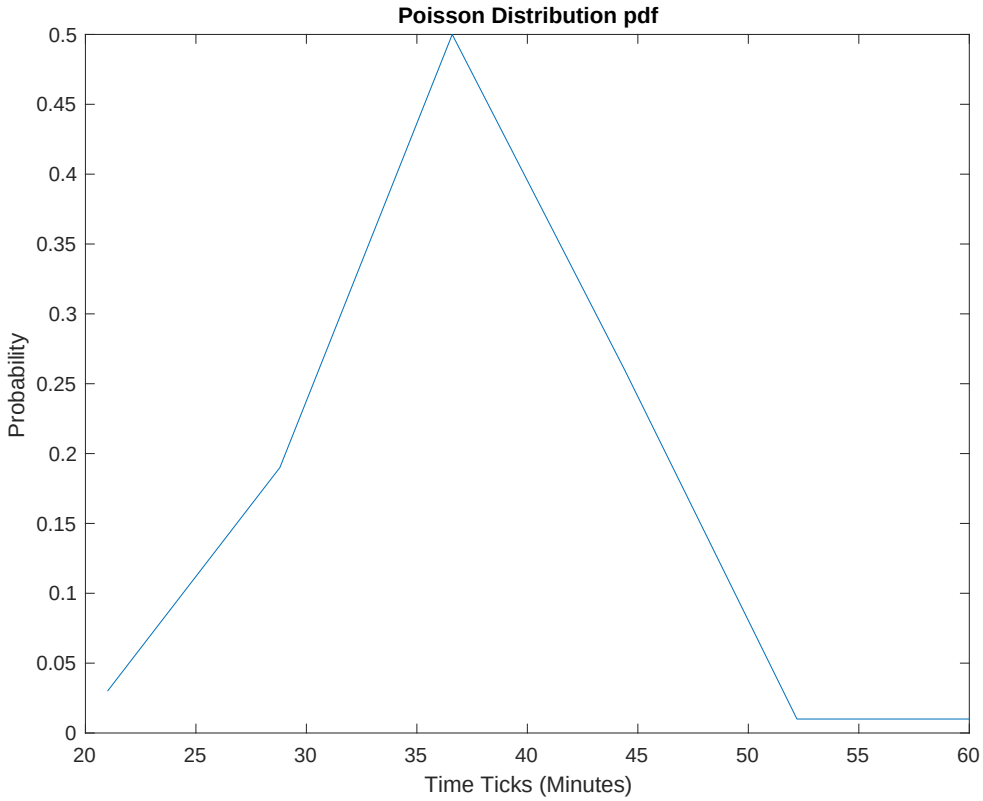


Figure 4.8: A Poisson Distribution PDF

satisfactory level of average time spent in system.

4.4.2 Number of Servers Per Passenger

Figure 4.15,4.16, 4.17 illustrate the number of servers as number of patient increases. It is clearly shown that the number of servers reaches its maximum during times where the arrival rate is dense which is mirrored in the Figures of *Time Spent in Queue*. Note that each increase in number of servers are aligned with the increasing trend of the number of arrivals. The total time in system does not become high during the center, but the arrival density increases which causes the demand for servers to be much higher. The increase in servers counters the increased arrival

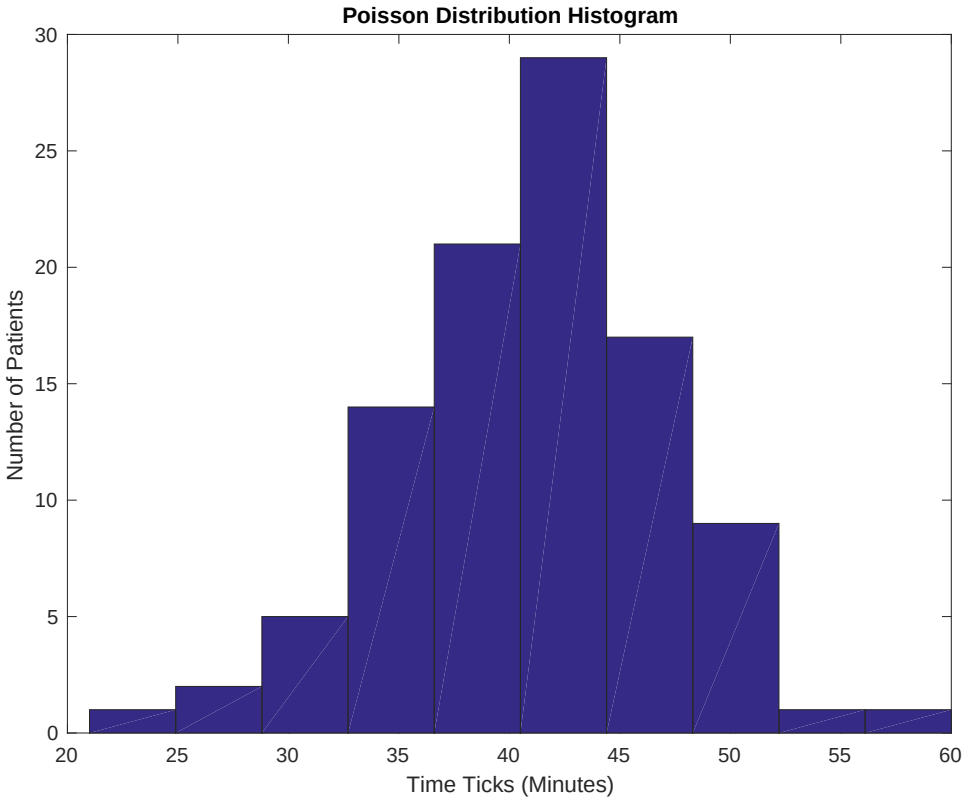


Figure 4.9: A Poisson Distribution Histogram

time at the center which is the reason for the total time in the system as shown that does not change significantly. It is clear that this developed queueing model is dynamic.

4.4.3 Server Utilization Per Passenger

In this work, the server utilization rate is identified as Lyapunov Characteristic Number, as mentioned in previous sections. The Figure 4.18, 4.19, 4.20 illustrate such ratios. The server utilization determines whether the number of servers needs to be changed to meet the system demands. The volatility of the server utilization ratios are all kept under 1 since the adjustment of number of servers is meant to

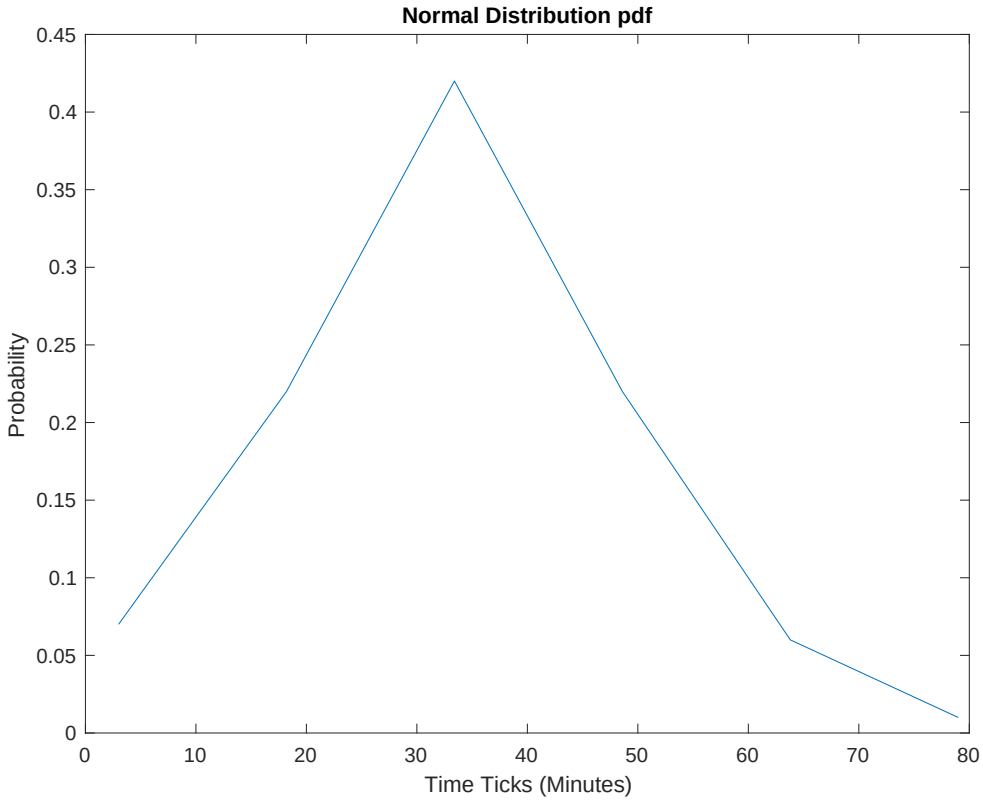


Figure 4.10: A Normal Distribution PDF

keep it at that value. Note that this is for illustration purpose and this threshold can always be changed. In the case when a server needs to take a break, the decrease in number of server will be reflected to utilization ratio and thus notify other standby servers to go online. The low utilization ratios at the beginning are due to the fewer arrival numbers, but with more passengers, the system levels out.

4.4.4 Time Spent in Queue

The average time spent in queue are very low, shown in Figure 4.21, 4.22, 4.23. Also, the mean waiting value of this model for Poisson distribution is 0.3576 minutes with random network uncertainty values added. The average waiting time for

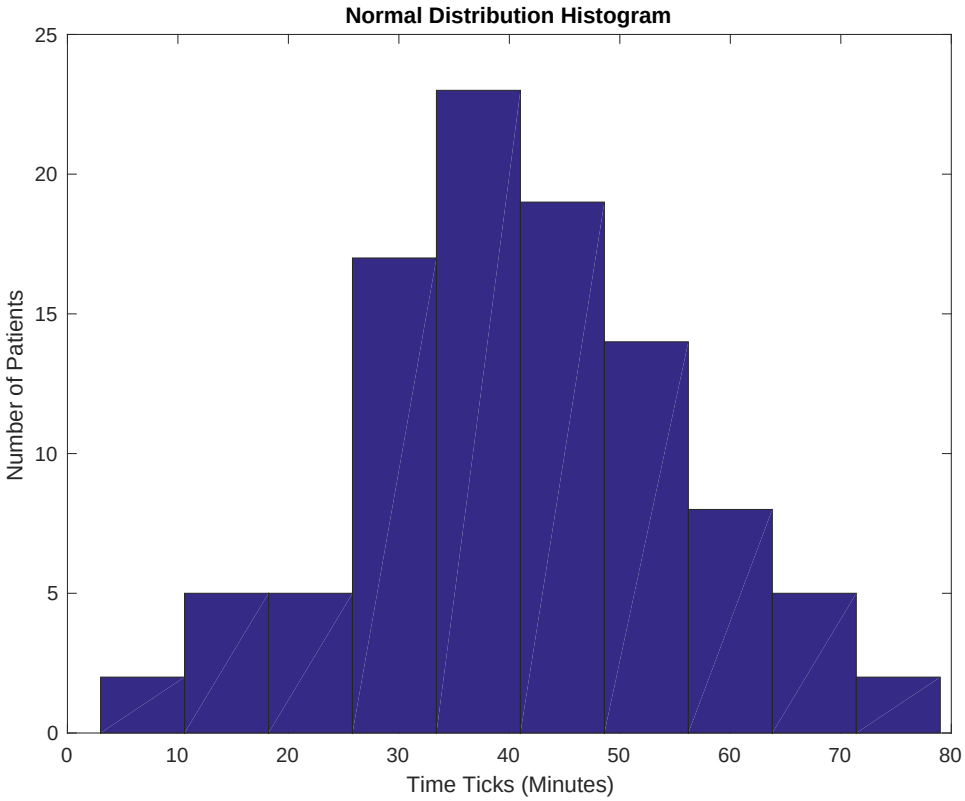


Figure 4.11: A Normal Distribution Histogram

level 3 patients is 0.1031 minutes.

The mean waiting value of this model for Normal distribution is 0.2848 minutes with random network uncertainty values added. The average waiting time for level 3 patients is 0.0703 minutes.

The mean waiting value of this model for Multimodal distribution is 0.3140 minutes with random network uncertainty values added. The average waiting time for level 3 patients is 0.0927 minutes.

Without validating the actual service rate of doctors, we can clearly see that patients with level 3 risk spend far less time in queue, almost 1/3 of the time-in-queue of patients with level 2.

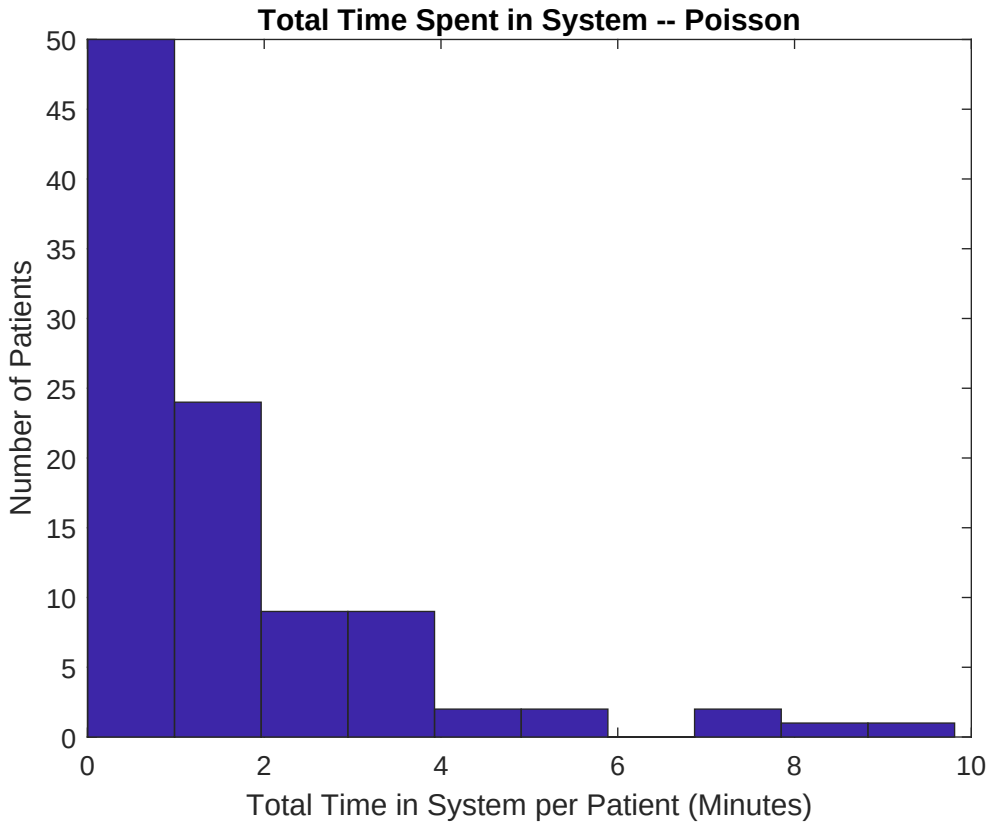


Figure 4.12: Time Spent in System Histogram – Poisson

4.4.5 End User Application

The results and discussions in previous section show very important and valuable information for both doctors and patients. The dynamic nature of this queueing system allows the doctors, caregivers and healthcare providers to use it as a decision support system for efficient use of the human resources. Adjusting server number by server utilization enables the healthcare provider to anticipate the instability of the system before it gets too high as opposed to adjusting the servers by maximum queue size. At the same time, the healthcare provider can cut down on work hours or energy costs if the system is stable.

