Feature Selection for Document Classification: Case Study of Meta-heuristic Intelligence and Traditional Approaches

Khin Sandar Kyaw

A Thesis Submitted in Fulfillment of the Requirement for the Degree of

Doctor of Philosophy in Computer Engineering

Prince of Songkla University

2020

Feature Selection for Document Classification: Case Study of Meta-heuristic Intelligence and Traditional Approaches

Khin Sandar Kyaw

A Thesis Submitted in Fulfillment of the Requirement for the Degree of

Doctor of Philosophy in Computer Engineering

Prince of Songkla University

2020

Thesis Title	Feature Selection for Document Classification: Case Study of Meta-heuristic Intelligence and Traditional Approaches
Author	Ms. Khin Sandar Kyaw
Major Program	Computer Engineering

Major Advisor

……
(Dr. Somchai Limsiroratana)

Examining Committee:

……Chairperson
(Assoc.Prof. Dr. Ponrudee Netisopakul)

……Committee
(Dr. Somchai Limsiroratana)

……Committee
(Asst. Prof. Dr. Panyayot Chaikan)

……Committee
(Asst. Prof. Dr. Pichaya Tandayya)

……Committee
(Dr. Anant Choksuriwong)

The Graduate School, Prince of Songkla University, has approved this thesis as fulfillment of the requirements for the Degree of Doctor of Philosophy in Computer Engineering.

……
(Prof. Dr. Damrongsak Faroongsarng)
Dean of Graduate School

This is to certify that the work here submitted is the result of the candidate's own investigations. Due acknowledgement has been made of any assistance received.

…….....Signature
(Dr. Somchai Limsiroratana)
Major Advisor

…….....Signature
(Ms. Khin Sandar Kyaw)
Candidate

I hereby certify that this work has not been accepted in substance for any degree and is not being currently submitted in candidature for any degree.

…….……Signature
(Ms. Khin Sandar Kyaw)
Candidate

Thesis Title	Feature Selection Process for Document Classification: Case Study of Meta-heuristic Intelligence and Traditional Approaches
Author	Ms. Khin Sandar Kyaw
Major Program	Computer Engineering
Academic Year	2019

ABSTRACT

Nowadays, the culture for accessing news around the world is changed from paper to electronic format and the rate of publication for newspapers and magazines on website are increased dramatically. Meanwhile, text feature selection for the automatic document classification (ADC) is becoming a big challenge because of the unstructured nature of text feature, which is called "multi-dimension feature problem". On the other hand, various powerful schemes dealing with text feature selection are being developed continuously nowadays, but there still exists a research gap for "optimization of feature selection problem (OFSP)", which can be looked for the global optimal features. Meanwhile, the capacity of meta-heuristic intelligence for knowledge discovery process (KDP) is also become the critical role to overcome NP-hard problem of OFSP by providing effective performance and efficient computation time. Therefore, the idea of meta-heuristic based approach for optimization of feature selection is proposed in this research to search the global optimal features for ADC.

In this thesis, case study of meta-heuristic intelligence and traditional approaches for feature selection optimization process in document classification is observed. It includes eleven meta-heuristic algorithms such as Ant Colony search, Artificial Bee Colony search, Bat search, Cuckoo search, Evolutionary search, Elephant search, Firefly search, Flower search, Genetic search, Rhinoceros search, and Wolf search, for searching the optimal feature subset for document classification. Then, the results of proposed model are compared with three traditional search algorithms like Best First search (BFS), Greedy Stepwise (GS), and Ranker search (RS). In addition, the framework of data mining is applied. It involves data preprocessing, feature engineering, building learning model and evaluating the performance of proposed meta-heuristic intelligence-based feature selection using various performance and

computation complexity evaluation schemes. In data processing, tokenization, stop-words handling, stemming and lemmatizing, and normalization are applied. In feature engineering process, n-gram TF-IDF feature extraction is used for implementing feature vector and both filter and wrapper approach are applied for observing different cases. In addition, three different classifiers like J48, Naïve Bayes, and Support Vector Machine, are used for building the document classification model. According to the results, the proposed system can reduce the number of selected features dramatically that can deteriorate learning model performance. In addition, the selected global subset features can yield better performance than traditional search according to single objective function of proposed model.

ACKNOWLEDGEMENTS

Firstly, I would like to express my deepest gratitude toward the tireless and prompt help of my supervisor, Dr.Somchai Limsiroratana, who leads me to the world of research in the specific area of data mining and has always allowed me complete freedom to define and explore my own direction for doing my research. Without his endless enthusiasm, motivation, generosity, understanding, and support, this study would hardly have been completed. In addition, acknowledges to his valuable suggestions, guidelines, and explanation patiently. Above all, my invaluable insight of life tremendously gained throughout of doing research and even preparing publication and writing of thesis report.

Secondly, I must express my special gratitude to head of TEH-AC Scholarship Organization (Grant No.058/2016) and other decision makers in Department of Computer Engineering (CoE) at Prince of Songkla University for giving me the opportunity for Doctoral program and offer me full scholarship to cover the tuition fee and living cost for three years here. Also, I would like to acknowledge PSU Graduate School for thesis research funding. In addition, I would like to thank our department of Computer Engineering under the Faculty of Engineering for providing funding of trip for attending international conferences.

Thirdly, I would like to express my warmest gratitude to my thesis committee members, Assoc.Prof. Dr. Ponrudee Netisopakul, Asst. Prof. Dr. Pichaya Tandayya, Asst. Prof. Dr. Panyayot Chaikan, Dr. Anant Choksuriwong, and all teachers in the Department of Computer Engineering at PSU. Thanks for their hard work in teaching, advices, reviews and valuable comments for technical proof and English proofreading, which are very helpful to finish my thesis successfully. In addition, I would like to acknowledge all staffs in Department of Computer Engineering, and special thanks to Ms. Bongkot Prucksapong, for dealing with the help for the arrangement of academic documents through out of my academic research work at CoE.

Moreover, thanks to all my friends and fellow classmates in PSU. I feel so good to have you guys to accompany me all the time. Furthermore, thanks to other

competitors in Kaggle and many open source contributors, for sharing their ideas and answering questions in forum. Moreover, special thanks to my families who support me so selflessly all the time. No matter where I am, they are always giving a hand to me when I need help, and I would never have been able to finish this thesis without their helps. I love all of them, always and forever.

Last, but not least, I would like to thank all people who help, support, encourage me directly or indirectly through out of my Ph. D research at PSU.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	Term
A	Accuracy
ACO	Ant Colony Optimization Algorithm
ABC	Artificial Bee Colony Algorithm
ADC	Automatic Document Classification
ADC-OFSMI	Automatic Document Classification - Optimization of Feature Selection based on Meta-heuristic Intelligence
ADC-OFSABC	Automatic Document Classification - Optimization of Feature Selection based on Artificial Bee Colony
ADC-OFSACO	Automatic Document Classification - Optimization of Feature Selection based on Ant Colony
BO	Bat Optimization algorithm
BFS	Best First Search
CSE	Classifier Subset Evaluation
CCI	Correctly Classified Instance
CFS	Correlation-based Feature Subset Selector
CO	Cuckoo Optimization Algorithm
EO	Elephant Optimization algorithm
EA	Evolutionary Algorithm
FO	Firefly Optimization Algorithm
FPO	Flower Pollination Optimization Algorithm
GA	Genetic Algorithm
GS	Greedy Stepwise
ICCI	Incorrectly Classified Instance
NI	Iteration Number
KDP	Knowledge Discovery Process
MAE	Mean Absolute Error
MI	Meta-heuristic Intelligence
NB	Naïve Baye

Abbreviation	Term
NL	Number of Leaves
NF/NSF/SF	Number of Selected Feature
OF	Objective Function
NP	Population Number
P	Precision
PCA	Principal Component Analysis
RS	Ranker Search
R	Recall
RO	Rhinoceros Optimization Algorithm
RMSE	Root Mean-Squared Error
RRSE	Root Relative Squared Error
ST	Size of Tree
SVM/SMO	Support Vector Machine
TF-IDF	Term Frequency Inverse Document Frequency
TLM/TCM	Total Computing Time for Classification Model
WI-OMFS	Wolf Intelligence based Optimization of Multi-dimensional Feature Selection
WO	Wolf Optimization algorithm

CHAPTER 1
INTRODUCTION

1.1 Introduction and Motivation

Nowadays, data growth on World Wide Web (WWW) is increased explosively by collecting a huge amount of various information via some modern techniques like OLTP (Online Transaction Processing), e-commerce, blogging, social communication network, online news and education channels, and data warehouse [1]. As a consequence, the role of discovery knowledge from documents becomes the hot research topic today for building the automatic prediction system for information retrieval (IR). This process is known as text mining. In addition, the extraction of knowledge from document require more preprocessing stages before deep jumping into the layer of mining process because text feature is complex and high- dimensional property which can happen the NP-hard problem. Hence, the exploration of feature selection process is regarded as significant issue for document classification.

To overcome this issue, the representative features from hypothesis must be selected to reduce the feature dimension because irrelevant features hurt the document classification performance. The purpose of feature selection is to obtain excellent performance and computation of learning model by removing the irrelevant and/or redundant features that lead confusion to draw the boundary line for classifying different objects. To achieve the relevant features that can reflect the type of object correctly, various feature selection techniques can be chosen according to the types of interested dataset such as text, video, image, etc., and the behavior of problem likes simple feature, multi-dimensional feature, and so on. Therefore, features analysis process should be made by visualization techniques to learn the characteristic of interested dataset.

Meanwhile, the searching policy should be adapted to dynamic nature for finding the optimal solution with effective performance and computing time, which is called “feature selection optimization”. The policy of randomization search for candidate solution selection rather than the bias searching policy such as hill-climbing search, exhaustive search, etc., should be employed. In order to develop the search

model described above, meta-heuristic intelligence (MI) can be applied for searching process in data mining research area because it can support randomization policy in terms of the various natural intelligence such as decentralizing the jobs to individual agents for searching the local and global optima for the best food source randomly.

1.2 Background: Data Mining and Text Mining

Today is the age of information becoming true by the explosive growth of available data, volume and gigantic body of data from time to time continuously in different areas such as business, agriculture, science, society, medicine, military, academic, almost every sector of daily life as a consequence of pouring data into the computerization of our network and society, the World Wide Web, and the rapid development of powerful data collection and storage tools [2]. In a widely accepted methodology, namely knowledge discovery process [3] which is the discovery of knowledge from data which becomes part of the natural evolution of information technology. It is to be a high demand for various areas of prediction in future event and analysis of data that have meaningful information as a global challenge for moving forward from data to information age, which has led to the birth of data mining.

In the state of art the of data mining, it moves from the simple data mining technology to more sophisticated mining schemes depending on the nature of data and their complexity in the range of most basic forms of data for mining applications like database data, data warehouse data, and transactional data, to other versatile forms and structures of data and rather different semantic meanings such as data streams (e.g., video surveillance and sensor data), graph or networked data (e.g., social and information networks), text data, ordered/sequence data, spatial data (e.g., maps), hypertext and multimedia data (e.g., text, image, video, and audio data), engineering design data (e.g., design of buildings, system components, or integrated circuits) and the Web (e.g., a big, widely distributed information repository made available by the Internet). In other words, the trend of data mining will exactly continue to embrace new data types in order to handle data carrying not only special structures (e.g., sequences, trees, graphs, and networks) but also rich structures and semantics (e.g., text, image, audio, video and connectivity).

Text mining is a subset of text analytics to extract the key phrases, concepts, etc. It is focused on applying data mining techniques in the domain of textual information using NLP and machine learning. In addition, text mining uses some methodologies from various areas such as information extraction, information retrieval, computational linguistics, categorization, clustering, summarization, topic tracking, and concept linkage.

1.3 Problem Statements

Since text document includes complex and high-dimensional feature, the exploration of feature selection process for multi- dimensional feature set is becoming the significant issue for the development of automatic document classification model. In other words, a challenge of computation exists in finding a global optimal solution from a tremendously huge search space. Formerly, conventional search for the optimal in an enormous search space became impracticable approach for NP-hard problem and therefore many researchers seek for the feature selection model for solving the optimization problem. The nature of conventional search is bias searching policy such as hill-climbing search, exhaustive search. As a good finding, meta-heuristic intelligence based global optimization approach is advanced significantly for solving the complex problem such as multi-dimensional feature selection. It can look for the global optimal solution from a large space of candidate solutions. MI is the category of non-deterministic algorithm that consists of a group of search agents for exploring the feasible region based on both randomization and some rules. The purpose of feature selection is to remove irrelevant and/or redundant features that can hurt for classification performance. In order to develop the search model described above, meta-heuristic intelligence-based optimization of feature selection process is proposed for document classification in this thesis. The nature of meta-heuristic algorithms can provide better performance for searching global optimal feature subset from multi-dimensional feature selection because it can support various natural intelligence such as decentralizing the task for near-optimal solutions.

1.4 Objectives

This thesis includes the state of the art for biological behavior of meta-

heuristic algorithms and optimization of multi-dimensional feature selection process for document classification based on meta-heuristic intelligence search policy. The main objective of this research is to describe the apprehension of artificial intelligence (AI) community to the investigation of feature selection process in accompany with advanced searching capability based on meta-heuristic schemes that can improve the searching ability for hypothesis of high-dimensional feature space in document classification problem. Through this purpose, the review of contemporary modern solutions with divergent types of searching policy is studied, but homogeneous objective of optimization; to facilitating feature searching for the discovery of optimal feature in feature engineering process. Although the critical objective of this thesis is intended to show the ability of meta-heuristic search in multi-dimension of complex feature space, several diverse characteristics of feature selection schemes for the measurement of text feature do bear in mind to investigate, which are described in following:

(A). Universal elucidation of problem definition and role of supreme which are related with feature selection and searching processes for AI and data mining community;

(B). State of the art of searching methodologies for feature selection process with respect to the purpose of optimization problems;

(C). Applied areas of feature selection with meta-heuristic searching scheme based on the swarms' intelligence and others natural intelligence in document classification problem;

(D). Different methods for the calculation of text feature extraction based on statistical machine learning models and various feature selection schemes such as filter and wrapper.

1.5 Research Questions and Contributions

When the trend for classification of document is moved toward automatic era, the technology for mining the data is also changed adaptably from simple to more sophisticated models in according to the society requirement and the new characteristic of data for both prediction in the classification area and analysis in the

clustering area (descriptive). Therefore, the general research question for this thesis is how to predict the category of testing text documents based on the training documents automatically. In addition, the more specific research question is how to optimize the feature selection process in order to achieve global optimal feature subset for the multi-dimensional document classification. To answer the research questions, meta-heuristic intelligence based optimization of feature selection process is proposed for automatic document classification as it can support the global optimal feature subset by using the capability of meta-heuristic search policy such as randomization and distributed search in multi-dimension feature space.

1.6 Benefits and Scope of Thesis

As an expected benefit of this thesis, the proposed system can reduce the number of selected features dramatically and it can support better optimal classifier performance than traditional search approaches. The scope of the proposed optimization of feature selection process is tested on news documents of five categories.

1.7 Thesis Outlines

The structure of thesis will be described in five chapters procedurally started from the history and background of research as a motivation sector of the thesis to the end of expected outcome. It includes several chapters about problem discovering and introduction, literature reviews, system design and implementation for expected outcome, evaluation of the proposed system with results and discussion, and implication of the thesis with conclusion and future work.

Chapter 1 introduces motivation, background, problem statements, objectives, research questions and contributions, and benefits and scope.

Chapter 2 focuses on literature reviews which includes attribute selection scheme for text mining, text document classification systems, optimization of feature selection using MI in various sectors, traditional search and meta-heuristic based search schemes.

Chapter 3 describe system design and implementation which includes basic concepts and theories related with document classification framework, dataset, library files, and parameters setting for experiments of proposed system.

Chapter 4 discusses several case studies results for proposed system using eleven meta-heuristic based search schemes, comparison of results with traditional search-based models using various evaluation schemes.

Chapter 5 presents the conclusion of the thesis with the area of implication, and future work.

CHAPTER 2

LITERATURE REVIEWS

This chapter provides the background information related to this thesis which includes text mining and attribute selection schemes, text document classification, feature selection optimization process, traditional search, and advance search based on meta-heuristic intelligence.

2.1 Text Mining and Attribute Selection Schemes

In data mining field, there are various branches for mining process based on the type of data such as relational databases (DB), data warehouses, advanced DB and information repositories, object-oriented and object-relational databases, transactional and spatial databases, heterogeneous and streaming database, and text databases. Text mining is a concept like data mining, but instead of looking for patterns in data. Text mining involves finding patterns in text. In this regard, text is very amorphous, and more difficult to deal with than numeric data in the process of data mining. In the process of text mining, the process of feature selection is important due to high-dimensional text features that can lead to low accuracy and high computation cost for the classification model. In addition, text mining uses methodologies from various areas such as information extraction, computational linguistics, categorization, clustering, topic tracking, etc. The domains for applications of text mining can be broadly organized into two groups: document exploration and analysis tools.

One of the most important tasks for document classification process is that the document representation and feature selection process (attribute selection). The document representation can be described by two general ways: a bag of words (BOW)- a document is described as a set of words accompany with their associated frequency in the document and is used commonly due to its simplicity; and direct representation of text as strings in which each document is the sequence of words. In this section, theory and calculation about feature selection methods for both supervised and unsupervised application is described in Section 2.1.1, 2.1.2, 2.1.3, 2.1.4 and 2.1.5. All methods calculate the score for individual features and then choose the features which are greater than the pre-defined threshold.

2.1.1 Information Gain

According to the theory in [4], information gain of a term is the measurement of the number of bits of information obtained for category prediction by the presence or absence of the term in a document. The simplest definition for information gain based on the pioneering work by Claude Shannon information theory is that "the value of information content" of messages. Assume P_i be the global probability of class i, and $P_i(w)$ be the probability of class i, given that the document contains the word w. Let F(w) be the fraction of the documents containing the word w. The information gain measures I(w) for a given word w is defined in Equation (2.1).

$$I(w) = -\sum_{i=1}^{k} P_i \cdot \log P_i + F(w) \cdot \sum_{i=1}^{k} p_i(w) \cdot \log\left(p_i(w)\right) + \left(1 - F(w)\right) \cdot \sum_{i=1}^{k} \left(1 - p_i(w)\right) \cdot \log\left(1 - p_i(w)\right) \quad (2.1)$$

The greater the value of the information gain I(w), the greater the discriminatory power of the word w. For a document corpus containing N documents and D words, the complexity of the information gain computation is O(N·D·k) [5].

2.1.2 Entropy

Entropy-based ranking method removes the feature according to the measurement of entropy reduction scheme. And, entropy is the most fundamental quantity in information theory which can be used to measure the amount of uncertainty of an unknown or random quantity. The entropy for a random variable X is defined in Equation (2.2):

$$H(X) = -\sum_{\text{all } x} p(x) \log_2 p(x) \quad (2.2)$$

where p(x) is the probability of each of these values occurring and used $\log_2$ because entropy is measured in bits for "presence or absence".

On the other hand, entropy is always measured relative to a probability distribution p(x), and therefore, it is not possible to consider the "true" probability of an event for many situations. Although entropy is useful for the measurement of uncertainty in a single variable, it does not provide how much uncertainty for one variable given knowledge of another. To overcome this situation, the "conditional entropy of X given Y" is defined for more than two variable case, in Equation (2.3).

$$H(X|Y) = -\sum_{\text{all } x,y} p(x,y) \log_2 p(x|y) \tag{2.3}$$

where $p(x|y)$ presents the probability of x given y, and the mutual information between two variables can reduce the uncertainty in one variable given another variable. Three different formats for mutual information can be described in Equation (2.4):

$$I(X;Y) = H(X) - H(X \mid Y) = H(Y) - H(Y \mid X) = H(X) + H(Y) - H(X,Y) \tag{2.4}$$

To be noted, the mutual information between two variables is "symmetric" likes I (X; Y) = I (Y; X) and if the random variables X and Y are independent, then their probability is looks like that p (x, y) = p(x) p(y) with the value of zero mutual information. Likely, if one can consider X exactly from Y and vice-versa, the mutual information is equal to the entropy of either of the two variables.

2.1.3 Mutual Information

In the theory of information [6], the mutual information can be defined as the calculation of correlation between the words or features and their corresponding classes. In other words, the pointwise mutual information $M_i(w)$ between the word, w, and the class, i, is defined as the basis of the level of co-occurrence between the class i, and word w. We can be denoted the expected co-occurrence of class i, and word w in term of mutual independence is given by P_i. F(w), and the true co-occurrence is defined by $F(w).P_i(w)$. In practice, the true co-occurrence may be larger or smaller than the expected co-occurrence. The mutual information between these two values can be defined specifically in Equation (2.5).

$$M_i\ (w) = \log\left(\frac{F(w).\,p_i(w)}{F(w).\,P_i}\right) = \log(\frac{p_i\ (w)}{P_i}) \tag{2.5}$$

We can conclude that the word w is positively correlated to the class i when $M_i(w)>o$, otherwise, the word w is negatively correlated to the class i when $M_i(w)<0$. In addition, the overall mutual information between the word w, and different classes and the maximum values of $M_i(w)$ over the different classes are defined as follow in Equations (2.6) and (2.7):

$$M_{avg}(w) = \sum_{i=1}^{k} P_i\ .M_i\ (w) \tag{2.6}$$

$$M_{max}(w) = \max_i \{M_i\ (w)\} \tag{2.7}$$

Either of these measures can be used for determining the relevance of the word w and the second one is very common use for the case of determination of high levels of positive correlation of the word w with any of the classes.

2.1.4 Gain Ratio

The extension of information gain is known as gain ratio which attempts to overcome the problem of biasing occurs in the case of information gain because it prefers to select the attributes having many values. In the process of gain ratio, it applies a normalization procedure to information gain using a "split information" value defined in Equation (2.8).

$$\text{SplitInfo}_A\ (D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \tag{2.8}$$

It is not the same with information gain, which measures the information with respect to classification that is acquired based on the same partitioning. It is defined in Equation (2.9).

$$\text{GainRatio}(A) = \frac{\text{Gain}(A)}{\text{SplitInfo}_A\ (D)} \tag{2.9}$$

If the split information approaches zero, the ratio becomes unstable and so the constraint value is added to overcome it, whereby the information gain of the selected must be as great as the average gain over all tests examined.

2.1.5 Gini Index

Gini-index is proposed for classification and regression tree (CART) in [7] for quantifying the discrimination level as a feature and let p_i (w) is the conditional probability that a document belongs to class i that contains the word w, and then its mathematical calculation is defined in Equation (2.10).

$$\sum_{i=1}^{k} p_i(w) = 1 \tag{2.10}$$

Then, Gini-index for the word w, denoted by G(w) is defined in Equation (2.11).

$$G(w) = \sum_{i=1}^{k} p_i(w)^2 \tag{2.11}$$

The range of Gini-index is always between 1/k and 1. By convention, the higher the Gini-index value, the greater discriminative power of the word w, for instance, when all documents which contain word w belong to a particular class, the value of G(w) is 1. When documents containing word w, are evenly distributed among the k different classes, the value of G(w) is 1/k. Normalization process should be considered for Gini-index in order to reflect the discriminative power of the attributes more accurately. Let P_1, ….., P_k denote the global distributions of the documents in the different labels of class. Then, the normalized probability value $\acute{p}_i(w)$ can be described in Equation (2.12).

$$\acute{p}_i(w) = \frac{p_i(w)/P_i}{\sum_{j=1}^{k} p_j(w)/P_j} \tag{2.12}$$

Then, the calculation of Gini-index is computed in terms of these normalized probability values. The accurate reflection of class-discrimination is to be ensured by using the global probability P_i for the case of biased class distributions in the whole document collection. The complexity of the information gain computation is O (n. d. k) for a document corpus containing n documents, d words, and k classes.

2.2 Text Document Classification

Though text document classification is a similar concept of data mining, it involves finding patterns in text instead of looking for patterns in data because of the characteristics of amorphous text. Figure 2.1 shows the functional diagram of document classification model. In the part of training, three basic stages are included. They are data cleaning and visualization (preprocessing and feature transformation phase), selection of relevance feature (feature engineering phase), training and testing the learning model on standard evaluation mode. In the stage of preprocessing, data transformation, noise removing, and feature extraction process are performed. In the stage of feature engineering, two main principles are considered. They are feature selection and reduction. The irrelevant features are removed in the process of feature selection while compressing the dimensionality of selected features in the feature reduction process. In the stage of building the learning model, the selected features for each class label are used to train the classifier model and evaluate the performance of classifier on the testing dataset using the 10-folds cross validation scheme. In the part of testing, it also follows the same procedure of the training model and generates the class of text document as output.

