MULTI-AGENT CLASSIFIER SYSTEM BASED ON HETEROGENEOUS CLASSIFIER

NUR AFIQAH BINTI MOHD DALI

BACHELOR OF COMPUTER SCIENCE

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : NUR AFIQAH BINTI MOHD DALI

Date of Birth : 03 AUGUST 1996

Title : MULTI-AGENT CLASSIFIER SYSTEM BASED ON HETEROGENEOUS CLASSIFIERS

Academic Session : 2017/2018

I declare that this thesis is classified as:

☐	CONFIDENTIAL	(Contains confidential information under the Official Secret Act 1997)*
☐	RESTRICTED	(Contains restricted information as specified by the organization where research was done)*
☑	OPEN ACCESS	I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

_______________________	_________________________
(Student's Signature)	(Supervisor's Signature)
960803-08-5852	DR.MOHAMMED FALAH MOHAMMED
_______________________	_________________________________
New IC/Passport Number	Name of Supervisor
Date:	Date:

Universiti Malaysia PAHANG
Engineering • Technology • Creativity

SUPERVISOR'S DECLARATION

I hereby declare that I have checked this thesis and in my opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Bachelor of Computer Science (Software Engineering).

____

(Supervisor's Signature)

Full Name : DR. MOHAMMED FALAH MOHAMMED

Date :

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

____

(Student's Signature)

Full Name : NUR AFIQAH BINTI MOHD DALI

ID Number : CB15136

Date :

MULTI-AGENT CLASSIFIER SYSTEM
BASED ON HETEROGENEOUS CLASSIFIERS

NUR AFIQAH BINTI MOHD DALI

Thesis submitted in fulfillment of the requirements
for the award of the degree of
Bachelor of Computer Science (Software Engineering)

Faculty of Computer System & Software Engineering
UNIVERSITI MALAYSIA PAHANG

DECEMBER 2018

ACKNOWLEDGEMENTS

I would like to thank to each and every person who individual who supported and assisted me in completing this thesis. Firstly, I would like to thank to my supervisor, Dr. Mohammed Falah Mohammed for assist, guidance moral support and advice that given to me to finish the development process of the application for this project. My special acknowledgement also goes to my parents, Encik Mohd Dali bin Bakar and Puan Sarimah binti Selaman, my family and my friends who support me mentally and physically throughout the process of completing this project.

ABSTRACT

The MAS model consists of several independents agents, and these agents has the ability to carry out a specific task and to make decisions. When working, these agents will share information with each other. Indirectly, this allows the system to get better predictions. When the constituent agents in a MAS model consist of classifiers, the resulting system is known as a multi-agent classifier system (MACS). In this project, our focus is about mutli-agent classifier system based on heterogeneous classifiers. This is because based on the previous analysis, previous MACS model that used homogeneous type of classifiers i.e., FMMs or EFMM have problem with noise effect and noise tolerance, where both classifiers have no mutant against noise. That could have a negative effect on the classification performance. In fact, learning with noise data can cause false knowledge which will be represented as noisy hyperbox in the topology of the classifier. In order to solve this problem we propose to use a heterogeneous classifiers with pruning strategy that have the ability to reduce noise effects. That could improve the MACS classification performance by overcomes the limitations of each classifier when handling different classification problems.

ABSTRAK

Model MAS terdiri daripada beberapa ejen bebas, dan ejen-ejen ini mempunyai keupayaan untuk menjalankan tugas khusus dan membuat keputusan. Apabila bekerja, ejen-ejen ini akan berkongsi maklumat antara satu sama lain. Secara tidak langsung, ini membolehkan sistem untuk mendapatkan ramalan yang lebih baik. Apabila ejen konstituen dalam model MAS terdiri daripada pengelas, sistem yang dihasilkan dikenali sebagai sistem pengelasan pelbagai agen (MACS). Dalam projek ini, tumpuan kami adalah mengenai sistem pengelas mutli-agen berdasarkan pengkelas yang berbeza. Ini kerana berdasarkan analisis terdahulu, model MACS terdahulu yang menggunakan jenis pengelas yang sama seperti FMM atau EFMM mempunyai masalah dengan kesan bunyi dan toleransi bunyi, di mana kedua-dua pengeluar tidak mempunyai mutan terhadap bunyi. Itu boleh memberi kesan negatif terhadap prestasi klasifikasi. Malah, pembelajaran dengan data bunyi boleh menyebabkan pengetahuan palsu yang akan diwakili sebagai hyperbox yang bising dalam topologi pengelas. Untuk menyelesaikan masalah ini, kami mencadangkan untuk menggunakan pengelas yang sama dengan strategi pemangkasan yang mempunyai keupayaan untuk mengurangkan kesan bunyi. Itu boleh meningkatkan prestasi klasifikasi MACS dengan mengatasi batasan setiap pengelas apabila mengendalikan masalah klasifikasi yang berbeza.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

MAS	Multi-Agent System
MACS	Multi-Agent Classifier System
MACS-CBS	Multi-Agent Classifier System based on Certified Belief in Strength
TNC-based MACS	Trust-Negotiation-Communication based Multi-Agent Classifier System
ANN	Artificial Neural Network
FMM	Fuzzy Min-Max
EFMM	Enhanced Fuzzy Min-Max
CBS	Certified Belief in Strength
NN	Neural Network

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Many researchers have pay great attention to multi-agent system (MAS) technologies, where various MAS models have been widely used in diverse fields such as power engineering, e-commerce and fault diagnosis. The MAS model consists of several independents agents, and these agents has the ability to carry out a specific task and to make decisions. When working, these agents will share information with each other. Indirectly, this allows the system to get better predictions. When the constituent agents in a MAS model consist of classifiers, the resulting system is known as a multi-agent classifier system (MACS) [1].

MACS consists of two layers. First layer known as manager, meanwhile second layer known as agents (classifier). Agents will undergo learning procedures according to training and prediction. After that, the agent will receive a test sample. Each agent will predict which hyperbox (hence the output class) the test sample that belongs to. Next, the decision made by the agent should be submitted to the manager. Then, the manager will determine the winner. In this model, neural network model used as the learning agents [1].

Artificial neural network (ANN) is a computational model that consists of an interconnected group of artificial neurons that simulates the biological neural system in our brain. Among the problems that occur in terms of ANN training are related to batch learning, which is catastrophic forgetting, where there is some issues in learning systems. In classification, there are two categories type of learning process including online and offline learning [4]. Online learning can be used to overcome stability plasticity dilemma. Fuzzy Min Max (FMM) is some of neural network that supports online learning..

The FMM structure comprises a number of hyperboxes. Since FMM support online learning, every time the data sample arrives, FMM will create a new hyperbox based on that data, and link them to a new class, or improve existing ones without retraining [3]. However, FMM suffers from some limitations belongs to its expansion, overlap test, and contraction processes. Hence, an Enhanced Fuzzy Min-Max (EFMM) network was proposed to overcome the FMM limitations.

In EFMM, the learning process is similar to that in FMM, but there are some improvements in the expansion, overlap test, and contraction processes [3]. EFMM has demonstrated its effectiveness in addressing the first three FMM limitations, but there are still unresolved issues that are related to noise tolerance. Hence, EFMM with pruning has been proposed to solve the limitations of EFMM and enhance its robustness for tackling pattern classification problems [5].

In fact, the technique that have been used to handle noise problem in EFMM with pruning known as pruning strategy. This strategy will reduce the complexity of the network associated with the presence of noise in the training data sample [5]. However, pruning strategy have a drawback related to the size of data and number of patterns, where using a small number of input pattern during the prediction stage could affect the performance quality.

Based on that, we can say that there are advantages and disadvantages for each neural network model. That advantages and disadvantages depends on the type of data sets. Based on that, we propose to use MACS based heterogeneous classifiers instead of using a homogeneous classifiers. That could improve the MACS classification performance by overcomes the limitations of each classifier when handling different classification problems.

1.2 PROBLEM STATEMENT

Based on the previous analysis, previous MACS model used a homogeneous type of classifiers i.e., FMMs or EFMM. The real problem with that design is the noise effect and noise tolerance, where both classifiers have no mutant against nose. That could have a negative effect on the classification performance. In fact, learning with noise data can cause false knowledge which will be represented as noisy hyperbox in the topology of the classifier. Because of that we propose to use a heterogeneous classifiers with pruning strategy that have the ability to reduce noise effects.

1.3 AIM AND OBJECTIVES

The aim of this project is to improve the classification accuracy of the MACS model by using different classifier models. The objectives of this project are:

1. To study and analyse some of the existing pattern classifiers and highlight their limitations and advantages.

2. To propose a heterogeneous MACS with the ability to deal with noise and free noise data sets for pattern classification problems.

3. To test and evaluate the performance of the model by using Iris datasets and Heart datasets.

1.4 SCOPE OF PROJECT

1. The study focus on heterogeneous classifier instead of homogeneous classifier.

2. The proposed model will be test using Iris datasets and Heart datasets.

1.5 SIGNIFICANCE

Proposing a new heterogeneous MACS with the ability to overcome the data noise problems.

1.6 THESIS ORGANIZATION

In chapter 1, we talk about the Multi Agent System in introduction. We also describe more about the neural network, and some of its models, also some explanation about homogeneous and heterogeneous model. Problem statement in this chapter describe the limitation in the homogeneous model and what is the techniques will be used to overcome the limitation. The objective is to set the goal of the system that need to be achieved.