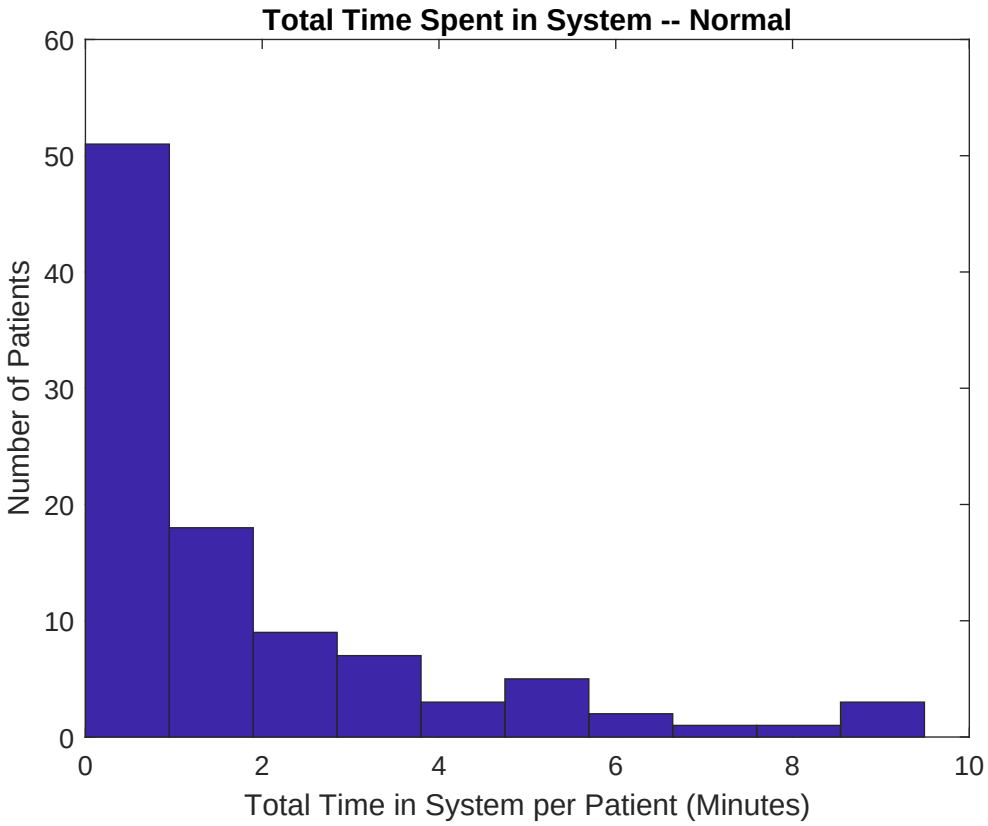


Figure 4.13: Time Spent in System Histogram – Normal

4.5 Chapter Summary

In the Phase 2, both a detailed general model and a simplified model for the online E-health queueing environment are developed. This model would better help the healthcare providers to use medical human resources efficiently. We have identified the characteristic number of the model, the server utilization, as the index number of suggesting the stability of the servers. We have shown a dynamic model that determines the optimal number of online doctors/caregivers at a given time that works for different arrival distributions. This can greatly help healthcare providers to recognize the patterns and optimize the duty shift to provide the optimal service.