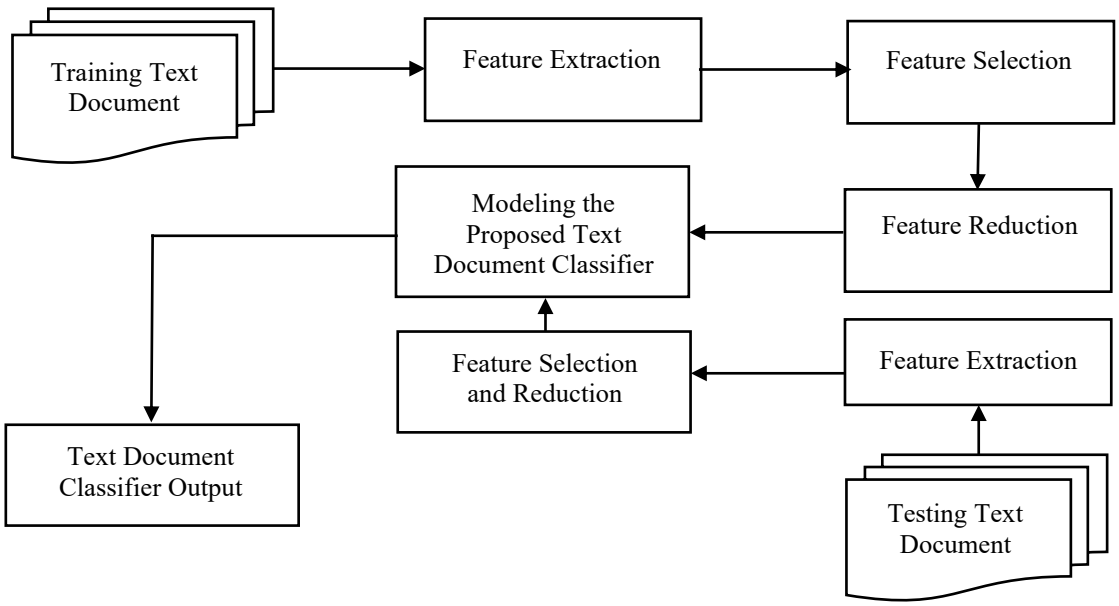


Figure. 2.1. Functional diagram of the document classification model

2.3 Feature Selection Optimization Process

Recently, feature selection for text document classification has become a critical challenge for automatic categorization of digital documents such as news, and blog information in order to simplify datasets by choosing only the relevant underlying features without sacrificing the prediction accuracy. Meanwhile, the roles of feature evaluation and searching process have become main topics for the feature selection. There are many searching approaches, but certain traditional approaches, such as the Best First Search approach (BFS) in which the feature is selected according to the highest score value, are no longer suitable for optimization problems because their way for searching is not suitable for multi-dimensional features. The properties of swarm intelligence such as natural distribution, self-organization, self-learning, simplicity and robustness can be applied to solve this problem. In this section, the work that is related to solving multi-dimensional features in various classification areas is described. It includes the inefficiency of conventional search-based feature selection for previous work, and the benefits of applications for solving complex problems in science and engineering using meta-heuristic intelligence such as flower pollination, elephant search, butterfly search.

In [8], greedy search based on sequential backward selection (SBS) and sequential forward selection (SFS) with the wrapper feature selection approach are proposed in which both are suffered from the issue called nesting effect. This means that the features cannot be chosen later after eliminating these features. In [9], one filter approach called Relief, is proposed to specify a weight to every attribute for presenting the relevance of the attributes to the target concept. However, it does not transact with redundant attributes because of the selection of all relevant attributes regardless of the redundancy among them. In [10], FOCUS filter technique has been proposed which determines all potential attribute subsets, and then select the minimal attribute subsets. However, it was not efficient for computation cost because of its exhaustive search. In addition, many researchers have used various searching approaches for handling different feature selection optimization problems in various areas of applications such as document classification [11] and clustering, pattern recognition, diagnosis of disease using medical data set [12], and several other applications of data mining fields [13].

One of the most popular meta-heuristic algorithms, Artificial Bee Colony (ABC), is used in many different sectors such as training the weight for Artificial Neural Network (ANN); classification of medical patterns; clustering problem to discover the k-best cluster; Travelling Salesman Problem [14], etc. In [15], they proposed the news web page classification system using the Ant Colony Optimization Algorithm (ACO), compared the results with C5.0 and investigated the pros and cons of reducing methods such as WordNet and other preprocessing stages for large numbers of attributes associated with web mining. In [16], they proposed the meta-heuristic based feature selection for sentiment analysis using Genetic algorithm (GA) and rough set theory which is intended to identify sentiment patterns on customers' opinions from websites, documents, discussion forums. In addition, they concluded that meta-heuristic algorithms-based sentiment analysis can provide better optimal subset of feature than traditional one. In [17], they proposed variable-length Particle Swarm Optimization (PSO) based feature selection model to reduce the memory consumption and computational cost that faced in fixed-length PSO. The significant good classification performance with shorter computation time is achieved by avoiding local optima on ten high-dimensional datasets testing.

In [18], the authors proposed the model for feature construction based on Genetic programming by using fuzzy-rough set feature selection. According to the results, the proposed method is more effective than other five methods while evaluating the results on six standard datasets. In [19] , the author employed a novel Chaotic Chicken Swarm Optimization algorithm (CCSO) for optimal feature selection in which logistic and chaotic mapping was applied to assist the local minimum problem in traditional CSO algorithm. The proposed system was compared with four feature selection algorithms: binary chicken swarm algorithm, bat swarm algorithm, particle swarm algorithm, and dragonfly. The results show that the proposed new algorithm achieves better feature selection results using logistic map than tent map, and CCSO achieves better classification accuracy than benchmarks over the different datasets. In [20], they proposed the hybrid meta-heuristic algorithm to optimize weights for multi-layer perceptron (MLP) network for the sentiments of twitter datasets. According to the results, Glowworm Swarm Optimization (GSO) based MLP is outperformed Genetic

algorithm based MLP (GA-MLP) and Biogeography-based Optimization (BBO-MLP) algorithms.

In [21], the authors proposed the Genetic operators for improving PSO's searching ability to solve local the local optima problem. The proposed Crossover-Mutation based PSO (CMPSO) is compared with three recent PSO based feature selection on eight datasets. According to the results, the Genetic operators assist CMPSO is promised better solution than the original PSO. In [22], they proposed PSO for feature selection and weighting (PSO-FSW) in high-dimensional clustering with new validation measure such as fitness function. The overall results comparison of proposed model showed significant improvement in F-score and Silhouette over all representative baselines. In [23], the authors proposed the two stages feature selection model based on PSO for text mining. It used correlation, and information-based measures in the first stage, and both error rate and number of selected features for classification as fitness function for PSO based algorithm in the second one. And the results are compared with four traditional feature selection methods on Reuter-21578 dataset. According to the results, the first stage of proposed system can significantly reduce the original feature set and the second one can further remove features and improve the classification performance.

In [24], wrapper feature selection with Whale optimization algorithm is proposed to find the best feature subset, and the results were compared to PSO and GA using 16 different datasets from UCI data repository. The results demonstrated that the accuracy of the proposed system is above those of the other optimizers. In [25], the author has proposed an improved feature selection algorithm using Ant Colony Optimization (FACO) with support vector classifier for detecting network intrusion. According to the results, FACO is more accurate for classification compared with the multi-objective Ant Colony Optimization algorithm (MOACO) and backup path planning approach (BPPA-ACO) algorithms, and it can reduce the rate of false alarm.

In [26], the author designed Artificial Fish Swarm Optimization algorithm (AFSO) with wrapper approach and support vector machine classifier in order to obtain promising results. The testing was made on nine different datasets from UCI machine learning repository, SRBCT microarray dataset from the database of

Shenzen University, churn from Telecommunication Company, and svmguid3 from LibSVM database, respectively. The accuracy for prediction has been improved from 3.15% to 22.84% higher than other existing algorithms. In [27], the author focused on controlling the big data streaming classification by using swarm search-based feature selection algorithms instead of a traditional one. They described the nature of big data streaming and challenges for the trend of technology innovation to overcome the problems for high dimensionality, memory consumption, and computation time of data streaming classification. In [28], the authors developed a particle swarm optimization learning model to detect the faults for web applications, and they found that it outperforms the term frequency-inverse document frequency (TF-IDF) filter-based classifiers with an average accuracy gain about 11% and 26% for average feature reduction. In addition, this proposed model provided the highest accuracy of 93.35% with the use of decision tree algorithm.

In [29], the author applied Artificial Bee Colony algorithm to determine the set of channels that are useful to discriminate different mental tasks for the development of Brain Computer Interfaces (BCI), while fractal dimension methods were used for feature extraction stage. In addition, the test for proposed methodology was evaluated on the dataset Iva from BCI international competition III. According to the results, the average accuracy for two conditions: rest vs movement, movement vs moment, with only 15 channels that could reduce the feature dimension to 87%. In [30], the binary version of hybrid Particle Swarm Optimization and Grey Wolf Optimization (BGWOPSO) was proposed to find the best feature subset with wrapper method and k-nearest neighbor classifier with Euclidean separation matric. The eighteen standard benchmark datasets from UCI repository were employed for testing, and the results showed that BGWOPSO significantly outperforms the binary GWO, the binary PSO, the binary GA, and the Whale optimization algorithm in terms of accuracy and computation time.

In [31], a hybrid swarm intelligence-based scheme, called Ant Colony and Artificial Bee Colony optimization algorithm (AC-ABC) optimization algorithm is proposed to optimize the feature selection process. According to the experimental results, the proposed method can support the promising classification accuracies by

selecting the optimal features from thirteen UCI (University of California, Irvine) benchmark datasets. In [32], Whale optimization algorithm (WOA) based wrapper feature selection approach is proposed to search the optimal feature subsets for classification process. The proposed system used two variants searching operators: Tournament and Roulette Wheel selection mechanism, and crossover and mutation operators. The results of proposed methods are compared with Particle Swarm Optimization (PSO), Genetic Algorithm (GA), and Ant Lion Optimizer (ALO), and five feature selection methods. The experimental results showed that the efficiency of the proposed approaches for optimal feature subsets searching is better than the other previous works.

In [33], they proposed Evolutionary Computation (EC) algorithms-based feature selection framework intended to overcome the worse classification performance of different emotion which is due to the high dimensionality features of electroencephalogram (EEG) signals. They applied the proposed model on two public datasets (MAHNOB, DEAP) and a new dataset accepted from a mobile EEG sensor. According to the results, the new findings of proposed EC algorithms can provide high performance for the selection of best channels of EEG over a four-quadrant emotion classification problem. It became significant development for the future of EEG-based emotion classification accompany with the popularity of low-cost mobile EEG sensors with fewer electrodes for many new application areas. In [34], a modified Cuckoo search (CS) algorithm is presented to deal with high dimensionality data through feature selection. CS used the rough sets (RS) theory to build the fitness function for selecting the optimal feature subsets which are used for training two classification algorithms: k-nearest neighbors (KNN) and support vector machine to evaluate the performance of proposed CS-based feature selection model on benchmark datasets of UCI repository. Moreover, the significant classification performance result was provided when compared with the six existing algorithms: RS, GARS, PSORS, GRSARS, IHSRS and FARS.

In [35], a novel metaheuristic method based on K-means and Cuckoo search is proposed to find the optimum cluster-heads from the sentimental contents of Twitter dataset. The results of proposed system outperformed PSO, differential

evolution (DE), CS, improved CS, gauss-based CS, and two n-grams methods, with theoretical implications for designing of analytical data model for any social issues in future. In [36], the authors proposed feature selection scheme based on hybrid of PSO with Genetic operators (H-FSPSOTC) to improve the performance of text clustering and reduce computational time by selecting more informative features. In addition, K-means clustering is applied to evaluate the quality of selected feature subset by proposed hybrid algorithm (H- FSPSOTC). Better results were provided when compared with four meta-heuristic optimization algorithms: Harmony search-based feature selection algorithm (FSHSTC), Genetic algorithm (FSGATC), and Particle Swarm Optimization (FSPSOTC). According to the results, H-FSPSOTC is adapted to text feature selection methods and yielded accurate clusters.

In [37], competitive swarm optimizer (CSO) was developed for overcoming a combinatorial optimization problem by adapting feature selection process and reduced the computation cost by eliminating irrelevant learning performance deteriorating features. In addition, six benchmark datasets are used to evaluate the proposed system performance for selected optimal features and better classification performance was achieved with significantly reduced number of selected features than the results of classification performance using canonical PSO and a state-of-the-art PSO variant. In [38], GA-based feature selection is proposed for breast cancer diagnosis medical problem. Elimination of insignificant features is performed by extraction of informative features in the first stage and several data mining techniques are employed to build the knowledge model for breast cancer diagnosis. From the results obtained, GA-based feature selection on 14 types of feature of two different Wisconsin Breast Cancer datasets (WBC DIAGNOSTIC, WBC Original dataset) from UCI machine learning repository, achieved highest classification accuracy (99.48%) for the rotation forest learning model. Results of the proposed system got better accuracy performance than results of previous works.

In [39], the modified Firefly algorithm-based (FA) feature selection system has been proposed that can provide the adaptive balance search between exploitation and exploration process for optimal solution. The idea of FA includes the rhythm, the rate and the duration of flashing form part of the signaling system that

brings two fireflies together. The proposed system used fitness function by incorporating both classification accuracy and feature reduction size and it was tested on eighteen data sets. The results prove advance over other search methods as Genetic algorithm (GA) and Particle Swarm Optimization (PSO) for various evaluation indicators. In [40], the authors proposed Cuckoo search optimization-based feature selection for solving the complex feature problem in lung cancer diagnosis system. CS algorithm is based on the breeding behavior of certain cuckoo species. And, many important features of the nodule of interest are extracted, and the proposed system selected the optimal features for providing better classification performance of lung cancer. Early Lung Cancer Action Program (ELCAP) public database is used for testing the proposed system performance. According to the results, the good total sensitivity and specificity are attained with the values of 98.13% and 98.79% respectively. Moreover, 98.51% for training and testing in a sample of 103 nodules is obtained for 50 exams, and a high receiver operating characteristic (ROC) of 0.9962 has been achieved.

In [41], a new hybrid binary version of Bat and enhanced Particle Swarm Optimization algorithm (HBBEPSO) has been proposed to overcome the problem of feature selection in which the echolocation capacity of Bat algorithm is used for exploring the feature space and enhanced PSO is used to converge the best global solution in the search space. A set of assessment indicators were used for evaluation process on 20 standard UCI datasets. The experimental results were outperformed when comparing to the ones of previous work. In [42], the authors have been proposed Elephant hearding optimization (EHO) for the intelligence human emotion recognition system in which electroencephalography (EEG) signals features are extracted by discrete wavelet transform. EHO included two stages: fine-tune regression parameters of the support vector regression (SVR), and selection of relevant features from extracted all 40 EEG channels. EHO-SVR prove for better accuracy with 98.64% and it is suitable for the prediction of emotion as quantifiable continuous variables rather than classification of emotion into discrete values.

In [43], binary variants of the Butterfly optimization algorithm (BOA) with wrapper approach for feature selection optimization are described. The proposed

two variants of BOA can provide the efficient exploration of optimal feature subset by maximizing the classification while minimizing the size of selected features. In addition, a various assessment indicator is utilized to compare with five state-of-the-art approaches and four latest high performing algorithms on 21 UCI datasets. The experimental results provided the better classification accuracy than other wrapper-based algorithms. In [44], the authors have been proposed an improved Flower Pollination algorithm (FPA) with AdaBoost algorithm for solving the problem of great bulk of feature space for text document classification. The testing for proposed FPA was conducted on Reuters-21578, WEBKB and CADE 12 datasets. And the experimental results showed that higher detection accuracy is achieved when comparing to KNN-K-Means, NB-K-Means and other learning models.

In this thesis, eleven meta-heuristic search policies are applied like Ant Colony algorithm, Artificial Bee Colony algorithm, Bat algorithm, Cuckoo algorithm, Evolutionary algorithm, Elephant algorithm, Firefly algorithm, Flower Pollination algorithm, Genetic algorithm, Rhinoceros algorithm, and Wolf algorithm, in order to optimize the high-dimensional feature selection process. It can support the selection of global optimal feature subsets for classification model. Then, their performance is compared with conventional search policies such as Best First Search, Greedy Stepwise search, and Ranker.

2.4 Traditional Search: Best First Search (BFS), Greedy Stepwise (GS), Ranker

"Search method" attempts to look forward the suitable candidate feature subset from the hypothesis of search space, for example, exhaustive search that finds the appropriate feature by considering the possible combination of different features. In traditional search approaches, most of them use nature of local search and therefore, final solution are always depending on the initial starting points. In addition, it tends to be problem-specific because it uses some information such as derivatives about the local objective landscape. Furthermore, it cannot solve nonlinear and multimodal problems, and it always struggle to cope with discontinuity problems when the gradients are needed. Except for hill-climbing with random restart, most of traditional algorithms are deterministic algorithms. And so, the final solutions will be identical if initial points are started the same. Since the randomization scheme is not used, the

diversity of the obtained solution can be limited. The fundamental control parameters and their description for traditional search are summarized in Table 2.1. In addition, Figure 2.2 shows the framework of traditional search approach using feature selection process.

Table 2.1 Control parameters for traditional search

Parameters	Functional Description
Direction Control	Guide the searching paths with three options: forward, backward and bi-directional.
Processing Control	Define the search process flow with search termination and size of lookup cache.
Search Index Control	Select the index of the start set for searching with three options: empty index for forward search, full set index for backward search, and range set index for bi-directional search.

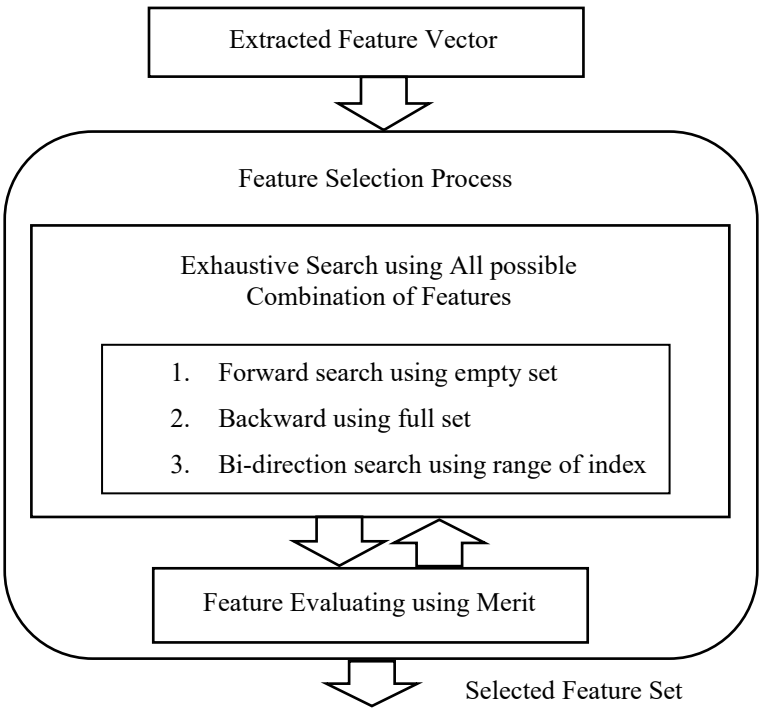


Figure 2.2: Framework of traditional search approach for feature selection process

Traditional search includes two main types of searching approaches: sequential forward search (SFS) which are started search from an empty feature subset; and sequential backward search (SBS) which are started search from a full set of

features. In addition, bi-directional search can be applied which is used the range of index for searching.

Common search methods for attribute selection is Best-First search (BFS) [45] which searches the candidate subset solution of feature through the search space by using the local changes to the current subset of features likes greedy hill-climbing. It may start with the empty set of attributes for forward search or start with the full set of attributes for backward search or start at any point bi-directional search. It moves through the search space by making local changes to the current feature subset. However, it can back-track to a more promising previous subset and continue the search from there if path being explored begins to look less promising. It will explore the feature subset through the entire search space if the enough time is set up for searching process, which is also common to use as a stopping criterion.

Greedy Stepwise search (GS) considers local changes to the current feature subset, which includes addition (forward selection) or deletion (backward elimination) method of a single feature from the subset. It is started with no or all attributes or from an arbitrary point in the space by using the backward and/or forward searching capabilities to control the number of consecutive non-improving nodes, and adding the new items that are deemed relevant or removing the redundant ones and generate the ranked list of attribute by traversing the space from one side to the other. Moreover, stepwise bi-directional search uses both addition and deletion. It stops when the consecutive number of non- improving nodes is found. It encompasses each of these variations like the consideration of all possible local changes to the current subset and then choose the best, or the first change that improves the merit of the current feature subset. When a change is identified, it is never reconsidered for both cases. It can be suffered from the nesting effect because it only considers sub-optimal in which the features that were already selected or deleted cannot be make discarded or re-selected.

In Ranker search [46] that executes a forward selection to sort the list of features according to the value of information and linearly evaluates the candidate feature in terms of ascending order of their sizes and use in conjunction with attribute evaluators such as gain ratio, mutual information, relief, entropy, and so on. It also

happens the overfitting for learning model if the dataset is unbalanced and has high dimensionality.

2.5 Advanced Search Based on Meta-heuristic Intelligence

In searching process, state of equilibrium is very important for seeking the optimum solution in nature. Moreover, all optimum seeking should have achievement of objectives and satisfaction of constraints within the optimum must be found. In general, there are two categories for solving optimization problem: deterministic and non-deterministic (stochastic) algorithms. Deterministic algorithms can provide the same solution in different runs by following more rigorous procedures and repeating the same path every time. Conventional algorithms based on mathematical programming are deterministic algorithms, for instance, linear programming, convex programming, integer programming, quadratic programming, dynamic programming, non-linear programming, and gradient-free methods. It can support accurate solutions for problems in a continuous space. They need the gradient information of the objective function and constraints and an initial point. However, nondeterministic methods exhibit randomness and generate different solutions for different runs. The special ability of exploration for solution is performed on several regions of the search space to avoid local optima. Therefore, it can be employed for handling NP-hard problems that is the problems that have no know solutions in polynomial time.

There are two main types of non-deterministic algorithms: heuristic algorithm and meta-heuristic algorithm. In the former one, discovery of solution is performed by means of trial and error and it cannot provide guarantee for reaching optimal solutions although quality solutions are found within a reasonable time. Local search, divide and conquer, branch-and-bound, cut and plan, and dynamic programming, are examples of heuristic search algorithms. In later one, it can solve more complex problems and it consists of search agents for exploring the feasible solution by taking account of both randomization and some particular rules. And, it includes repeated evaluations of the objective function and heuristic guidelines for estimating the search direction. The inspiration of natural phenomena is used for defining the rules. Evolutionary algorithm, scatter search, guided local search, hill

climbing, iterated local search, and stochastic algorithm are the examples of meta-heuristic algorithms. The group of meta-heuristic algorithm can be classified based on population and neighborhood. Simulated annealing, and tabu search, are the example of neighborhood-based meta-heuristics in which one potential solution is evaluated at one time and the solution moves through a trajectory with nonzero probability in the space of solution that can reach the global optimum. Genetic algorithm and Particle Swarm Optimization are the examples of population based meta-heuristic algorithms. The main components meta-heuristic algorithm is shown in Table 2.2 and the detailed individual components for each MI algorithm will be described in Section 3.2.3.

Table 2.2 Main strategies for meta-heuristic based search

Strategies	Functional Description
Agent	Perform sub tasks according to the work assignment.
Collaboration	Share information among the agents directly or indirectly to obtain intelligence behavior which is used for future decisions of the population.
Exploration	Search the optimal solution globally from the defined search space.
Exploitation	Intensify the search locally from the region selected by exploration stage.
Fitness Function	Perform evaluation functions that are associated with each candidate solution.

2.6 High-Dimensional Feature and MI Categories

Meanwhile, searching the optimal subset of features from high-dimensional feature space is become a hot challenge for NP-hard computational problem in the process of feature selection. In the case of NP-hard problems, it is computationally infeasible to achieve the best solution by brute-force. Meanwhile, the exhaustive brute-force approach for searching feature from the huge hypothesis of feature is become inefficient though sometime the data is not big, but it has many features such as text dataset. Therefore, the meta-heuristic based advanced searching has become a solution to overcome the optimization problem. The important factor for optimization algorithms is the heuristic searching policy, proposed by Glover for solving hard optimization problems, which can provide the capability of searching the better feature subset within reasonable time constraints for a complex feature space by imitating certain strategies taken from nature, social behaviors, physical laws, etc.