In chapter 2, we discuss about Multi-Agent Classifier System with its existing model. We also discuss about Fuzzy Min-Max and models that have been proposed. We discuss all of it in details. Every model, there must be some limitations. So, once identified the shortcomings, a study will be made to propose a new model for solving the previous problem.

In chapter 3, we dicussed the methodology used in this work. Flowchart and pseudocode for proposed model are also included. In addition, the hardware and software used are also described here.

In chapter 4, we will state the results we get for the proposed model. From these results, analysis and discussion will be made.

In chapter 5, we will discuss the constraints related to this project. Further work will also be stated.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

Before start the project, some analysis for previous work should be done. From analysis that have been done, our view for the project will be more clear. This chapter is discussing three existing MACS models, and three existing Neural Network models.

2.2 MULTI-AGENT CLASSIFIER SYSTEM (MACS)

Multi-agent systems (MASs) have gained much interest of researchers over the last decade. Many researchers have pay great attention to MAS technologies. The MAS model consists of several independents agents, and these agents has the ability to carry out a specific task and to make decisions. When several agents operate in a common environment, sharing resources and information among agents becomes possible, allowing the system to arrive at a better prediction. When the constituent agents in a MAS model consist of classifiers, the resulting system is known as a multi-agent classifier system (MACS) [1]. In this part, it is discussion for three types of MACS model, Multi-Agent Classifier System based on Certified Belief in Strength (MACS-CBS), Multi-Agent Classifier System using Trust-Negotiation-Communication model (MACS using TNC model) and Multi-Agent Classifier System with a Bayesian Formalism.

2.2.1 Multi-Agent Classifier System based on Certified Belief in Strength (MACS-CBS)

Among the MACS models that we will discuss are Multi-Agent Classifier System based on Certified Belief in Strength (MACS-CBS). MACS-CBS has been selected and used in this work, but we only referred it as MACS only. Figure 2.1 shows MACS-CBS model.

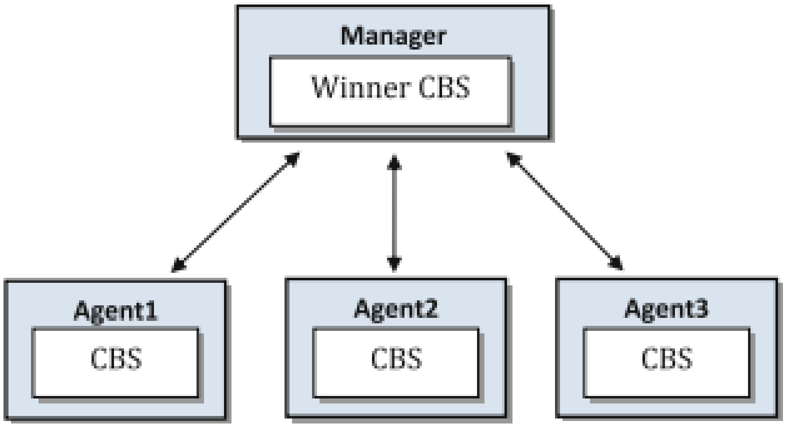


Figure 2.1 MACS-CBS Model [1]

Figure above shown MACS-CBS consist two layers, that is agent layer and manager layer. This model have three agents, and FMM used as learning agent. After undergo traing and receive real sample, each agent will make a decision, and manager will choose the winning agent. CBS trust measurement method is the nucleus of each agent [1].

Since manager is the one who will select the winner, agents will send CBS value to the manager. Then, manager will determine the winning agent based on the first bid auction by choosing the highest CBS value. In this model, each agent will participate in the auction by starting with a specific net worth defined by the manager. Net worth also known as strength (S). Each agent will receive initial strength value, 100. After manager select the winner, the correct result will be sent whether the winner is correct or not. If winning agent make a correct decision, manager will be reward, else manager will be penalize. Other than that, each agent also will be reward or penalize, where their strength value will be update.

2.2.2 Multi-Agent Classifier System using Trust-Negotiation-Communication model (MACS using TNC model)

Here, Multi-Agent Classifier System using Trust-Negotiation-Communication model (MACS using TNC model) will discussed. Figure 2.2 shown the model.

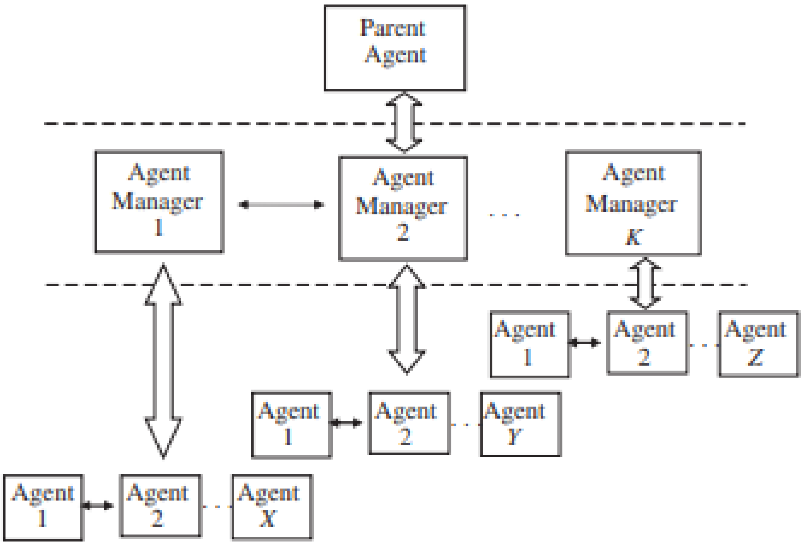


Figure 2.2 MACS using TNC model [7]

Different with the MACS-CBS model that consists of only two layers, this model contains three layers. First layer known as parent agent, second layer as a hidden layer, consists of team manager, and third layer as team agent. In this model, two neural network model used as agents, it is fuzzy min max (FMM) and and fuzzy ARTMAP (FAM). In this model, the decision will be made by parent agent. The parent agent will decide on the decision made by the manager, meanwhile the manager will decide on the decision made by the team agent. [7].

Even though MACS with TNC and MACS-CBS is a difference models and have some differences, there are still some similarities between them. Both of them using an auction which “sealed bid-first price auction” method. Agent who bid with the highest price will be the winner [7].

But in MACS with TNC, it happens twice in different phases [7]. For first phase, team agent will make a prediction and manager will make decision by choosing prediction that has high value. In second phase, decision that have been made by each manager will submit to parent agent, and parent agent will make the final decision [7].

2.2.3 Multi-Agent Classifier System with a Bayesian Formalism

Here is discussion for Multi-Agent Classifier System with Bayesian Formalism model. Figure 2.3 shown the model.

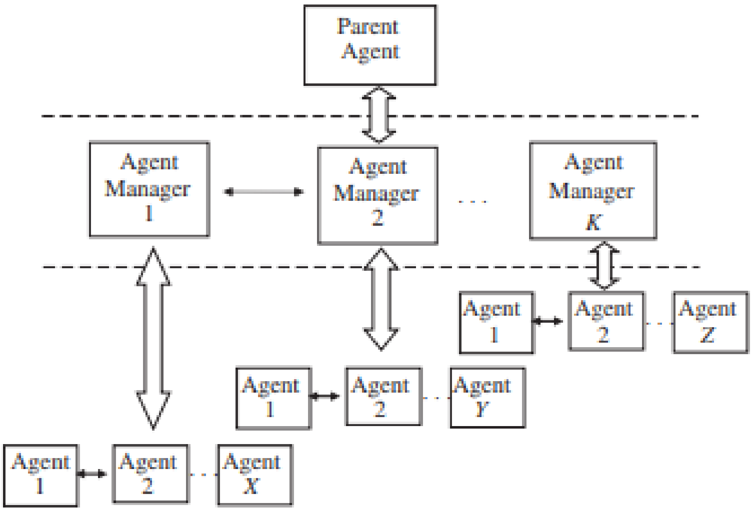


Figure 2.3 Multi-Agent Classifier System with Bayesian Formalism

Based on above figure, it is known that this model is a bit similar with MACS using TNC model. MACS with Bayesian Formalism also consists of three layers, parent agent layer, team manager layer and team agent layer [7].

This model proposed an auction method for negotiation and a novel method for measuring and propagating trust within the TNC model. Auction is one of the most popular and widely used in MAS [10]. In this model, auction occurs in two separate phase. For first phase, after get real sample, each agent from team agent layer will make a prediction associated with a trust value. The prediction will submit to manager. Then manager will choose prediction with highest trust value. This phase is first auction process. After that, second auction process begin where decision that have been made by manager will submit to parent agent along with their trust and reputation value. Based on the information received, the parent agent will makes a final decision, and assigns a predicted output class for the input sample.

There are two neural network used in the model. First team is using FMM agents, while second team using modified FMM (MFMM) agents.

2.3 NEURAL NETWORK (NN)

Artificial neural network (ANN) is a computational model that consists of an interconnected group of artificial neurons, and it has widely used in various areas. Pattern classification is one of the active ANN application domains [2].