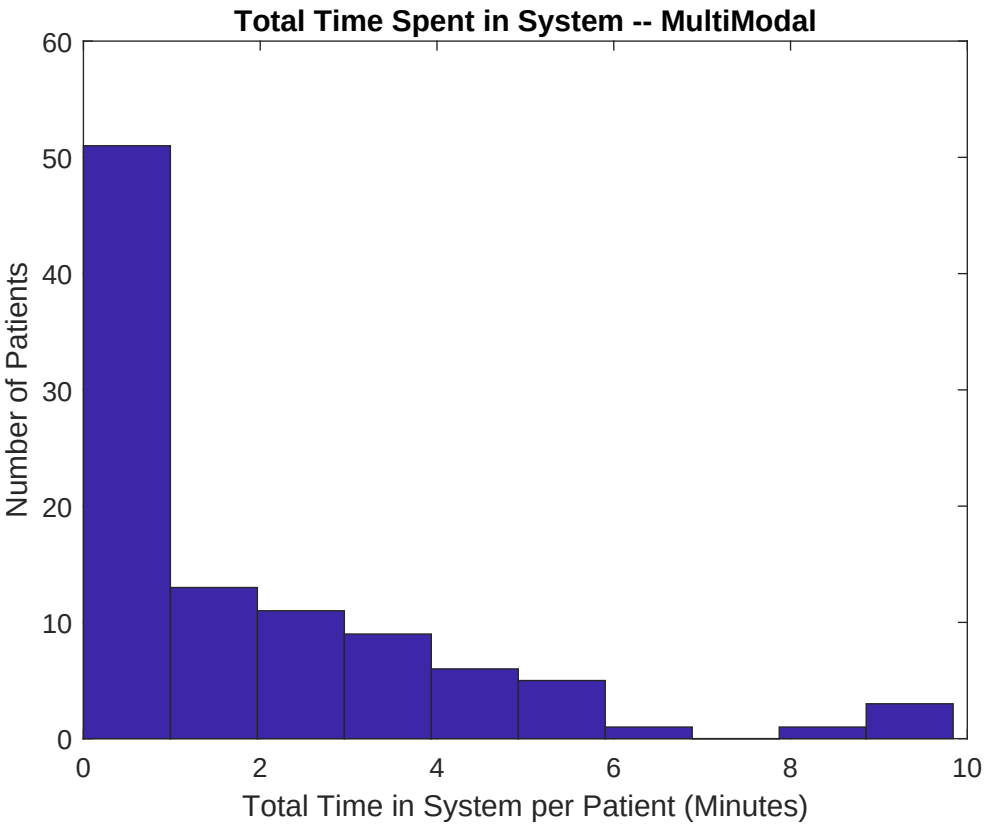


Figure 4.14: Time Spent in System Histogram – MultiModal

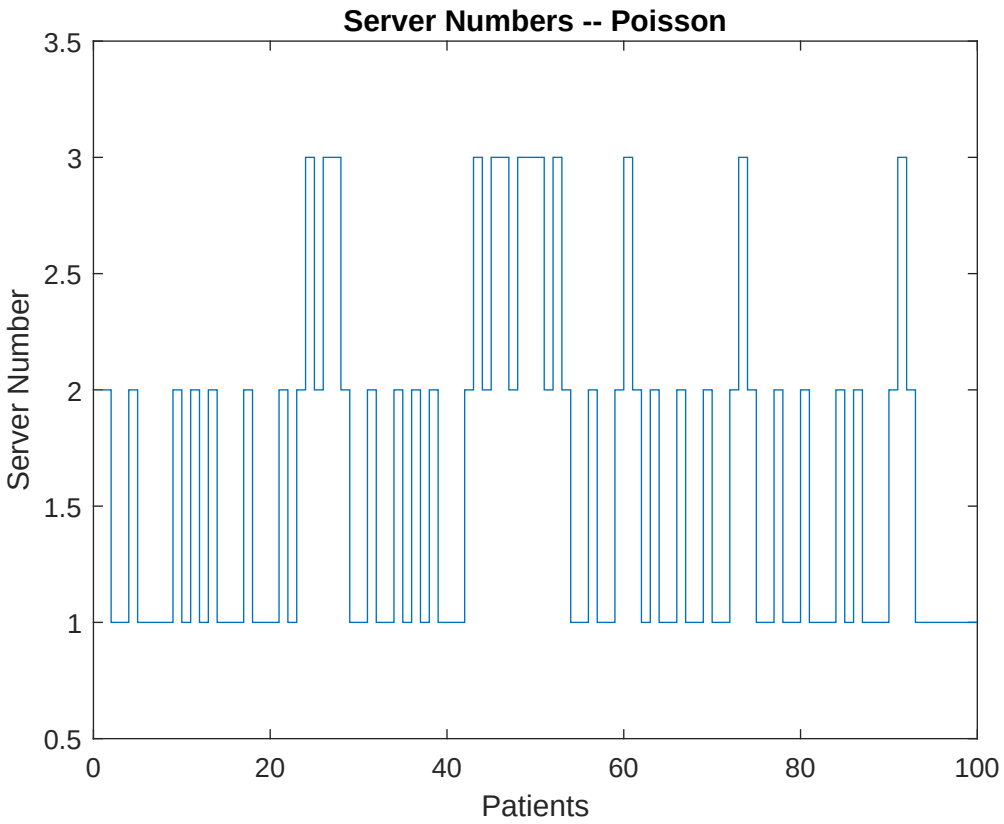


Figure 4.15: Number of Active Servers – Poisson

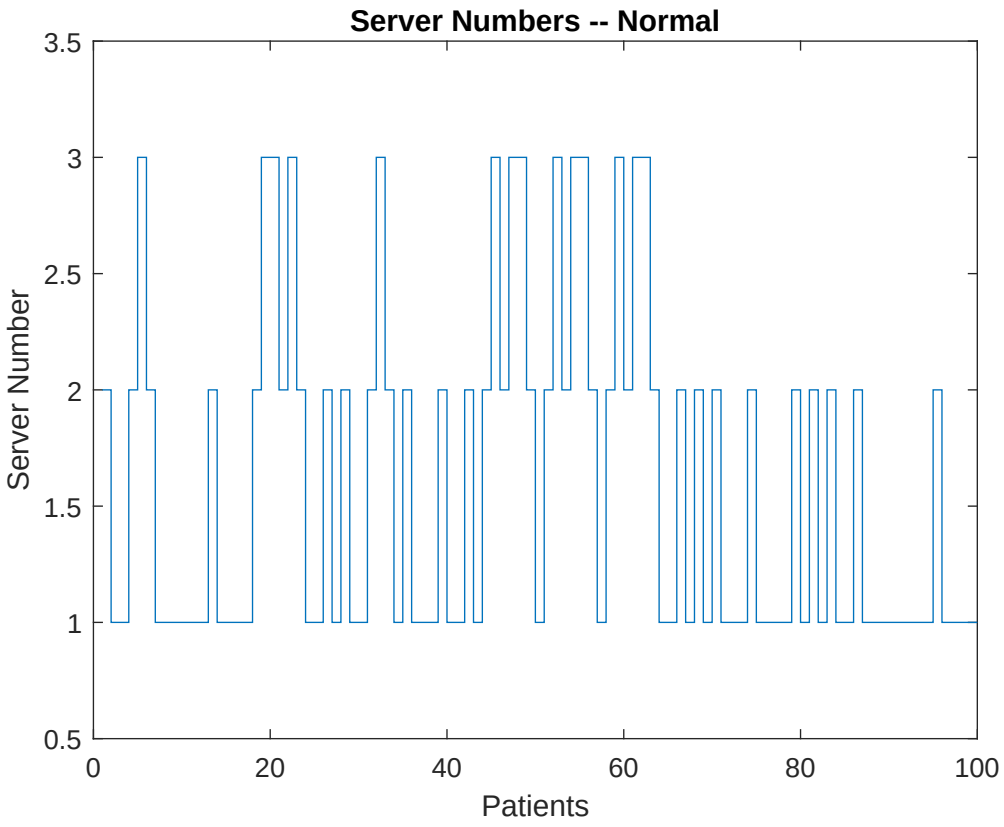


Figure 4.16: Number of Active Servers – Normal

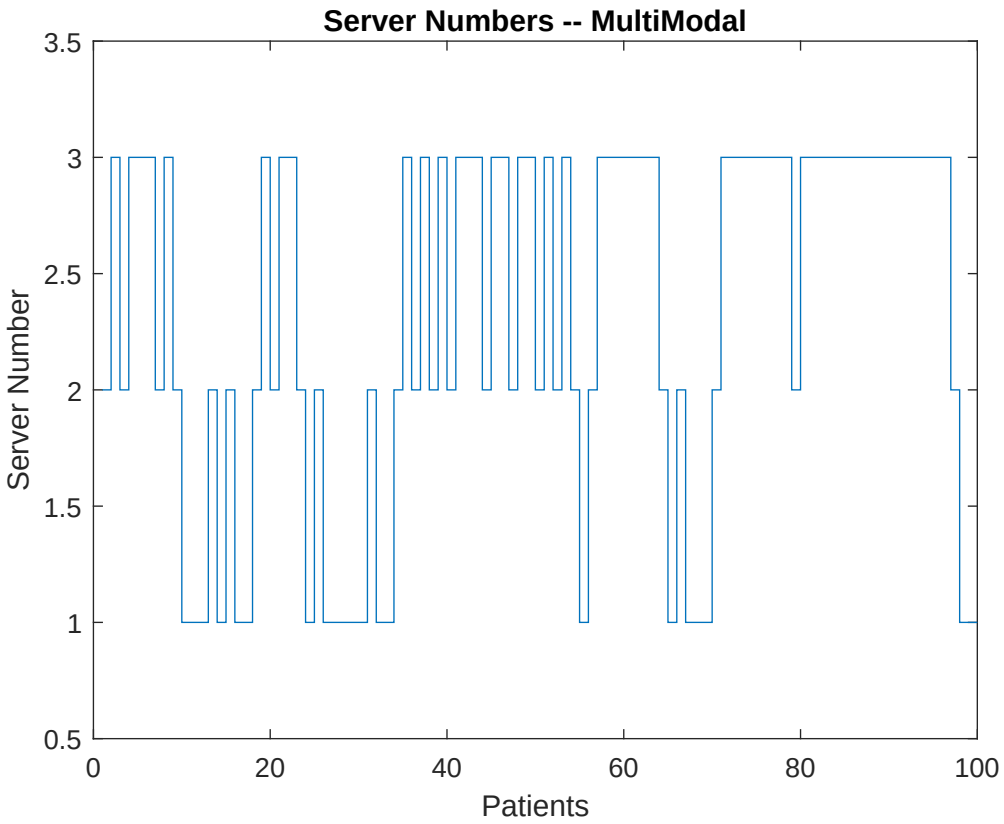


Figure 4.17: Number of Active Servers – MultiModal

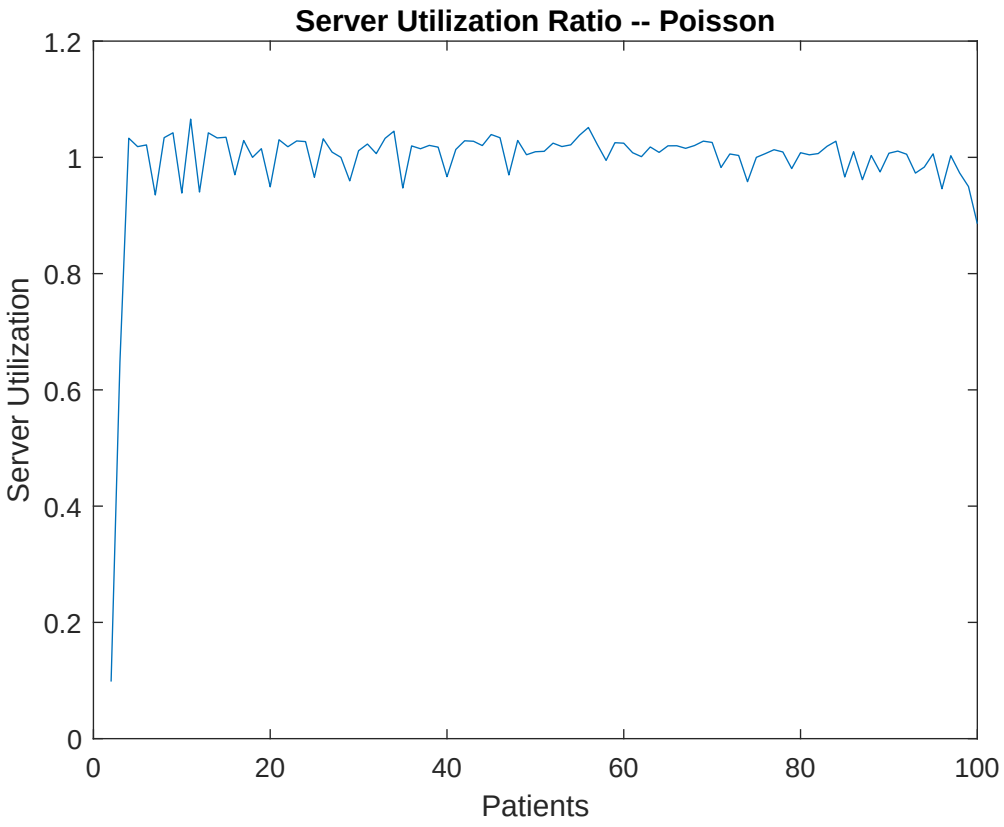


Figure 4.18: Server Utilization – Poisson

Figure 4.19: Server Utilization – Normal

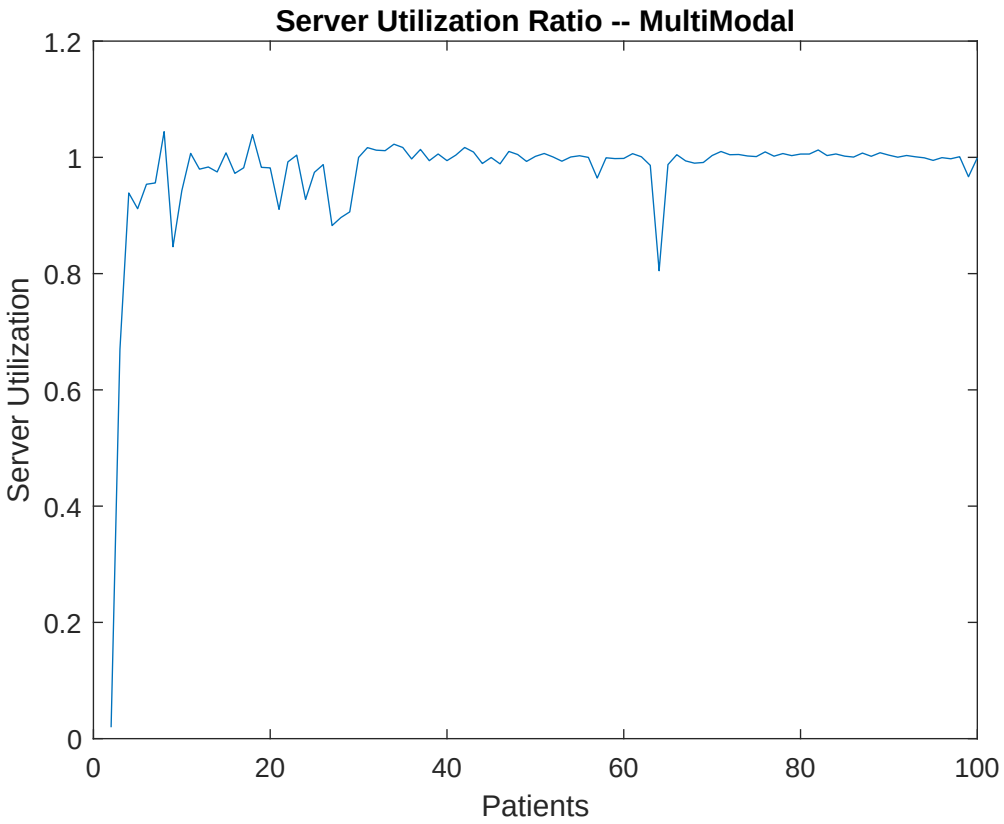


Figure 4.20: Server Utilization – MultiModal

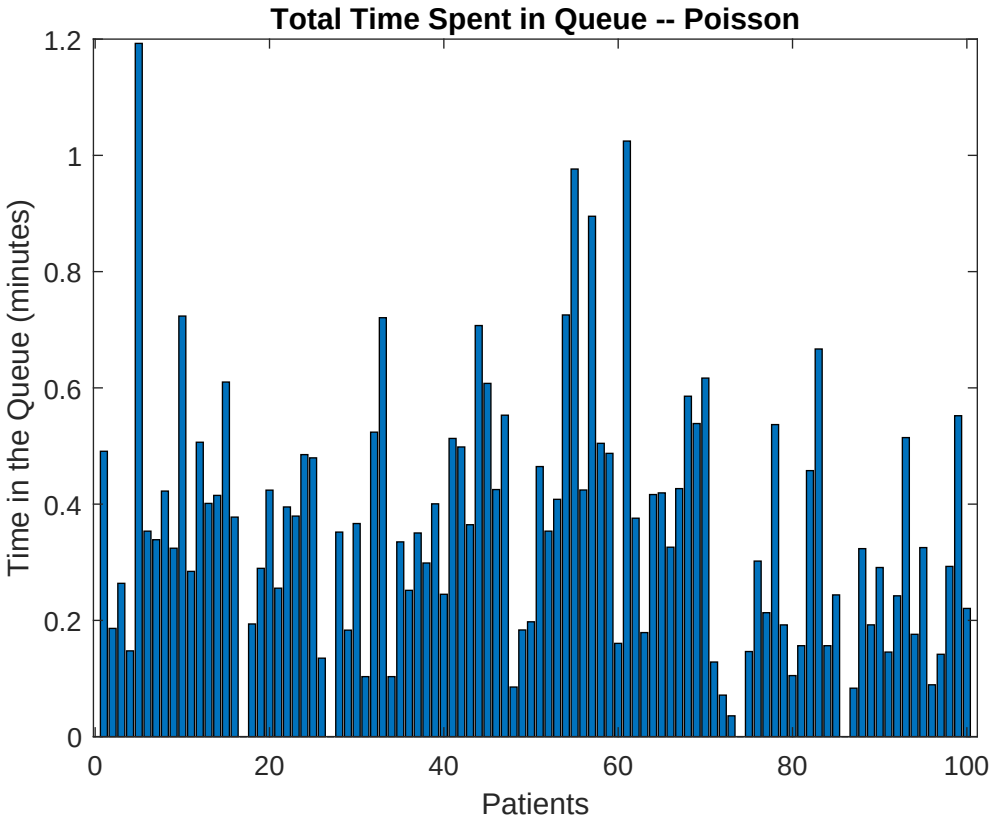


Figure 4.21: Time Spent in Queue Histogram – Poisson

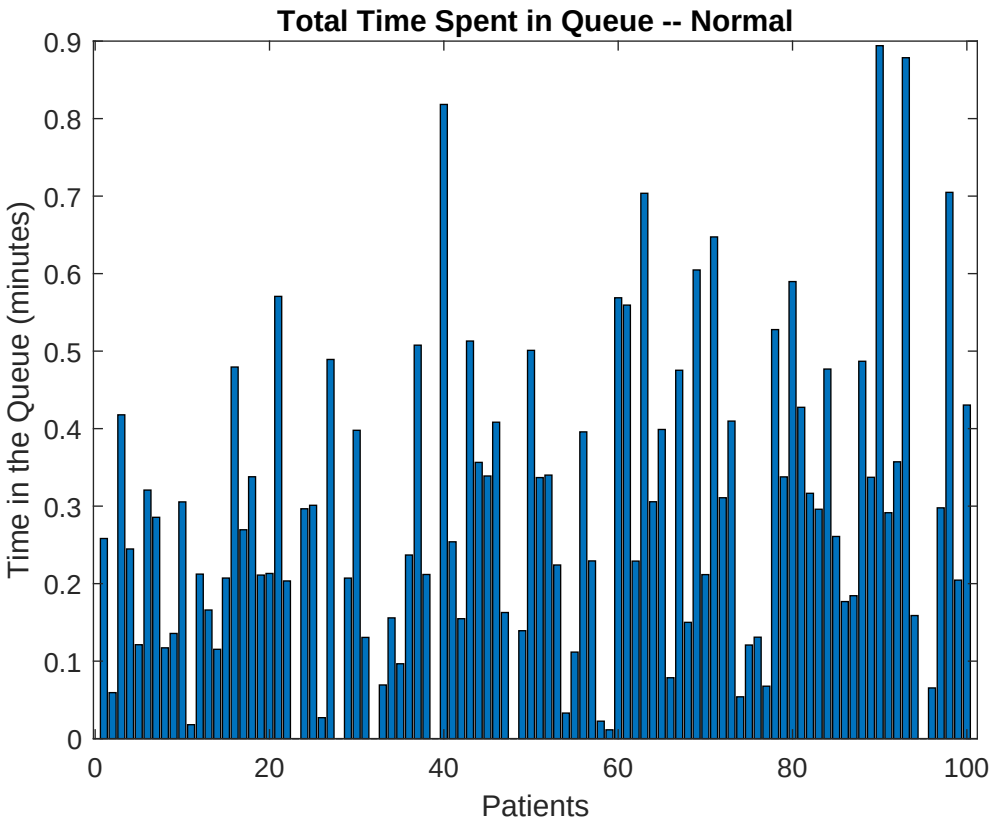


Figure 4.22: Time Spent in Queue Histogram – Normal

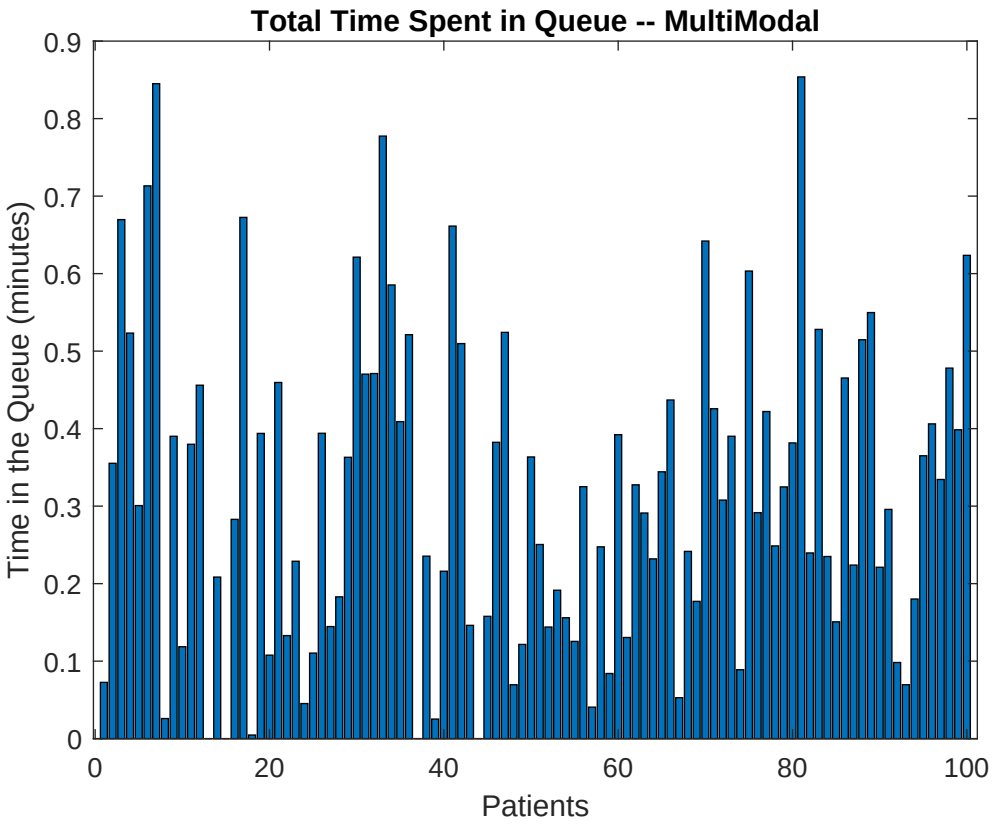


Figure 4.23: Time Spent in Queue Histogram – MultiModal

Chapter 5

Regional Health Alert Level Decision Support

Phase 3: Decision Making on Regional Healthcare Alert Level.

For the last two chapters, we have covered the expected behavior of two most involved actors in an E-health Cyber Ecosystem, as well as the corresponding frameworks for both parties. In this chapter, we look into an extremely important but often neglected party in this ecosystem: the healthcare providers. Healthcare providers can commonly infer the following involvers: Hospitals, Nursing Homes, Home health agencies, Dialysis facilities, Inpatient Rehabilitation Facilities, Hospice Agencies, *etc*.

Although decisions are needed for all the mentioned individual healthcare providers, in this thesis, we mainly focus on a greater scale and develop a decision making scheme for regional healthcare providers. As describe in Chapter 2, The National Association of Strategic Regional Healthcare Organizations (SRHO) is one of such examples that this decision making model supports.

In this chapter, a decision-making scheme is proposed to provide two alternative choices suggestions for regional healthcare providers such as SRHO. There is no given time frame for the model to make the choice, but as long as the thresholds are reached, the corresponding choices will be prompted to healthcare providers.

Before covering the technical details, it should be noted that this model uses the

property of the Spiking Neural Network. Just as how we consider the collective risk level dynamics can alter the decision needed for the regional healthcare providers, here we imagine that each patient wearing the vital sign measuring devices to be a single neuron unit in the network. Therefore, the E-health Cyber Ecosystem network will provide the risk level information of each patient in a distributed manner and thus each patient's risk level contributes to the process of suggesting a decision for the regional healthcare providers. Intuitively, more patients with high risk levels in the system should faster drive to a decision suggestion that medical resources will be needed, and vice versa.

5.1 Technical Overview

This chapter proposes a novel collective decision-making scheme to solve the multi-agent drift-diffusion model problem with the help of spiking neural networks. The exponential integrate-and-fire (EIF) model is used here to capture the individual dynamics of each agent in the system, and we name this new model the Self-Organized Decision Making (SODM) model. Introduced by Fourcaud-Trocme *et al.* [24], the nonlinear EIF model is experimentally verified to be able to accurately capture the response properties. To be specific, the nonlinearity of the exponential integrate-and-fire (EIF) model is incorporated here to replace the stochastic spiking scheme in DDM proposed by [9].