In addition, the role of optimization algorithm is to find the optimal solution for specific purpose of applications in any area likes profit, quality and time under various constraints. The range of optimization algorithm are wide from conventional algorithms to modern meta-heuristic algorithms [47], and from deterministic to stochastic respectively. Metaheuristic search scouts the search space in distributed fashion for a current best solution which is refined itself by trying out new solutions from the unexplored search terrains repeatedly. Along the past 20 years, the evolution of various meta-heuristic algorithms for various communities including artificial intelligence, computational intelligence and soft computing, have been witnessed as the intersection of several fields. In addition, researchers recently have invented a collection of heuristic optimization methods inspired by nature of animals and insects, for instance, firefly, cuckoos, with the advantages of efficient computation and easy implementation. Figure 2.3 represents branches of meta-heuristic algorithms.

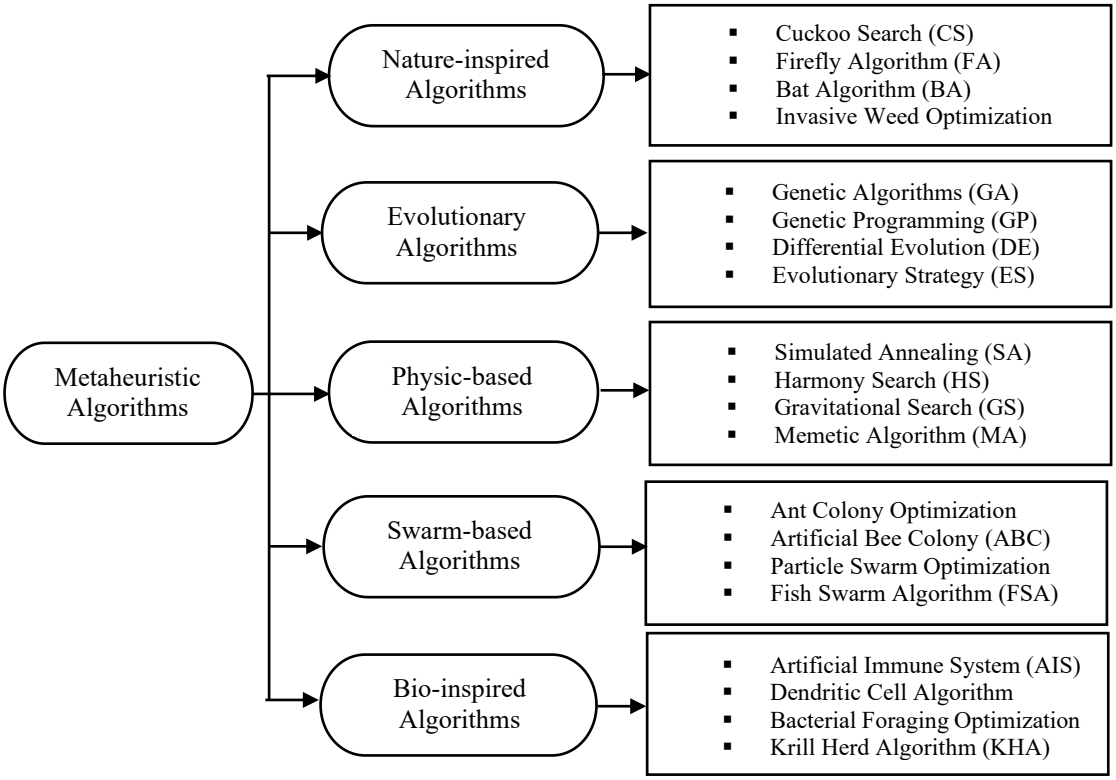


Figure 2.3: Branches of meta-heuristic algorithms

2.7 Meta-heuristic based Feature Searching Process

Figure 2.4 shows the framework of advanced search approach for optimizing feature selection process which includes four general phases for searching

the global optimal feature subset using meta-heuristic search. In the phase of initialization, the parameters are defined such as population size, number of iterations, and so on. In the phase of defining food source, the possible number of solutions is defined such as the hypothesis of feature space. In the phase of defining fitness value, the objective function is defined depending on the problem domains such as classification, clustering, and so on, and characteristic of problem such as single objective or multi-objective optimization. In the phase of defining searching parameters, the rules for searching the optimal solution are defined such as parameters for stop criteria, condition for local to global search, modification rate and so on. Merit function is used to evaluate the selected subset of feature.

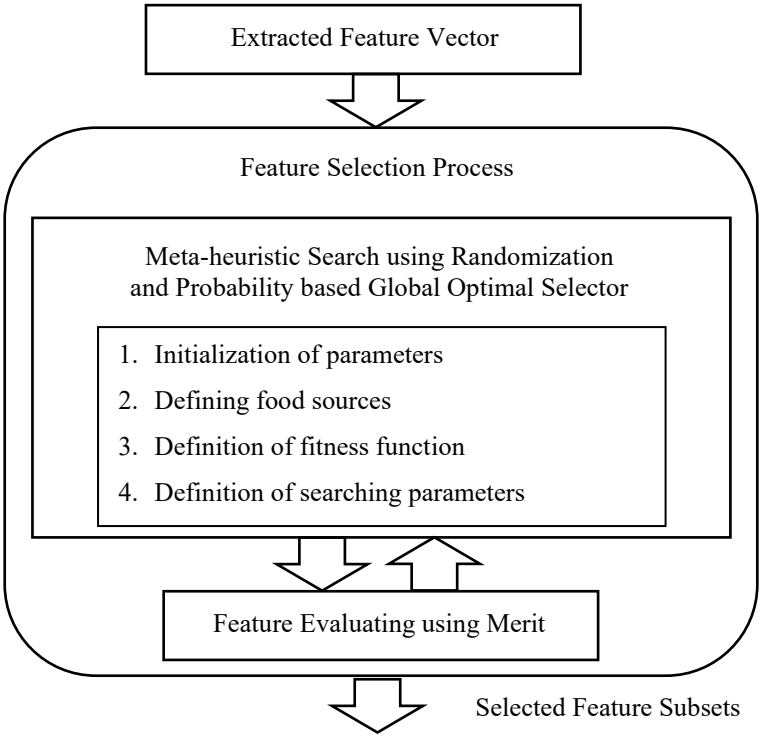


Figure 2.4: Framework of advanced search for optimizing feature selection process

In the meta-heuristic approach, random search mechanism is used in which it starts with the initialization of the random feature subset, scouting the one-fourth or 25% of its neighbors for generating a candidate subset, and then comparing the selected randomization of neighborhood candidate subset with an initialized feature subset. If the value of new subset of neighbor is greater than the old one, update the new candidate solution instead of old one, and this process is repeatedly until the reach

of end criterion such as the maximum number of cycle or 25 % of the search space, and achieved the optimal candidate solutions. In this research, eleven meta-heuristic algorithms, namely, Ant Colony Optimization algorithm (ACO) [48], Artificial Bee Colony algorithm (ABC) [49], Evolutionary algorithm (EA) [50], Flower Pollination Optimization algorithm (FPO) [51], Rhinoceros Optimization algorithm (RO) [52], and Wolf Optimization algorithm (WO) [53], Elephant Optimization algorithm (EO) [54], Cuckoo Optimization algorithm (CO) [55], Genetic algorithm (GA) [56], Firefly Optimization algorithm (FO) [57], Bat Optimization algorithm (BO) are investigated for optimizing the feature selection for document classification problem with single objective function (accuracy).

2.7.1 Ant Colony Optimization Algorithm (ACO)

ACO is motivated by foraging behavior of ants choosing the shortest path and has been popular in the late 80s and in the 90s. In the process of searching food of ACO, the agents of ant are distributed to search the food randomly. Whenever the food is found, the ants mark the way by leaving pheromones on the path when they return to the nest in order to guide other ants in search for food. They follow the path according to the probability proportional to the concentration of the pheromones. The concentration level of pheromone is increased when more ants find the path. The rate of evaporation is increased according to the time and distance of the path which consists of n features, and m ants are placed randomly in n feature nodes at the initial moment. The nodes that individual ant has visited which are recorded in the list of tabu. In addition, concentration of pheromone τ_{ij}, is used to select the next node. The probability for the movement from feature i to feature j during t iteration is shown in Equation (2.13).

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{\tau_{ij}^{\alpha}(t)\,\eta_{ij}^{\beta}(t)}{\sum_{s \notin tabu_k} \tau_{is}^{\alpha}(t)\,\eta_{is}^{\beta}(t)}, & j \notin tabu_k \\ 0 & \text{else} \end{cases} \tag{2.13}$$

where η_{ij} is heuristic information, which is generally $\frac{1}{d_{ij}}$; d_{ij} is the Euclidean distance between two features, and $\tau ij(t)$ is the pheromone concentration from feature i to j for t iterations. In addition, α and β are the information heuristic factor and expectation heuristic factor, which are referred to distribution of weights for heuristic information and pheromone concentration. After completing for traversal of ant, the information concentration for each path is updated by Equation (2.14).

$$\tau_{ij} \leftarrow (1-p)\tau_{ij} + p\sum_{k=1}^{m}\Delta\,\tau_{ij}^{k} \tag{2.14}$$

where p is the weight coefficient with ($0 < p < 1$). $\Delta\,\tau_{ij}^{k}$ is the pheromone increment of the path between feature i and j during the traversal, which is expressed below in Equation (2.15).

$$\tau_{ij}^{k} = \begin{cases} \frac{Q}{L_k}, & (i,j)\ \epsilon \text{ path of } k \\ 0 & \text{else} \end{cases} \tag{2.15}$$

Q is a constant, and L_k is the length of path k.

2.7.2 Artificial Bee Colony Algorithm (ABC)

The computational intelligence of ABC is inspired by the foraging behavior of honeybee. It includes three main categories of foraging task: employed bees which are responsible for exploiting the food and recruiting the others by dancing; onlooker bees choose the food by watching the movement of employed bees' dancing; and scout bees perform the exploration process. Employed bee becomes a scout bee in case of food source exhaustion. The population of food source is generated randomly using Equation (2.16).

$$x_{ij} = x_j^{min} + \text{rand}\,(0,1)\left(x_j^{max} - x_j^{min}\right) \tag{2.16}$$

where i = 1…SN, j = 1…D, SN is the number of food sources, D is the number of design parameters, x_j^{min} and x_j^{max} are lower and upper boundary of j^{th} dimension, correspondingly.

Employed bees performing local search for neighborhood sources, is known as exploitation, using Equation (2.17).

$$v_{ij} = x_{ij} + \varphi_{ij}(x_{ij} - x_{kj}) \tag{2.17}$$

where i is the current solution, k is a neighbor solution chosen randomly, and φ_{ij} is a real random number of uniform distributions in the range [-1,1]. Then, a Greedy selection is applied between the current and its mutant solutions to select the better one which is kept in the population. In addition, the combination of local search and Greedy selection are applied for every food source in the population. The onlooker bees search the better solution among the neighborhood of the food sources and is selected stochastically depending on their fitness values using the Equation (2.18). One of the selection schemes such as roulette wheel, tournament, stochastic university sampling, ranking based or others, is employed to select a better solution after calculating the value of individual food source probability.

$$p_i = \frac{\text{fitness}}{\sum_{i=1}^{SN} \text{fitness}_i} \tag{2.18}$$

If the solution cannot be improved by the local search for both phases of employed bees and onlooker bees, the counter is increased by one. The counter keeps the exploited and retained number of solutions in the population, which is used for determining exploitation sufficiency and exhaustion. The exhausted solution is replaced by the new one produced randomly by Equation (2.16) if the counter exceeds the limit.

2.7.3 Wolf Optimization Algorithm (WO)

Wolf optimization search is the social predators that hunt in packs. It is inspired by the hunting behavior of wolves in which each searching agent hunts for a prey individually, silently and they merge by moving their current positions to their peers' positions if the new terrains are better than the old ones. In the food searching mode, visual range and Lévy flight are used by wolves. In addition, a random hunter behavior such as jump out of wolf's current visual range to a random position upon encounter is considered in order to stay out of a local subspace.

Three rules are presented that govern the idea of the Wolf optimization algorithm. First, a fixed visual area with a radius has defined by v for X as a set of

continuous possible solutions for each wolf. In 2D, the area of a circle by the radius v is the coverage while the distance would be estimated by Minkowski distance in hyper-plane, where multiple attributes dominate, by using Equation (2.19).

$$v \leq d(x_i, x_c) = \left(\sum_{k=1}^{n} |x_{i,k} - x_{c,k}|^{\lambda} \right)^{1/\lambda}, x_c \in X \tag{2.19}$$

where x_i is the current position; x_c are all the potential neighboring positions near x_i and the absolute distance between the two positions must be equal to or less than v; and λ is the order of the hyper space. Second, the fitness of the objective function presents the quality of the wolf's current position. The wolf always tries to move to better terrain but rather than choose the best terrain it opts to move to better terrain that already houses a companion. If there is more than one better position occupied by its peers, the wolf will choose the best terrain inhabited by another wolf from the given options. Otherwise, it will continue to move randomly. Third, the wolf will sense an enemy at some point.

The wolf will then escape to a random position far from the threat and beyond its visual range. For discrete solutions, an enumerated list of the neighboring positions would be approximated. Each wolf can only sense companions who appear within its visual circle and the step distance by which the wolf moves at a time is usually smaller than its visual distance. Furthermore, the potential contributions of Wolf optimization can be applied to finding optimal solutions in different applications such as quadratic assignment problems, travelling salesman problems, job scheduling problems and sequential ordering problems, and so on.

2.7.4 Flower Pollination Optimization Algorithm (FPO)

Flower pollination optimization algorithm (FPO) is inspired by the pollination characteristics of flowering plants and it is one of the population-based metaheuristic algorithms. FPO includes characteristics of biotic and abiotic pollination as well as the co-evolutionary flower constancy between the species of flower and pollinators like insects, birds, bats, other animals or winds. The global search of FPO is carried out by using Equation (2.20).

$$x_i^{t+1} = x_i^t + \gamma L(\lambda)(g_* - x_i^t) \quad (2.20)$$

where γ is a scaling parameter, $L(\lambda)$is the random number vector drawn from a Lévy distribution governed by the exponent λ. Also, g_* is acts as a selection mechanism for the best solution found so far. The current solution x_i^t is replaced by varying step sizes. This is because Lévy flights can have a fraction of large step sizes in addition to many small steps. The local search of pollination and flower constancy is figured out by using Equation (2.21).

$$x_i^{t+1} = x_i^t + U(x_j^t - x_k^t) \quad (2.21)$$

where U is a uniformly distributed random number. And, x_j^t and x_k^t are solutions for pollen from different flower patches. Though switch probability can be used for activating pollination, the equations are linear in terms of solutions x_i^t, x_j^t and x_k^t is preferred for simplicity.

FPO can typically provide not only a higher explorative ability but also strong exploitation ability. There are various applications of FPO for solving many optimization problems such as economic and emission dispatch, solar photovoltaic parameter estimation, and EEG-based identification. Furthermore, FPO can also be used for the purpose of multi-objective optimization.

2.7.5 Rhinoceros Optimization Algorithm (RO)

In RO, it can provide the abstract and simplification of rhinoceros' groups foraging and other natural behaviors. In addition, it also provides its strong ability in dealing with high-dimensional optimization problems with promise computation time. According to natural behavior of rhinoceros, an idea of distributing different gender distinct roles can achieve inclination of global exploration and local intensification. To simplify the working process, three general assumptions are defined according to rhinoceros' behavior. First, all agents are doing Lévy flight in their predefined search range. Second, male agents have bounce mechanism while female not. Third, every agent will die, and reborn mechanism is implemented in each epoch (a possibility of 0.05), and the group leader is abandoned in this assumption in favor of the computing speed.

In the stage of search range definition, test functions are used which can be calculated using Equation (2.22).

$$\text{Search Range} = \frac{\text{upper bound} - \text{lower bound}}{\text{D}} \tag{2.22}$$

where, D is the function's dimension. If the problem is low dimension, the search range will be large that is more search range is considered. In the case of high-dimensional optimization problem, same search range for low dimensional problem will make each agent heavy burdened in global search that leads to the result unstable. This problem can be overcome by defining different dimension D depend on the size of problem.

In the stage of collision bounce mechanism, two male rhinoceros bounced off in opposite direction of each other which is similar a form of elastic collision. The radius of the moving bodies is the range of visual range. The bounce mechanism should be avoided male rhinoceros search range's overlapping to save computation ability and enlarge the whole search space. Euclidean distance is used to calculate the distance between two rhinoceros. In addition, random walk exploration from current location to a new one is one of the activity of rhinoceros to transit information between the male or female rhinoceros. Lévy flights can be used for random walk calculation. It is a class of non-Gaussian random process where random walks are drawn from Lévy stable distribution. Furthermore, Lévy probability distribution that has power tail than normal Gaussian distribution because the probability of returning to previously visited site is smaller, and so benefit when target sites are sparsely and randomly distributed. For male agents, they do randomly Lévy flight while a possibility of 0.9 of female agents do Lévy flight around group leader and a possibility of 0.1 that they do random Lévy flight. The purpose of female rhinoceros' concept is to avoid the group falling into local optimum easily.

In the stage of die and born mechanism, the children will get the information from himself when one male agent die (personal best location) as well as the group leader (the best location). However, children will get information from itself and the male agent with best location when a female agent die. This design aims to accelerate the converge process and reflect the phenomenon of natural selection that is

good agent will have larger chance for transmitting good information or abilities to its descendent.

2.7.6 Evolutionary Algorithm (EA)

In the process of feature selection, searching an optimal feature subset can be provided by heuristic concept of evolutionary search to enhance the classification accuracy for the problem of feature selection from a huge number of attributes (complex feature). Evolutionary Algorithm (EA) is an idea of natural evolution which can be used as a generic optimization technique. EA involves three basic concepts: crossover- generates offspring from parents; mutation- undergoes small changes for individuals; selection- chooses higher likelihood for survival for fitter individuals. The implementation of EA involves many operators such as uniform random initialization, binary tournament selection, single point crossover, bit flip mutation and generational replacement with elitism. In the initialization stage, the population is generated randomly. Then, the parents are picked from population for crossover. There is various type of crossover operators such as single split point. Crossover can create large jumps in the fitness landscape which allows EA to cope much better with multi-modal fitness landscapes and have less likelihood to get stuck in a local extreme. Also, mutation is performed by flipping a single bit from 0 to 1 or the other way around. The likelihood for flipping the selection of a single attribute is 1/m if m is the number of attributes. The purpose of mutation is to happen the small movement in the fitness landscape to climb up towards a close-by extremum. Then, evaluation is performed on current individual population by using the learning model on a cross-validation mode. Finally, the selection of the optimal feature subset is performed iteratively in which various selection operators can be applied such as tournament selector. Then, the loop will end when it meets the stopping criterions like the maximum number of generations or no more improvement till reach a time limit for optimization.

2.7.7 Genetic Algorithm (GA)

Genetic Algorithm (GA) is popular algorithm of MI and it is set up on the resemblance to natural selection. In basically, GA operates with population set of chromosomes, and fitness function (objective function) in order to find the satisfactory

solutions for interested problems. A chromosome is the sequential of gene, parameters of solution. GA searches best solution until satisfactory results are reached, and fitness function includes two operators (crossover and mutation functions) which is used to estimate the critical of the solution in the evaluation step. In crossover operation, jointing of distinct features from subsets pair into a novel subset is performed. In mutation, randomized updating of genes is carried out. Reproduction of chromosomes is taken by finding the fitness value in which bigger fitness value is selected which has higher probability of chromosome by using the Roulette wheel or the tournament calculation. In addition, the substitution of population process is performed by using the elitism or variety replacement strategy to generate a novel population in the new generation, called offspring.

2.7.8 Elephant Optimization Algorithm (EO)

Elephant Optimization Algorithm (EO) is one of the nature-inspired meta-heuristic algorithms and inspired by the habitual features of elephant herds search, male elephant scouts to look forward new habitat while female elephant surround the group leader step by step move towards new habitat according to the responsibility of different gender agents. The gender ratio is used for handling the algorithm's inclination on global exploration or local intensification. EO can be designed using many mechanisms, but not limited to bounce mechanism, death and reborn walk mechanism, and random walk mechanism. However, the feature provides broad applicability depend on the different types of optimization problems. Consequently, time consumption for EO is too much.

In the summary of effective EO search, three unique rules are carried out. First, the updating solution process should be performed during the normal update cycles, but with some major reforms by large cycles represented by the lifetimes of the searching elephants. Second, the intensive local searches at places in the search space that have higher likelihood of yielding the best result are led by some chief female elephants. Third, the male elephants lead the whole elephant clan out of local optima by serving as rangers to venture out to the far space.

EO attempts to bridge the advantages of swimming ability of PSO and evolutionary ability of GA. In other word, EO taps on the dual benefits of swarm

movements and evolution. The solutions of EO are enhanced across the spatial domain by allowing the position of best fitness in swarm and a better generation by the principle of retaining only the fittest in evolution. The spatial enhancement is carried out by the female elephant gang using bi-velocities, local and global. There is a leader female with the best fitness like Firefly algorithm.

In addition, the exploration process is performed by male elephants by taking their individual course without swarming near the female circle. The local search and the global search are influenced by the two velocity vectors which are corresponded to individual female elephants and the whole swarm. The implementations of global and local searches are separated more explicitly in EO likes the males who solely explore and the females who undertake intensive local searches.

Furthermore, other auxiliary functions that need to be assumed, for instance, the expulsion of two male elephants that encounter each other. And, the male elephants should widespread as far apart as possible for achieving maximum exploration coverage. In the case of collide, two male elephants bounced off in opposite direction of each other where their visual ranges set the radius of the moving bodies. They scatter off with unequal masses which are fitness values of the two colliding elephants. The worse fitness bounds off is existed at a higher velocity in a multi-dimensional hyperspace. The best heuristic information is carried forward in the future generation, through some evolutions, by updating the current elephants intermittently with the best-found positions.

EO may not fully converge into a single position because of the separation of exploitation and exploration. However, the operation of EO may run through a certain amount of cycles for promising the best fitness value converges to some maximum threshold. Moreover, the assumption of lifespan for elephant is defined by a probabilistic death rate which is bounded by a negative exponential distribution. The birth rate of new elephant should be the same rate while the elephants vanish randomly according to the probability distribution. The male leader ventures far out to explore for better terrains while the leader of the female elephant herd performs local search in depth. When better solutions are found, the herd migrates towards there. The unique feature of EO is that male elephants will leave the group when they grow to

maturity and reincarnating into male elephant infants of the same herd in future generations.

2.7.9 Firefly Optimization Algorithm (FO)

Firefly algorithm (FO) is one of the nature-inspired meta-heuristic algorithms. It was based especially on the flashing patterns and behavior of fireflies. It includes three basic rules. First, since fireflies are unisex and so, firefly will have attracted each other regardless of their sex. Second, the level of attractiveness is proportional to the brightness. Therefore, for any two flashing fireflies, the less bright of firefly will move towards the brighter one. However, the firefly will move randomly if there is no brighter one. Third, the brightness of a firefly is determined by the landscape of the objective function.

The attractiveness is directly promotional to the light intensity of adjacent fireflies and the variation of attractiveness β for distance *r* can be defined using Equation (2.23).

$$\beta = \beta_0 e^{-\gamma r^2} \tag{2.23}$$

where $\beta_0$is the attractiveness at distance r = 0.

Moreover, the movement of firefly *i* to brighter firefly *j* is defined by using Equation (2.24).

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2} \left(x_j^t - x_i^t\right) + \alpha_t \epsilon_i^t \tag{2.24}$$

where the second term is due to the attraction and the third term is randomization with respect to α_t being the randomization parameter. Also, ϵ_i^t is a vector of random numbers derived from a Gaussian distribution or uniform distribution at time t. If β_0 = 0, it becomes a simple random walk. On the other hand, if γ = 0, it reduces to a variant of particle swarm optimization. Furthermore, the randomization ϵ_i^t can easily be extended to other distributions such as Lévy flights.

In the point of view for application areas, the FO algorithm has attracted much attention to many applications such as digital image compression with least computation time. Meanwhile, it is used for feature selection with consistent and better performance by means of time and optimality.