In terms of ANN training, one of the major problems related to batch learning is catastrophic forgetting, where the learning system cannot remember what has been learned previously each time new information is absorbed. Because of that, some ANN models have been proposed, including the adaptive resonance theory (ART) networks, and fuzzy min-max (FMM) networks. Specifically, there are two FMM networks, i.e., a supervised classification model and an unsupervised clustering model [8].

2.3.1 Fuzzy Min-Max (FMM)

The supervised FMM network (hereafter addressed as FMM) requires a dynamic network structure with online learning capabilities [3]. FMM network uses hyperbox fuzzy sets to create and store knowledge its network structure [5]. The FMM structure consists of several hyperboxes. As FMM supports online learning, learning models can create new classes, while at the same time refine the existing class without damaged the previous knowledge. This effect allows FMM to add new classes or improve existing classes without requiring retraining. FMM successfully tackle the stability-dilemma, and this enable learning model absorb new information continuously without damaged the previous knowledge [3].

After receiving data samples, FMM will create multiple hyperboxes based on the data. Each hyperbox is represented by the minimum and maximum points in the n-dimensional space within a unit hypercube (I^n). There are three steps in FMM learning, hyperbox expansion, hyperbox overlap test, and hyperbox contraction [3].

First step, hyperbox expansion will be done to include input patterns in each hyperbox class. However, it must be ascertained that the hyperbox size does not exceed the expansion coefficient, $\boldsymbol{\Theta}$, where $0 \leq \boldsymbol{\Theta} \leq 1$. Therefore, when hyperbox B_j is expanded to include new input

pattern A_h, the following constraint must be met [3]:
$n\Theta \geq \sum_{i=1}^{n}(\max(w_{ji}, a_{hi}) - \min(v_{ji}, a_{hi}))$. (1)

However, if the constraints are not met, a new hyperbox will be created to encode the patterns. Due to this process, an overlapping between hyperboxes will occur. Therefore, an overlap test will be performed after the expansion process to check if there is an overlap between expanded and existing hyperboxes that fall into other classes.

If there is an overlap, but it is occur between same hyperbox classes, the overlap is allowed, as shown in Figure 1.1 below. However, if the overlapping occur between different hyperbox classes, the hyperbox contraction process will be made to eliminate overlapping regions [2].

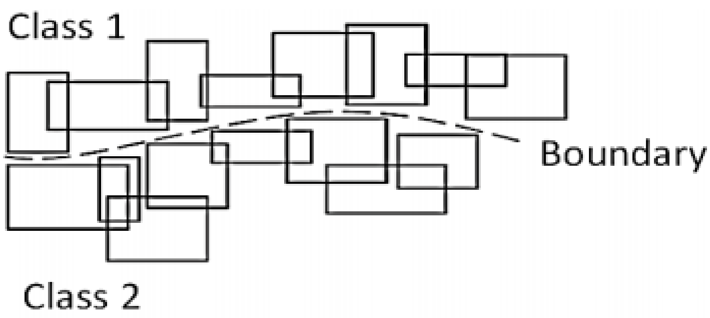


Figure 2.4 Example of the FMM hyperbox boundary of a two-class problem [2]

However, there are still some limitations in FMM dynamic learning. The current hyperbox expansion process lead to increasing overlap between different classes, and this will negatively affect FMM. Although the hyperbox overlap test rule have been applied, it has been found that the test cannot detect all overlapping regions. And this will affects the subsequent hyperbox contraction process. As a result, an Enhanced FMM (EFMM) network is proposed [3].

2.3.2 Enhanced Fuzzy Min-Max (EFMM)

Enhanced Fuzzy Min-Max (EFMM) model is proposed to solve the limitations in FMM [5]. EFMM will undergo same process as FMM. Three steps, hyperbox expansion rule, hyperbox overlap test rule, and hyperbox contraction that has been used in FMM also used in EFMM, but there will have some improvement to overcome the limitations in FMM.

During the expansion process, FMM calculates the sum of all dimensions (as in (1)) and compares the resulting score by ($n\mathbf{\Theta}$). That process has been found lead FMM to have multiple overlapping in different classes. Thus, a new constraint has been introduced to solve the problem:

$$\mathrm{Max}_n(W_{ji}\ ,\ a_{hi}) - \mathrm{Min}_n(V_{ji}\ ,\ a_{hi}) \leq \mathbf{\Theta}. \tag{2}$$

EFMM considers each dimension individually and checks the difference between the maximum and minimum points of each dimension against separately. Since EFMM considers each dimension individually, the difference between the maximum and minimum points of each dimensional will be checked separately (as in (2)). If all the hyperbox dimensions do not exceed $\mathbf{\Theta}$, the expansion process will be performed.

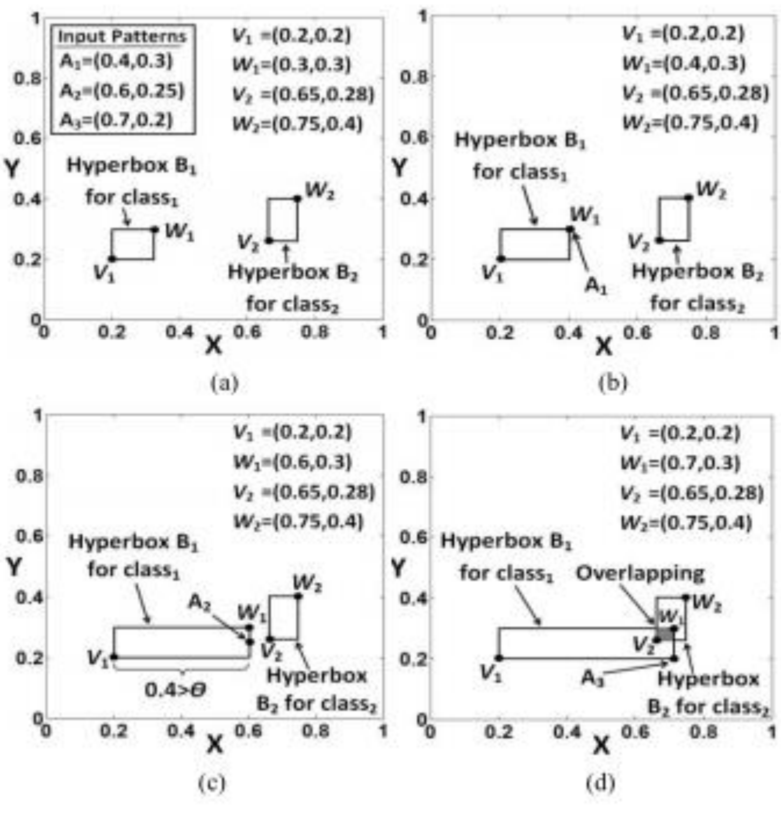


Figure 2.5 FMM Expansion Process [3]

If the input pattern does not belong the any existing hyperboxes, a new hyperbox will created. However, the expansion process leads to overlaps among hyperboxes, and this can cause overlapping between different classes. To check whether there is any overlapping between classes, overlap test will be performed.

It has been found that hyperbox overlap test rule in FMM cannot detect all overlapping regions. If the test is conducted, and the overlap test rule cannot detect overlapping regions that occur, FMM will expect there is no overlapping, and this cause FMM to stop the overlap test [3].

To deal with this problem, some cases have been added, and this allows the overlap to be detected [3]. Note that cases (3) and (4) also have been used in FMM, while (5) – (11) are new cases introduced.

Case 1:

$$V_{ji} < V_{ki} < W_{ji} < W_{ki}\ ,\ \delta^{new} = \min(W_{ji} - V_{ki}\ ,\ \delta^{old}). \tag{3}$$

Case 2:

$$V_{ki} < V_{ji} < W_{ki} < W_{ji}\ ,\ \delta^{new} = \min(W_{ki} - V_{ji}\ ,\ \delta^{old}). \tag{4}$$

Case 3:

$$V_{ji} = V_{ki} < W_{ji} < W_{ki}\ ,\ \delta^{new} = \min(\min(W_{ji} - V_{ki}\ ,\ W_{ki} - V_{ji}),\ \delta^{old}). \tag{5}$$

Case 4:

$$V_{ji} < V_{ki} < W_{ji} = W_{ki}\ ,\ \delta^{new} = \min(\min(W_{ji} - V_{ki}\ ,\ W_{ki} - V_{ji}),\ \delta^{old}). \tag{6}$$

Case 5:

$$V_{ki} = V_{ji} < W_{ki} < W_{ji}\ ,\ \delta^{new} = \min(\min(W_{ji} - V_{ki}\ ,\ W_{ki} - V_{ji}),\ \delta^{old}). \tag{7}$$

Case 6:

$$V_{ki} < V_{ji} < W_{ki} = W_{ji}\ ,\ \delta^{new} = \min(\min(W_{ji} - V_{ki}\ ,\ W_{ki} - V_{ji}),\ \delta^{old}). \tag{8}$$

Case 7:

$$V_{ji} < V_{ki} \leq W_{ki} < W_{ji}\ ,\ \delta^{new} = \min(\min(W_{ji} - V_{ki}\ ,\ W_{ki} - V_{ji}),\ \delta^{old}). \tag{9}$$

Case 8:

$$V_{ki} < V_{ji} \leq W_{ji} < W_{ki}\ ,\ \delta^{new} = \min(\min(W_{ji} - V_{ki}\ ,\ W_{ki} - V_{ji}),\ \delta^{old}). \tag{10}$$

Case 9:

$$V_{ki} = V_{ji} < W_{ki} = W_{ji}\ ,\ \delta^{\text{new}} = \min(W_{ki} - V_{ji}\ ,\ \delta^{\text{old}}). \tag{11}$$

Assuming at the beginning, $\delta^{\text{old}} = 1$. Then, check dimension-by-dimension, if found $\delta^{\text{old}} - \delta^{\text{new}} < 1$, means there is overlapping regions. Next, set setting $\Delta = i$ and $\delta^{\text{old}} = \delta^{\text{new}}$, then perform overlap test to check the next dimension. If no overlapping area is detected, the test will stop. However, if there is overlapping between different hyperbox classes, a hyperbox contraction process will be made to eliminate the overlap regions, by using cases that have been improved from FMM [3].