We demonstrate analytically and experimentally that the gating variable for instantaneous activation follows Boltzmann probability distribution, and the collective system reaches meta-stable critical states under the Markov chain premise. With mean field analysis, we derive the global criticality from local dynamics and achieve a power-law distribution. It is also demonstrated that neural sampling

and mean field branching can be derived with the Boltzmann distribution. This suggests that critical behavior of SODM exhibits the convergent dynamics of Boltzmann distribution, and we conclude that the SODM model inherits the property of self-organized criticality, i.e., the system will eventually evolve toward criticality. The proposed SODM model, therefore, reaches a set of absorbing states; and the corresponding global criticality follows power-law distribution, thus attaining the SOC behavior and provides a decision support. Last but not least, the semistability and consensus problem under semistable equilibrium state are discussed to further present the stability analysis of the SODM model.

At this point, it should be more clear that each patient is considered as a single neuron in a spiking neural network; and each neuron inherits the generalized EIF property. With this in mind, a modified DDM is used here to share the EIF terms. The reason for combining these will be clarified later. Then the collective behavior of such a spiking neural network follows the SOC behavior and distributedly achieve a decision, proved by convergence analysis. This decision suggests the Regional Health Care Alert Level and help the regional healthcare providers to better make the decisions.

5.2 Technical Explanation

Self-organized criticality (SOC), a groundbreaking achievement of statistical physics, is receiving growing interest as to its application to neural firing and brain activity [29]. Bak's hypothesis [6] and recent studies [29, 61] all suggest that criticality is evolutionarily chosen by human brains for optimal computational power and fast response time, and that our brains are always balanced precariously at the critical point.

The critical dynamics emerge during the phase transition between randomness (subcritical) and order (supercritical), which usually follow power-law distribution or exhibit similar spatio-temporal properties [61]. A dynamic network system with the SOC behavior has the potential to be scale free, spatially and/or temporally [66], and thus can quickly switch between phases to acquire optimal computational capability. This offers a possible approach to modeling a decision-making process.

The systems exhibiting SOC behavior are usually highly dimensional and slowly driven, with nonlinear properties [6, 66]. To this end, Brochini *et al.* [11] have discussed the phase transitions and SOC in stochastic spiking neural networks. Also, Bogacz *et al.* [9] demonstrated that the standard drift-diffusion model (DDM) can be used for stochastic spiking dynamics, and they relate DDM to a highly interactive "pooled inhibition" model.

However, to the best of the authors' knowledge, although the SOC has been recognized as a fundamental property of neural systems [29], there has yet to be a decision-making model capitalizing on the SOC property.

5.3 Preliminaries

The model in chapter 3 covers a lot of topics and perceived a overlapping property of them, as shown in Figure 5.1. Each topic described in the circle, as well as the methods of adjusting and incorporating the topics will be covered in this section.

5.3.1 Notation and Preliminaries

Here we use a classical directed graph representation, $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$, with a nonempty finite number of nodes and edges. Specifically, $\mathcal{V}$ is the set of nodes, $\mathcal{E}$ is the set of directed edges, and $\mathcal{A} = [a_{ij}]$ is an adjacency matrix with weights $a_{ij} > 0$

if, $\langle i, j \rangle \in \mathcal{E}$, an edge from node i to node j. Note that the assumed graph is simple, i.e., $\langle i, i \rangle \notin \mathcal{E}$, $\forall i$, with no multiple edges going in the same direction between the same pair of nodes and no self-loop. In this case, the diagonal elements of $\mathcal{A}$ are zero. In addition, the Laplacian matrix of $\mathcal{G}$ is denoted by L.

5.3.2 Self-Organized Criticality

Self-organized criticality describes a self-tuned internal interactions that show critical dynamics in complex systems [6]. The interacting node groups are called active sites while the nodes that are less sensitive to the input are called inactive sites. In the sand pile model, each agent has their own steep slope which represents the membrane potential of the spiking neurons. When a certain threshold is hit and the sand in that specific area is steep enough, i.e., $Z_{local} > Z_{critical}$, the avalanches will be triggered, which follows a power-law distribution of $1/f$ noise. Plenz and Beggs [8] observed a similar pattern of avalanches in the cortical neural electrical activity, which was the first evidence that the brain functions at criticality.

For the spiking neural network sense, if an agent activates too many neighbor-

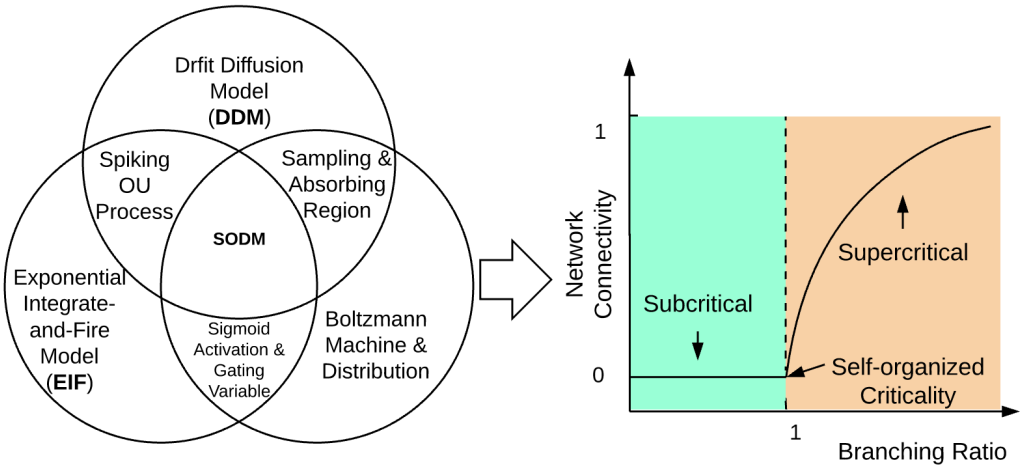


Figure 5.1: SODM model shows SOC property

ing neurons (super-critical), it leads to the massive activation of the entire network, while if too few neurons are activated (sub-critical), propagation dies out too fast [9]. In this model, the local rigidity level is expressed by two terms: the firing threshold of each agent, and the correlation between each two agents in the same local active sites.

5.3.3 Boltzmann Machine

Boltzmann machine (BM) is a special type of stochastic recurrent neural network based on non-stochastic Hopfield nets. In recent years, BM's property of binary output has been attracting more attention in both the theoretical neuroscience and high dimensional parallel stochastic computation [1, 13].

Boltzmann machine is proven to be efficient for the models with connectivity properly constrained. To be specific, machine learning and probability inference are two major applications. In this paper, we apply neural sampling and show that the probability function of the gating current variable for the activation term follows the Boltzmann functions.

The global energy function of Boltzmann machine is defined as:

$$E = -\sum_{i<j} h_{ij} s_i s_j - \sum_i b_i s_i$$

, where E is the global energy, h_{ij} is the connection strength between Unit j and Unit i, $s_i \in \{0, 1\}$ is the state of Unit i, and b_i is the bias of Unit i.

As for the Boltzmann distribution, the probability that the ith unit is on is

$$P_{i=on} = \frac{1}{1 + \exp(-\triangle E_i / T)}, \tag{5.1}$$

where T is the temperature of the system. The probability is calculated using only the information on the energy difference from the initial state and the current temperature.

In this framework, the term $-\triangle E_i/T$ is considered as a logistic activation function, similar to [7, 60]. It has already been shown in [60] that some stochastic neurons sample from a Boltzmann distribution. Ideally, after a long running period and without further inputs, the probability of a global state will not be affected by other terms, i.e., time constants and conductance values in the EIF model. At this stage, the system is at its "thermal equilibrium", and converges to a low temperature distribution where the energy level hovers around a global minimum.

This feature presents behavior similar to SOC if we consider criticality as the thermal equilibrium around which the energy level fluctuates. Also, the log probabilities eventually become a linear term, which helps us to simplify the exponential term in the EIF model. Further discussion will be given in later sections.

Moreover, the neural sampling technique in the later section incorporates the Boltzmann machine according to some local switching, with conditional probability integrated. The multivariable Boltzmann joint distribution has the form [43]

$$P_m = \frac{e^{-\epsilon_m/kT}}{\sum_{n=1}^{M} e^{-\epsilon_n/kT}}, \tag{5.2}$$

where P_m stands for the probability of state m, ϵ_m is the energy at state m, k is a constant, and M is the total number of the states.

5.3.4 Drift-Diffusion Model (DDM)

The DDM has been applied to a Two-Alternative Forced-Choice (TAFC) task in an extensive amount of work (see [9, 55] for instance). The fact that DDM integrates the difference between two choices according to one or two thresholds makes it possible to describe a decision making process in a spiking neural network.

In the pure DDM, the accumulation of the unbiased evidence has the form

$$\mathrm{d}x = g\mathrm{d}t + \beta\mathrm{d}w, \;\; x(0) = 0, \tag{5.3}$$

where $\mathrm{d}x$ represents the changes in difference over the time interval $\mathrm{d}t$, g is the increase in evidence supporting the correct choice each time, w is the independent, identically distributed (i.i.d.) Wiener process, and β is the standard deviation. The probability density $P(x, t)$ is normally distributed with mean gt and standard deviation $\beta\sqrt{t}$.

Since the second term in (5.3) is represented by a standard Wiener process that describes the noise, it is common to consider $\mathrm{d}x$ in DDM to be the change in membrane potential within a certain amount of time [41].

Here we consider a network system with N agents. Each agent relates the DDM model to the nonstationary dynamics of the firing activity of an EIF spiking neural model. While the forced-response protocol is usually considered, we follow the free-response protocol, that each consecutive fires determine the range of the time interval. The common assumption made for this equation usually considers $g > 0$ to support the first choice, and $g < 0$ for the other [9]. The term g can either be a constant for inactive nodes, or a function for active nodes that depends on membrane potential.

While (5.3) only describes the dynamics of a single DDM system, we need extra

terms to capture the impact from neighbors. We have the following stochastic diffusion process with an initial condition x_0:

$$\mathrm{d}x = (\alpha(x(t), t)(x(t) - x_0) + g(x(t), t))\mathrm{d}t + \beta(x(t), t)\mathrm{d}w, \tag{5.4}$$

where $\alpha(t)$ is a measurable gain function that models the external input to accelerate the potential increment, and the linear drifting term $g(t)$ represents the dynamic drifting variable of the node itself. In this regime, we have transferred our model to a Ornstein-Uhlenbeck (OU) process, which is known to be the solution to the famous Fokker-Planck equation. Here, to further simplify the model, we may eliminate the afterhyperpolarization, that is, let $x_0 = 0$.

For the model proposed above, it is possible to receive the spike generations with arbitrary shape, i.e., different spiking time intervals and different incremental speed of membrane potential. With a properly defined activation function, which will be discussed later, the behavior of each single stochastic diffusion process can be bounded. Before further discussing the individual dynamics and their boundedness properties, we need to look into the specific local dynamics by applying the most commonly used neuron model.

5.3.5 Generalized Exponential Integrate-and-Fire Model

Exponential integrate-and-fire is a well developed biological neuron model introduced by Fourcaud-Trocme *et al.* [24] as an extension of the standard leaky integrate-and-fire model. As concluded in several studies [7, 60], EIF is a suitably simple model for very large scale network simulations. For the generalized EIF [60], arbitrary spike shapes are allowed; and gated currents usually reach a steady state with the nonlinear voltage activation function.

The EIF model holds a nonlinearity property consisting of a linear leakage term combined with an exponential activation term, which follows simple RC circuit dynamics before V, the membrane potential, reaches a fixed threshold V_T. After reaching the threshold, it can be considered that the neuron has fired; and its membrane potential is then set to a resting voltage, V_R, approximately -60 mV [7, 60] or 0 mV [41] by different assumptions.

The dynamics of the membrane potential are given by

$$C\frac{\mathrm{d}V}{\mathrm{d}t} = -\varrho_L(V - V_L) + \varrho_T\Delta_T \exp\left(\frac{V - V_T}{\Delta_T}\right) + I_{ion}. \tag{5.5}$$

In this equation, C is the membrane capacitance; V_T is the membrane potential threshold; Δ_T is the sharpness of action potential initiation, or slope factor; V_L is the leak reversal potential; ϱ is the conductance, and I_{ion} is input current. While I_{ion} only represents synaptic current in Fourcaud's model, here we have extended the ionic current by summing up input current, I_{neib}, from neighbors with connectivity, external noise current, I_{noise}, that integrates the i.i.d. Wiener process, and synaptic input, I_{syn}, that incorporates the drifting term, serving as a bias. We have

$$I_{ion} = I_{neib} + I_{noise} + I_{syn}$$

where the term I_{neib} takes identical form as leakage current in the first term of (5.5), while

$$I_{syn} = \varrho_{syn}\Gamma \cdot (V - V_{syn})$$

represents the slow voltage activated current with a gating variable $\Gamma(V, t)$. The term

$$\Gamma_\infty(V) = \lim_{t\to\infty} \Gamma(V, t)$$

can be used to describe the instantaneous activation.

Here, we alter the usually constant conductance, ϱ, and change it to a function of V and t. Multiplying $\mathrm{d}t$ to both sides of the (5.5), we now have

$$\begin{aligned} C\mathrm{d}V = &-\varrho_L(V - V_L)\mathrm{d}t + \varrho_T\Delta_T \exp\left(\frac{V - V_T}{\Delta_T}\right)\mathrm{d}t \\ &+ \varrho_{syn}\Gamma(V,t)(V - V_{syn})\mathrm{d}t + \varrho_{neib}(V - V_{noise})\mathrm{d}t + I_{noise}\mathrm{d}t \end{aligned} \tag{5.6}$$

It is clear that most terms in (5.6) have very similar forms to those in (5.4). As most current terms do not need to be altered to fit in (5.4), the conductance term can change over time and become a function. Functions ϱ and α are sometimes interchangeable. However, the exponential term can be tricky to work around; and we will talk about it in a later part.

Henceforth, for simplicity purposes, we refer to this EIF and DDM combined model as the Self-Organized Decision Making (SODM) model.

5.4 Decision Making Dynamics

The commonly discussed decision-making process is an adaptive behavior that makes use of a series of external input variables and then leads to an optimal or sub-optimal choice of action over other competing alternatives.

We begin by discussing the optimal decision rule. There are two thresholds z_i in the DDM model, with the same magnitude but different signs to represent different choices. In SODM, we consider this choice to be optimal if the threshold of the correct choice is reached or sub-optimal, if our expectation value, $\mathbb{E}$, ends up with the same sign as the correct threshold but with smaller magnitude.

Now we start solving the OU process described in (5.4).