2.7.10 Cuckoo Optimization Algorithm (CO)

Cuckoo optimization algorithm (CO) is a novel population based stochastic global search meta-heuristic algorithm. It is inspired by the natural of breeding behavior of some cuckoo species that lay their eggs in the nests of host birds. In addition, an individual egg represents a solution, and a cuckoo egg represents a new solution. The purpose of CO is to use new and potentially improved solutions to replace worse solutions in the nests. Three rules of CO can be briefly described. First, each cuckoo lay one egg at a time, and dumps it in a randomly chosen nest. Second, the best nests with high quality of eggs (solutions) will carry over to the next generations. Third, the number of available host nests is fixed, and the egg laid by a cuckoo is discovered by the host bird with a probability $p_a \geq (0,1)$. In this case, the host bird can either get rid of the egg, or simply abandon the nest and build a completely new nest.

As a further estimation, a fraction p_a of the n host nests can be replaced by new nests with new random solution. In a problem of maximization, the fitness function of a solution is directly proportional to the value of the objective function. In the implementation of CO, the simple assumption is used such as each egg in a nest represents a solution, each cuckoo can lay one egg, the objective function is to look forward the new potential better solutions for replacing a not-so-good solution in the nests. However, this basic CO can be extended to more complicated one by modifying some parameters such as the assumption of each nest has multiple eggs representing a set of solutions.

In the calculation of CO, combination of a local random walk and the global explorative one is performed by switching parameter pa in order to have balance search. And, the local random walk can be defined by using Equation (2.25).

$$x_i^{t+1} = x_i^t + \alpha s \otimes H(p_a - \epsilon) \otimes (x_j^t - x_k^t) \quad (2.25)$$

where x_j^t and x_k^t are two different solutions selected randomly by random permutation, H(u) is a Heaviside function, ϵ is a random number drawn from a uniform distribution, and s is the step size. Meanwhile, the global random walk is figured out by applying Lévy flights in Equation (2.26).

$$x_i^{t+1} = x_i^t + \alpha L(s,\lambda) \tag{2.26}$$

where the Lévy flights are calculated randomly by using Equation (2.27).

$$L(s,\lambda) \sim \frac{\lambda\Gamma(\lambda)\sin(\pi\lambda/2)}{\pi} \frac{1}{s^{1+\lambda}}, (s \gg s_0 > 0) \tag{2.27}$$

Here $\alpha > 0$ is the step size scaling factor, which should be related to the scales of the problem of interest. In most cases, $\alpha = O(L/10)$ is used, where L is the characteristic scale of the problem of interest, while in some cases $\alpha = O(L/100)$ can be more effective and avoid flying too far. Obviously, the α value in these two updating equations can be different, α_1 and α_2. However, $\alpha_1 = \alpha_2 = \alpha$ can be used for simplicity.

The benefit of CO is that Lévy flights is used for global search rather than standard random walks. It can provide infinite mean and variance that can help the exploration of more efficient search than standard Gaussian processes. In addition, it can provide global convergence by combing the capability of local and global search.

In the point of view for application, CO has been used in various areas of optimization and computational intelligence with good efficiency, for instance, engineering design applications. Moreover, it can also be used for training spiking neural network model, optimizing semantic web service composition processes, optimizing design for embedded system, selection of optimal machine parameters in milling operation, and generating independent paths for software testing and data generation. In addition, modified CO is used for solving non-linear problem. On the other hand, a discrete CO is used to solve nurse scheduling problems. Furthermore, a variant of CO in combination with the quantum-based approach is used to solve the problems of Knapsack. In the point of view for algorithm analysis, CO search and differential evolution algorithms can provide more robust result than PSO and ABC. In complex phase equilibrium applications, CO search can offer a reliable method for solving thermodynamic calculations while it is used for solving a six-bar double dwell linkage problem and solving distributed generation allocation problem in distribution networks with good convergence rate and performance. As a future enhancement, the multi-objective CO search is used for designing engineering applications such as scheduling problems.

2.7.11 Bat Optimization Algorithm (BO)

Bat optimization algorithm (BO) is one of the bio-inspired algorithms which was developed by Yang in 2010. The idea of BO was based on the echolocation features of microbats where used a frequency-tuning technique to increase the diversity of the solutions in the population and it also employed the automatic zooming in order to balance exploration and exploitation for search process by mimicking the variations of pulse emission rates and loudness of bats when looking forward prey.

Three basic rules are used to develop BO algorithm. First, echolocation is used by all bats to sense distance between food/prey and background barriers in some magical way. Second, bats fly randomly using velocity v_i at position x_i with a frequency f_{min}, varying wavelength λ and loudness A_0 to search for prey. They can automatically adjust the wavelength (or frequency) of their emitted pulses and adjust the rate of pulse emission $r \in [0,1]$, depending on the proximity of their target. Third, the assumption of loudness variation from a large (positive) A_0 to a minimum constant value A_{min} is used although the loudness can vary in many ways. In the simple model of BO, ray tracing is not used though it can form an interesting feature for further extension. Though ray tracing can be computational extensive, it can be a very useful feature for computational geometry and other applications. Furthermore, frequency is always intrinsically linked to a wavelength. Therefore, frequency f or wavelength λ should be changed depending on the ease of implementation and other factors for different applications.

The velocity v_i^t and a location x_i^t, at iteration t, in a d dimensional search or solution space are always important factors for the consideration of each bat. There exists a current best solution x_* for all the bats. Therefore, the following three Equations (2.28), (2.29), and (2.30).

$$f_i = f_{min} + (f_{max} - f_{min})\beta \tag{2.28}$$

$$v_i^t = v_i^{t-1} + (x_i^{t-1} - x_*)f_i \tag{2.29}$$

$$x_i^t = x_i^{t-1} + v_i^t \tag{2.30}$$

where $\beta \in [0,1]$ is a random vector drawn from a uniform distribution.

The implementation of BO can be used depending on the domain size of the problem of interest. Since frequency is assigned randomly for each bat, frequency-

tuning algorithm should be used to support a balanced combination of exploration and exploitation. In addition, the loudness and pulse emission rates can be used for controlling and auto zooming into the region with promising solutions.

Since the loudness decreases when a bat has found prey while the rate of pulse emission increases, the loudness A_i and the rate of pulse emission r_i must vary between A_{min} and A_{max} during the iterations. In the case of $A_{min} = 0$, a bat has just found the prey and temporarily stop emitting any sound and it can be defined using Equation (2.31).

$$v_i^{t+1} = \alpha A_i^t, \qquad r_i^{t+1} = r_i^0 (1 - e^{-\gamma t}) \tag{2.31}$$

where α and γ are constants. For any $0 < \alpha < 1$ and $\gamma > 0$, it can be defined using Equation (2.32).

$$A_i^t \to 0, \qquad r_i^t \to r_i^0, \qquad as\ t \to \infty \tag{2.32}$$

where $\alpha = \gamma$ can be used for simple case. In addition, BO algorithms can be applied in various area of optimization, scheduling, feature selection, classification, datamining, image processing, and others.

CHAPTER 3

SYSTEM DESIGN AND IMPLEMENTATION

This chapter presents the detail description of individual components for the proposed automatic document classification using optimization of feature selection process. However, the practical model implementation process includes multiple steps that are related borderline to many different theories and concepts such as data preprocessing, feature extraction, feature engineering, and learning models and evaluation measurements.

First, the overall architecture of meta-heuristic-based optimization of feature selection for document classification is described and it includes several stages such as tokenization process, normalization process, stop words handling process, stemming and lemmatization process. In addition, feature extraction stage using n-gram TF-IDF and feature engineering is also described.

Second, the various powerful feature selection schemes are presented by using both filter and wrapper feature evaluation with various meta-heuristic based randomized searching capability. Then, MI-based optimization of feature selection process is explained. In addition, three groups of MI-based feature selection system are described which include swarm intelligence-based feature selection system, evolutionary intelligence-based feature selection system, and modern MI-based optimization of feature selection system. Moreover, feature reduction process, and learning models and various evaluation methods are also explored.

Then, system implementation section composes with the description of dataset, library file and parameter setting of experiments for the proposed system. In the last part of this section, the detailed parameter settings for filter and wrapper feature selection approaches, traditional search approaches, and meta-heuristic intelligence search approaches are described.

3.1 System Design

In this section, meta-heuristic based optimization of feature selection for web document classification is depicted in Figure 3.1. Firstly, the text value includes

many feature values such as many words for training phase, and therefore the important words are extracted by removing irrelevant and redundant feature, which is called data preprocessing and feature extraction step. Secondly, feature selection process is performed using filter or wrapper approach such as the correlation-based feature subset selection (CFS) or classifier subset evaluation (CSE) with meta-heuristic based searching policy to look for the global optimal feature subset. Thirdly, the selected features are compressed by using Principal Component Analysis (PCA) to speed up the building of classification process. Finally, J48, Naïve Baye, and support vector machine classifiers are selected for studying the performance of selected subset features that are driven from the feature selection process.

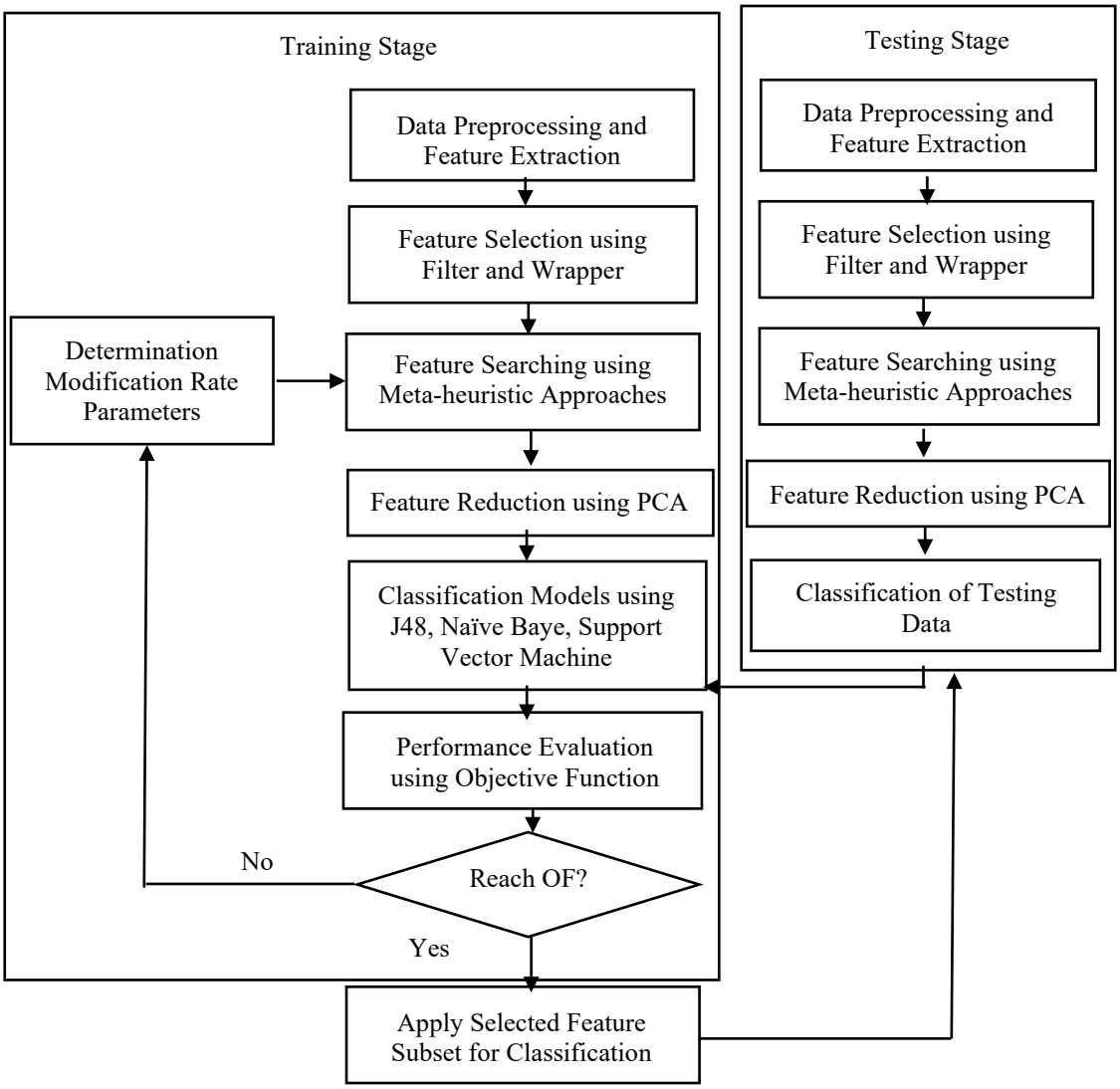


Figure 3.1: Meta-heuristic-based feature selection for document classification

The objective function (OF) for the proposed model is the accuracy of classifier. When the testing data is input, the same processing that applied in training stage are used to select the optimal feature subset. And then, the classification model is applied on testing data to evaluate the performance of classification model on 10-folds cross validation. If the result is not reached the defined objective function, modification rate parameters is determined in order to find the optimal feature subset and this process is performed repeatedly until the maximum number of iterations or achieved objective function.

3.1.1 Data Preprocessing and N-Gram TF-IDF Feature Extraction

Since data comes from heterogenous, and multiple sources, with huge sizes, they are highly susceptible to noisy, missing, and inconsistent data called noise. In addition, the nature of text is unstructured and complex, and therefore, the process of data preprocessing and feature extraction is important for document classification. Figure 3.2 shows feature extraction process for feature vector implementation which contains several stages: word tokenization is used to split the word; stop words handler, stemmer and lemmatizing are used to collect the word of bag; normalization process is applied to give all attributes an equal weight; and term frequency- inverse document frequency (TF-IDF) [58] is performed to calculate the score of text.

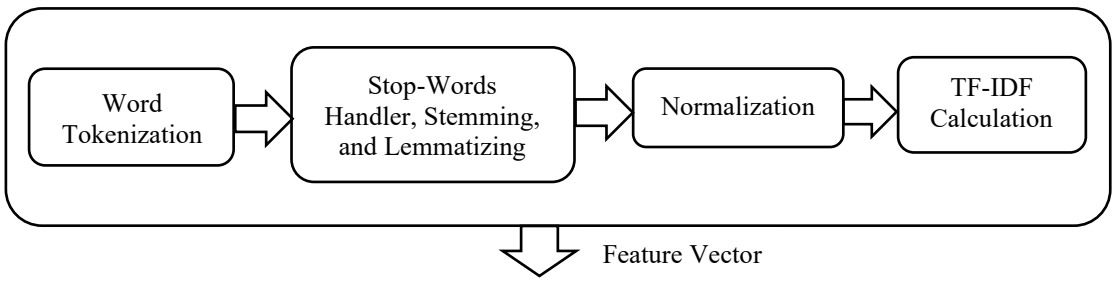


Figure 3.2: Feature extraction process for feature vector implementation

3.1.1.1 Tokenization, Stop-Words Handling, Stemming and Lemmatizing

In the process of tokenization, dividing text strings into lists of substrings is performed by using different schemes such as word tokenizer, n-gram tokenizer, character n-gram tokenizer and alphabetic tokenizer, depends on the nature of dataset and problem. In addition, stop-words handler removes the words that are

commonly appeared in text documents without dependency to an individual topic. Moreover, stemming is performed by removing suffixes and prefixes from index terms before assigning the terms to reduce inflectional forms and derivationally related forms of a word to a common base form. However, lemmatizing is used to identify word form from its root form by considering the part of speech and the context of the word in the sentence.

3.1.1.2 Normalization

Normalization or standardization process is an essential step for classification model in order to avoid dependencies of measurement units. It attempts to give all attributes an equal weight to speed up the learning stage. There are various schemes for normalization with different calculations such as min-max normalization which performs a linear transformation on the original data and it will face an "out of bounds" error if a future input for normalization falls outside of the original data range; z-score normalization or zero-mean normalization which uses the mean or standard deviation of attributes and useful when the minimum and maximum of attribute are unknown or when outliers that dominate the min-max normalization; and normalization by decimal scaling which is performed by removing the decimal point of values of attribute, etc.

3.1.1.3 Feature Extraction: Term Frequency-Inverse Document Frequency

An initial set of raw data is reduced to more manageable groups for processing, is known as feature extraction. TF-IDF is a popular and commonly used a statistical weighting method for retrieving the importance of a term in a document for text mining. TF is the number of occurrences of this term in a specific document, but IDF is a measure of the importance of a term in the whole collection. The term frequency is the number of occurrences of the term t in document d divided by the total number of tokens in the document which can be calculated using Equation. (3.1):

$$\mathrm{tf}_{\mathrm{d,t}} = \frac{\mathrm{n}_{\mathrm{d,t}}}{|\mathrm{d}|} \tag{3.1}$$

where $n_{d,t}$ is number of occurrences of t in d. The inverse document frequency is calculated by using Equation. (3.2):

$$\mathrm{idf_t} = \log \frac{|\mathrm{D}|}{|\{\mathrm{d} : \mathrm{t} \in \mathrm{d}\}|} \tag{3.2}$$

where $|D|$ is the number of categories and $|\{d : t \in d\}|$ if the number of documents with term t occurrences. Finally, TF-IDF can be calculated by multiplying TF with IDF in Equation (3.3):

$$\mathrm{tf-idf_{d,t}} = \mathrm{tf_{d,t}} * \mathrm{idf_t} \tag{3.3}$$

In the calculation of TF-IDF, it considers weighting for individual term based on its inverse document frequency. In other words, if the more documents a term appears in, the less important that term will be, and the weighting will be less. The calculation of TF-IDF for each attribute, a_{ij} ,can be depicted in Equation. (3.4):

$$\mathrm{a_{ij}} = \mathrm{tf_{ij}} * \log\left(\frac{\mathrm{N}}{\mathrm{n_j}}\right) \tag{3.4}$$

where, tf_{ij} refers the term frequency of term j in document i, N represents the total number of documents in the dataset, n_j represents the number of documents that term i appears. We need to apply some smoothing techniques in a small dataset for the case when N equals n_j, then a_{ij} becomes zero by using the Equation. (3.5) which is described as follows:

$$\mathrm{a_{ij}} = \log\left(\mathrm{tf_{ij}} + 1.0\right) * \log\left(\frac{\mathrm{N} + 1.0}{\mathrm{n_j}}\right) \tag{3.5}$$

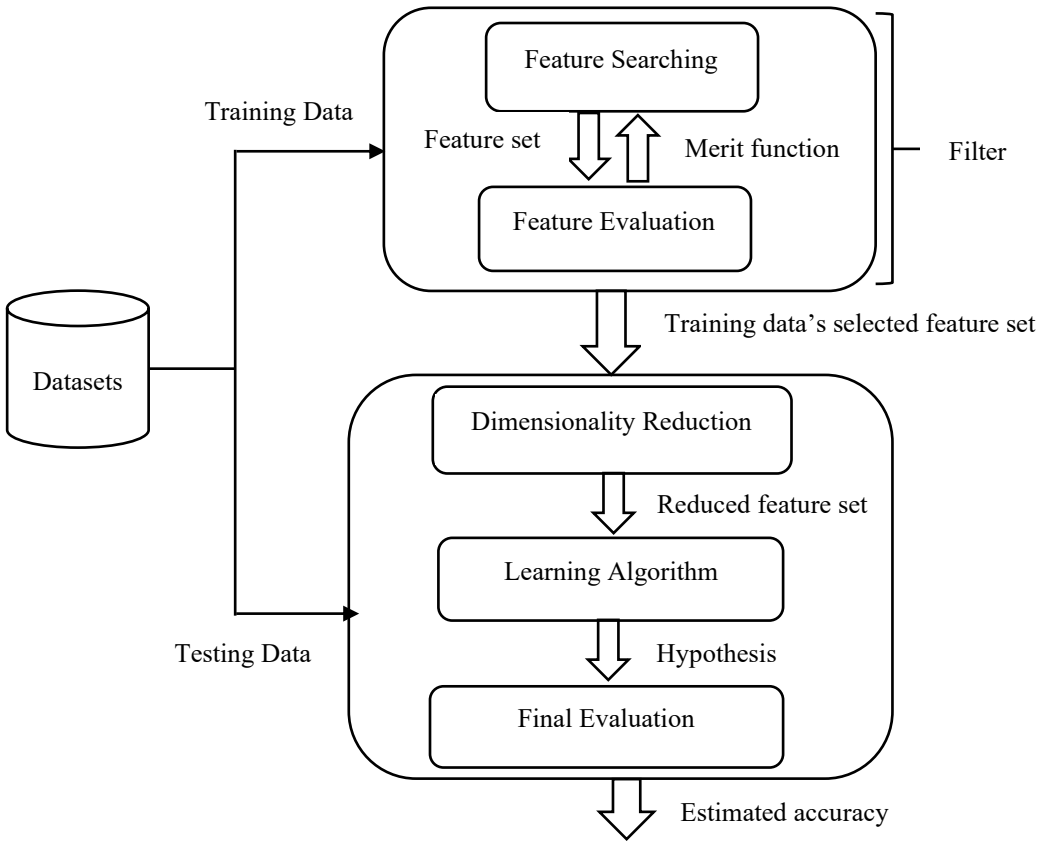
3.1.2 Feature Engineering using Data Mining Framework

A high number of features in a dataset, comparable to or higher than the number of samples, can turn into poor results for the validation datasets due to high complexity for computation. In [59], the main operation for multi-sector application areas in real-world from statistics, data mining, and knowledge discovery to machine learning areas, is to find the population or feature subsets that to be worthy of focused analysis. Both theoretical analyses and experimental studies show that many of the feature selection algorithms scale worse in the domains of large numbers of irrelevant and/or redundant features and consequently the development of additional procedures and methods becomes the important topic in data pre-processing, discovering, or

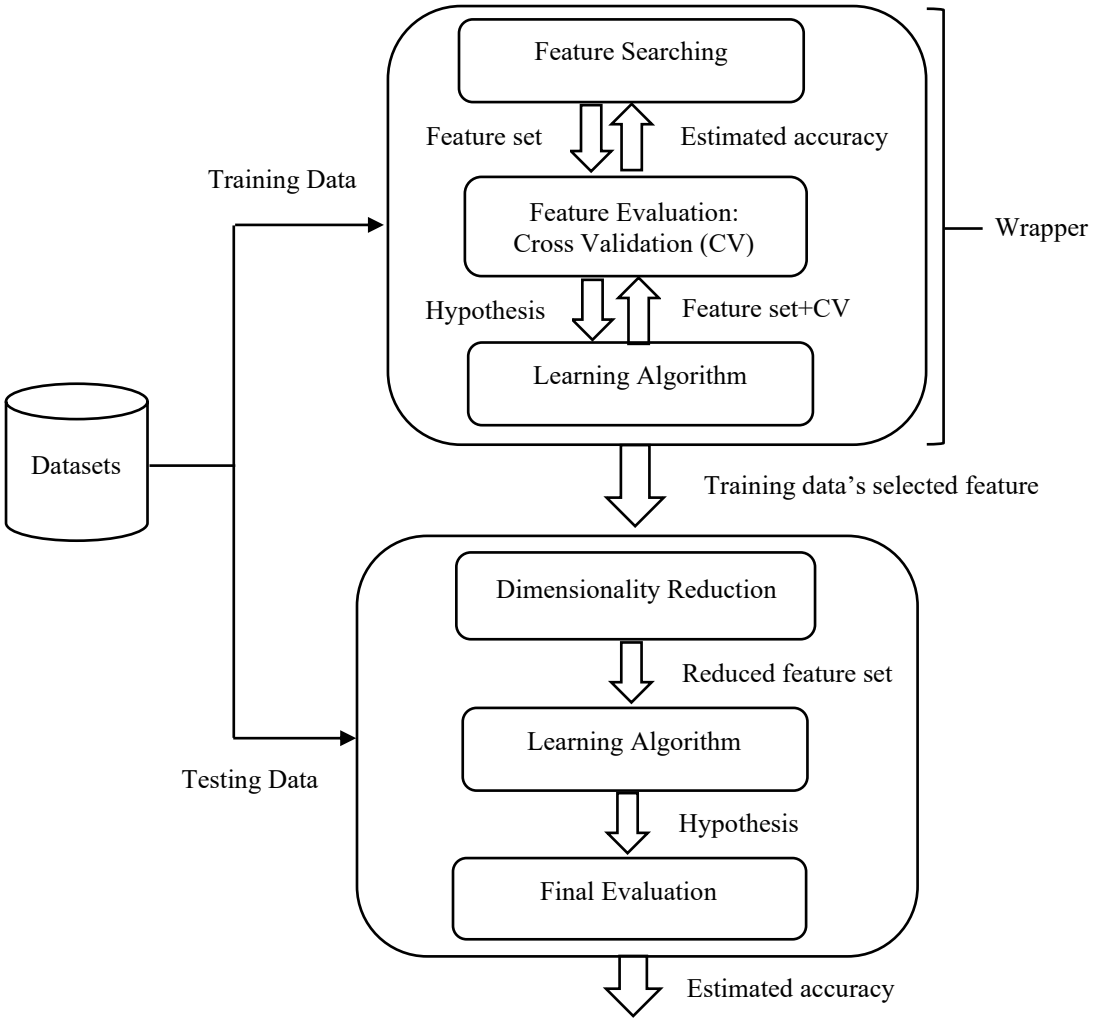
visualizing algorithms. Several stages of feature engineering process for this research will be explained in the following subsections.