To determine the correct adjustment, all cases developed will be checked. Note that cases (12) and (13) also have been used in FMM, while (14) – (23) are newly proposed cases.

Case 1:

$$V_{j\Delta} < V_{k\Delta} < W_{j\Delta} < W_{k\Delta}\ ,\ W_{j\Delta}^{new} = V_{k\Delta}^{new} = \frac{W_{j\Delta}^{old} + V_{k\Delta}^{old}}{2} \tag{12}$$

Case 2:

$$V_{k\Delta} < V_{j\Delta} < W_{k\Delta} < W_{j\Delta}\ ,\ W_{k\Delta}^{new} = V_{j\Delta}^{new} = \frac{W_{k\Delta}^{old} + V_{j\Delta}^{old}}{2} \tag{13}$$

Case 3:

$$V_{j\Delta} = V_{k\Delta} < W_{j\Delta} < W_{k\Delta}\ ,\ V_{k\Delta}^{new} = W_{j\Delta}^{old} \tag{14}$$

Case 4:

$$V_{j\Delta} < V_{k\Delta} < W_{j\Delta} = W_{k\Delta}\ ,\ W_{j\Delta}^{new} = V_{k\Delta}^{old} \tag{15}$$

Case 5:

$$V_{k\Delta} = V_{j\Delta} < W_{k\Delta} < W_{j\Delta}\ ,\ V_{j\Delta}^{new} = W_{k\Delta}^{old} \tag{16}$$

Case 6:

$$V_{k\Delta} < V_{j\Delta} < W_{k\Delta} = W_{j\Delta}\ ,\ W_{k\Delta}^{new} = V_{j\Delta}^{old} \tag{17}$$

Case 7(a):

$V_{j\Delta} < V_{k\Delta} \leq W_{k\Delta} < W_{j\Delta}$ and

$$(W_{k\Delta} - V_{j\Delta}) < (W_{j\Delta} - V_{k\Delta}),\ V_{j\Delta}^{new} = W_{k\Delta}^{old} \tag{18}$$

Case 7(b):

$V_{j\Delta} < V_{k\Delta} \leq W_{k\Delta} < W_{j\Delta}$ and

$$(W_{k\Delta} - V_{j\Delta}) > (W_{j\Delta} - V_{k\Delta}),\ W_{j\Delta}^{new} = V_{k\Delta}^{old} \tag{19}$$

Case 8(a):

$V_{k\Delta} < V_{j\Delta} \leq W_{j\Delta} < W_{k\Delta}$ and

$$(W_{k\Delta} - V_{j\Delta}) < (W_{j\Delta} - V_{k\Delta}),\ W_{k\Delta}^{new} = V_{j\Delta}^{old} \tag{20}$$

Case 8(b):

$V_{k\Delta} < V_{j\Delta} \leq W_{j\Delta} < W_{k\Delta}$ and

$$(W_{k\Delta} - V_{j\Delta}) > (W_{j\Delta} - V_{k\Delta}),\ V_{k\Delta}^{new} = W_{j\Delta}^{old} \tag{21}$$

Case 9(a):

$$V_{j\Delta} = V_{k\Delta} < W_{j\Delta} = W_{k\Delta},\ W_{j\Delta}^{new} = V_{k\Delta}^{new} = \frac{W_{j\Delta}^{old} + V_{k\Delta}^{old}}{2} \tag{22}$$

Case 9(b):

$$V_{k\Delta} = V_{j\Delta} < W_{k\Delta} = W_{j\Delta},\ W_{k\Delta}^{new} = V_{j\Delta}^{new} = \frac{W_{k\Delta}^{old} + V_{j\Delta}^{old}}{2} \tag{23}$$

Case 9(a) will be used to make contraction if maximum point (W_j) of one or more dimensions owned by the hyperbox (i.e., H_j) overlap with other hyperbox (i.e., H_k). Whereas, if a minimum point (V_j) of one or more dimensions owned by H_j overlapping with H_k , case 9(b) will be used.

Although EFMM successfully improved the FMM limitation, there are still some issues not resolved by EFMM, as inherited from FMM. The limitation will weaken the EFMM performance. The issue related is about noise problem. In EFMM, learning with a sample of noisy data results in false knowledge stored as a hyperbox in the network structure. Due to this limitations, Enhanced Fuzzy Min Max with pruning (EFMM with pruning) has been proposed [6].

2.3.3 Enhanced Fuzzy Min-Max with Pruning (EFMM WITH PRUNING)

The previous section has described the technique used by EFMM to addressing the first three FMM limitations. However, due to the expansion rule of the original FMM as adopted in EFMM, some limitations remain unsolved, that is noise tolerance [5]. In EFMM, learning with a sample of noisy data results in false knowledge stored as a hyperbox in the network structure [5]. Due to this limitations, Enhanced Fuzzy Min-Max with pruning (EFMM with pruning) has been proposed [5].

This model proposed pruning strategy to deal with complexity and noise problems. Pruning will identify low hyperbox in accuracy, which is typically due to noise and outliers, and will remove them from the network; thereby reducing network complexity in the existence of noisy data [5].

The pruning strategy used here has been introduced by Carpenter and Tan (1995). In this strategy, the sample data provided will be divided into three subset, that is, training, prediction, and test. Firstly, the training set will be used for learning. Then, the prediction set is used to facilitate pruning of the trained network structure. And finally, the test set is used to evaluate network performance. Similar to the methods employed in Carpenter and Tan (1995), a Hyperbox Accuracy (HA) score is calculated for each hyperbox using prediction set, as follows.

$$HA_j = \left(\frac{\sum_{i=1}^{n} CP_{ji}}{\sum_{i=1}^{n} (CP_{ji} + ICP_{ji})} \right) * 100 \qquad (24)$$

Where *HA* is the hyperbox accuracy, *CP* is the number of correct predictions, *ICP* is the number of incorrect predictions, *j* is the hyperbox index, and *i* is the number of input samples in the prediction set classified by hyperbox (*j*), where $i = (1, 2, ..., n)$. After acquiring *HA* for each hyperbox, the confidence factor of each hyperbox is calculated using Equation (25) [5].

$$CF_{kj} = \frac{HA_{kj}}{Max\ (HA_k)} \qquad (25)$$

Where CF_{kj} is confidence factor of the hyperbox *j* possessed by class *k*. The hyperbox with CF_{kj} is lower than the user-defined pruning threshold ($CF_{kj} < \delta$) will removed. When there are less number of hyperboxes, the process will be faster. Confidence factor finds the hyperbox which are frequency used and the yield higher accuracy [9].

As a result from EFMM with pruning discussion, we find that pruning strategy helps reduce the network complexity in the presence of noisy data [5]. However, some further work still require for improve the robustness of EFMM with pruning.

2.4 CONCLUSION

In conclusion, there are advantages and disadvantages for each neural network model. Based on that, we propose MACS based heterogeneous classifiers instead of using a homogeneous classifiers. That could improve the MACS classification performance by overcomes the limitations of each classifier when handling different classification problems. Figure 2.6 shows the proposed model, MACS with heterogeneous classifiers.

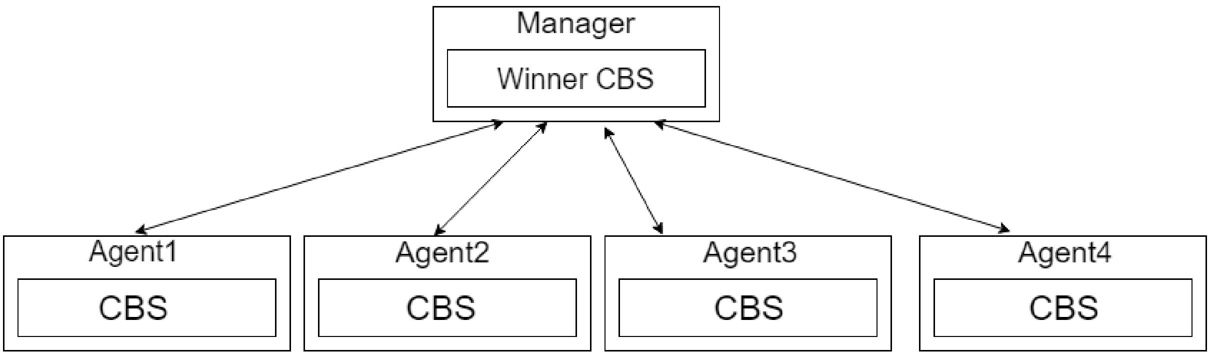


Figure 2.6 MACS with Heterogeneous Classifiers

In this work, our proposed model only using 2 layer, consist of agent layer and manager layer. For neural network model, there are four different neural network models used as learning learning agent, FMM, FMM with pruning, EFMM and EFMM with pruning. By using heterogeneous classifiers, the classification performance could improve because the limitations of each classifiers when handling different classification problem could be overcome.