Lemma 1. *The solution of the collective decision-making system in (5.4) is given by*

$$x = e^{\alpha(t)}\left(c + \int_0^t e^{-\alpha(t-\eta)}g(\eta)d\eta + \int_0^t e^{-\alpha(t-\eta)}\beta(\eta)dw_\eta\right) \tag{5.7}$$

where w_η represents each individual Wiener process, which has a similar form as in [5], with the updated expectation

$$\mathbb{E}(x(t)) = \int_0^t e^{-\alpha(t-\eta)}g(s)ds. \tag{5.8}$$

Proof. Let $\phi(t)$ be a fundamental solution matrix.

Also, let Y be

$$c + \int_0^t \phi(\eta)^{-1}g(\eta)\mathrm{d}\eta + \int_0^t \phi(\eta)^{-1}\beta(\eta)\mathrm{d}w_\eta.$$

Then Y has the stochastic differential equation

$$\mathrm{d}Y = \phi(t)^{-1}(g(t)\mathrm{d}t + \beta(t)\mathrm{d}w_t),$$

which implies

$$\begin{aligned} x &= \phi(t)Y \\ &= \phi(t)\left(c + \int_0^t \phi(\eta)^{-1}g(\eta)\mathrm{d}\eta + \int_0^t \phi(\eta)^{-1}\beta(\eta)\mathrm{d}w_\eta\right). \end{aligned}$$

Furthermore, combining all of the above together, the straightforward conclusion is (5.7). □

5.4.1 Behaviors

Here we consider a very special but popular network system.

Assumption 2. *For a high dimensional dynamical network system with N agents described by (5.4), each unit integrates an inward stimulus α_i and receives signals I_{neib} and noise $\beta_i dw_i^j$ from local j neighbors. The dynamics of w^j follow Correlated Brownian motions, with standard correlation col $\in (-1, 1)$.*

In [42], the authors proposed an Itô consensus stochastic differential equation (S.D.E) formalized by N^2-dimensional standard white noise, and expanded the right-hand side terms in (5.3) to a matrix form with graph theory, i.e., $\alpha(t)Lx(t)$. Different from their encoded gain function, we now assume that all of the state information is available to others.

Assumption 3. *For the considered network system with N agents, the state of each agent is observable by others.*

Extending (5.4), then we have the dynamic equation for each agent with neighbors' dynamics added

$$\mathrm{d}x_i = \left(\sum_{j=1}^{K} \alpha(x_i(t), t) l_{ij}(y_{ji} - x_i(t)) + g(x_i(t), t) \right) \mathrm{d}t + \beta(x_i(t), t) \mathrm{d}w_i, \tag{5.9}$$

where K is the total number of agents connecting the agent i, l_{ij} are elements in the Laplacian matrix L, y_{ji} denotes the observed membrane potential of jth agent by the ith agent.

In [16], Charalambos *et al.* have shown a method of solving optimization problems under a reference probability measure by transferring continuous and

discrete-time stochastic dynamic decision systems, via Girsanov's measure transformation. To this end, we have the following claim.

Proposition 1. *The collective stochastic dynamic decision system with common team optimality can be transformed to the equivalent static optimization problem with independent distributed sequences. Under the reference probability space, states and observations are independent Brownian motions.*

Consider a series of d inputs, $\mathbf{X}_i(t) = [x_{i1}(t), ..., x_{id}(t)]$, for N agents, each of which has two possible states

$$s_i = \begin{cases} 1, & \text{if } t \in (t - \tau_{ref}, t], \\ 0, & \text{otherwise.} \end{cases} \tag{5.10}$$

In addition, $\mathbf{Y}_i$ and $\mathbf{w}_i$ are similarly defined. The firing during τ_{ref} is sometimes called the absolute refractory period.

Such series of inputs can be modeled as a discrete decision making scheme, upon which we have applied the free-response paradigm, with adaptive prescribed time interval, τ_x, for each agent in the network system. Equation (5.9) then becomes the following distributed protocol

$$\mathrm{d}\mathbf{X}_i = \Big(L \cdot \alpha(\mathbf{X}_i(t), t) \cdot (\mathbf{Y}_i(t) - \mathbf{X}_i(t)) + g(t) \Big) \mathrm{d}t + \beta(\mathbf{X}_i(t), t) \mathrm{d}\mathbf{w}_i. \tag{5.11}$$

5.4.2 Sampling with EIF

Recall from Proposition 1 that, since the Brownian motions are indpendent of all other team decisions, it opens up the possibility for Markov-chain-related methods. For efficiency and flexibility purposes, Markov chain Monte Carlo (MCMC) has

been applied in sampling the spiking network of neurons [13].

In [60], Richardson has shown the equilibrium value of a slow-driven, voltage-activated, current gating variable with the form

$$\tau_\Gamma \frac{d\Gamma}{dt} = \Gamma_\infty - \Gamma$$

, where $\tau_\Gamma(V)$ is an adaptive time constant characterized by different voltage values; and Γ_∞ is the equilibrium value. Then we have

$$\Gamma = \Phi((V_\Gamma - V)/\Delta_\Gamma)$$

, where Φ is the sigmoid function that digests a membrane potential function into a probability density function with range $[0, 1]$.

$$\Gamma_\infty = \frac{1}{1 + e^{-(V - V_\Gamma)/\Delta_\Gamma}}. \tag{5.12}$$

Here, the equilibrium term Γ_∞ holds a very similar form to the Boltzmann probability distribution as in (5.1).

It is clear that (5.12) is a slowly varying function, which can be proved simply by applying the definition. Now we can use the property of the slow varying function to deal with the exponential term in (5.6). The Karamata representation theorem is one of the most used properties of slow varying functions that transfer a function into a general exponential form. In our case, Γ_∞ is expressed as:

$$\Gamma_\infty = \exp\left(\hbar(\Gamma) + \int_B^\Gamma \frac{\varepsilon(t)}{t}\, \mathrm{d}t\right)$$

, for some $B > 0$, where $\hbar(\Gamma)$ is a bounded measurable function converging to a

finite number, and $\varepsilon(t)$ is a bounded measurable function converging to 0.

Here, since the exponential term is only a property of membrane potential increment that adds to nonlinearity, the potential accumulation of each agent does not affect collective network decisions as much during the firing period $(t - \tau_{ref}, t]$. And for the absolute refractory period, the neuron model is guaranteed not to fire. For such a piece wise continuous function, if we only consider the time interval to be one firing, then the variable V can be bounded. Taking out the exponential term in (5.6), we have

$$\frac{C\mathrm{d}V_e}{\varrho_T(V_e,t)\Delta_T dt} = \exp\left(\frac{V_e - V_T}{\Delta_T}\right),$$

$$\begin{aligned}\lim_{t\to t_{end}} \exp\left(\frac{V_e - V_T}{\Delta_T}\right) &= \lim_{t\to t_{end}} \exp(\hbar(V_e)) \\ &= \exp\left(\frac{V_{end} - V_T}{\Delta_T}\right) = \mathcal{C} \\ &= \tau_\Gamma \frac{\mathrm{d}\Gamma}{\mathrm{d}t} + \Gamma.\end{aligned}$$

where t_{end} is the end of the refractory time, V_{end} is the membrane potenial at the end of the refractory period, and V_e is the membrane potential incremented by exponential terms only. Since $\hbar$ converges to a finite number and $\varepsilon(t)$ converges to 0, the limit of the exponential term converges to a constant $\mathcal{C}$ during the refractory period.

It can be thought of as entering an absorbing state where its behavior at infinity is very similar to the behavior of converging to infinity. Therefore, at the absolute refractory period, we can safely ignore this exponential term in (5.6), and treat it just as other linear terms.

For simulation, we have used the same parameters as the work done by Barranca

et al. [7]. As shown in Fig. 5.2, this monotonically increasing activation function has a sigmoidal general form, with an upper bound 1 and a lower bound 0. In some cases, as mentioned above, we may set the initial condition to be 0 mV to eliminate the afterhyperpolarization. For our simulation, we have removed such constraints to show that the dynamics of the membrane potential can be bounded by the activation function with arbitrary initial conditions. This is because a higher threshold in the EIF model always requires higher voltage to be breached, and the logistic activation function describes such positive correlation well enough.

In fact, very similar dynamics of the activation function have been capitalized in the neural sampling framework [60, 13, 52]. Thus, our collective decision-making model can be thought of as a network consisting of N agents (or neurons) sampling from a probability distribution p using the stochastic dynamics carried from the DDM.

Proposition 2. *The firing activity of the generalized EIF model, which represents each agent in a collective DDM network system, follows a Markov chain process.*

With the information-coded signal from each DDM agent of the system, in other words, the firing information within the time interval $(t - \tau_{ref}, t]$, the neural sampling follows conditional probability distribution, and most of time is a Boltzmann distribution.

For each agent, we consider the collective behavior of connected nodes as an accelerator/damper. For instance, if an agent is surrounded by nodes with higher membrane potential, it receives more current than normal drifting, and vice versa.

$$p(s_i = 1 | x_i(t-1)) = \frac{x_i(t-1) + g(t) + \sum_{j=1}^{K} \alpha(t) l_{ij} (y_{ji} - x_i(t))}{V_T + \dfrac{\sum_{j=1}^{K} \alpha(t) l_{ji} (y_{ij} - x_j(t))}{K}}. \tag{5.13}$$

5.5 Collective Behavior

For most multi-agent dynamic network systems, it is common that new communication links can be established over two agents with no previous connection. In the previous sections, we have assumed that the connection is known at each state. Now we define the coupling and connection behavior among the agents in the system.

5.5.1 Coupling and Connectivity

Assumption 4. *For the system described in this paper, agent i forms at most K outward links randomly at $t = 0$. When $s_i = 1$, Agent i tries to establish new connections with new neighbors, for instance, connecting to Agent j with the coupling probability $\mathcal{P}_{ij}$ depends on the voltage difference. When successful, the equal number of previous connections are lost according to a decoupling probability function $\mathcal{Q}$. When $s_i = 0$, Agent i will not actively modify its neighboring connections.*

It is worth pointing out that both $\mathcal{P}$ and $\mathcal{Q}$ follow a sigmoid (or reverse sigmoid) relation Φ (or $1 - \Phi$). For function $\mathcal{P}$, the greater the difference in membrane potentials, the higher the probability. The situation is reversed for function $\mathcal{Q}$.

Then we have the following equations

$$\mathcal{P}_{ij} = \frac{x_j - x_i}{V_{Ti} + V_{Tj}} \frac{K - \mathcal{K}_i}{K},$$

$$\mathcal{Q}_{ij} = \left(1 - \frac{x_i - x_j}{V_{Ti} + V_{Tj}}\right) \frac{\mathcal{K}_i}{K},$$

where $\mathcal{K}_i$ is the number of current-established connections of node i. In the case of probability values that are less than zero or greater than one, we simply set

them to 0 and 1, respectively, for mean field analysis. The negative probability is also provided in Fig. 5.4 . Different from the common equation for branching probability, our process is not a tree like process; and the maximum connectivity is defined by K. Therefore they are Markovian processes with respect to the number of connected nodes.

5.5.2 Mean Field Analysis

In the mean field analysis, the individual drifting variable usually follows a distribution with the average. In our case, we assume that the expanded term follows the average distribution of g_i/W, where W can be considered as the sum of the total drift. Here g_i is still a gain function representing the external input exerted by neighbors. However, due to the free-response property, g_i is essentially a piece-wise continuous gain function representing synaptic strength.

Recalling the term "active site" described in Section 5.3.2, the local active site for each agent can be thought of as the local field:

$$\mathcal{F}_i = -\sum_{j=1}^{N} a_{ji} g_j / W$$

, where a_{ji} consists of the elements in an adjacency matrix, and the global field has the average

$$\bar{\mathcal{F}} = \sum_{i=1}^{N} \left(\mathcal{F}_i/K\right)/N$$

Therefore, with regard to a sequence of d numbers of inputs, the probability of a single DDM having the on state $s_i = 1$, represented by a single generalized EIF

neuron, has approximately the following probability with respect to the field.

$$p_i(s) = S^{-1} \exp\left(-\gamma \left(\sum_{j=1}^{N \setminus K} \bar{\mathcal{F}} \mathcal{P}_{ij} - \sum_{j=1}^{K} \mathcal{F}_j \mathcal{Q}_{ij} + \sum_{j=0}^{d} b_j \right) \right), \tag{5.14}$$

where S is some partition function, b_i is the bias term that supports the correct choice, γ is a thermodynamic beta in the Boltzmann factor with the form

$$\gamma = 1/(k_B \mathcal{F}_i)$$

, and k_B is a Boltzmann constant.

Putting aside the coupling strength, the mean activity of this network is measured as $\sum_{i=1}^{N} s_i/N$. However, for such a spiking neural model, it is not always practical to have a normally distributed probability density function; and in fact, most of these processes are stochastic with certain thresholds or even highly constrained. That being said, we would have a stochastic Itô-based integral for probability density for the local mean activity:

$$\mathcal{H}_i = \int_{-\infty}^{\infty} \Phi(x) p_i(s|x) \mathrm{d}x$$

5.6 Convergence Analysis

There are two main evidences for systems presenting the SOC behavior [6, 66]. The first is the power-law distribution, and the second is the critical dynamics, which we consider as converging to absorbing states in the SODM model. In this section, we expand the results from Sections 5.4 and 5.5, and examine the global convergence behavior of the collective SODM model. We then provide both pieces of evidence to show the SODM system has the SOC behavior.

5.6.1 Global Criticality from Local Dynamics

Recall that in SOC, active nodes trigger self-avalanches when a threshold is reached and update the information of all connected active nodes. Also, nodes in nearby inactive sites are communicated to establish more connections if needed.

Moreover, each active site has their own dynamics of reaching out to other active or inactive sites. Inspired by Harris's theory of branching processes [28], we use a branching parameter σ that captures the subsequent activity of connectivity triggering or dying out [10]. The *local branching ratio* and *global branching ratio* have the form

$$\sigma_j(t) = \sum_{i=1}^{K} \mathcal{P}_{ij}(t)$$

$$\tilde{\sigma}(t) = \frac{1}{N-1} \sum_{j=1}^{N} \sigma_j(t),$$

respectively. As discussed in [28, 10], the system exhibits criticality at $\sigma = 1$, and is subcritical (supercritical) for $\sigma < 1$ ($\sigma > 1$).

In our simulation, sub-critical dynamics are characterized by low potential and rapidly decaying agents' neuron firing distributions, while super-critical dynamics are characterized by high potential and slowly decaying firing activities. Critical dynamics are characterized by firing activity that follows power-law distributions.

As shown in Fig. 5.3, it can be easily recognized that the collective behaviors of the firing density in the SODM model follow a power-law distribution, $\mathcal{D}(z) \sim z^{\Lambda}$ with different cluster sizes, z, and scaling factors, $\Lambda < 0$, which is primary evidence supporting the SOC behavior [66].

5.6.2 Absorbing States

Recalling the avalanches described in self-organized criticality, the system keeps tuning itself to one of many meta-stable states, which commonly have lifespans shorter than ground states and longer than excited states [6]. And without further inputs, the distribution reaches meta-stability, a very special energy well that is able to temporarily trap the system for a limited number of states.