3.1.2.1 Feature Selection Scheme

In the process of feature selection, it can be separated into two parts which are feature evaluator- evaluating the relevance of each feature in the candidate subset; and search methods- searching the appropriate features for the next suitable candidate feature subset in the search space. Types of feature selection algorithm can be grouped into three as a general: filter, wrapper, and embedded selector. In Figure 3.3 (a), filter considers the individual feature evaluation process independent from the learning model which includes ranking the lists of features from evaluated score and select the fine ones that above the threshold value. In Figure 3.3 (b), wrapper operation is depended on the type of selected classifier to evaluate feature sets, and it can provide more reliable. In embedded filter process, it selects the feature during the process of learning such as ANN.



(a)



(b)

Figure 3.3: Common feature selection approaches: (a) Filter, (b) Wrapper

3.1.2.2 Filter Approach: Correlation-Based Feature Subset Selector (CFS)

Filter approach selects feature set for any learning algorithm in order to overcome the biasing problem for selecting features. The searching process for feature selection proceeds until a pre-specified number of features is reached or some thresholding criterion is achieved. Filter approach has faster running time than the wrapper approach and so it is suitable for large dataset containing many features. CFS is one of the popular filter algorithms and its hypothesis is built using heuristic approach which considers the features that are highly correlated with predictive label, but uncorrelated with other labels. In [61], the concept of CFS calculation includes feature

evaluation and search for feature subset space using the correlation between features. In feature evaluation process, the merit function is used to formulize the heuristic in which the numerator can be assumed as giving an indication of how predictive of the class a group of features are, and the denominator represents how much of the redundancy among the features. Irrelevant and redundant features are removed according to the statements of heuristic. The mathematical expression of merit function is shown using Equation. (3.6):

$$M_s = \frac{k\bar{r}_{cf}}{\sqrt{k + k(k-1)\bar{r}_{ff}}} \tag{3.6}$$

where, M_s refers to the heuristic "merit" function for feature subset S with k features, $\bar{r}_{ff}$ means the average intercorrelation between features, and $\bar{r}_{cf}$ brings up the mean of feature to class correlation in which f belongs to S. In the feature subset space searching process, the entropy of Y [H(Y)], and correlation between discrete random variables X and Y [H (Y|X)], can be calculated using Equations. (3.7) and (3.8), respectively:

$$H(Y) = -\sum_{y \in Y} p(y)\ \log_2 p(y) \tag{3.7}$$

$$H(Y|X) = -\sum_{x \in X} p(x) \sum_{y \in Y} p(y|x)\ \log_2 p(y|x) \tag{3.8}$$

Information gain (IG) can be calculated using the following mathematical expressions in Equation. (3.9).

$$IG = H(Y) - H(Y|X) \tag{3.9}$$

where H (Y|X) refers to the probability of Y is changed according to the occurrence of X (dependency).

In order to normalize the feature values to the range [0, 1], symmetrical uncertainty can be applied that is depicted in Equation. (3.10).

$$\text{symmetrical uncertainty} = 2.0 \times \frac{\text{gain}}{H(Y) + H(X)} \tag{3.10}$$

Figure 3.4 illustrates correlation-based feature selection process and it is one common filter algorithm that is performed by coupling the ranking process of feature subset in according to the correlation-based evaluation formula with an

acceptable correlation measure and approach of heuristic search. The merit function calculation urges a ranking on feature subsets in the hypothesis of feature subsets search space. The acceptance rules are defined according to the two rules which are irrelevant feature- they should be ignored as their correlation with the interested test is low; and redundant features- they should be screened out because they are highly correlated feature with each other. Moreover, the condition of feature acceptance is always relying on the prediction of class in the scope of instance space, but not considered on other features. In most of the domains, the CFS normally can eliminate the over half of the features from hypotheses of original dataset and therefore execution time for the CFS is faster than wrapper for large scale of datasets.

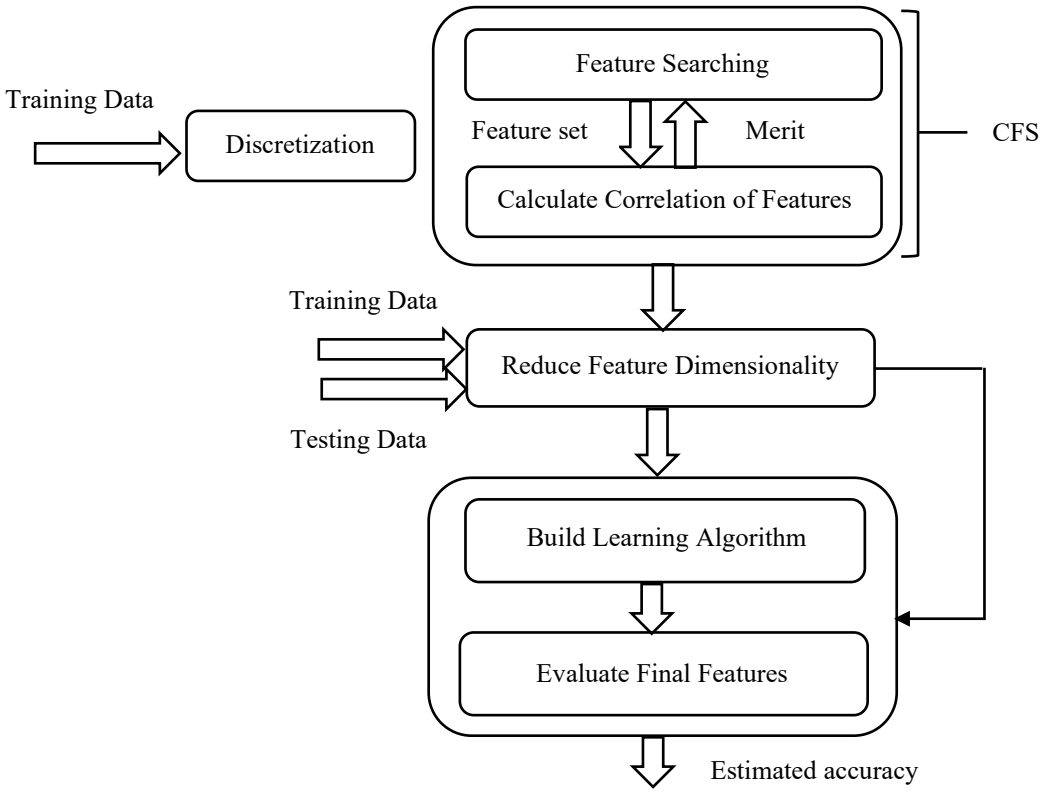


Figure 3.4: Correlation-based feature selection process

3.1.2.3 Wrapper Approach: Classifier Subset Evaluation (CSE)

In wrapper approach, searching through the space of feature subsets is performed depending on the individual learning algorithm to inform the search. The estimated accuracy of the learning algorithm for each feature are added to or removed

from the feature subset. Evaluation for accuracy measurement is performed on cross validation of the training set. When the estimated accuracy of adding any feature is less than the estimated accuracy of the feature set already selected, the wrapper process is ended. The wrapper approach of CSE is generally considered to produce better feature subsets, but it is not cost effective for running time because the learning algorithms is called recursively. In this thesis, CSE is used with MI-based searching policy which is shown in Figure 3.5. In the proposed system, the computational intelligence helped to search the optimal features which can provide the good performance of specific learning algorithms by evaluating the selected features on 10-folds cross validation in the training stage. In testing stage, the selected features from the training stage are applied to estimate the accuracy through the stages of dimensionality reduction, learning algorithm and evaluation of selected feature subsets.

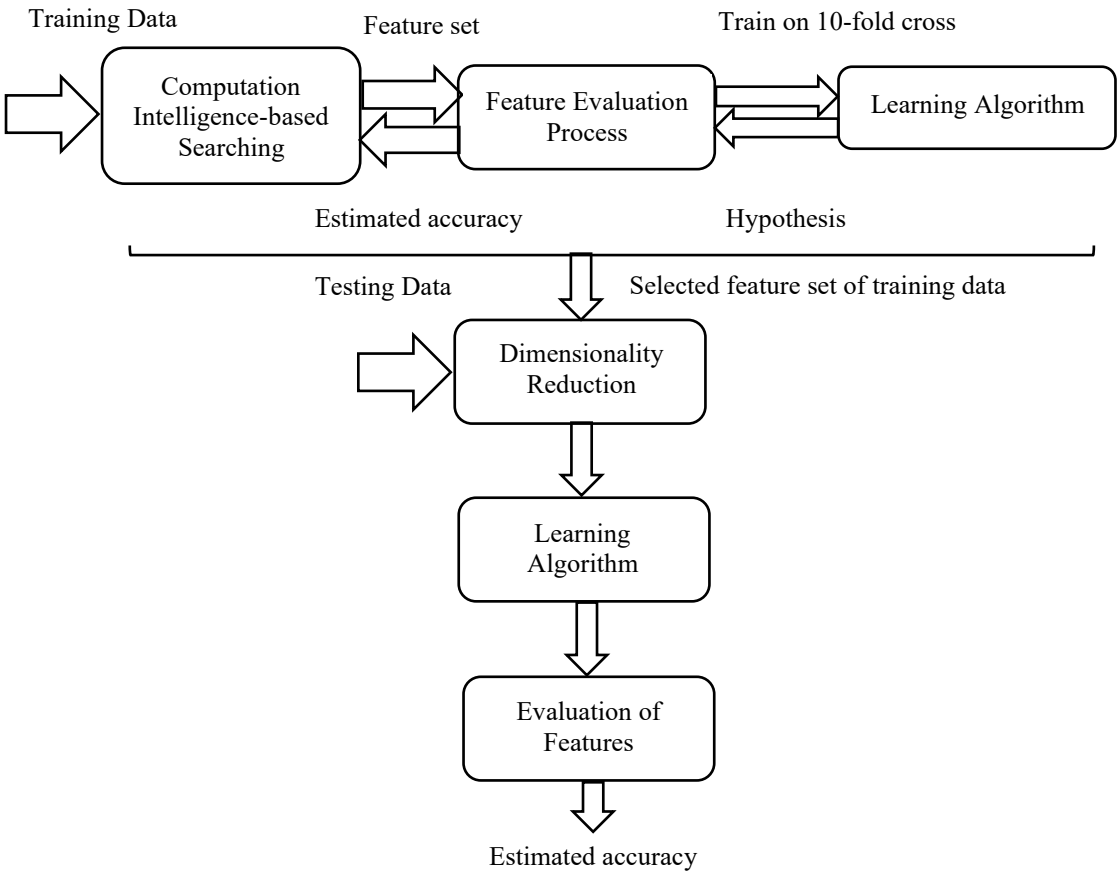


Figure 3.5: Feature selection using CSE with MI-based search

3.1.2.4 Multi-Dimensional Feature and MI-Based Optimization of Feature Selection

Text is high dimensionality feature and selection of distinctive features is essential for text classification. The selection of feature is performed by considering two criteria: feature evaluation and feature searching. Meanwhile, searching the optimal solution has become a hot topic like treasure hunting. In the case of blindfolded search without any guidance, the pure random search is used that is not efficient. On the other hand, searching process is directly climb up to the steepest cliff in order to find the treasure at the highest peak of a known region, for instance, classic hill-climbing approaches. However, most of our search should have been these two extremes like a random walk while looking for some hints. In this type of search, treasure hunting alone can be performed if the whole path is a trajectory-based search such as simulated annealing. In contrast, a group hunting is performed by sharing the information likes swarm intelligence. Though the global optimal solution can be found theoretically for the assumption of unlimited time with any accessible region, the search process will take a very long time. In fact, all modern meta-heuristic algorithms tried to use the optimal solutions by randomized search and replacement of better solution in the place of not-so-good ones while evaluating individual competence (fitness) in combination with the use of system memory. With such a balance, the effort for building better and efficient optimization algorithm is the main objective.

The principle objective of this research is to observe the ability of meta-heuristic based search policies for solving the problem of optimization for feature selection from high dimensionality of feature space. To achieve optimal features for document classification system, the proposed searching model can be worked by synchronizing with the feature evaluation process by checking it according to defined objective function. If the selected subset feature cannot reach to the defined objective function, the modification rate parameters is tuned for metaheuristic-based searching until the maximum number of iterations or objective function is achieved, in order to provide the new subset features for optimal classification.

3.1.2.5 Swarm Intelligence Based Feature Selection System

A challenge for solving optimization problem which includes various decision variables and complicated structured objectives and constraints, has become a hot demand in different fields today. One way to relate with real- world optimization application is by taking inspiration from nature for the development of computational algorithms. Swarm-based algorithms are computational algorithms which are based on nature-inspired algorithms based on population. In the process of swarm-based algorithm, a good enough solution is generated by cooperating among populations of individual potential candidate solutions, and then generating better solutions over generations. In this experiment, swarm intelligence-based optimization of feature selection system is performed, and the detailed process is shown in Figure 3.6.

After cleaning the data and other tasks in preprocessing stage, feature extraction process is performed by n-gram TF-IDF calculation in order to build the feature vector. The extracted features are fed to the feature selection process to reduce the number of dimensions of feature vector. In the proposed system, correlation-based feature subset evaluation process (CFS) is used to guide swarm-based searching with the purpose of exploring the optimal subset in a space of given feature set that has highly correlated with the class are investigated, but uncorrelated with each other. The acceptance rules are defined according to the two aspects of point of view which are features irrelevant - they should be ignored as their correlation with the interested test is low; and features redundant - they should be screened out. The selected feature subset is evaluated in order to measure the performance in terms of classification results and monitor the score value for selected feature subset (FS_{score}) by measuring the performance of the classification process. Then, the output of optimal feature subset is used as a recommended set of features for classification.

$$\text{Fitness Function} = \text{Merit (D)} \tag{3.11}$$

where D is the set of total number of features in the hypothesis.

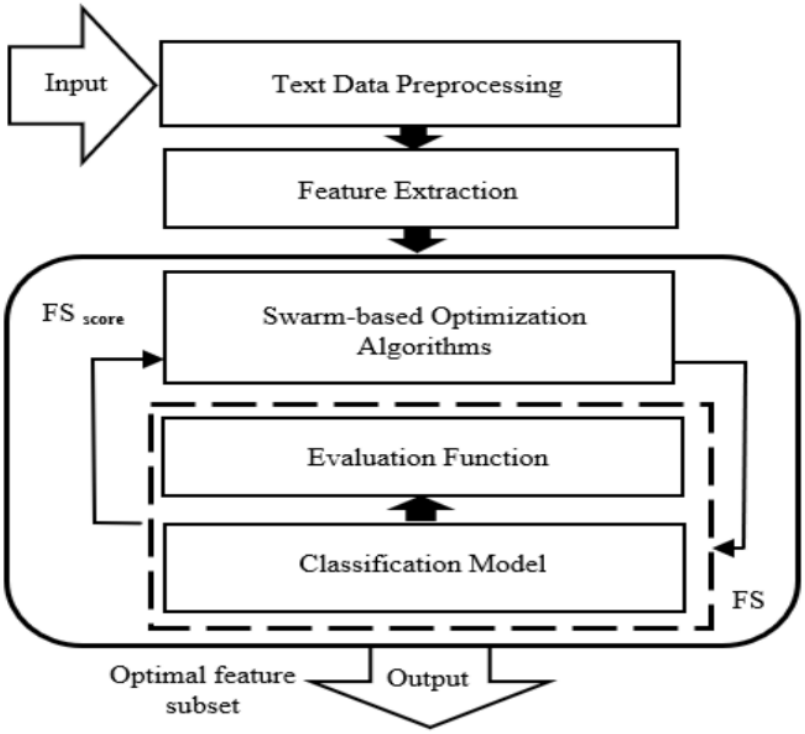


Figure 3.6: Swarm intelligence-based feature selection system

3.1.2.6 Evolutionary Intelligence Based Feature Selection System

In Figure 3.7, searching an optimal feature subset can be provided by heuristic concept of evolutionary intelligence to enhance the classification accuracy for the problem of feature selection from a huge number of attributes (complex feature). In the initialization stage, the population (feature set) is generated randomly, and the parents are picked from population to evaluate the value of individual chromosome (feature). In the evaluation and exploration process, selection of the optimal feature subset is performed iteratively in which various selection operators can be applied. And, single point crossover is performed to achieve new offspring (feature). Then, mutation is performed on new offspring to avoid the process in initialization stage. After finishing evaluation and exploration of global offspring (feature), all selected global offspring (feature) are fed back to evaluation and do the evaluation until reach to objective function (merit in filter or accuracy in wrapper) or maximum iteration.

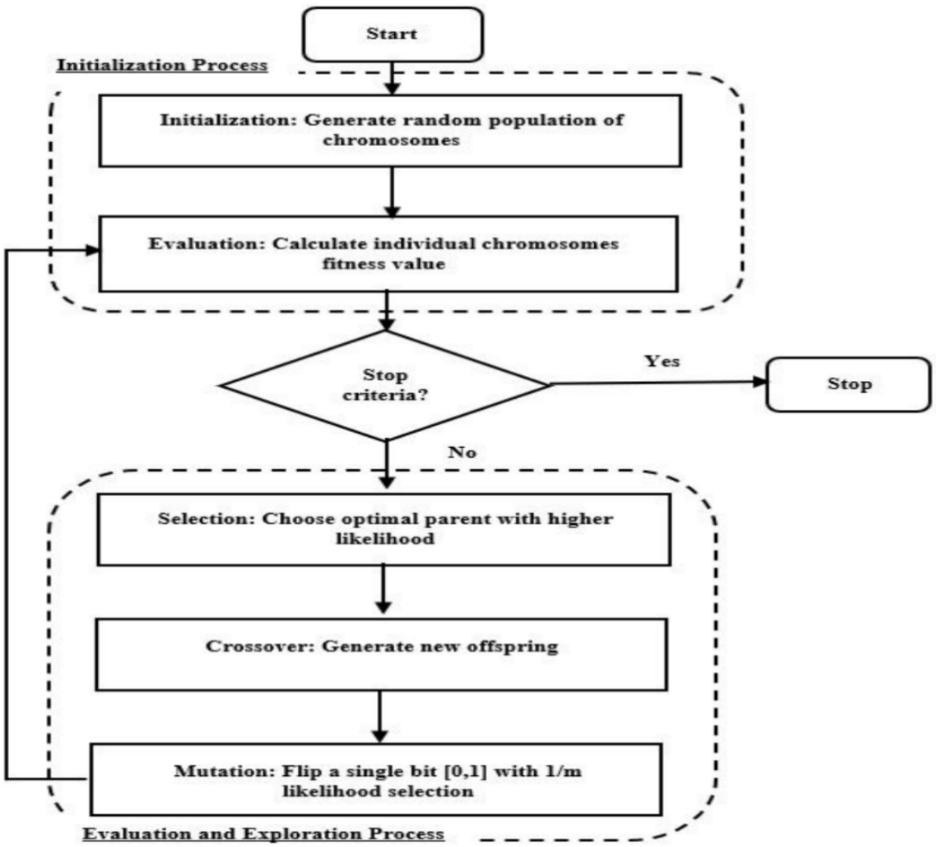


Figure 3.7: Evolutionary-based feature selection system

3.1.2.7 Modern MI-Based Optimization of Feature Selection System

The proposed system is designed by two main stages: elimination of redundant and irrelevant features by looking forward the optimal features from text documents and conduct to the highly effective classifiers. Basically, optimization in second stage is provided guideline towards enhancing the accuracy of classification process given the preselected classifiers. However, the time consumption for individual evaluation in second stage is taken more time than first one. In the initialization stage for modern MI-based feature selection, all common variables are used like swarm-based search except a few value and calculation are different. In addition, the rule for search policy is quite different depend on the nature of individual animals (Section 2.7). In addition, MI-based filter approach used merit function to guide heuristic search in which the selected feature subset from individual search process is evaluated using merit function to measure the quality of selected feature subset which is correlated to individual class highly. Meanwhile, MI-based wrapper approach sends all selected

feature subset to specific classifier-based feature evaluator in order to measure the quality of feature subset. In addition, accuracy is used as objective function in order to provide heuristic search. In the proposed case-study system, number of population and iteration is used as stop criteria for search process and selected all global feature subset within defined iteration and population.

3.1.2.8 Feature Reduction Scheme

In the case of text document classification, dimensionality reduction process has become an important concept to reduce the features from high dimensionality space. Then, Principal Component Analysis (PCA) is a popular algorithm that can create new attributes by using linear combining the original attributes. PCA is looking for a set of new attributes that meets the following criteria: (i) linear combinations of the original attributes, (ii) orthogonal to each other, and (iii) capture the maximum amount of variation in the data. In this thesis, PCA is used which is a linear model for performing randomness extracting the uncorrelated or orthogonal principal components in the high dimensional space in order to reduce the dimension of selected features and the complexity. If applying PCA in classification, the attribute selection for noise removing is applied firstly, and then use PCA for reducing the feature dimension because PCA is not the feature extraction technique and therefore some information will be lost when compression on original data.

Figure 3.8 demonstrates the process of Principal Component Analysis (PCA). It is the process of mapping the original attributes onto a new synthetic form. It is performed by combining the information from subsets of the original ones with statistical properties without changing the property of original data. In the calculation of the PCA, means value is calculated to normalize the features, and the standard deviation between individual variables is figured out to implement the covariance matrix which is used to measure the variance of attributes together. Then, engine value and vector calculation are performed for data transformation and they are non-negative descending ordered eigenvalues. Then, the components are chosen in order to form a feature vector or a column vector. Finally, principal components are formed by multiplying the transformed feature vector with transformed scaled features.

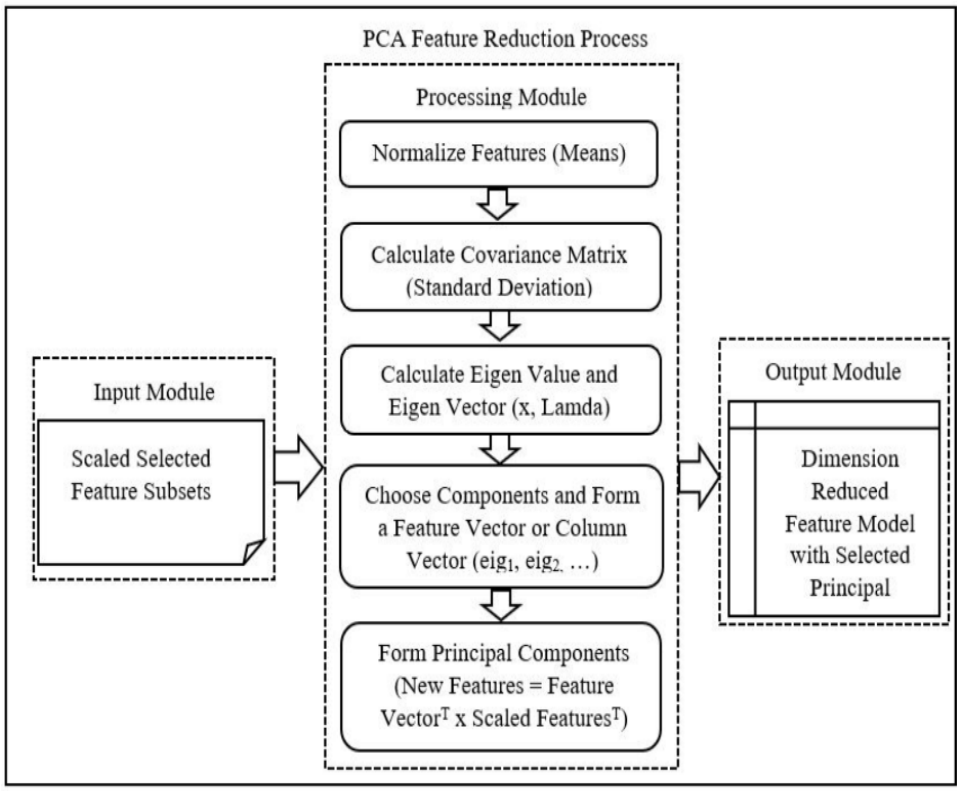


Figure 3.8: Feature reduction using principal component analysis (PCA)

3.1.3 Learning Models and Evaluation Measurements

In this document classification system, various learning algorithms can be applied for categorizing document label. They are Naïve Baye, support vector machine, and decision tree (J48). The detail concepts and mathematical models are explained in Subsections 3.1.3.1, 3.1.3.2, and 3.1.3.3 respectively.