Table 2.1 shows differences between the previous model and our proposed model.

Table 2.1 Differences between model

Model	Layer	Type of classifiers	Learning Agent
MACS with CBS	2 layers	Homogeneous classifiers	FMM
MACS using TNC	3 layers	Heterogeneous classifiers	One team FMM, one team FAM.
MACS with Bayesian Formalism	3 layers	Heterogeneous classifiers	First team FMM, second team MFMM
MACS with heterogeneous classifiers	2 layers	Heterogeneous classifiers	FMM, FMM with pruning, EFMM, EFMM with pruning.

CHAPTER 3

METHODOLOGY

3.1 INTRODUCTION

This chapter will discuss about the complete methodology and approach used on the flow of this project. It is begin with the discussion of the methodology used in this work. In this chapter also, we will discuss on hardware and software used in this project. Gantt chart that shows every phase and time used during this work also is shown here.

3.2 METHODOLOGY

To get the best results in a job, some steps need to followed. Figure 3.1 shows the methodology used in this work. There are six phases used in this work process. The steps are make background study, analysis and data gathering, design, implementation, testing and final step is maintenance. Some of the phases will discuss in details below.

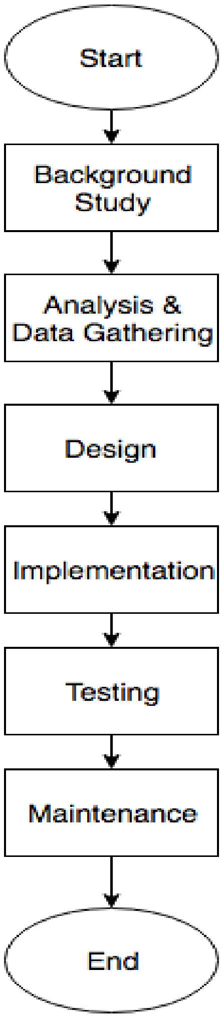


Figure 3.1 Research Methodology

3.2.1 Background Study

Previous model have limitations where the models cannot show good perfomace for dataset that have problem of noise. This will make whole model suffer from noise, and it will negatively effect the classification accuracy. Based on that, we propose to use heterogeneous model. Purpose of this work is to improve the classification accuray and overcome the noise problem.

3.2.2 Analysis and Data Gathering

At this phase, each relevant data needs to be collected for analysis. The data needed is data that will be used from the beginning of the work start until get the final result.

3.2.2.1 Literature Review

After conducting an analysis, it has been found the old model of MACS that using FMM and EFMM as learning agent faces some problems in terms of noise. Thus, the steps have been taken by identifying new models that can cope with this limitation. It has been found EFMM with pruning strategy can solve the problem. A decision has been made to make EFMM with pruning as learning agent in MACS. But, it also has been found, if the data set does not have noise problem, pruning strategy could effect the performance of that data. Due to that, here it is proposed to use heterogeneous classifier rather than homogeneous classifier, so that each one of the model can show a good performance based on type of data set. In this work, two benchmark problems taken from the UCI machine learning repository will be use to evaluate the performance of the proposed model.

3.2.2.2 Data Collection

In this project, two benchmark problems taken from UCI machine learning repository. The data sets taken were Iris and Heart(Statlog) datasets. Table 3.1 shows the information associated with the benchmark data sets.

Table 3.1 Information associated with the benchmark data sets

Benchmark data	Samples	Attributes	Classes
Iris	150	4	3
Heart(Statlog)	270	13	2

1. *Iris*: the Iris data set contains 150 samples, each with four features, from three classes. They are Iris Setora (50 samples), Iris Versicolor (50 samples), and Iris Virginica (50 samples).

2. *Heart(Statlog)*: the Heart(Statlog) data set contains 270 samples, thirteen features, from three classes.

3.2.3 System Design

In design phase, developer can plan every step of a system better. Indirectly, the understanding of the system will increase.

As discussed earlier, MACS-CBS models have been selected in this work. And we will use heterogeneous system to overcome noise problem and improving the accuracy of the classifier. Here, FMM, EFMM, and EFMM with pruning used as a learning agent.