This can be modeled as a Boltzmann distribution with a global energy level in a simulated annealing system from any initial conditions. With the results from (5.13) and (5.14), as well as the exponential property of the generalized EIF system, the Lyapunov based semistability [34] can be achieved with great potential, but due to the page restriction, this concept is not discussed in this paper. Nevertheless, we are going to show that the system described in this paper converges to some absorbing states.

In mean field theory, as discussed in [10], the absorbing state becomes unstable when the probability of a node creating connection with neighbors is greater than $1/2$. In our case, this can be thought of as the coupling probability $\mathcal{P} > \mathcal{P}_{critical} = 1/2$.

Lemma 2. *For finite number of total states, the usually unique absorbing state becomes a small range containing a set of states in system presenting SOC nature. The attractor of the system is a set of discrete states.*

Proof. If a non-conserving system, such as the DDM-based SODM model described in this paper, has shown a temporary stable configuration after the avalanches, then the system is at least at a critical point. The critical and supercritical sessions are usually slow driving [10]. So there must be a drift load and a diffusion dissipation

fluctuating to keep all of the nodes in the system from either forming active sites or staying completely quiescent.

Thus, if a system presents the thermodynamic behavior as a Boltzmann distribution with simulated annealing, there can exist an infinite number of infinitesimally varied absorbing states around the thermodynamic limit. For a finite number of states, this absorbing phase becomes a set of discrete states. □

This obeys another property of SOC, that is, the dynamical system with a critical point as an attractor, is able to keep itself at the critical point between two phases which, in our case, are the active phase and the absorbing phase.

Proposition 3. *Meta-stable states generally hold more energy than the ground states and less energy than the excited states. Therefore, SOC is the process of the SODM model losing global energy and falling into a certain set of absorbing states, regardless of guaranteed stability.*

Each agent in the SODM model is essentially a drift diffusion term taking input variables from the EIF markup. It is clear that when the individual thresholds are reached, agents will initiate the spike and send information-coded signals (current) to connected nodes or nodes with great probability of establishing connectivity. The excited states usually carry higher membrane potential than the incremental states. Then the fired node resets its membrane potential to V_R and enters a refractory period. Therefore, the states during this absolute refractory period, τ_{ref}, can be considered as comparable meta-stable states that trap the dynamics of each node for τ_{ref}. And since we have proved Lemma 2, the system self-organizes, instead of fine-tuning, to a small region of absorbing criticality, or in another word, metastability.

Since the system is slow varying at the absorbing state, the fundamental solution is independent of t, that is, $\alpha(x(t), t)) = \alpha(x)$. And (5.7) becomes

$$x = e^{\alpha(t-\tau_{ref})}c + \int_{t_0}^{t} e^{-L\alpha(t-\eta)}(g(\eta)\mathrm{d}\eta + \beta(\eta)\mathrm{d}w_\eta). \tag{5.15}$$

For the slow-varying, time-independent, absorbing states mentioned in Proposition 3, it is natural to assume that the dimension of β is 1. Using the corollary in [5],

$$\phi(t) = \exp\left(\int_{(t_0)}^{t} \alpha(\eta)\mathrm{d}\eta\right).$$

Then we can further turn (5.7) to be

$$x(t) = \exp\left(\int_{(t_0)}^{t} \alpha(\eta)\mathrm{d}\eta\right)\left(c + \int_{t_0}^{t} \exp\left(\int_{t_0}^{s} A(u)\mathrm{d}u\right)(g(\eta)\mathrm{d}\eta + \beta(\eta)\mathrm{d}w_\eta)\right). \tag{5.16}$$

Note that, the convergence dynamics of these absorbing states show Boltzmann distribution as well, i.e.,

$$P_i(V_x|s_i = 0) = e^{-x_i/k_B\bar{\mathcal{F}}} \Big/ \left(\sum_{j=1}^{N} e^{-x_j/k_B\bar{\mathcal{F}}}\right).$$

As shown in Fig. 5.5, the branching pattern of the SODM model across multiple fires follows the SOC behavior and eventually evolves to a certain set of absorbing states, known as the recurrence sets. Also, these stationary distributions of network states can be convergent from any initial state.

Therefore, without the presence of a proper controller, the system fine-tunes itself and then converges to exponent

$$\bar{u} = \lim_{t\to\infty}\left((1/t)\sum_{i=0}^{t-1} \ln\sum(\mathcal{K}_i \cdot P_i)\right),$$

with $\bar{u} = 0$ suggests critical branching process.

5.7 Semistability

Since theoretical and experimental studies have demonstrated that critical systems often optimize computational capability, it is promising to suggest that the system with the SOC behavior is both robust and flexible enough to ensure homeostatic stability.

In fact, due to the nature of absorbing states and criticality property, any initial conditions of a spiking network decision-making system can converge and/or fluctuate around a set of states, potentially semistability [34]. The convergence property of such a model can be useful for fault pre-screening and is, in a way, robust to quantified uncertainties.

Since SODM model has a continuum of equilibria, in this section we examine the semistability potential of this stochastic model. Semistability is defined as the property of a dynamical system whereby its trajactories converge to Lyapunov stable equilibria that are not necessarily isolated [34]. Instead of using widely adopted asymptotic stability, we use semistability here to properly capture the SODM model with a continuum of equilibria. Semistability serves as a stronger rigorous proof of the stability analysis of SODM in addition to the convergence analysis done theoretically and experimentally.

Semistablity for deterministic dynamical systems was originally proposed by the supervisor of this thesis work, Dr. Qing Hui, in 2007 on American Control Conference [33] and in 2008 on IEEE Transactions on Automatic Control [34], and have attracted a huge amount of attentions and citations. It describes a property that the solutions of a dynamical system converge to Lyapunov stable quilibrium

points based on system's different initial conditions [34].

5.7.1 Lyapunov Theory for Semistability

Note that, semistability mentioned above is not merely asymptotic stability of a set of equilibria [59]. In addition, semistability can be used to describe a trajectory converging to its corresponding set of equilibria, without the commitment of converging to any specific equilibrium point. Thus, semistability is a perfect tool to describe the SODM model with a set of absorbing states. Also note that, semistability and the stability of the set of equilibrium points are independent notations [59].

Let us recall the Equation 5.9, which satisfies the following definition.

Definition 1. *Consider the nonlinear stochastic dynamical systems of the form*

$$dx(t) = f(x(t))dt + \mathbb{D}(x(t))dw(t), x(0) = x_0, t \in \tau_0 \tag{5.17}$$

where for all $t \in \tau_0$, $\mathfrak{H}$ *is used here to represent Hilbert space,* $x(t) \in \mathfrak{H}_{\mathfrak{D}}$ *is a measurable random system state vector for the Hilbert space (an extension of Euclidean space with infinite dimensions),* $\mathfrak{D}$ *is an open set with* $0 \in \mathfrak{D}$, $w(t)$ *is an independent standard Wiener process (i.e., Brownian motion) defined on a probability space with the same dimension of the first term in this dynamic system, we assume this number of dimension to be d from now on,* $x(0)$ *is independent of* $(w(t) - w(0)), t \geq 0$, $f : \mathfrak{D} \to \mathbb{R}^n$ *is continuous on* $\mathfrak{D}$, $\mathbb{D} : \mathfrak{D} \to \mathbb{R}^{n \times d}$ *is continuous on* $\mathfrak{D}$, $f^{-1}(0) \cap \mathbb{D}^{-1}(0) \triangleq \{x \in \mathfrak{D} : f(x) = 0 \text{ and } \mathbb{D}(x) = 0\}$ *is nonempty, and* $\tau_0 \in [0, \infty]$ *serves as the maximal interval of existence for the solution, which in our case, the time between two consecutive fires.*

Then a equilibrium point $x_e \in \mathbb{R}^n$ of the system 5.17 has the property of

$f(x_e) = 0$ and $\mathbb{D}(x_e) = 0$. We name the set of equilibrium points of system 5.17 to be $\mathbb{S} \triangleq \{\omega \in \Omega : x(t, \omega) = x_e\} = \{x_e \in \mathfrak{D} : f(x_e) = 0 \text{ and } \mathbb{D}(x_e) = 0\}$.

From now on, we are interested in Itô integrals in Equation 5.7. Here we go over some other definitions that are necessary for the stability analysis.

Definition 2. *([49][Definition 7.7]). Let $x(\cdot)$ be a time-homogeneous Markov process in $\mathfrak{H}_{\mathfrak{D}}$ and let $V : \mathfrak{D} \to \mathbb{R}$. Then the infinitesimal generator $\mathfrak{L}$ of $x(t), t \geq 0$, with $x(0) = x_0$ is defined as*

$$\mathfrak{L}V(x_0) \triangleq \lim_{t \to 0^+} \frac{\mathfrak{E}^{x_0}[V(x(t))] - V(x_0)}{t}, x_0 \in \mathfrak{D} \tag{5.18}$$

where $\mathfrak{E}^{x_0}$ denotes the conditional expectation with respect to the probability measure $\mathbb{P}^{x_0}$.

Also, the probability measure means we need the definition of *Lyapunov stable in probability* to be properly defined. Here, we are only interested in standard Lyapunov stable in probability, and ignore the *asymptotically stable in probability*.

Definition 3. *([40]). The equilibrium solution $x(t) \equiv x_e$ is called Lyapunov stable in probability if, for every $\varepsilon > 0$ and $\varrho > 0$, there exist $\delta = \delta(\varepsilon, \varrho) > 0$ such that, for all x_0 belong to the Borel sets (sets that can be formed using finite iteration of union, intersection and complement operations), the probability measure*

$$\mathbb{P}^{x_0}\left(\sup_{t \geq 0} \|x(t) - x_e\| > \varepsilon\right) \leq \varrho$$

Here we can easily define the following assumption to apply the Lyapunov stable in probability Theory.

Assumption 5. *Assume that for every open subset $\mathfrak{N}_\epsilon$ of the absorbing set $\mathfrak{G}$ containing x, there exists a different open subset $\mathfrak{N}_\delta$ of the absorbing set $\mathfrak{G}$ containing x such that*

the probability measure $\mathbb{P}^{x_0}$ *has a upper bound of* $\varrho \leq \mathbb{P}_{BM}$*, the probability defined by Boltzmann Machine in Equation 5.2.*

With this assumption made, we can easily achieve that

$$\mathbb{P}^{x_0}\left(\sup_{t\geq 0}\|x(t)-x_e\|>\varepsilon\right)\leq \mathbb{P}_{BM}\leq 1$$

where ε is the afterhyperpolarization states' values to which the superior is greater. We can conclude that there exists an equilibrium point $x \in \mathfrak{G}$ in SODM model is *Lyapunov Stable in Probability*.

In the previous sections, it has been proved that there exists an absorbing set through the Boltzmann Machine premises; and all trajectories of the initial conditions in this subset $\mathfrak{U}$ fluctuate around this set. Since there exists an equilibrium point in SODM that all initial conditions converge to, this point is a Lyapunov stable in probability equilibrium point and $x \in \mathfrak{U}$. From now on, we call this point to be Absolute Self-Organized Criticality (ASOC) Point.

With some minor modifications, we have the updated Definition 4 to accomodate the Itô integrals used in this thesis. We simply get rid of the ε term and replace δ with a class function $\aleph$. The proof can be referred to in [59].

Definition 4. *([59]). The equilibrium solution* $x(t) \equiv x_e$ *is called Lyapunov stable in probability if and only if, for every* $\varrho > 0$*, there exists a class function* $\aleph(\cdot)$ *and a constant* $c = c(\varrho)$ *such that, for all* x_0 *belong to the Borel sets, the probability measure then becomes*

$$\mathbb{P}^{x_0}\left[\sup t \geq 0\|x(t)-x_e\|>\aleph(\|x_0-x_e\|)\right]\leq \varrho$$

Then, with these definitions, we are ready to develop a sufficient condition for stochastic semistability.

Theorem 1. *([59]). Consider the nonlinear stochastic dynamical system 1. Assume that there exists a two-time continuously differentiable function $V : \mathfrak{Q} \to \mathfrak{N}$, where $\mathfrak{Q}$ is an open neighborhood of $\mathfrak{R}$, such that*

$$V'(x)f(x) + \frac{1}{2}tr\mathbb{D}^T(x)V''(x)\mathbb{D}(x) < 0, x \in \mathfrak{Q}\backslash\mathfrak{N} \tag{5.19}$$

If every equilibrium point of dynamical system 1 is Lyapunov stable in probability, then 1 is stochastically semistable.

Example: Consider a simplified version of SODM model with the nonlinear stochastic dynamic system given by following, where Laplacian matrix and summation sign is expanded

$$\mathrm{d}x_1(t) = \Big(\alpha_{12}(x_2(t)) - \alpha_{21}(x_1(t)) + g(x_1)\Big)\mathrm{d}t + \varpi(x_2(t) - x_1(t))\mathrm{d}w(t), \tag{5.20}$$

$$\mathrm{d}x_2(t) = \Big(\alpha_{21}(x_1(t)) - \alpha_{12}(x_2(t)) + g(x_2)\Big)\mathrm{d}t + \varpi(x_1(t) - x_2(t))\mathrm{d}w(t), \tag{5.21}$$

where the unity coefficients scaling $\alpha_{ij}(\cdot)$, $\{i = 1, j = 2.$ or $i = 2, j = 1\}$ are Lipschitz continuous. Here, the measurable gain function is represented by the unity coefficients scaling $\alpha_{ij}(\cdot)$; the connectivity can be captured by the Laplacian matrix to which two equations we have already separated; the drifting term g_i is similarly defined as in Equation 5.9; and the standard deviation term is represented by $\varpi : \varpi > 0$ times the difference term, as $(y_{ji} - x_i)$ in Equation 5.9.

Equations 5.20 and 5.20 describe the collective dynamics of a simplified SODM model with two agents exchanging information with each other. To be specific,

the unity coefficients scaling $\alpha_{ij}(\cdot)$ is 1 when two agents are connected, and this term becomes 0 when two agents are disconnected [59]. Note that between two consecutive fires, the communication topology is fixed.

To show that this example is stochastic semistable, we apply the Theorem 1 to analyze the consensus behavior. Here, we follow the similar assumptions made in [59]. First is that $\alpha_{ij}(x_j) - \alpha_{ji}(x_i) = 0$ if and only if $x_i = x_j, i \neq j$, and the second as $(x_i - x_j)[\alpha_{ij}(x_j) - \alpha_{ji}(x_i) + g(x_i)] \leq -\varpi^2(x_1 - x_2)$. The first assumption follows the zeroth law of thermodynamics, that the information is not exchanged between agents with equal energy. At this step, we do not consider drifting since we focus more on the energy level over the piece-wise gain dynamics than the dynamic drifting. The second assumption follows the second law of thermodynamics, that the information flows from high energy instances to low energy instances, which is capture by the denominator in Equation 5.13. In addition, the second assumption ensures that the term $\alpha_{ij}(x_j) - \alpha_{ji}(x_i) + g(x_i)$ is bounded by the negative intensity of the diffusion coefficient $\frac{1}{2}tr\mathbb{D}^T(x)\mathbb{D}(x)$ [59]. Note that the energy for the i_{th} term is a balance between the energy level of i and its own dynamic drifting, which is perfectly satisfied by the term $g(t)$ in Equation 5.13.