On the other hand, the evaluation scheme for knowledge discovery process (KDP) is very critical to determine the performance of proposed system. Though there are many different types of measurement for evaluating any built model, the selection of correct scheme, depending on the application, is more important. In addition, the performance of the proposed feature selection system is evaluated on confusion matrix. To describe scheme in specific, accuracy [correctly classified instance (CCI) and incorrectly classified instance (ICCI) the quantity of confusion matrix in percentage; mean absolute error (MAE) the average over the verification sample of absolute values differences between forecast and the corresponding observation; root mean-squared error (RMSE) a quadratic scoring rule which measures

the average magnitude of the error; and time consumption the total computing time for classification model (TLM/TCM) in second; root relative squared error (RRSE) the rate of error difference between predicted value and target value, are used to evaluate the performance of the proposed system. In addition, precision (P), recall (R), F_1 score, number of selected feature (SF), number of leaves (NL), and size of the tree (ST) are also used. The calculations for each measurement are shown in Equations (3.12), (3.13), (3.14), (3.15), (3.16), (3.17), (3.18), (3.19) and (3.20) respectively.

$$\text{Accuracy or CCI}(\%) = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{FP} + \text{TN} + \text{FN})} * 100\% \tag{3.12}$$

$$\text{Error rate or ICCI}(\%) = \frac{(\text{FP} + \text{FN})}{(\text{TP} + \text{FP} + \text{TN} + \text{FN})} * 100\% \tag{3.13}$$

$$\text{P} = \frac{\text{TP}}{(\text{TP} + \text{FP})} \tag{3.14}$$

$$\text{R} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.15}$$

$$\text{F1} = \frac{2 * \text{P} * \text{R}}{\text{P} + \text{R}} \tag{3.16}$$

where TP is the number of true positive, TN is the number of true negative, FP is the number of false positive, and FN is the number of false negative.

$$\text{MAE} = \frac{1}{\text{n}} \sum |\text{y} - \hat{\text{y}}| \tag{3.17}$$

where, n is the total number of samples, y is actual output value, and $\hat{y}$ is the predicted output value.

$$\text{RMSE} = \sqrt{\frac{\sum_{\text{i}=1}^{\text{N}} (\text{Predicted}_\text{i} - \text{Actual}_\text{i})^2}{\text{N}}} \tag{3.18}$$

$$\text{TLM or TCM} = \text{Time to build model on full dataset} \tag{3.19}$$

where TLM or TCM is the total computation time for classification model (second).

$$\text{RRSE} = \sqrt{\frac{\sum_{i=1}^{N}(O_i - P_i)^2}{\sum_{i=1}^{N}(O_i - \overline{O}_i)^2}} \tag{3.20}$$

where O_i is the predicted values, P_i is the target values, and $\overline{O}_i$ is the mean of target values.

3.1.3.1 Naïve Baye Algorithm

Naïve Bayes classifier [62] is a simple probabilistic classifier which uses the Bayes theorem in which the strong independent assumptions among the features are selected to build the learning network. The Naïve Bayes (NB) algorithm estimates the individual class prior probabilities by considering the number of class occurrence in the training, then the conditional probabilities of the independent features are calculated and apply the conditional and prior probabilities generated from the training data to make a prediction for the testing data. In addition, it can handle missing values by removing the probability and it is easy to use.

NB has been proven very effective for text categorization in which $d \in D$, where D denotes the training document set and d represented as a bag of words. In addition, individual word $w \in d$ where w is feature words. Each document d is concerned with a class label $c \in C$, where C denotes the class label set. Naive Bayes classifiers estimate the conditional probability P(c|d) which represents the probability of a document d belongs to a class c, that is shown in Equation. (3.21):

$$P(c|d) \; \alpha \; P(c) \; . P(d|c) \tag{3.21}$$

In addition, P(w|c) can be calculated using Laplacian smoothing which is depicted in Equation. (3.22):

$$P(w|c) = 1 + n(w, c) \; |W| + n(c) \tag{3.22}$$

where n(w,c) is the number of the word positions that are occupied by w in all training instance with class c. Finally, |W| is the total number of distinct words in the training set, and n (c) is the number of class.

3.1.3.2 Support Vector Machine Algorithm

Support vector machine [63] is a popular supervised learning algorithm for text classification, and it selects a modest amount of significant limit samples from all classes and constructs a linear discriminant model called the maximum margin hyper plane. The greatest margin hyper plane can provide the most supreme division among the classes. It does not go nearer to any than it ought. To be more precise, the convex hull of a group of points is the most stable enclosing convex polygon. If the systems exceed the restrictions of linear limits, nonlinear function terms such as quadratic, cubic and higher-order decision limits can be used to establish the margin of hyper plane.

The maximum margin hyper plane between individual hyper plane dividing the classes is the one being the furthest away from both convex hulls and vertical bisector of the least distanced line linking the hulls. In the satisfactory mapping selection, the input examples become linearly or approximately linearly divisible in the high-dimensional plane. The optimal hyper plane that has maximizes the distance between the instances of different classes can be computed. In order to get optimal approach, the task is to maximize the distance from the separating boundary to the support vector- the points which are closest to the separating hyperplane. The process of transformation is made by “Kernel Function” which includes Polynomial Learning Machine, Radial-basis function network, and two-layer Perceptron.

3.1.3.3 Decision Tree Algorithm (J48)

A decision tree learning (J48) [64] is the predictive modelling technique for class-labeled training samples. Figure 3.9 illustrates decision tree for training and testing data for news classification which includes two main parts. Classification model is constructed by learning from a training dataset and their associated class labels. This model is used to predict class labels for given data and estimate the accuracy of the learning model in testing part.

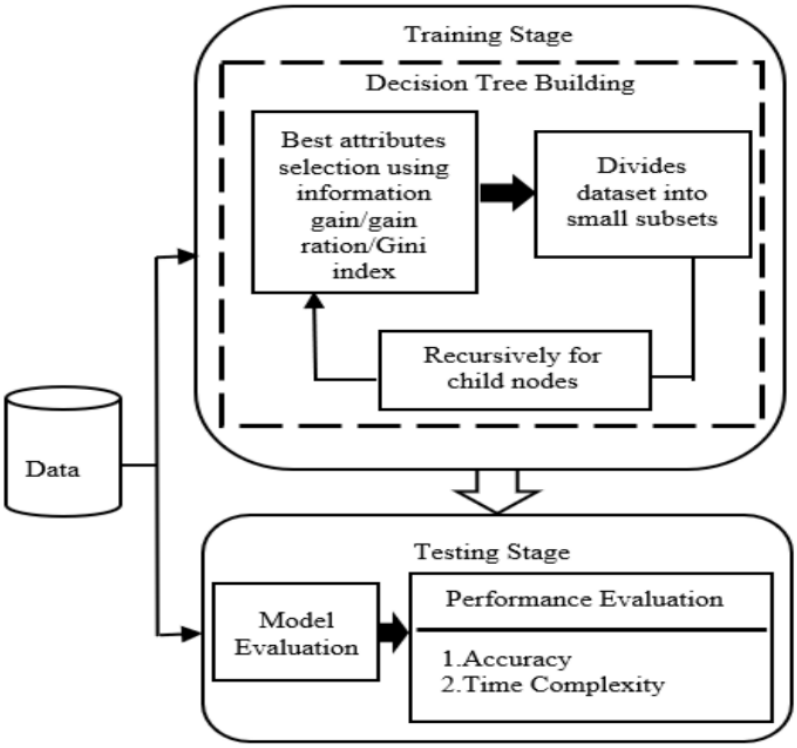


Figure 3.9: Functional diagram for decision tree classifier

In the proposed classification process, J48 decision tree is used because it does not require any domain knowledge and it can handle multidimensional data. In addition, the representation of acquired knowledge is intuitive and easy to assimilate by human. The conceptual decision tree which includes the internal node for testing on individual attribute; branches to represent the outcome of the test; and leaf nodes to hold the class labels. The topmost root node which includes more generalized value and the lowest leaf node which involve more specific value.

It can be used for classification commonly, and others such as clustering by applying the “divide and conquer” technique to split the problem search space into sub space [65], in order to select the best split attributes. In the case of classification process, the attribute values of the tuple for which the associated class label is unknown, are tested against the decision tree by tracing from the root to a leaf node and converted to classification rules. There are different schemes such as entropy, mutual information, information gain, for calculating the quantify information to select the optimal values induction, depending on the version of decision tree.

3.2 System Implementation

The principle objective of this thesis is to observe the ability of meta-heuristic based search policies for solving optimization problem of feature selection in the part of document classification. To achieve optimal features, the proposed searching model is worked by synchronizing with the classification model in order to know the accuracy of classification result which are evaluated on testing data and checked it according to defined objective function is shown in Equation. (3.23). If the selected subset feature cannot reach to the defined objective function, the modification rate parameters is tuned for MI-based searching until the maximum number of iterations.

$$\text{Objective or Fitness Function} = \text{Accuracy or Correctly Classified Instance}(D) \quad (3.23)$$

where D is the set of features. In addition, Machine Learning technique is applied to learn about characteristic of interested data. The common characteristics are derived from news are text-based feature which has the unstructured nature that can happen the multi-dimensional complex features. In addition, we have investigated preprocessing case which includes tokenization process, normalization process, stop words handling process, stemming and lemmatization process, and feature extraction using TF-IDF. Meanwhile, the powerful feature selection scheme is defined by using filter or wrapper with effective meta-heuristic based randomized searching capability to select the optimal features. In addition, the various powerful classifiers are used to build the classification models and evaluated them on testing data using 10-folds cross validation. In the rest of this section, we will describe dataset, system library file and parameters setting for various experiments of the proposed model.

3.2.1 Dataset

In this experiment, BBC news dataset [66] is used as benchmarks for machine learning research. It consists of the top five popular topics: business (510), entertainment (386), politics (417), sport (509), tech (400), containing 2,222 documents in total. The attributes for training model are long strings of text and their corresponding class. The training data type is nominal, which is passed into several stages for preparing feature vectors. In the first stage, data cleaning is performed by the activities of stop word handling, stemming and normalization. In the second one, feature is extracted

using n-gram tokenizer with TF-IDF, and 2,591 features are achieved for building the feature vector.

3.2.2 Library File

The very common data mining and machine learning workbench for building the data mining and machine learning application, which is known as Waikato Environment for Knowledge Analysis, WEKA in short, is described in this section. It is fully developed using Java and distributed under the terms of General Public License, and flexible on every platform like Linux, Windows, Macintosh operating systems, and even on a personal digital assistant. Three main things by the help of WEKA is observed which are analysis of the data nature; prediction of new instance and event using the learning model; evaluation process of different models to choose the best suitable model for specific target application of dataset on hands.

First, WEKA workbench is a suit of the state-of -art of machine learning algorithms and data preprocessing tools, which can provide the whole process of experimental data mining to try out every methods of machine learning schemes on new datasets in easy and flexible ways in which contain preparation of input data, evaluation of learning schemes statistically, and visualization of the input data and the result of learning. In addition, it can provide the various learning algorithms accompany with a large range of preprocessing tools and is accessed through a common interface in order to compare different methods and can describe the most appropriate for the problem on hand.

Second, WEKA can provide the data mining methods for preprocessing, attribute selection, building models, and others, to solve the different real-world problems such as regression, classification, clustering, association rules mining, and; so on. WEKA support their service in different forms to users not only graphical user interface which is easy to use for machine learning beginner to understand the different learning models accompany with complete flow of data mining process but also library files with different packages which is intended for the developer who want to build the specific application completely by collaborating with other useful frameworks and

tools. WEKA can support more than 100 packages related with data mining and machine learning area for different areas of application.

The structure of WEKA is look like the organization of Java Programs which include Classes- a collection of variables along with some methods, Instances- object of the class, and Packages- a directory containing a collection of related classes. The implementation of a learning algorithm is organized by a class and each package is organized in a hierarchy, for instance, the J48 package is a sub-package of the classifiers package, and the classifier package itself is be a sub-package of the overall WEKA package. In WEKA, it provides two indexes for building the own application, which are “Interface Index” and “Class Index” in which the former lists all the interfaces and the latter one lists the all class contained within the package respectively. In addition, the WEKA library file is used for implementation of proposed system. Among the different packages supported by Weka, “Attribute Selection Package” is used for selecting features. “Classification Package” is used to build the classification model.

“Attribute Selection Package” is one of the common used package for building the data mining application, which includes two portions: “Attribute Evaluator”- a Java class for assessment of feature subsets methods can be derived, for instance, CfsSubsetEval- selects the highly correlated subset features with the target class and less correlated feature subset with the other features, ClassifierSubsetEval which is for evaluating the feature subset on training tuples or another hold-out testing tuples using a specified classifier; and “search method” which attempts for searching the suitable candidate feature subset from the hypothesis of search space, for example, exhaustive search- search the appropriate feature by considering the possible combination of different features, heuristic search- search the best feature by refining their selections Iteratively, and so on. The common search methods for attribute selection in WEKA is “Best-First Search”, “Greedy Stepwise search” and rank search.

In the upgraded version of WEKA, it supported the feature of meta-heuristic approach for searching process which is intended for the attribute selection process and it added nine new methods to the WEKA machine workbench recently, to supplement the existing search methods in order to have efficient and effective feature

selection process in data mining models. This new package of “metaphor search” is introduced in 14 September 2016 at Sourceforge which is compatible with the version of Weka 3.7.4 and above, and it added the new meta-heuristic methods which are Wolf Search [67], Harmony Search [68], Flower Search [69], Firefly Search, Elephant Search, Cuckoo Search [70], Bee Search [71], Bat Search [72] and Ant Search [73] respectively. This package is very useful for different areas of application to build different optimal models according to the specific data sets like medical dataset, social media dataset, text dataset, and so on; and targets for optimization such as optimization for accuracy in classification model; optimization of computation time and work management in work flow problem like air ticket reservation, production flow in industry, etc. The development of meta-heuristic package in WEKA can enhance the data mining model performance dramatically. In meta-heuristic approach, “random search” mechanism is used in which it starts with the initialization of the random feature subset, scouting the one-fourth or 25% of its neighbors for generating a candidate subset, and then comparing the selected randomization of neighborhood candidate subset with an initialized feature subset. If the value of new subset of neighbor is greater than the old one, update the new candidate solution instead of old one, and this process is performed repeatedly until the reach of end criterion such as the maximum number of cycle or 25 % of the search space, and got the optimal candidate solution.

3.2.3 Parameters Setting of Experiments for the Proposed System

In this section, the various experimental parameters setting related with feature selection for document classification using meta-heuristic intelligence and traditional approaches are described. It includes the detailed parameters setting and descriptions about filter and wrapper approaches, and traditional and eleven meta-heuristic based search approaches.

3.2.3.1 Parameters Setting: Filter and Wrapper

Table 3.1 describes the common parameters for filter and wrapper feature selection schemes. In the process of feature evaluation for filter, the correlation matrix score is implemented firstly, and merit function use the correlation matrix score in order to consider the feature that are highly correlated with predictive label, but uncorrelated with other labels. In other word, features from the searching process are

sent to the evaluation process of filter in which merit function is used as objective function and individual feature that has a high correlation with the class is predicted and added as selected feature subsets. Meanwhile, the processing unit for wrapper approach uses the accuracy value as an evaluation measurement of selected feature by searching process. If the wrapper approach is used with meta-heuristic based search, the merit objective function is used for our proposed system, but the evaluation of accuracy performance is depended on the classifier that evaluated the selected feature from searching process.

Table 3.1 Filter and wrapper parameters setting

Filter	**Setting**
Evaluation	Correlation matrix scores
Number of threads	1
Wrapper	**Setting**
Evaluation	Accuracy (A)

3.2.3.2 Parameters Setting: Traditional Search Approach

Table 3.2 summarizes the main parameters along with their conceptual values for three traditional searches. In BFS-based search process in filter and wrapper, it is started from empty set with forward direction and look forward the features by expanding the number of nodes until reaching to five consecutive non-improving nodes (CNI: 5). In GS-based feature selection process, backward search approach is used with full set of feature and search process is ended when the addition or deletion of any remaining attributes results in a decrease in evaluation process. Then, a ranked list of attributes is produced by traversing the space from one side to the other and recording the order that attributes are selected. In addition, GS for our proposed model used the parameter for selecting all attributes that matched with the default thresholds value. Meanwhile, RS-based feature selection process is started with empty set and used forward search in order to look forward feature through whole attribute search space and generated ranked features according to the score value of information gain.

Table 3.2 BFS and GS parameters setting

Parameters	BFS	GS	RS
Direction Start	Forward Empty Set	Backward Full Set	Forward Empty Set
Termination	Consecutive non-improving nodes (CNI: 5)	CNI	Whole feature vector

3.2.3.3 Parameter Setting: Meta-heuristic Intelligence Search Approach

This section includes four main summarization parameter tables for meta-heuristic intelligence search according to their nature of search such as evolutionary intelligence in Table 3.3, swarm intelligence in Table 3.4, nature inspired intelligence in Table 3.5, and modern nature inspired intelligence in Table 3.6 respectively.

Table 3.3 shows the parameters setting for evolutionary search in which the default crossover probability (0.5), and mutation probability (0.1) are used as average rate for crossover and mutation process. In the process for feature crossover, two features are selected randomly (initialization operator) and combined one half from parent $feature_A$ (crossover probability = 0.5) and another half from parent $feature_B$ by using single point crossover. In the process of feature mutation, bit inversion is used in which the selected bit rate is inverted with the mutation rate of 0.1. In the process of optimal feature selection, (binary) tournament selection operator is used in which the feature with the best fitness value is selected by comparing the current generated feature with the previous best feature. In order to achieve the updated best feature in new population generation, elitism operator (generational) copies the best feature to new population to compare the old recorded best feature with the new generated feature value.

Table 3.3 Evolutionary intelligence parameters setting

Parameters	Values
Crossover Operator	Single point crossover
Crossover Probability	0.5
Generation	20,…., N
Initialization Operator	Random
Mutation Operator	Bit-flip
Mutation Probability	0.1
Population Size	20,….,N
Replacement Operator	Generational
Selection Operator	Tournament Selection

In Table 3.4, the parameters for driving the searching process based on swarm intelligence are described. It includes two swarm intelligence-based search algorithms that we applied for our case study. In both ACO and ABC search process for global optimal feature subset selection, the merit function is used to guide the searching process for the case of filter feature selection approach. The parameter values for evaporation, heuristic and pheromone are defined according to the equations that described in Section 2.7.1. In the searching process of ACO, the number of population and iteration are defined randomly in the initialization process, for example, in the range of 20 to 200 population and iteration values are used for our case study. To explain clearly for the number of output for selected feature subset (380 features = 20 features or population * 19 iteration search) for 20 value in population and iteration in filter feature selection process, the random 20 features are selected from the feature vector and do the search process until defined maximum iteration (20). In addition, the update process of global optimal feature (shortest path in nature of ant search) is performed by measuring pheromone level. Meanwhile, the heuristic value guide ACO to search the most promising feature (solution). Objective function (merit) is used to provide heuristic value (0.7) for searching the promising feature value among several local best feature for individual class. In addition, evaporation of pheromone is used in order to overcome local optima because it can be used for forgetting for old high pheromone value in local search in order to lead flavoring the exploration search in new areas. In the evaporation value (0.9) is used for this case study and it is about half of pheromone

value for avoiding the convergence to a local optimal feature. In other word, the previous path selected by first ant will continue to be extensively to the following ants if without considering evaporation of pheromone.

In the process of ABC search, three types of bee perform local and global searching together in which the initialization components and searching, selection, and updating rules are also involved to achieve the global optimal feature subset. The selected feature from search process is evaluated by filter or wrapper and they can also guide search process heuristically by feeding back results to search agent. Similar to ACO search, merit function is used to guide the heuristic search in filter and the specific learning algorithm is also applied to evaluate the selected feature subset from search process. In contrast, mutation probability, mutation type, radius mutation and radius damp are used in ABC in which radius damp is defined the chance of crossover in the initial time and bit flip mutation is intended for updating the individual feature value by comparing two features. In addition, the mutation probability (0.01) is used for each bit inversion rate and the discount factor of chance of crossover based on iteration is defined as radius mutation (0.8).

Table 3.4 Swarm intelligence parameters setting

Parameters	**ACO**	**ABC**
Accelerate Type	Normal	Normal
Chaotic Coefficient	4	4
Chaotic Parameter	Normal	Normal
Chaotic Population Type	Normal	Normal
Chaotic Type	Logistic mapping	Logistic mapping
Iteration	20,….,N	20,….,N
Objective Type	Merits	Merits
Population Size	20,….,N	20,….,N
Seed	1	1
Start Set	Empty	Empty
Search Direction	Forward	Forward
Others	Evaporation (0.9) Heuristic (0.7) Pheromone (2)	Mutation Probability (0.01) Mutation Type (Bit-flip) Radius Mutation (0.8) RadiusDamp (0.98)

In Table 3.5, parameters setting for RO, WO, FPO-based nature-inspired intelligence search are summarized. All nature-inspired intelligence search has the common parameters for initialization process. However, the searching rule and selection parameters are different for individual nature-inspired algorithm. In RO-based feature selection, all common parameters are used to generate optimal global feature subset, but rule for searching is different from others (Section 2.7.5). However, WO-based feature selection used other three variables such as absorption, beta min, and escape for supporting the measurement for individual search rules. The absorption parameter is intended to flavor for avoiding local search, and betaMin set the zero-distance attractiveness of the rhinoceros population members. In addition, escape parameter defines the probability from avoiding enemy by escaping to a random position far from the threat and beyond its visual range. Meanwhile, FPO-based feature selection used pollination rate is used in order to measure the local and global feature.

Table 3.5 Nature-inspired intelligence parameters setting

Parameters	**RO**	**WO**	**FPO**
Accelerate Type	Normal	Normal	Normal
Chaotic Coefficient	4	4	4
Chaotic Type	Logistic mapping	Logistic mapping	Logistic mapping
Iteration	20,….,N	20,….,N	20,….,N
Mutation Probability	0.01	0.01	0.01
Mutation Type	bit-flip	bit-flip	bit-flip
Objective Type	Merits	Merits	Merits
Population Size	20,….,N	20,….,N	20,….,N
Others		Absorption (0.001) Beta Minimum (0.33) Escape (0.8)	Pollination (0.33)

Table 3.6 describes the critical parameters setting for four modern nature-inspired approaches which are based on the animal intelligence search such as Firefly, Elephant, Cuckoo, and Bat. Although most of the parameters of are quite similar to the approaches of nature-inspired algorithms that described above, their searching rule is different which are already described in Section 2.7. In BO-based

feature selection, frequency and loudness values of 0.5 are used to search for prey (feature subset). In the feature selection process based on CO search intelligence, the default constant rate ($p_a = 0.25$) and (sigma = 0.69657) is used to define the third rule of CO search (Section 2.7.10). In EO-based search, the common parameters for nature-inspired search algorithms except the rules for searching and selection is different (Section 2.7.8). In addition, the coefficient of absorption and betaMin values of 0.001 and 0.33 are used as parameters for FFO-based search with their corresponding firefly nature-based search policy (Section 2.7.9).

Table 3.6 Prominent parameters setting for modern nature-inspired intelligence

Search Approach	**Parameter**	**Value**
BO	Frequency	0.5
	Loudness	0.5
CO	Constant rate (p_a)	0.25
	Constant rate (sigma)	0.69657
EO	Coefficient of chaotic	4.0
FFO	Coefficient of absorption	0.001
	Coefficient (betaMin)	0.33

CHAPTER 4

RESULTS AND DISCUSSION

This chapter discusses the results of automatic document classification using optimization of feature selection process based on meta-heuristic intelligence (ADC-OFSMI). First, the results of document classification by using various types of meta-heuristic intelligence are explained in Sections 4.1, 4.2, 4.3, 4.4, and 4.5 respectively. In all experiments, various types of evaluation schemes for document classification models such as performance evaluation in terms of accuracy and error rate, and computation complexity in terms of time taken for built classification model and others includes number of leaves (NL), size of tree (ST), are used. In the final section, the summary of discussions for our proposed model testing results are described.