3.2.4 Sequence Diagram

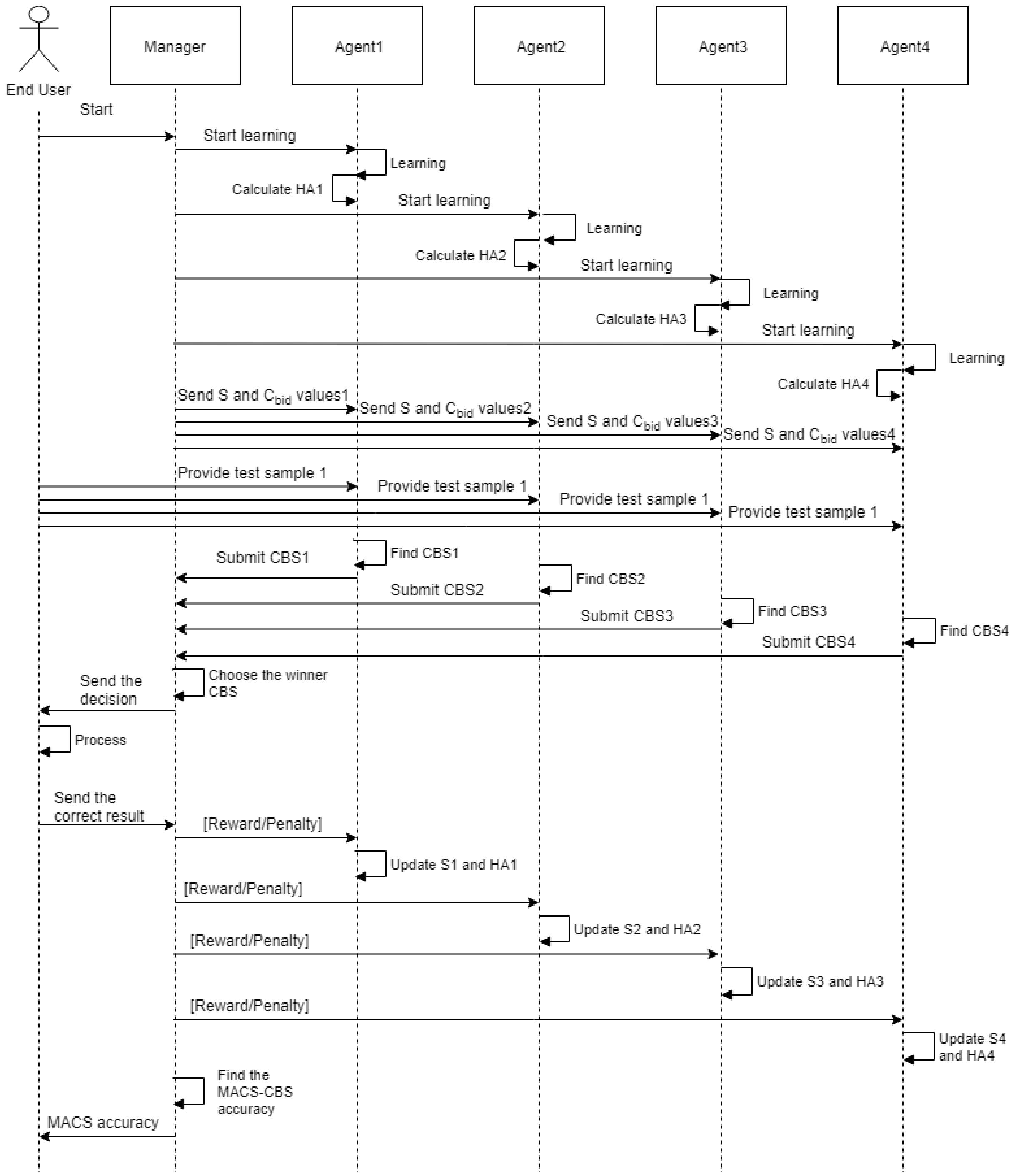


Figure 3.2 Sequence Diagram

3.2.5 Flowchart

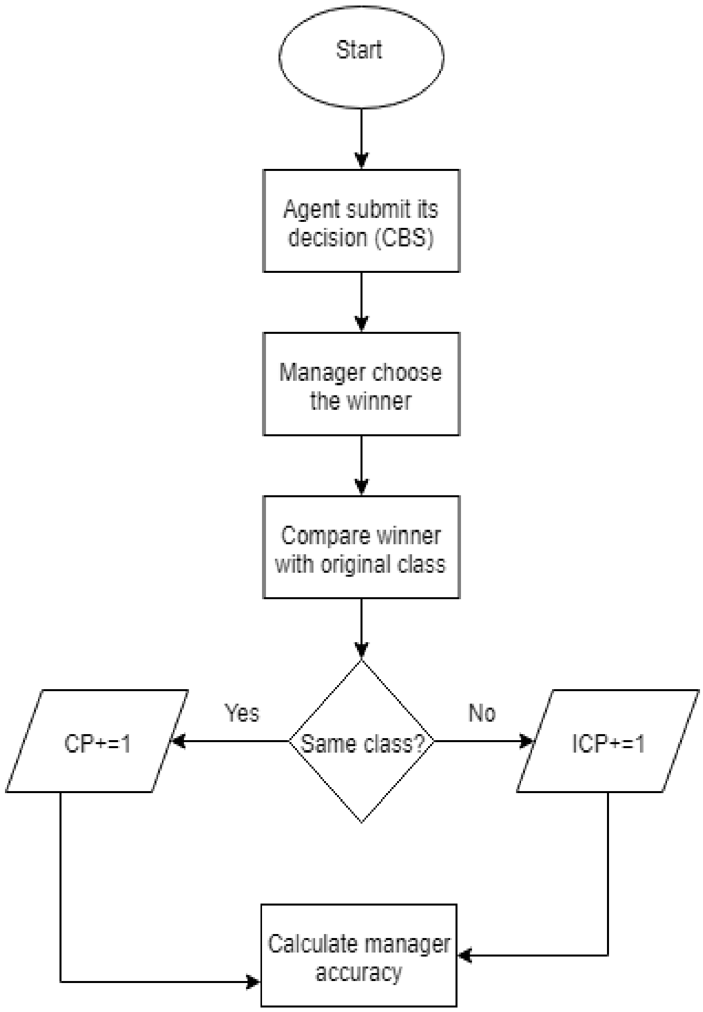


Figure 3.3 Flowchart

Flowchart above shows how the flow of the project. At the beginning, each agent will start learning. Then, real sample test will given to each agent. Each agent will make a decision and send the decision (CBS) to the manager. Next, manager will choose the winning agent by choosing the highest CBS. After get real result, the result will send to the manager. Each agent will be reward or penalize based on their decision, here the strength of each agent is updated. Manager also will be reward or penalize. If the winning agent make a correct decision, manager will be reward, else manager will be penalize. Equation update strength value:

$S_i\,(t + 1) = S_i(t) - P_i(t) - T_i(t) + R_i(t)$

where P is penalty, T is tax, R is reward, i is the classifier index, and t is the time step.

If the prediction is correct, $P_i = 0$ while $R_i = B_i$; otherwise, $P_i = B_i$; while $R_i = 0$. In both condition, $T = 0$. After update S, the agents bid again when receive next test sample. The final test accuracy rate is as follows:

$$\text{Test accuracy} = \left(\frac{\sum_{j=1}^{n} CPTS_j}{\sum_{j=1}^{n} (CPTS_j + ICPTS_j)} \right) \text{ x } 100$$

Where CPTS is number of correctly and ICPTS is incorrectly predicted test samples. In case where all the CBS values are equal, the agent with highest Strength value is selected to be the winner.

3.2.6 Pseudo-code

```
Start learning
Initialize strength
    Agents submit CBS value
    Manager choose the highest CBS value
        Compare  manager with original class
                If same class with original class
                    Correct Prediction = Correct Prediction + 1
                Else
                    Incorrect Prediction = Incorrect Prediction + 1
        Compare agents with original class
                If same class with original class
                    Reward the agent
                Else
                    Penalize the agent
        Calculate manager accuracy
                Manager accuracy = ( Correct prediction / (Correct prediction+Incorrect
predition) ) * 100
```

Figure 3.4 Pseudo-code

3.3 HARDWARE AND SOFTWARE

In this work, some hardware and software are needed. It is very important to ensure every phase in this work runs smoothly from the beginning to the end. So in this section, we will discuss some of the necessary hardware and software.

3.3.1 Hardware Specification

In this work, some hardware has been used. Each of the functionality of the hardware used is different, but it all carries the same goal, which is to facilitate and smooth the work. Table 3.2 shows the hardware used during this work.

Table 3.2 Hardware Specifications

Hardware	Purpose
Laptop	Work with documentation, and do some research
Printer	Print any research paper and documentation which is related to PSM

Personal laptop is used to make documentation and do some research. While, printer is used to print any research paper as references, or to submit the decimation to supervisor and faculty.

3.3.2 Software Tools

Other than that, some software also used in this work. Each of the selected software is important to ensure every phase in this project work smoothly. Table 3.5 shows the software and specifications used during the development.

Table 3.3 Software Tools

Software	Purpose
Matlab	To make the coding for the project

Microsoft Word	Used for documentation
Microsoft Project	To create Gantt Chart
Draw.io	To create flowchart and sequence diagram

Matlab is the main software used in this work for implementation. While, Microsoft Word used for documentation, and Microsoft Project used to produce a Gantt chart timeline. In this project also, Draw.io used to create a flow chart and sequence diagram.

3.4 GANTT CHART

Refer appendix A

CHAPTER 4

RESULTS AND DISCUSSION

4.1 INTRODUCTION

This chapter discussed the result of the implementation. This result will be discussed and conclusion are made to compare both results.

4.2 RESULT

The implementation used MATLAB to run the code. Table below shows the result.

Table 4.1 Iris Datasets

Agent1	Agent2	Agent3	Agent4	MACS
100	100	100	100	100
100	100	100	100	100
100	100	100	100	100
100	100	100	100	100
100	100	100	80	100
100	100	100	83.33	100
100	100	100	85.71	100
100	100	100	87.50	100
100	100	100	88.89	100
100	100	100	90	100
100	100	100	81.82	100
100	100	100	83.33	100
100	100	100	76.92	100

100	100	100	78.57	100
100	100	100	80	100
100	100	100	81.25	100
100	100	100	82.35	100
100	100	100	83.33	100
100	100	100	84.21	100
100	100	100	85.00	100
100	100	100	85.71	100
100	100	100	86.36	100
100	100	100	86.96	100
100	100	100	87.50	100
100	100	100	88.00	100
100	100	100	88.46	100
100	100	100	88.89	100
100	100	100	89.29	100
100	100	100	89.66	100
100	100	100	90.00	100

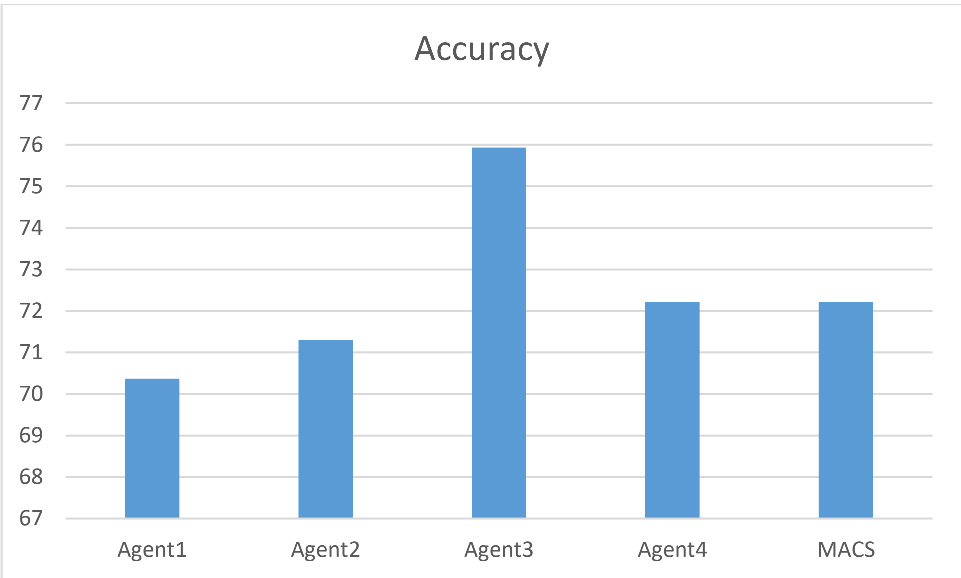


Figure 4.1 Result based on Iris Datasets

Table 4.2 Heart Datasets

Agent1	Agent2	Agent3	Agent4	MACS
100	100	100	100	100
100	100	100	100	100
100	100	100	100	100
100	100	100	100	100
80	80	80	80	80
83.33	83.33	83.33	83.33	83.33
85.71	85.71	85.71	85.71	85.71
87.50	87.50	87.50	87.50	87.50
77.78	77.78	77.78	77.78	77.78
80	80	80	80	80
81.82	81.82	81.82	81.82	81.82
75	75	75	75	75
76.92	76.92	76.92	76.92	76.92
78.57	78.57	78.57	78.57	78.57
80	80	80	80	80
81.25	81.25	81.25	81.25	81.25
82.35	82.35	82.35	82.35	82.35
83.33	83.33	83.33	83.33	83.33
84.21	84.21	84.21	84.21	84.21
85	85	85	85	85
85.71	85.71	85.71	85.71	85.71
86.36	86.36	86.36	86.36	86.36
86.96	86.96	86.96	86.96	86.96
87.50	87.50	83.33	87.50	83.33
88	88	84	88	84
88.46	88.46	84.62	88.46	84.62
88.89	88.89	85.19	88.89	85.19
85.71	85.71	85.71	85.71	82.14
86.21	86.21	82.76	86.21	82.76