Therefore, we convert the $f(x)$ and $D(x)$ to matrix forms.

$$f(x) = \begin{bmatrix} \alpha_{12}(x_2(t)) - \alpha_{21}(x_1(t)) \\ \alpha_{21}(x_1(t)) - \alpha_{12}(x_2(t)) \end{bmatrix}$$

$$\mathbb{D}(x) = \begin{bmatrix} \varpi(x_1 - x_2) \\ \varpi(x_2 - x_1) \end{bmatrix}$$

Note that, the stochastic term $\mathbb{D}(t)dw$ now stands for probabilistic measure of information flow variations between two agents.

Now we are ready to find the Lyapunov function $V(x_1, x_2)$ for the example. Recall that $f^{-1}(0) \cap \mathbb{D}^{-1}(0)$ is nonempty, we let this intersection set be $\{(x_1, x_2) \in \mathfrak{R}^2 : x_1 = x_2 = \Upsilon, \Upsilon \in \mathfrak{R}\}$. Let us consider the Lyapunov function candidate $V(x_1, x_2) = \frac{1}{2}(x_1 - \Upsilon)^2 + \frac{1}{2}(x_2 - \Upsilon)^2$. Then the infinitesimal generator $\mathfrak{L}$ defined in Definition 2 follows the Equation 5.19:

$$\mathfrak{L}V(x_1, x_2) = (x_1 - \Upsilon)[\alpha_{12}(x_2(t)) - \alpha_{21}(x_1(t)) + g(x_1)] + \tag{5.22}$$

$$(x_2 - \Upsilon)[\alpha_{21}(x_1(t)) - \alpha_{12}(x_2(t)) + g(x_2)] + \tag{5.23}$$

$$\frac{1}{2}[(\varpi(x_1 - x_2))^2 + (\varpi(x_2 - x_1))^2] \tag{5.24}$$

$$= (x_1)[\alpha_{12}(x_2(t)) - \alpha_{21}(x_1(t)) + g(x_1)] + \tag{5.25}$$

$$(x_2)[\alpha_{21}(x_1(t)) - \alpha_{12}(x_2(t)) + g(x_2)] + (\varpi(x_1 - x_2))^2 \tag{5.26}$$

$$= (x_1 - x_2)[\alpha_{12}(x_2) - \alpha_{21}(x_1) + g(x_1)] + \varpi^2(x_1 - x_2) \tag{5.27}$$

$$\leq 0 \tag{5.28}$$

where $(x_1, x_2) \in \mathfrak{R}^2$, which proves that $x_1 = x_2 = \Upsilon$ is Lyapunov stable in probability.

Further, we infer that $\mathfrak{L}V(x_1, x_2)$ is strictly less than 0 because when $x_1 \neq x_2$, $\mathfrak{L}V(x_1, x_2) \neq 0$. Henceforth, using the Theorem 1, we conclude that $x_1 = x_2 = \Upsilon$ is stochastically semistable for all $\Upsilon \in \mathfrak{R}$.

Remark 3. *Each individual agent's behavior in SODM model, with neighbors' dynamics added, described by dynamic equation 5.9 is stochastic semistable, because every equilibrium point of SODM is Lyapunov stable in probability*

Note that, a *globally stochastic semistable equilibrium* cannot be achieved here, since trajectory subest $\mathfrak{U} \neq \mathfrak{D}$, the open set, at the moment of neuron firing.

Nevertheless, we have rigorously proved that the SODM model makes decisions properly with consensus analysis under the free-response paradigm. Again, the free-response paradigm is useful since we only want to receive proper decision support when indicated. With the current setup, the decision makers will get neutral suggestions when neither thresholds for two choices are reached. This is helpful in the healthcare scheme since the healthcare providers do not wish to redistribute the medical resources too often and thus increase the cost. Our decision making scheme is helpful to provide decision support only when needed. However, this does not necessary mean that a decision will never be made. In our example, we proved that the SODM model is stochastic semistable, which guarantees that a decision support is offered to healthcare providers in E-health Cyber Ecosystems.

5.8 Chapter Summary

In this chapter, a collective decision making model named SODM is proposed with the integration of a specific type of spiking neurons, exponential integrate-and-fire. Our method is based on the well-known two-alternative forced choice task solver – drift-diffusion model. We recognize that DDM and EIF share very common terms in their dynamic equations, and the exponential term in EIF can be ignored during the absolute refractory period. We have derived the probability of each agent's firing based on a Markov chain conditional premise. Then the mean field theory is used to approximate the global criticality from local dynamics.

With analytical reasoning, experimental simulation and theoretical proof, it is found that the global branching ratio follows a power-law distribution; and the SODM system eventually evolves to a set of absorbing states, which are two

main evidences suggesting the self-organized criticality behavior. The activation function follows the Boltzmann state probability, and the convergence dynamics of absorbing states follow Boltzmann distribution. Then the detailed theoretical proof is given to show that the SODM system also achieves semistability and arrives a semistable equilibrium state for consensus problems.

It has been successfully shown that the SODM model is able to provide decision supports for healthcare providers with collective spiking behavior of each patients in a proper and timely manner.

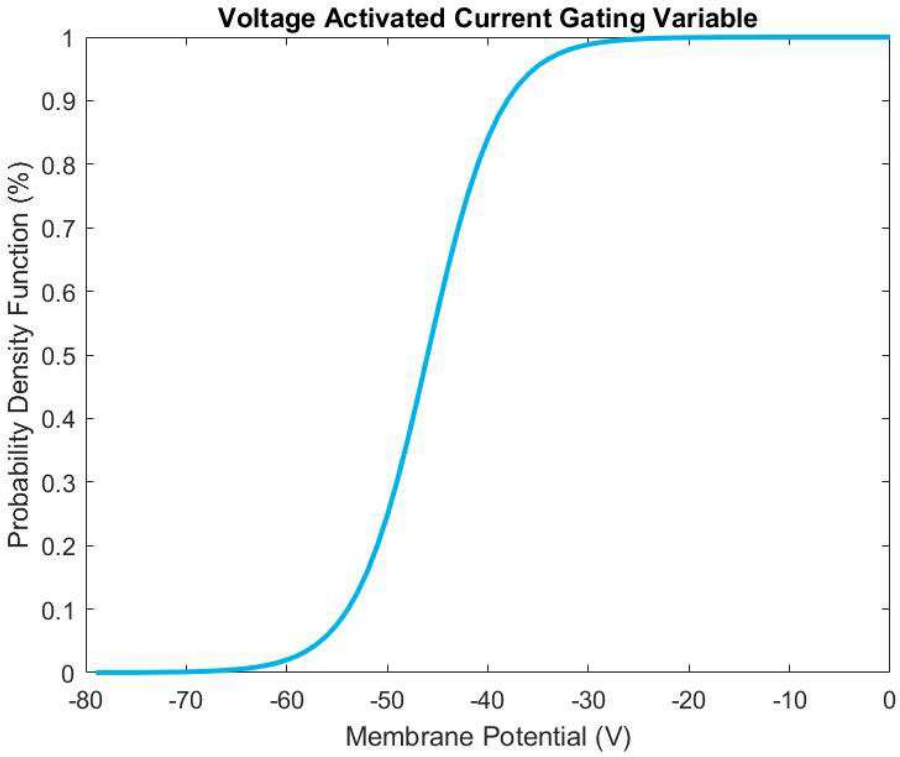


Figure 5.2: Voltage Activated Current Gating Variable

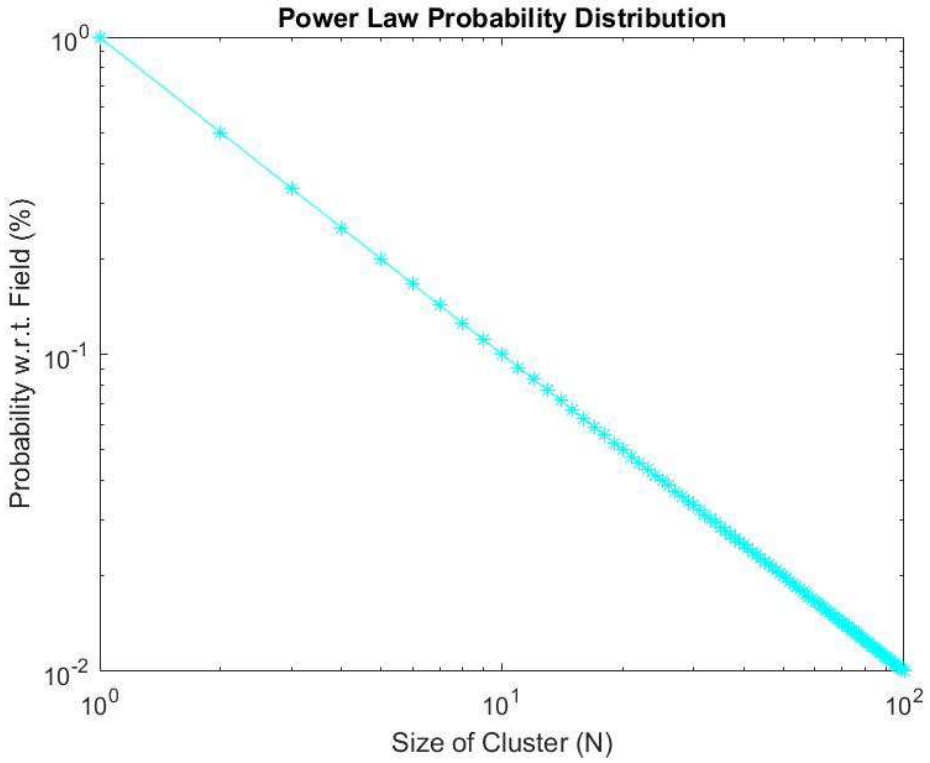


Figure 5.3: Power-law Probability Distribution

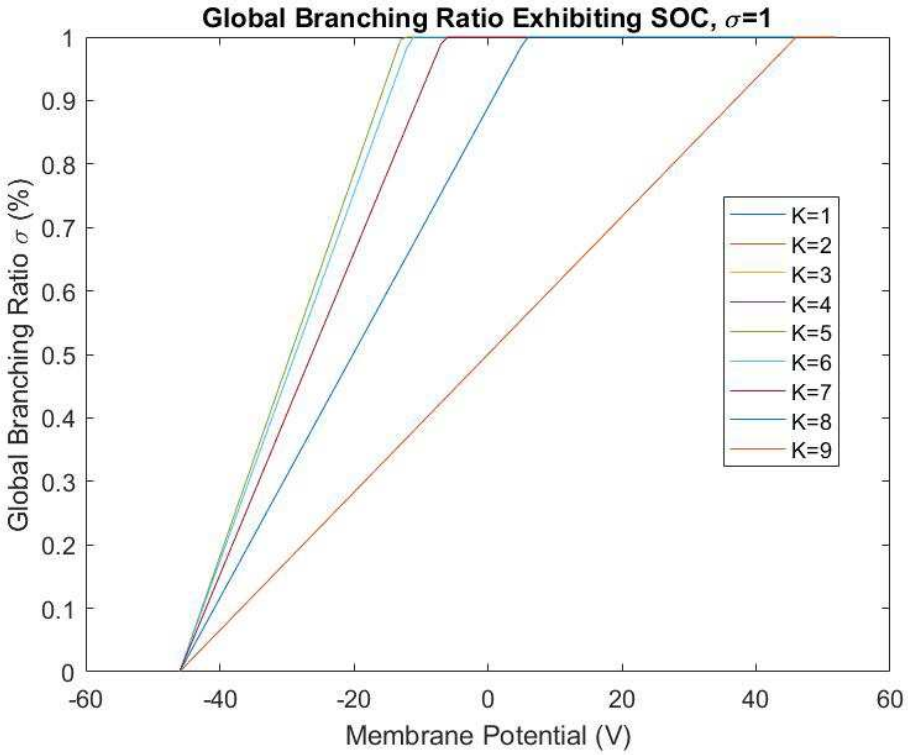


Figure 5.4: In this Figure, as the firing probability increases with the membrane potential, σ converges to 1 with proper connectivity constraints. Cases with different numbers of active and connected neighbors are shown in a network system with $N = 10$.

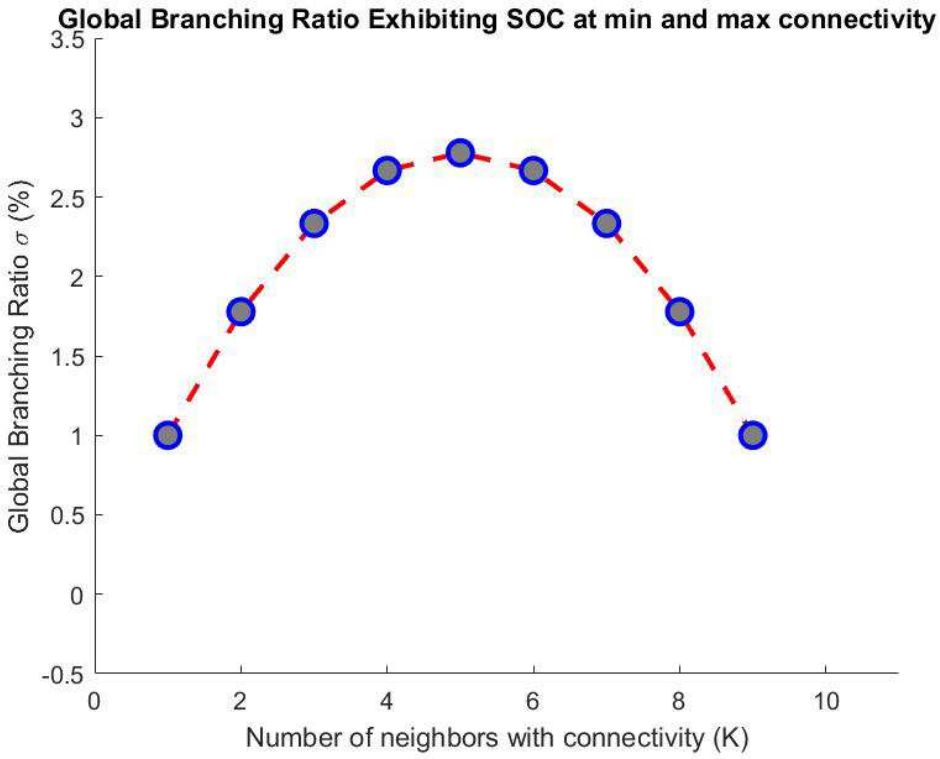


Figure 5.5: In this figure, we remove the constraint that the cumulation of the local branching ratio in each iteration caps at 1. It is clear that as the number of connected neighbors increases, the network system enters the active/loading phase first, and then evolves to the dissipation/absorbing phase. The system clearly shows SOC dynamics, that is $\sigma = 1$ at both minimum and maximum connectivity.