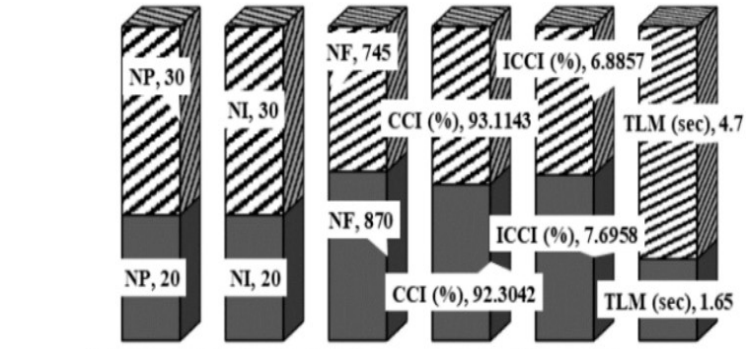
4.1 ADC-OFSMI System Testing using Swarm Intelligence: ACO and ABC

In this experiment, swarm-based searching policy which includes Ant Colony Optimization (ACO) and Artificial Bee Colony (ABC) are discovered to overcome the local optimization problem of high dimensional feature selection for BBC news document classification. Then, the results are compared using various evaluation measurements for performance and computation cost. The detail results are discussed in the following sub-sections.

4.1.1 Experimental Results and Discussion: ADC-OFSABC and ADC-OFSACO

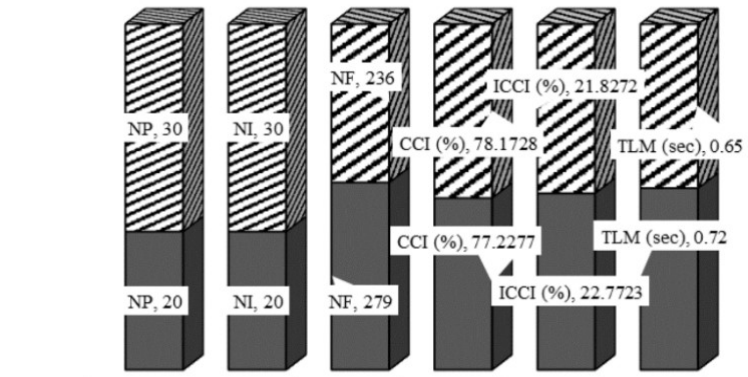
The time series results of the performance and computation cost for ACO and ABC according to the rate of change of factors for population number (NP) and iteration number (NI) are shown in Figure 4.1 and Figure 4.2 respectively. In the experiment for ACO-based feature searching approach, the accuracy (fitness function) for classification has been improved by the reduction in the number of selected features (NF), as NP and NI have increased. However, the computation and hardware costs have become high, while NP and NI have increased. When NP and NI increased from 20 to 30, the CCI increased from 92% to 93%, while NF were reduced from 870 to 745. Meanwhile, TLM has significantly increased from 1.65 to 4.7 seconds when increasing NP and NI. Similarly, the accuracy for ABC increased from about 77 to 78%, while the

number of features was reduced from 279 to 236 whenever we increased NP and NI for running ABC algorithm. However, the computation cost (NF and TLM) was reduced though the performance (accuracy) of ABC-based feature selection approach is lower than ACO one at the same NP and NI parameters setting. To overcome the limitation of proposed model using ACO and ABC, it should be extended to automatic multi-objective optimization model as the future work by considering the adaptation of parameters setting and fitness function in order to have a more benefit results of optimal accuracy with the better computation cost. In addition, we can except better performance result for ABC by increasing NP and NI because of the experimental results for performance is increased from about 77 to 78% according to the increment of NP and NI in Figure 4.2.



	NP	NI	NF	CCI (%)	ICCI (%)	TLM (sec)
Series2	30	30	745	93.1143	6.8857	4.7
Series1	20	20	870	92.3042	7.6958	1.65

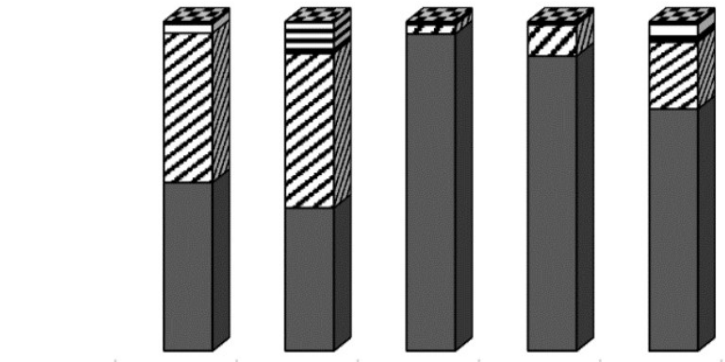
Figure 4.1: ACO-based feature selection for document classification



	NP	NI	NF	CCI (%)	ICCI (%)	TLM (sec)
Series2	30	30	236	78.1728	21.8272	0.65
Series1	20	20	279	77.2277	22.7723	0.72

Figure 4.2: ABC-based feature selection for document classification

Figure 4.3 shows the comparison for traditional search and swarm-based search in terms of four-dimensional measurements: TLM, CCI, ICCI, and NF. According to the results, Greedy Stepwise search takes the shortest time for feature selection, because it only selects the highest score value of features depending on the defined threshold level, with no consideration to the whole of the hypothesis of feature. However, the CCI (accuracy) percentage has not shown a substantial difference between BFS and ACO, but if we increase NI and NP, better accuracy can be provided by BFS according to the characteristics of swarm-based algorithm. Although the accuracy of ABC is not better than BFS for the NP of 20, 30 and NI of 20, 30, the accuracy can be increased according to the probability of time series results when we increase NP and NI. However, expensive hardware will be needed for computation. The ranker search provided worse accuracy because it only considered selection on every feature ordered in a descending score. Although the accuracy for Greedy Stepwise search is good with a reasonable number of selected features and computation time, it cannot provide the adaptative property when unknown testing data is applied to system. However, ACO can provide the better accuracy with the property of adaptative searching according to the NP and NI and other parameters tuning than Greedy Stepwise.



	Best First	Greedy Stepwise	Ranker	Ant Colony	Bee Colony
TLM (sec)	0.26	0.25	0.56	1.65	0.72
ICCI (%)	6.9307	16.8317	24.4374	7.6958	22.7723
CCI (%)	93.0693	83.1683	75.5626	92.3042	77.2277
NF	105	77	2591	870	279

Figure 4.3: Traditional search and swarm-based search: results comparison for performance and computation time with NF

4.2 ADC-OFSMI System Testing using Evolutionary Intelligence: EA and GA

In this experiment, the exploration of MI for knowledge discovery process is carried out using two popular algorithms like EA and GA with two feature selection approach, filter (CFS) and wrapper (CSE), and two classifiers like NB and SMO for the area of text document classification by using four cases: EA + CFS + SMO/NB, EA + CSE + SMO/NB, GA + CFS + SMO/NB, and GA + CSE + SMO/NB. In the case of EA + CFS and EA + CSE, 852 and 127 total number of features are selected respectively. In the case of GA + CFS and GA + CSE, 572 and 70 total number of features are selected correspondingly. According to the selected number of features, MI can reduce size of features dramatically. In addition, their corresponding results comparison using different evaluation schemes are described in the following sections.

4.2.1 Experimental Results for Computation Complexity (Time)

According to the results in Figure 4.4 and Figure 4.5, building the model using NB was faster than using SMO for both EA and GA with CFS filter approach. Meanwhile, the better time consumption is accessed by NB learning model for both EA and GA with CSE wrapper approach. On the other hand, SMO-based classification model for both EA and GA with CFS or CSE needed more consumption time. According to the results, NB learning model is more suitable for this area of experiments that are carried out in this case study.

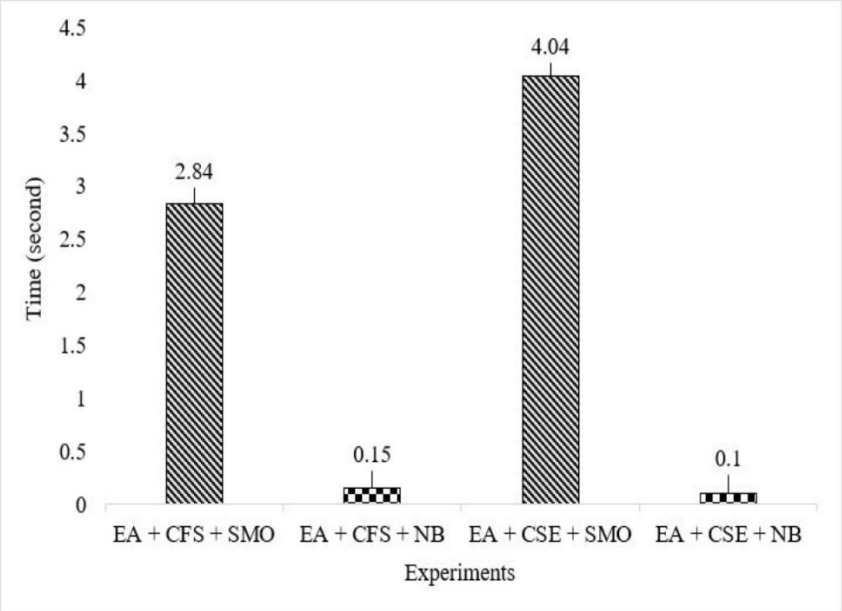


Figure 4.4: Computation complexity comparison using EA

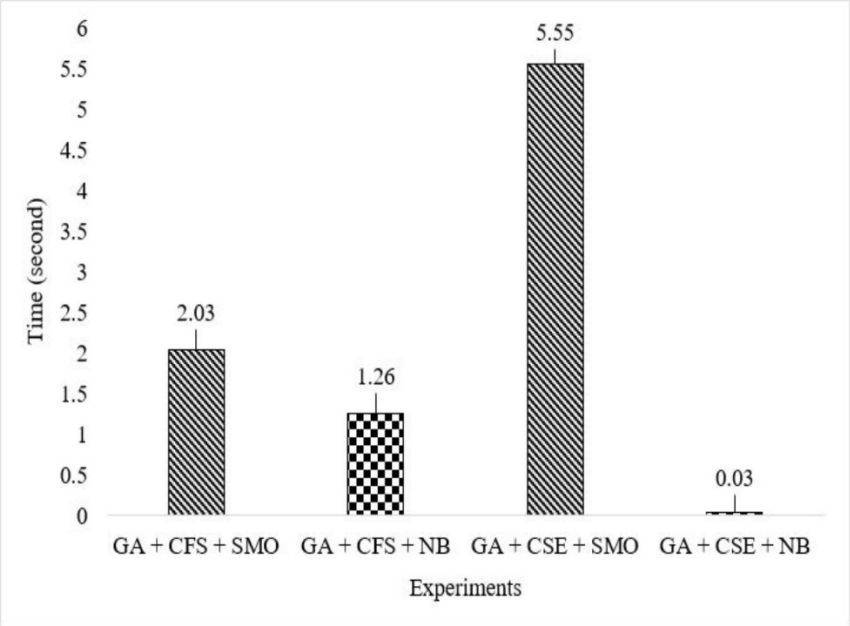


Figure 4.5: Computation complexity comparison using GA

4.2.2 Experimental Results for Error (MAE, RMSE)

In the case of MAE measurement shown in Figure 4.6 and Figure 4.7, EA with CFS shows less error rate for testing case on a 10 folds-cross validation than EA with CSE. This is because the features are selected considering generalization of feature score which are correlated to the individual labels of class, but unrelated with each class in term of the property of filter with evolutionary computation intelligence. Similarly, the results for RMSE using filter with EA is better than wrapper feature selection scheme because the features are selected according to the specific type of classifier and the error rate is increased for the 10- folds cross validation on testing datasets. However, the case for GA and CFS with SMO cannot provide better error rate when compared to CSE with SMO. On the other hand, the error rate can be reduced using GA and CFS with NB than CSE with NB. Similar results were observed in the case of RMSE using GA and CFS with SMO/NB, and GA and CSE with SMO/NB.

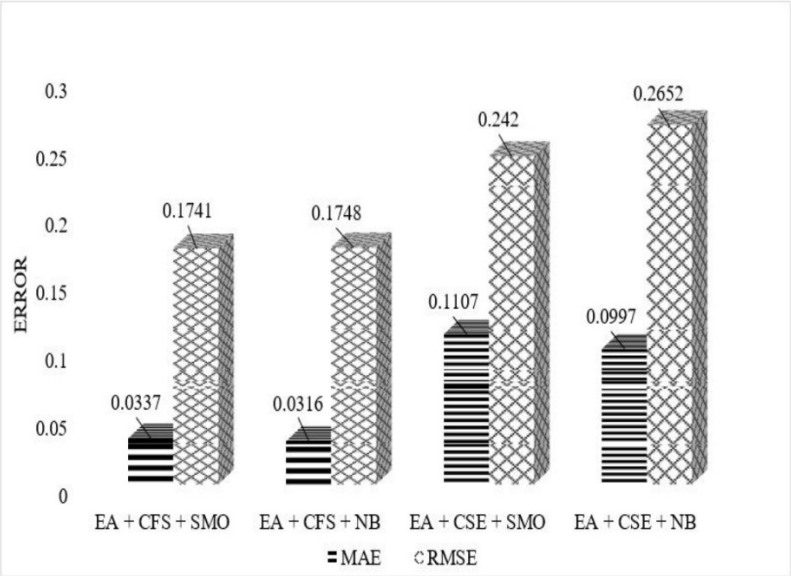


Figure 4.6: Error comparison using EA

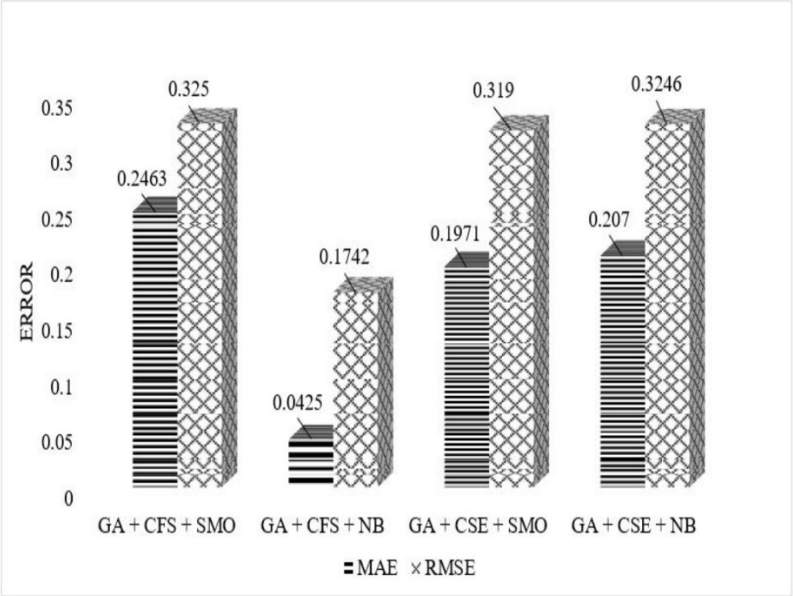


Figure 4.7: Error comparison using GA

4.2.3 Experimental Results for Performance (Accuracy)

In the measurement of evaluation scheme, the feature searching process is carried out by the intelligence of Evolutionary and Genetic computation with filter and wrapper approaches. According to the experimental results in Figure 4.8 and Figure 4.9, accuracy performance using both Evolutionary and Genetic computation with CFS filter was better than CSE wrapper for 10 folds-cross validation on testing model by selecting the optimal feature subset which are distributed over the labels of class for classification learning model without bias for searching feature from high

dimensionality hypothesis. In addition, NB classifier is more suitable for text classification than SMO. On the other hand, the EA showed more efficient accuracy than GA for both CFS and CSE feature selection approaches.

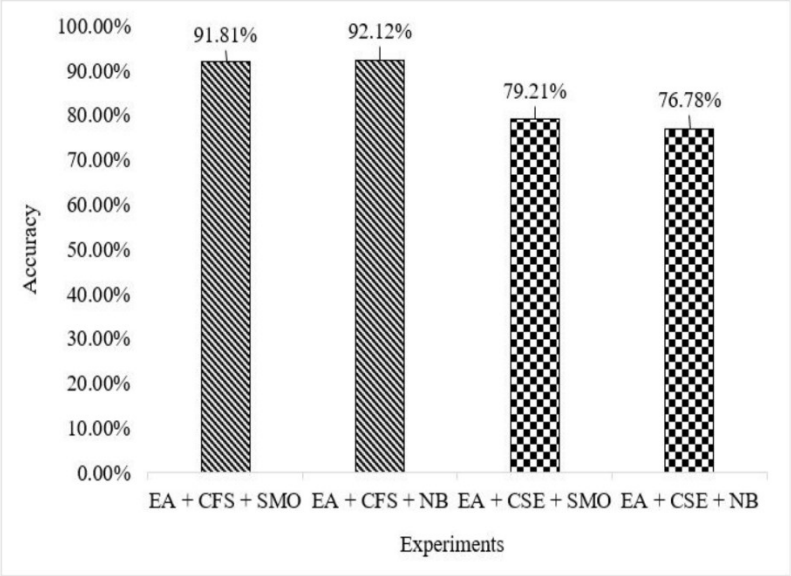


Figure 4.8: Performance comparison using EA

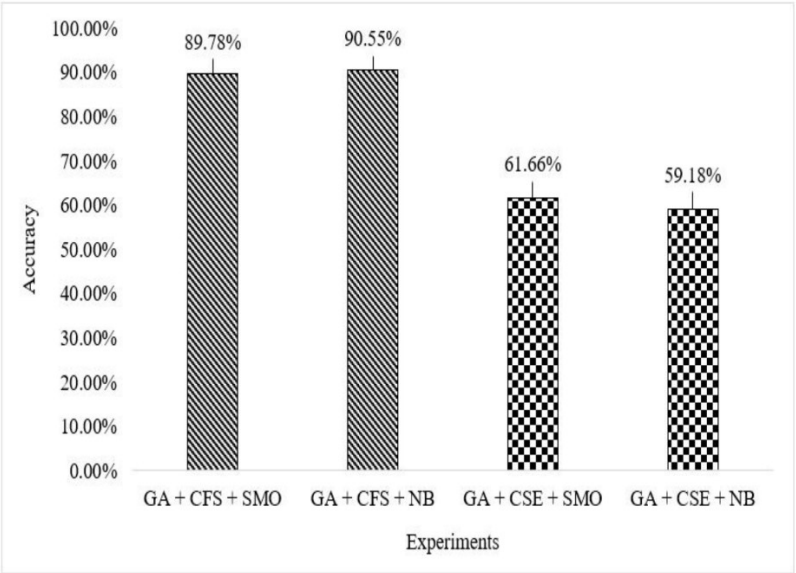


Figure 4.9: Performance comparison using GA

4.3 ADC-OFSMI System Testing using Traditional vs Nature-inspired Intelligence

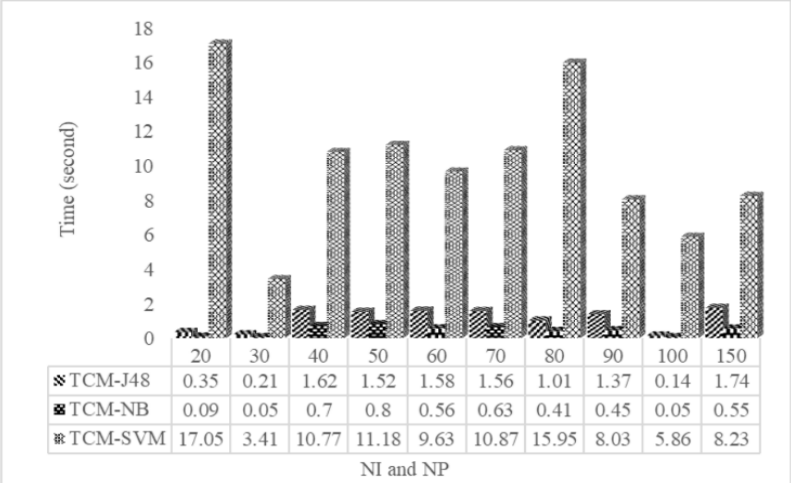
In this experiment, nature-inspired based optimization of multi-dimensional feature selection approach is proposed for document classification and compared the results of performance to conventional search-based feature selection

approach. To be specific description, wolf intelligence is used to solve the research problem is called wolf intelligence-based optimization of multi-dimensional feature selection (WI-OMFS). Wolf algorithm that imitate the way wolves search for food and survive by avoiding their enemies. The performance (accuracy) is used as fitness function. Moreover, various measurements of performance and computation complexity are also used to evaluate the proposed system. The contribution of this experiment is two-folds: 1. the efficacy of the wolf algorithm is verified by testing quantitatively and compared to other conventional algorithms using various evaluation indicators and analysis; 2. wolf algorithm is investigated with respect to the rate of increment for the size of populations and iterations in order to looking forward superior results. In addition, two feature selection schemes like filter and wrapper, and three classification algorithms like Naïve Baye (NB), support vector machine, and J48, are applied. According to the experimental results, WI-OMFS can provide robustness for performance according to the objective function. In addition, the better performance can be achieved using wolf search with filter approach than conventional search. The detail results comparison using different evaluation schemes are described in the following sections.

4.3.1 WI-OMFS Filter Experimental Results

In the case of WI-OMFS filter, the computation cost (TCM) results using three different types of classifiers are shown in Figure. 4.10. According to the NI and NP, TCM values are changed as time series. Among of three classifiers, SVM has taken most computation time than others (J48 and NB). The peak TCM value for SVM is happened at lowest NI and NP (20) and its computation cost is 17.05 second. However, the values of TCM- SVM are fluctuated from the values of NI and NP 20 to 150, for instance, the second highest is at 80 and the third one is at 50. In contrast, the bottom values of TCM can be achieved in the case of NB classifier and its value is 0.05 at (NI and NP = 30, and NI and NP = 150). Even the highest TCM value of NB (0.8) is better than the lowest TCM- SVM (3.41). Meanwhile, the cost of TCM-J48 can be regarded as medium line between highest cost line of TCM- SVM and lowest one of TCM-NB. The lowest value (0.14) and highest one (1.74) of TCM-J48 are occurred at (NI and NP = 100, and NI and NP = 150) as continuous sequence.

According to the measurement of performance for three classifiers using WI-OMFS in Figure. 4.11, Figure. 4.12, and Figure. 4.13, the best accuracy (A) values can be achieved for J48 in the range of 91.63% (highest) to 53.74% (lowest) with respect to the parameter tuning of NI and NP. However, the overall accuracy value for the testing of NB with respect to the increment of NI and NP, are the poorest among others. Its peak and bottom values of A are 81.28% and 53.20% respectively. In contrast, the accuracy results for SVM is followed the nearest accuracy values of best J48 classifier results that means the range of SVM results in the range of 80% to over 90% like J48. Some case like the lowest accuracy of SVM (57.47%) is better than the lowest one of J48 (53.74%), at the same NI and NP (20). Similar trend of results for the other measurements of performance (P, R, and F_1) is occurred for three classifiers.