83.33	83.33	80.00	83.33	80.00
80.65	80.65	77.42	80.65	77.42
81.25	81.25	78.13	81.25	78.13
78.79	78.79	78.79	81.82	75.76
79.41	79.41	79.41	82.35	76.47
80.00	80.00	80.00	82.86	77.14
80.56	80.56	80.56	83.33	77.78
81.08	81.08	81.08	83.78	78.38
78.95	78.95	78.95	81.58	76.32
76.92	76.92	76.92	79.49	74.36
77.50	77.50	77.50	80.00	75.00
75.61	75.61	75.61	78.05	73.17
73.81	73.81	76.19	76.19	71.43
74.42	74.42	76.74	76.74	72.09
75.00	75.00	77.27	77.27	72.72
75.56	75.56	77.78	77.78	73.33
76.09	76.09	76.09	76.09	71.74
74.47	74.47	74.47	74.47	70.21
75.00	75.00	75.00	75.00	70.83
73.47	73.47	75.51	75.51	71.43
74.00	74.00	76.00	76.00	72.00
74.51	72.55	76.47	76.47	72.55
73.08	71.15	76.92	76.92	73.08
73.58	71.70	77.36	77.36	73.58
72.22	72.22	75.93	75.93	72.22
72.72	72.72	76.36	76.36	72.72
73.21	73.21	76.79	75.00	71.43
71.93	71.93	77.19	75.44	71.93
70.69	70.69	77.58	74.14	70.69
69.49	69.49	76.27	72.88	69.49
70.00	70.00	76.67	73.33	70.00
70.49	70.49	77.05	73.77	70.49

70.97	70.97	77.42	74.19	70.97
71.43	71.43	77.78	74.60	71.43
71.88	71.88	78.13	75.00	71.88
70.77	70.77	76.92	73.85	70.77
71.21	71.21	77.27	74.24	71.21
71.64	71.64	77.61	74.63	71.64
72.06	72.06	77.94	73.53	70.59
72.46	72.46	78.26	73.91	71.01
72.86	72.86	78.57	74.29	71.43
73.23	73.24	78.87	74.65	71.83
72.22	72.22	77.78	73.61	70.83
72.60	72.60	78.08	73.97	71.23
71.62	71.62	77.03	72.97	70.27
70.67	70.67	76.00	72.00	69.33
71.05	71.05	76.32	72.37	69.74
71.43	71.43	76.62	72.73	70.13
71.80	70.51	76.92	73.08	70.51
72.15	70.89	77.22	73.42	70.89
71.25	70.00	77.50	73.75	71.25
71.60	70.37	77.78	74.07	71.60
71.95	70.73	78.05	74.39	71.95
72.29	71.08	78.31	74.70	72.29
71.43	70.24	77.38	75.00	71.43
70.59	69.41	76.47	74.12	70.59
70.93	69.77	76.74	73.26	70.93
71.26	70.11	77.01	73.56	71.26
70.46	70.45	76.14	72.73	71.59
69.66	69.66	75.28	71.91	70.79
70.00	70.00	75.56	72.22	71.11
69.23	70.32	75.82	71.43	71.43
69.56	70.65	76.09	71.74	71.74
69.89	70.97	76.34	72.04	72.04

70.21	71.28	76.60	72.34	72.34
69.47	70.53	76.79	71.58	71.68
69.80	70.83	76.04	71.88	71.88
69.07	70.10	75.26	71.13	71.13
68.37	69.39	74.49	70.41	70.41
68.69	69.70	74.75	70.71	70.71
69.00	70.00	75.00	71.00	71.00
69.31	70.30	75.25	71.29	71.29
69.61	70.59	75.49	71.57	71.57
69.90	70.87	75.73	71.84	71.84
70.19	71.15	75.00	72.12	71.15
70.48	71.43	75.24	72.38	71.43
70.75	71.70	75.47	72.64	71.70
70.09	71.03	75.70	72.90	71.97
70.37	71.30	75.93	72.22	72.22

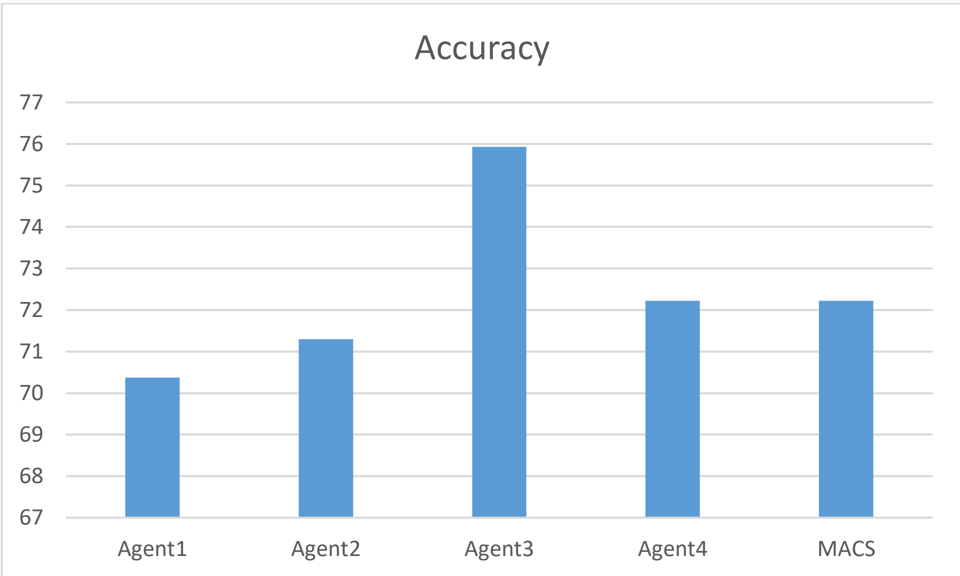


Figure 4.2 Result based on Heart Datasets

4.2.1 Result Discussion

Based on the table 4.1, it shows the Iris datasets where our proposed model consists of four heterogeneous classifiers, and CBS as the nucleus of each agent used to trained the Iris datasets to make the decision. The value in the table showed the total of accuracy of each agent. The accuracy value is updated each time receive new test sample. Based on the result, it shows that managers accuracy is always 100 percent which means all the agents that choosen by manager are correct. Based on that, we can say this model shows great performance because manager accuracy are always 100 percent.

Based on the table 4.2, it show the Heart datasets where our proposed model consists of four heterogeneous classifiers, and CBS as the nucleus of each agent used to trained the Heart datasets to make the decision. Based on the result, it shows that manager accuracy is fluctuated because sometimes it choosing wrong agents. However, in final test accuracy, manager's accuracy is 72.22 that are the second highest accuracy, where still can be consider as highest accuracy.

In comparison for both datasets, we can see that the proposed work perfectly with Iris datasets, however it does not show a good accuracy for the Heart datasets. In the project, I use Strength value instead of agent reputation to finalize the manager decision in selecting the winner agent. That could affect the classification accuracy in case of heart data set.

CHAPTER 5

CONCLUSION

5.1 INTRODUCTION

Based on the previous analysis, the previous MACS model is based on homogeneous classifiers. Both of these classifier cannot overcome the data noise problem, and this can affect MACS classification performance. Hence, we proposed to used EFMM with pruning for handling noise problem. However, pruning strategy have a drawback related to the size of data and number of patterns, where using a small number of input pattern during the prediction stage could affect the performance quality. Based on that, we found that each neural network model has its own advantages and disadvantages. Therefore, we proposed heterogeneous model to increase MACS classification performance. By using the new model, it helps to find one best decision (highest CBS value) from the multiple decisions that have been made by the agents. Each agent will produce its decision (CBS) to manager. Then, the manager will choose the winner based on the highest CBS value, and compare it with the original class. If its same class, the manager will be reward, else, the manager will be penalize. The implementation has be done by using MATLAB software.

From this project, it shows how important a methodology during doing the work. The flow of work was planned well, indirectly the work able to be completed immediately at the designated time. In addition, it is important to make an analysis first before doing a work. Throught the analysis, it will be easier to understand the model that we want to use and we will more understand and can see clearer the method that we used.

5.2 CONSTRAINT

1. Not suitable for some datasets

The proposed method is not suitable for some dataset and it have affect the classification accuracy.

5.3 FUTURE WORK

Some future work needs to be considered to improve this research project. In this project , the future work is focused on evaluating the resulting model using various benchmarks and real data sets. Other than that, it can be improve by using agent reputation other than Strength value to increase the performance.