Chapter 6

Conclusion

In this thesis, a complete end-to-end solution for E-health Cyber Ecosystem is proposed to deal with 3 phases. The Phase 1 mainly focuses on patients and is named as **Health Risk Identification**. A model based on machine-learning analysis is developed to monitor patients' risk level of having heart disease. Overall, 88.1% of prediction accuracy is reported, which is 10% higher than existing research work. The Phase 2 focuses on doctors/caregivers and is named **Risk Level Confirmation**. An online queueing model is developed to pair patients with doctors to reconfirm the high risk levels. This step mitigates the errors generated in Phase 1. The Phase 3 focuses on regional healthcare providers and is named **Regional Health Alert Level Decision Support**. A novel decision making paradigm is developed combing fields from neuroscience, machine learning, and statistical physics. Overall, all three phases achieve satisfactory results.

This thesis work has presented different methods with regard to different phases in E-health Cyber Ecosystem to solve for many problems. With the successful patching among all involved parties and different layers, we believe that further efforts to study the interactive and collaborative behaviors will pay off richly.

Bibliography

[1] David H. Ackley, Geoffrey E. Hinton, and Terrence J. Sejnowski. A learning algorithm for Boltzmann machines. *Cognitive Science*, 9(1):147–169, 1985. 5.3.3

[2] D Aha and Dennis Kibler. Instance-based prediction of heart-disease presence with the cleveland database. *University of California*, 1988. 3.2.1, 3.2.1, 3.2.3

[3] David W Aha, Dennis Kibler, and Marc K Albert. Instance-based learning algorithms. *Machine learning*, 6(1):37–66, 1991.

[4] Sultan Alodhaibi, Robert L. Burdett, and Prasad KDV. Yarlagadda. Framework for airport outbound passenger flow modelling. *Procedia Engineering*, 174:1100 – 1109, 2017. 13th Global Congress on Manufacturing and Management Zhengzhou, China 28-30 November, 2016.

[5] L. Arnold. *Stochastic Differential Equations*. Wiley, 1974. 1, 5.6.2

[6] P. Bak. *How Nature Works: The Science of Self-Organized Criticality*. Copernicus Series. Springer, 1996. 5.2, 5.3.2, 5.6, 5.6.2

[7] Victor J. Barranca, Daniel C. Johnson, Jennifer L. Moyher, Joshua P. Sauppe, Maxim S. Shkarayev, Gregor Kovačič, and David Cai. Dynamics of the exponential integrate-and-fire model with slow currents and adaptation. *Journal of Computational Neuroscience*, 37(1):161–180, Aug 2014. 5.3.3, 5.3.5, 5.4.2

[8] John M. Beggs and Dietmar Plenz. Neuronal avalanches in neocortical circuits. *Journal of Neuroscience*, 23(35):11167–11177, 2003. 5.3.2

[9] Rafal Bogacz, E Brown, Jeffrey M Moehlis, P Holmes, and JD Cohen. The physics of optimal decision making: a formal analysis of models of performance in two-alternative forced- choice tasks. *Psychological Review*, 113 (4):700–765, Oct 2006. 5.1, 5.2, 5.3.2, 5.3.4, 5.3.4

[10] Juan A Bonachela and Miguel A Muoz. Self-organization without conservation: True or just apparent scale-invariance? *Journal of Statistical Mechanics: Theory and Experiment*, 2009(09):P09009, 2009. 5.6.1, 5.6.2, 5.6.2

[11] Ludmila Brochini, Ariadne de Andrade Costa, Miguel Abadi, and Antonio C Roque. Phase transitions and self-organized criticality in networks of stochastic spiking neurons. *Nature Scientific Reports*, (6), Nov 2016. 5.2

[12] Herwig Bruneel and Sabine Wittevrongel. Analysis of a discrete-time single-server queue with an occasional extra server. *PERFORMANCE EVALUATION*, 116:119–142, 2017. 2.2

[13] Lars Buesing, Johannes Bill, Bernhard Nessler, and Wolfgang Maass. Neural dynamics as sampling: A model for stochastic computation in recurrent networks of spiking neurons. *PLOS Computational Biology*, 7(11):1–22, 11 2011. 5.3.3, 5.4.2, 5.4.2

[14] Juan Pablo Cavada, Cristián E. Cortés, and Pablo A. Rey. A simulation approach to modelling baggage handling systems at an international airport. *Simulation Modelling Practice and Theory*, 75:146–164, 2017.

[15] Wang Chao, Zhang Xinyue, and Xu Xiaohao. Simulation study on airfield system capacity analysis using simmod. *2008 International Symposium on Computational Intelligence and Design*, 2008.

[16] C. D. Charalambous and N. U. Ahmed. Equivalence of decentralized stochastic dynamic decision systems via Girsanov's measure transformation. In *53rd IEEE Conference on Decision and Control*, pages 439–444, Dec 2014. 5.4.1

[17] Gaurav Dangi, Tanupriya Choudhury, and Praveen Kumar. A smart approach to diagnose heart disease through machine learning and springleaf marketing response. In *Recent Advances and Innovations in Engineering (ICRAIE), 2016 International Conference on*, pages 1–6. IEEE, 2016. 3.2.1, 3.2.3.1

[18] Richard De Neufville. Airport systems planning and design. *Air Transport Management: An International Perspective*, page 61, 2016.

[19] Robert Detrano, Andras Janosi, Walter Steinbrunn, Matthias Pfisterer, Johann-Jakob Schmid, Sarbjit Sandhu, Kern H Guppy, Stella Lee, and Victor Froelicher. International application of a new probability algorithm for the diagnosis of coronary artery disease. *The American journal of cardiology*, 64(5):304–310, 1989. 3.2.1

[20] Bruce Edmonds. Using localised gossipto structure distributed learning. AISB, 2005. 3.2.1

[21] Anouk LM Eikendal, Karlijn A Groenewegen, Michiel L Bots, Sanne AE Peters, Cuno SPM Uiterwaal, and Hester M den Ruijter. Relation between adolescent cardiovascular risk factors and carotid intima-media echogenicity in healthy young adults: The atherosclerosis risk in young adults (arya) study. *Journal of the American Heart Association*, 5(5):e002941, 2016. 2.1

[22] Manuel Felix and Vasco Reis. A micro-simulation model for assessing the performance of check-in in airports. *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 253–258, 2016.

[23] Laura Fiorini, Raffaele Esposito, Manuele Bonaccorsi, Claudio Petrazzuolo, Filippo Saponara, Roberta Giannantonio, Gianluca De Petris, Paolo Dario, and Filippo Cavallo. Enabling personalised medical support for chronic disease management through a hybrid robot-cloud approach. *Autonomous Robots*, 41(5):1263–1276, 2017. 2.1

[24] N Fourcaud-Trocme, D Hansel, C van Vreeswijk, and N Brunel. How spike generation mechanisms determine the neuronal response to fluctuating inputs. *Journal of Neuroscience*, 23(5):11628–11640, 2003. 5.1, 5.3.5

[25] John H Gennari, Pat Langley, and Doug Fisher. Models of incremental concept formation. *Artificial intelligence*, 40(1-3):11–61, 1989. 3.2.1

[26] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009. 3.2.2

[27] Mark A Hall and Geoffrey Holmes. Benchmarking attribute selection techniques for discrete class data mining. *IEEE transactions on knowledge and data engineering*, 15(6):1437–1447, 2003.

[28] Theodore E Harris. *The Theory of Branching Processes*. Courier Corp., 2002. 5.6.1

[29] Janina Hesse and Thilo Gross. Self-organized criticality as a fundamental property of neural systems. *Frontiers in Systems Neuroscience*, 8:166, 2014. 5.2

[30] M. Hgberg, T. R. Bewley, and D. S. Henningson. Relaminarization of re=100 turbulence using gain scheduling and linear state-feedback control. *Physics of Fluids*, 15(11):35723575, 2003. 2.2

[31] M Shamim Hossain. Cloud-supported cyber–physical localization framework for patients monitoring. *IEEE Systems Journal*, 11(1):118–127, 2017. 2.1

[32] Q. Hui, W. M. Haddad, J. M. Bailey, and T. Hayakawa. A stochastic mean field model for an excitatory and inhibitory synaptic drive cortical neuronal network. *IEEE Transactions on Neural Networks and Learning Systems*, 25(4):751–763, April 2014.

[33] Q. Hui, W. M. Haddad, and S. P. Bhat. Finite-time semistability theory with applications to consensus protocols in dynamical networks. In *2007 American Control Conference*, pages 2411–2416, July 2007. 5.7

[34] Q. Hui, W. M. Haddad, and S. P. Bhat. Finite-time semistability and consensus for nonlinear dynamical networks. *IEEE Trans. Autom. Control*, 53:1887–1900, 2008. 5.6.2, 5.7

[35] Lucian Ionescu, Claus Gwiggner, and Natalia Kliewer. Data analysis of delays in airline networks. *Business and Information Systems Engineering*, 58(2):119133, 2015.

[36] Jinhee Jeong and Fazle Hussain. On the identification of a vortex. *Journal of Fluid Mechanics*, 285:6994, 1995. 4.2

[37] George H John and Pat Langley. Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pages 338–345. Morgan Kaufmann Publishers Inc., 1995.

[38] S. Sathiya Keerthi, Shirish Krishnaj Shevade, Chiranjib Bhattacharyya, and Karuturi Radha Krishna Murthy. Improvements to platt's smo algorithm for svm classifier design. *Neural computation*, 13(3):637–649, 2001.

[39] Artur Kierzkowski and Tomasz Kisiel. A model of check-in system management to reduce the security checkpoint variability. *Simulation Modelling Practice and Theory*, 74:80–98, 2017. 2.2

[40] Harold Joseph Kushner. Stochastic stability. In *Stability of stochastic dynamical systems*, pages 97–124. Springer, 1972. 3

[41] Petr Lansky, Charles E. Smith, and Luigi Ricciardi. One-dimensional stochastic diffusion models of neuronal activity and elated first passage time problems. *Research Trends In Biological Cybernetics*, 1989. 5.3.4, 5.3.5

[42] Tao Li and Ji-Feng Zhang. Mean square average-consensus under measurement noises and fixed topologies: Necessary and sufficient conditions. *Automatica*, 45(8):1929 – 1936, 2009. 5.4.1

[43] W. Maass. Noise as a resource for computation and learning in networks of spiking neurons. *Proceedings of the IEEE*, 102(5):860–880, May 2014. 5.3.3

[44] Ioanna E. Manataki and Konstantinos G. Zografos. Assessing airport terminal performance using a system dynamics model. *Journal of Air Transport Management*, 16(2):8693, 2010.

[45] Mic. The beginner programmer. (document), 4.3.4.1, 4.3

[46] Jesmin Nahar, Tasadduq Imam, Kevin S Tickle, and Yi-Ping Phoebe Chen. Computational intelligence for heart disease diagnosis: A medical knowledge driven approach. *Expert Systems with Applications*, 40(1):96–104, 2013. 2.1

[47] Ahlem Nasri and Abdelhabib Bourouis. A model driven engineering approach to reduce large queueing networks. *Journal of Information Technology Research (JITR)*, 10(2):1–18, 2017. 4.3.4

[48] Hannah Nichols. The top 10 leading causes of death in the united states, Feb 2017. 2, 2.1

[49] Bernt Øksendal. Stochastic differential equations. In *Stochastic differential equations*, pages 65–84. Springer, 2003. 2

[50] Jaymin Patel, Dr TejalUpadhyay, and Samir Patel. Heart disease prediction using machine learning and data mining technique. *Heart Disease*, 7(1), 2015. 3.2.1, 3.2.2

[51] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901. 3.2.3.2

[52] Mihai A. Petrovici, Johannes Bill, Ilja Bytschok, Johannes Schemmel, and Karlheinz Meier. Stochastic inference with spiking neurons in the high-conductance state. *Physical review. E*, 94 4-1:042312, 2016. 5.4.2

[53] Andrea M Plaza, Jessica Díaz, and Jennifer Pérez. Software architectures for health care cyber-physical systems: A systematic literature review. *Journal of Software: Evolution and Process*, 2018. 2.3, 4.3.2, 4.3.4

[54] David Portugal, Luís Santos, Paulo Alvito, Jorge Dias, George Samaras, and Eleni Christodoulou. Socialrobot: An interactive mobile robot for elderly home care. In *System Integration (SII), 2015 IEEE/SICE International Symposium on*, pages 811–816. IEEE, 2015. 2.1

[55] I. Poulakakis, L. Scardovi, and N. E. Leonard. Coupled stochastic differential equations and collective decision making in the two-alternative forced-choice task. In *Proceedings of the 2010 American Control Conference*, pages 69–74, June 2010. 5.3.4

[56] David Martin Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. 2011. 3.2.3.1, 3.2.3.1

[57] Nikolas Pyrgiotis, Kerry M Malone, and Amedeo Odoni. Modelling delay propagation within an airport network. *Transportation Research Part C: Emerging Technologies*, 27:60–75, 2013.

[58] J Ross Quinlan. C4. 5: Programming for machine learning. *Morgan Kauffmann*, 38:48, 1993.

[59] Tanmay Rajpurohit and Wassim M Haddad. Lyapunov and converse lyapunov theorems for stochastic semistability. *Systems & Control Letters*, 97:83–90, 2016. 5.7.1, 5.7.1, 4, 1, 5.7.1

[60] Magnus J. E. Richardson. Dynamics of populations and networks of neurons with voltage-activated and calcium-activated currents. *Physical Review E*, 80(2), 2009. 5.3.3, 5.3.5, 5.4.2, 5.4.2

[61] Mikail Rubinov, Olaf Sporns, Jean Philippe Thivierge, and Michael Breakspear. Neurobiologically realistic determinants of self-organized criticality in networks of spiking neurons. *PLOS Computational Biology*, 7:1–14, June 2011. 5.2

[62] Inas S. Khayal and Amro Farid. Architecting a system model for personalized healthcare delivery and managed individual health outcomes. 2018:1–24, 03 2018.

[63] Hamid R. Sayarshad and Joseph Y. J. Chow. Survey and empirical evaluation of nonhomogeneous arrival process models with taxi data. *Journal of Advanced Transportation*, 50(7):1275–1294. 4.3.4.1

[64] Walber. Precision and recall, Nov 2014. (document), 3.1

[65] Yu Wang, Jiafu Tang, Zhendong Pan, and Chongjun Yan. Particle swarm optimization-based planning and scheduling for a laminar-flow operating room with downstream resources. *Soft Comput.*, 19(10):2913–2926, October 2015. 2.2

[66] Nicholas W. Watkins, Gunnar Pruessner, Sandra C. Chapman, Norma B. Crosby, and Henrik J. Jensen. 25 years of self-organized criticality: Concepts and controversies. *Space Science Reviews*, 198(1):3–44, Jan 2016. 5.2, 5.6, 5.6.1

[67] Ward Whitt and Wei You. Using robust queueing to expose the impact of dependence in single-server queues. *Operations Research*, 2017. 4.3.4

[68] Yin Zhang, Meikang Qiu, Chun-Wei Tsai, Mohammad Mehedi Hassan, and Atif Alamri. Health-cps: Healthcare cyber-physical system assisted by cloud and big data. *IEEE Systems Journal*, 11(1):88–95, 2017. 2.1