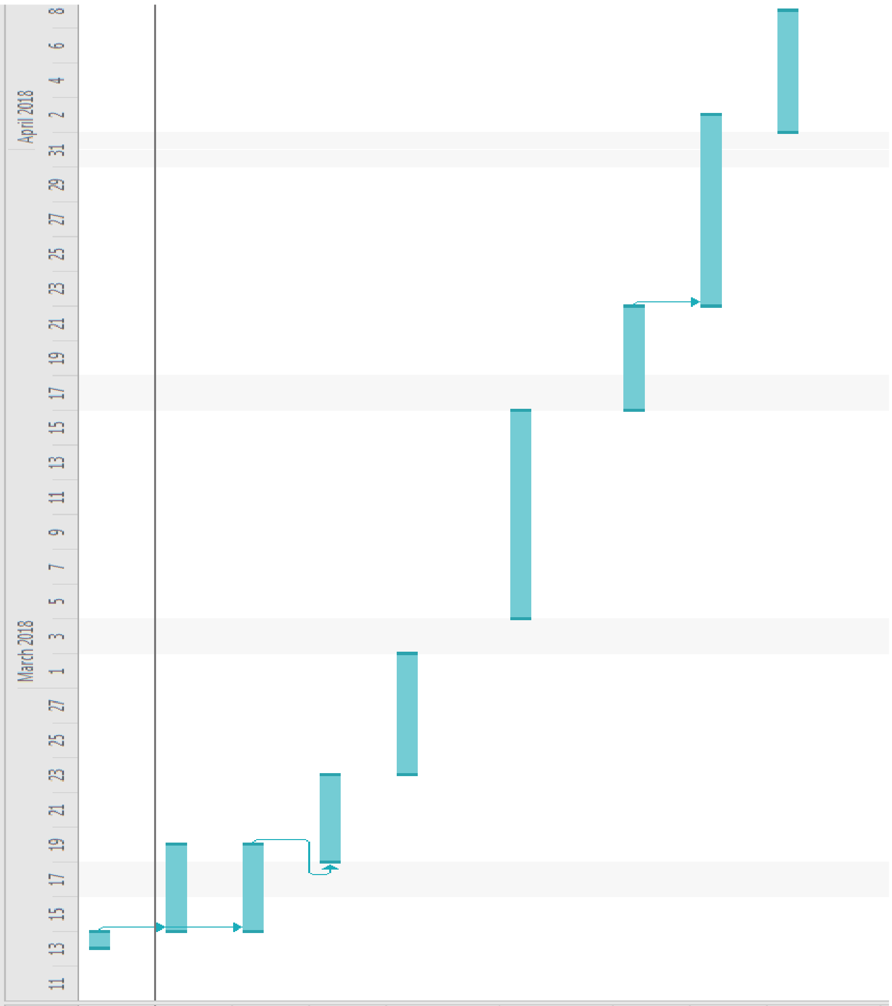
REFERENCES

1. Mohammed, M., Lim, C., & Quteishat, A. (2012) A novel trust measurement method based on certified belief in strength for a multi-agent classifier system. *Neural Computing And Applications*, 24(2), 421-429. http://dx.doi.org/10.1007/s00521-012-1245-2
2. Quteishat, A., Lim, C., Saleh, J., Tweedale, J., & Jain, L. (2010). A neural network-based multi-agent classifier system. *Neurocomputing*, 72(2), 1639-1647. https://doi.org/10.1016/j.neucom.2008.08.012
3. Mohammed Falah Mohammed, C. P. L. (2015). An Enhanced Fuzzy Min-Max Neural Network for Pattern Classification, *26*(3), 417–428. Retrieved from http://ieeexplore.ieee.org/xpls/icp.jsp?arnumber=6808500
4. Mohammed, M., & Lim, C. (2017). A new hyperbox selection rule and a pruning strategy for the enhanced fuzzy min-max neural network. *Neural Networks*, 86, 69-79. http://dx.doi.org/10.1016/j.neunet.2016.10.012
5. Mohammed, M., & Lim, C. (2017). A new hyperbox selection rule and a pruning strategy for the enhanced fuzzy min-max neural network. *Neural Networks*, 86, 69-79. http://dx.doi.org/10.1016/j.neunet.2016.10.012
6. Sean A. Gilpin & Daniel M. Dunlavy (2008). HETEROGENEOUS ENSEMBLE CLASSIFIATION. Retrieved from http://www.cs.sandia.gov/~dmdunla/publications/SAND2009-0203P.pdf
7. Quteishat, A., Lim, C., Saleh, J., Tweedale, J., Tweedale, J., & Jain, L. A neural network-based multi-agent classifier system with a Bayesian formalism for trust measurement. *Soft Computing,* 15(2), 221-231. http://dx.doi.org/10.1007/200500-010-0592-0
8. Mohammed, M., Lim, C. Improving the Fuzzy Min-Max neural network with a K-nearest hyperbox expansion rule for pattern classification. *Applied Soft Computing*, 52 135-145. https://doi.org/10.1016/j.asoc.2016.12.001
9. Ashish D. Bopte, Dr. D. M. Yadav, Prof. S. B. Shinde. An Enhanced Fuzzy Min-Max Neural Network Based on Pruning Algorithm for Pattern Classification. *International Journal of Advanced Research in Computer and Communication Engineering*, 5(6). https://doi.org/10.17148/ijarcce.2016.5677

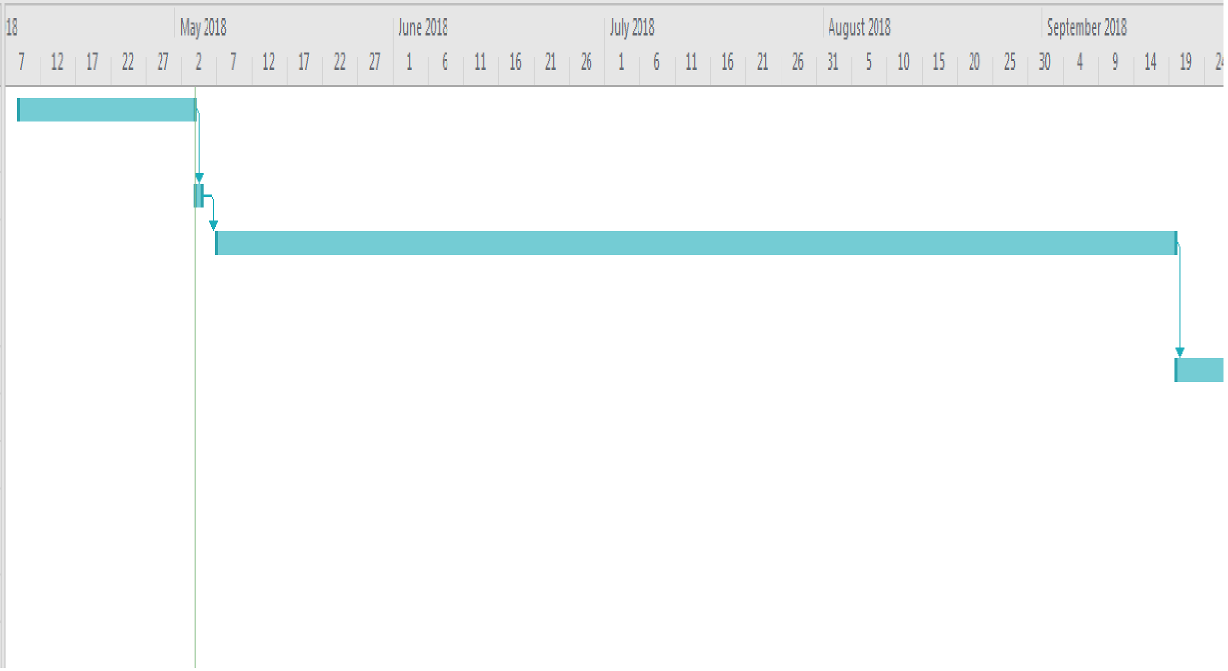
10. Walter I, Gomide F (2006). Design of coordination strategies in multiagent systems via genetic fuzzy systems. *Soft Comput Fusion Found Methodol Appl*, 10:903-915

APPENDIX A

	Task Name	Duration	Start	Finish	Predecessors
1	Discuss title with supervisor	1 day	Wed 2/14/18	Wed 2/14/18	
2	Analyse problem of the topic	3 days	Thu 2/15/18	Mon 2/19/18	1
3	Determine objective and scope	3 days	Thu 2/15/18	Mon 2/19/18	1
4	Details discussion about MAS	5 days	Mon 2/19/18	Fri 2/23/18	3
5	Details discussion about heterogeneous and homogeneous	6 days	Sat 2/24/18	Fri 3/2/18	
6	Comparison of existing neural network model	10 days	Mon 3/5/18	Fri 3/16/18	
7	Gather all data needed	5 days	Sat 3/17/18	Thu 3/22/18	
8	Details discussion for research methodology	7 days	Fri 3/23/18	Mon 4/2/18	7
9	Design the implementation method	6 days	Mon 4/2/18	Sun 4/8/18	



	Task Name	Duration	Start	Finish	Predecessors
10	Improve writing for documentation	19 days	Mon 4/9/18	Thu 5/3/18	
11	Finalize report PSM 1	1 day	Fri 5/4/18	Fri 5/4/18	10
12	Make some improvement and modification	98 days	Mon 5/7/18	Wed 9/19/18	11
13	Implement coding	20 days	Thu 9/20/18	Wed 10/17/18	12
14	Testing the code	17 days	Thu 10/18/18	Fri 11/9/18	13
15	Generate the result	6 days	Mon 11/12/18	Mon 11/19/18	14
16	Improve writing for documentation	15 days	Sat 11/17/18	Thu 12/6/18	
17	Finalize report PSM 2	1 day	Fri 12/7/18	Fri 12/7/18	16



	Task Name	Duration	Start	Finish	Predecessors
10	Improve writing for documentation	19 days	Mon 4/9/18	Thu 5/3/18	
11	Finalize report PSM 1	1 day	Fri 5/4/18	Fri 5/4/18	10
12	Make some improvement and modification	98 days	Mon 5/7/18	Wed 9/19/18	11
13	Implement coding	20 days	Thu 9/20/18	Wed 10/17/18	12
14	Testing the code	17 days	Thu 10/18/18	Fri 11/9/18	13
15	Generate the result	6 days	Mon 11/12/18	Mon 11/19/18	14
16	Improve writing for documentation	15 days	Sat 11/17/18	Thu 12/6/18	
17	Finalize report PSM 2	1 day	Fri 12/7/18	Fri 12/7/18	16