	20	30	40	50	60	70	80	90	100	150
TCM-J48	0.35	0.21	1.62	1.52	1.58	1.56	1.01	1.37	0.14	1.74
TCM-NB	0.09	0.05	0.7	0.8	0.56	0.63	0.41	0.45	0.05	0.55
TCM-SVM	17.05	3.41	10.77	11.18	9.63	10.87	15.95	8.03	5.86	8.23

Figure 4.10: Computation complexity: WI-OMFS filter

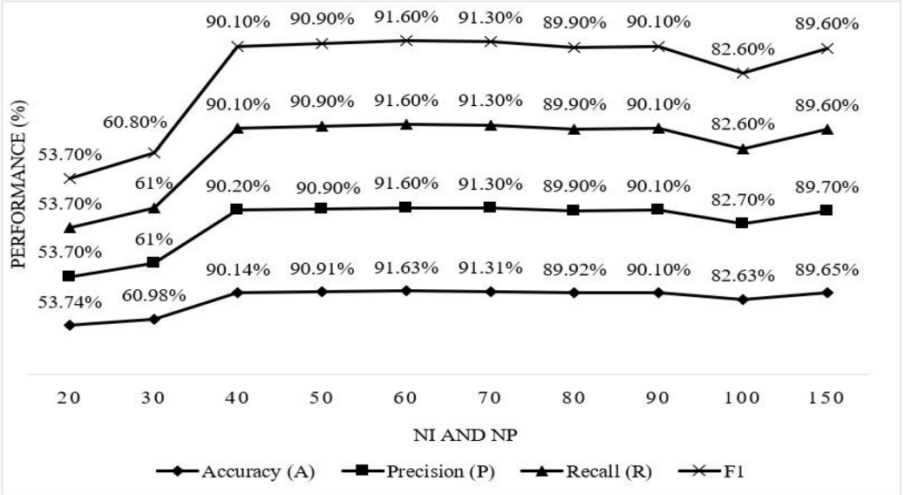


Figure 4.11: Performance results: WI-OMFS filter + J48

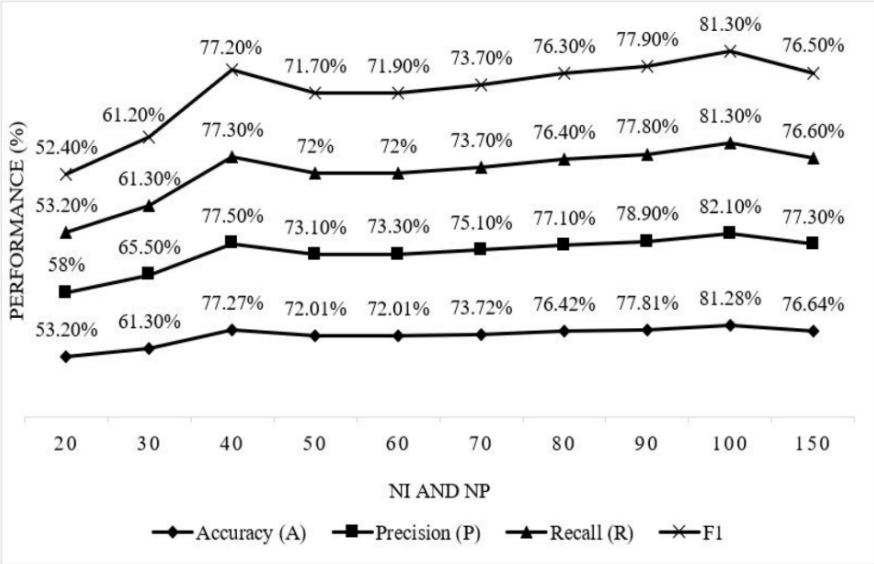


Figure 4.12: Performance results: WI-OMFS filter + NB

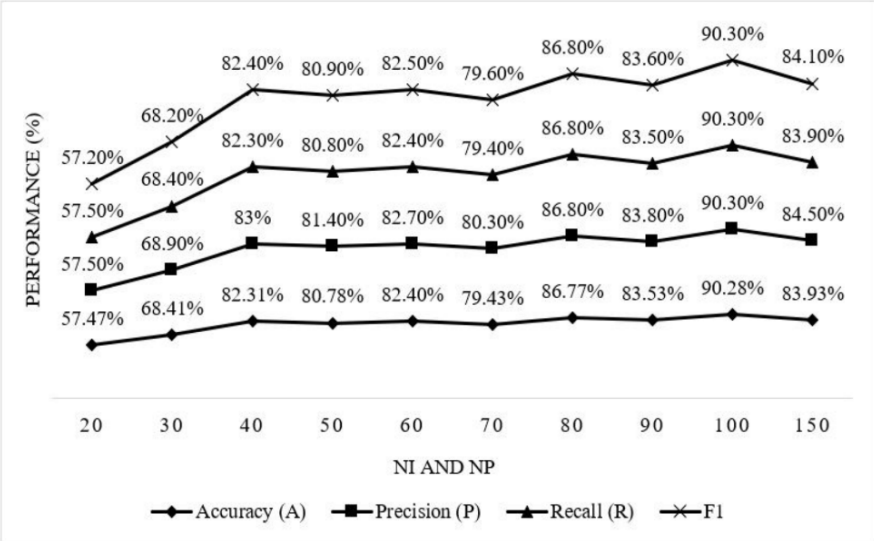


Figure 4.13: Performance results: WI-OMFS filter + SVM

4.3.2 Complexity and Performance: Traditional Search and WI-OMFS

In the case of complexity comparison between conventional search and WI-OMFS, the least number of selected feature (NF = ***55***) can be achieved by using WI-OMFS-Filter approach at the least number of iteration time (NI and NP = 20) because of the capacity of reducing feature using Wolf intelligence optimization. However, GS can provide the better TCM cost in the case of J48 and NB testing (***0.13***, ***0.03 second***). But WI-OMFS-wrapper with SVM for TCM cost can provide the faster time for computation (***5.48 seconds***). Though the second shortest TCM value in the

case of NB testing (**0.09 second**) is occurred the same in both cases of conventional search and WI-OMFS-filter, NF value for WI-OMFS (***55***) is less than BFS (***102***). The detail statistical results for conventional search and WI-OMFS is described in Table 4.1. Though highest NF values (NF = 198) at (NI and NP = 20) is occurred in according to the presented result in Table 4.1 when compared to the results of NF for CS (***BFS-NF = 102 and GS-NF = 60***) in Table 4.2, the least NF value can be obtained in WI-OMFS-Wrapper approach (***NF = 34***) at (NI/NP = 80) according to the results in Table 4.3.

In the point of view for performance, the average values of A, P, R, F_1 using both conventional search and WI-OMFS filter is almost the same for three different classifiers generally according to all statistical results in Table 4.4, Table 4.5, and Table 4.6, for instance, (accuracy, A, of ***92.08%*** in BFS- J48 and **91.63%** in WI-OMFS-filter-J48; ***92.08%*** in BFS- NB and **81.28%** in WI-OMFS-filter-NB ;***95.18%*** in BFS- SVM and **90.28%** in WI-OMFS-filter- SVM). Though their values are approximately same, the better reduced number of selected feature (NF = 86) in Table 4.3, can be achieved in the case of WI-OMFS-filter- NB and SVM at (NI and NP = 100) in Table 4.3 when compared to the testing of BFS (***NF = 102***) in Table 4.2. However, all results of performance in the case of WI-OMFS-wrapper is lower than the other three cases. Meanwhile, the results for GS with J48 and NB are almost the same, but the better result can be achieved using SVM.

Table 4.1 Computation complexity: Conventional search and WI-OMFS

Experiments	NF	TCM-J48	TCM-NB	TCM-SVM
BFS	102	0.34	**0.09**	10.89
GS	60	***0.13***	***0.03***	6.48
WI-OMFS-Filter	***55***	0.35	**0.09**	17.05
WI-OMFS-Wrapper	198	0.59	0.26	***5.48***

Table 4.2 Feature selection results for conventional search

Total no. of extracted features	2,591
NSF-BFS	***102***
NSF-GS	***60***

Table 4.3 Feature selection results for WI-OMFS

NI and NP	20	30	40	50	60	70	80	90	100	150
NF-Filter	***55***	61	1065	1064	965	1078	706	817	86	961
NF-Wrapper	198	47	47	43	48	48	***34***	43	48	46

Table 4.4 J48 performance results: Conventional search and WI-OMFS

Experiments	**A-J48**	**P-J48**	**R-J48**	**F1-J48**
BFS	***92.08%***	***92.10%***	***92.10%***	***92.10%***
GS	83.89%	83.90%	83.90%	83.90%
WI-OMFS-Filter	**91.63%**	**91.60%**	**91.60%**	**91.60%**
WI-OMFS-Wrapper	66.56%	66.60%	66.60%	66.60%

Table 4.5 NB performance results: Conventional search and WI-OMFS

Experiments	**A-NB**	**P-NB**	**R-NB**	**F1-NB**
BFS	***92.08%***	***92.10%***	***92.10%***	***92.10%***
GS	83.98%	84.10%	84.00%	83.90%
WI-OMFS-Filter	**81.28%**	**82.10%**	**81.30%**	**81.30%**
WI-OMFS-Wrapper	64.76%	67.30%	64.80%	65.00%

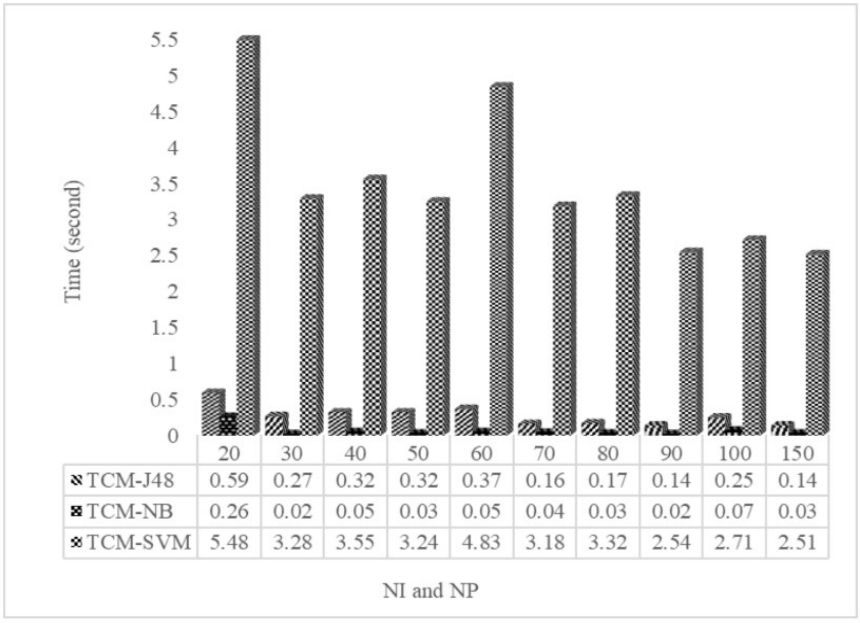
Table 4.6 SVM performance results: Conventional search and WI- OMFS

Experiments	**A-SVM**	**P-SVM**	**R-SVM**	**F1-SVM**
BFS	***95.18%***	***95.20%***	***95.20%***	***95.20%***
GS	90.73%	90.70%	90.70%	90.70%
WI-OMFS-Filter	**90.28%**	**90.30%**	**90.30%**	**90.30%**
WI-OMFS-Wrapper	79.03%	78.90%	79.00%	79.00%

4.3.3 WI-OMFS Wrapper Experimental Results

In the case of WI-OMFS wrapper, state of the cost of computation time is occurred the same in the case of WI-OMFS filter except the cost of TCM for wrapper is better than filters. Because wrapper only consider the features specifically to the characteristic of individual classifier. The detail statistical values for WI-OMFS based wrapper approach with three different classifiers are shown in Figure. 4.14. In contrast, the accuracy values for wrapper approach is not better than filter. Even the highest accuracy value of wrapper is occurred in SVM testing (79.03%) that cannot be reached

to the initial highest range of filter approach (80% to 90%). According to the overall result of performance, SVM is the best for wrapper approach while NB is grounded at bottom. And, the overall results of J48 is situated at middle stage. In addition, the values of P, R and F_1 is fluctuated with respect to the change of NI and NP, but not sharply. The way for trending the result graph is the same as the case in WI-OMFS-Filter. All results of performance for WI-OMFS- Wrapper are shown in Figure. 4.15, Figure. 4.16, and Figure. 4.17.



	20	30	40	50	60	70	80	90	100	150
TCM-J48	0.59	0.27	0.32	0.32	0.37	0.16	0.17	0.14	0.25	0.14
TCM-NB	0.26	0.02	0.05	0.03	0.05	0.04	0.03	0.02	0.07	0.03
TCM-SVM	5.48	3.28	3.55	3.24	4.83	3.18	3.32	2.54	2.71	2.51

Figure. 4.14. Computation complexity: WI-OMFS wrapper

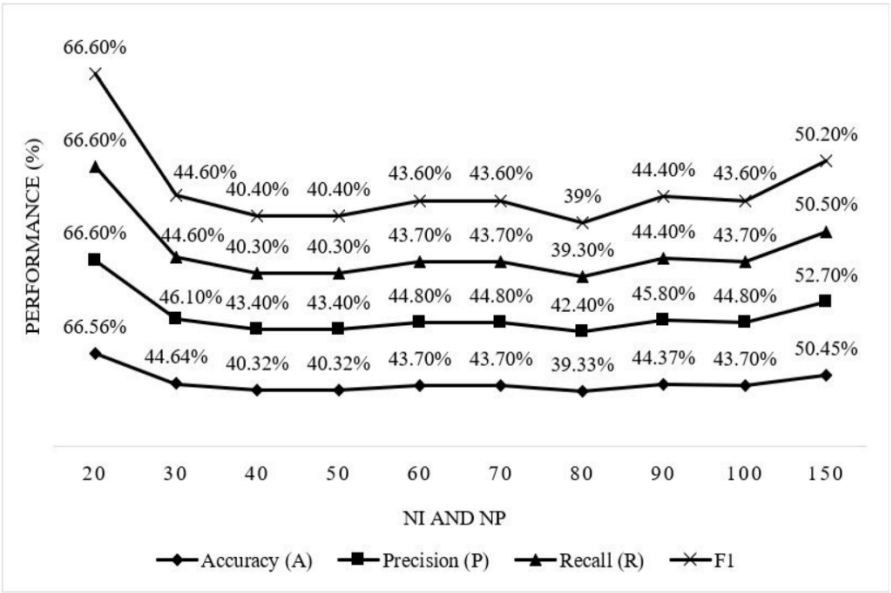


Figure. 4.15. Performance results: WI-OMFS wrapper + J48

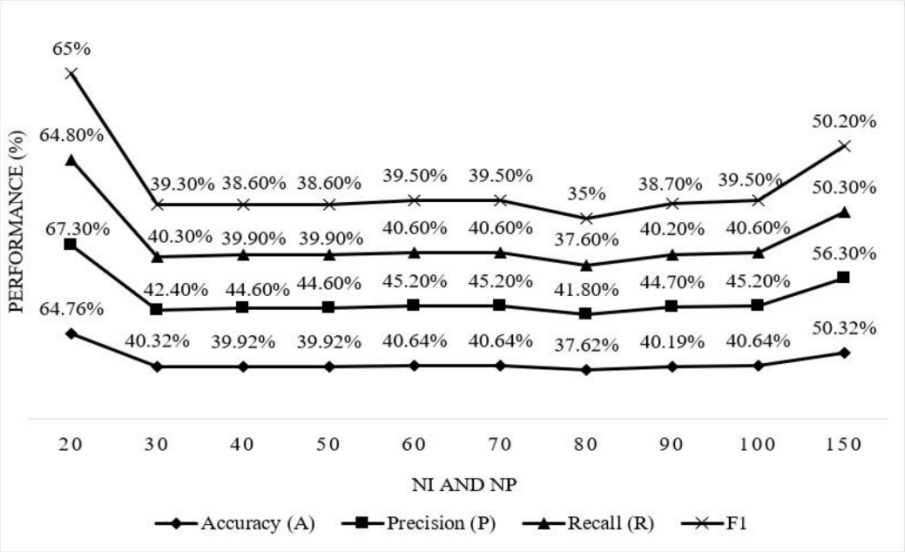


Figure. 4.16. Performance results: WI-OMFS wrapper + NB

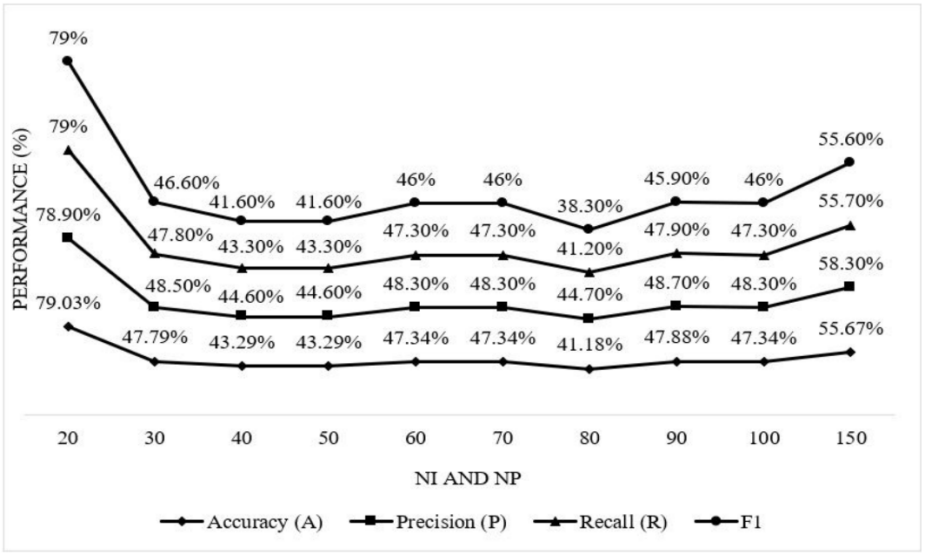


Figure. 4.17. Performance results: WI-OMFS wrapper + SVM

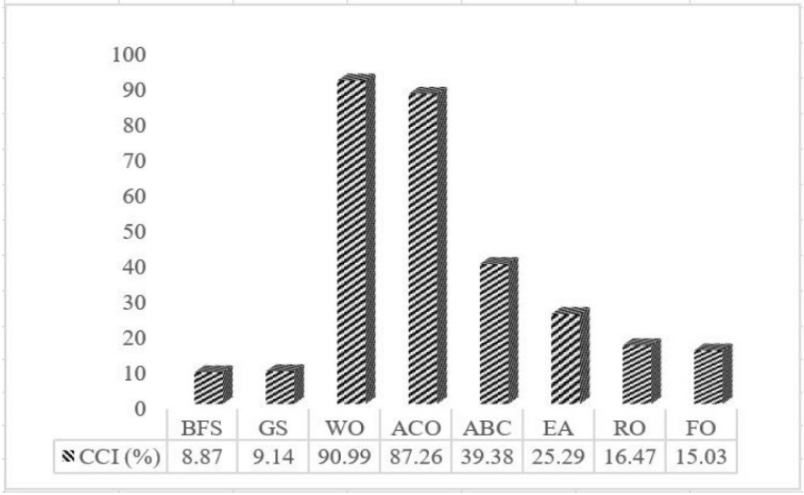
4.4 ADC-OFSMI System Testing using Traditional vs Advanced Search

In this experiment, six metaheuristic-based search policies based on evolutionary (EA), swarm intelligence (ACO and ABC), nature-inspired intelligence (WO, RO, and FO), are observed for achieving the global optimal feature subset according to the purpose of single objective function of accuracy in news classification problem. And, the results of advanced search based on meta-heuristic schemes at the values of 20 for iteration number and population size are compared with traditional one according to the following procedure: performance comparison, computation time comparison, and complexity comparison. In addition, the relationship between fitness

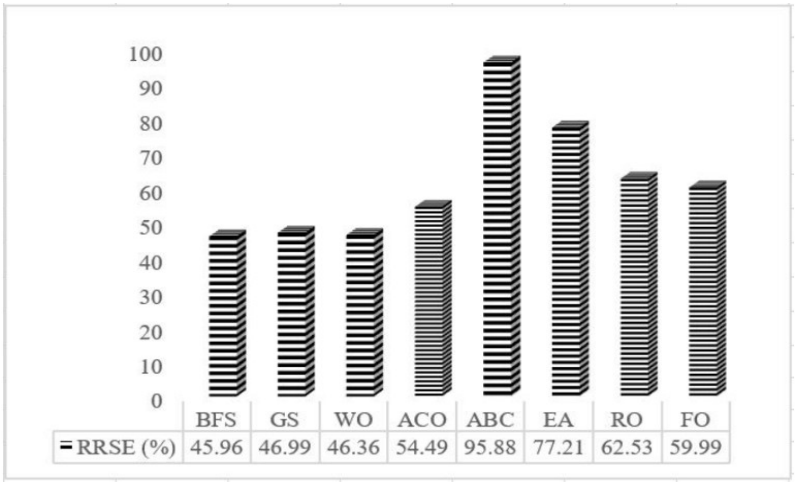
function value according to iteration number is also described to point out the important role of control parameter tuning in meta-heuristic algorithms. According to the experimental results, the proposed schemes for optimization of feature selection that can provide flexibility in integrating classifier in accordance with its objective function such as optimal classification performance by adjusting the rate of modification parameters for the testing data.

4.4.1 Performance Comparisons: Classification Accuracy and Error Rate

According to the results of performance comparisons (CCI) in Figure 4.18 (a), the accuracy percentage of traditional search approach such as BFS and GS are lower than 10% for testing dataset on 10-folds cross validation because it selects only the local optimal subset features according to the ascending order of highest score values for whole of the feature space. It can lead the selection of irrelevant features when evaluating randomly on testing datasets with the ratio of 30 to 70 percent folding (10-folds cross validation). To overcome this hardness, the powerful advanced search capability based on meta-heuristic search is applied for finding the global optimal feature. It can take advantage of flexibility with integration of classifier as its objective function and installation of any meta-heuristic algorithm for facilitating heuristic search. According to the results, the first to third optimal classification accuracy is provided by WO, ACO, and ABC respectively. Meanwhile, the result of RRSE for WO is still better than the traditional search approach according to the results performance comparisons (RRSE) in Figure 4.18 (b).



(a)



(b)

Figure 4.18: Performance comparisons (%) (a). CCI (b). RRSE

4.4.2 Computation Time Comparison

According to the results of computation time comparison (second) in Figure 4.19, GS search takes the shortest time for feature selection because it only selects the highest score value of features depending on the defined threshold level, with the lack of consideration for the whole hypothesis of feature. However, the second shortest computation time is provided by ABC by applying the distributed searching for optimal subset features over the defined labels of class. Like the concept of GS, BFS takes the fast computation time than the other meta-heuristic search approaches. Nevertheless, some of the meta-heuristic search can provide the reasonable computation time with optimal performance of classification.

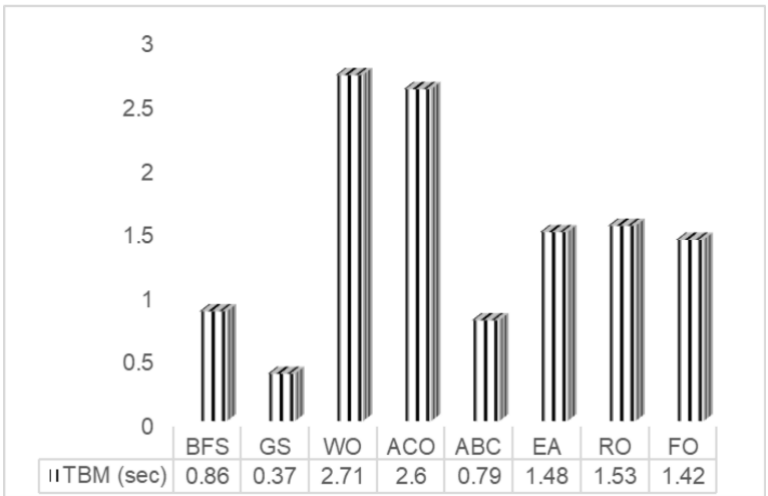
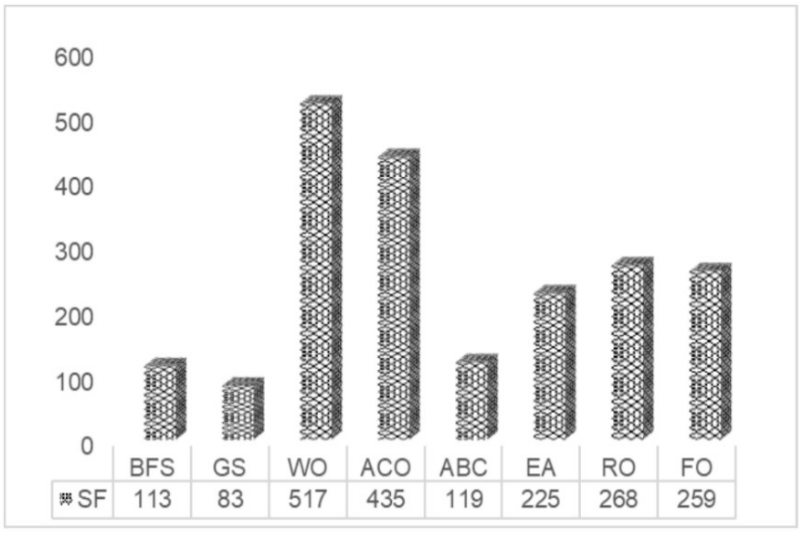


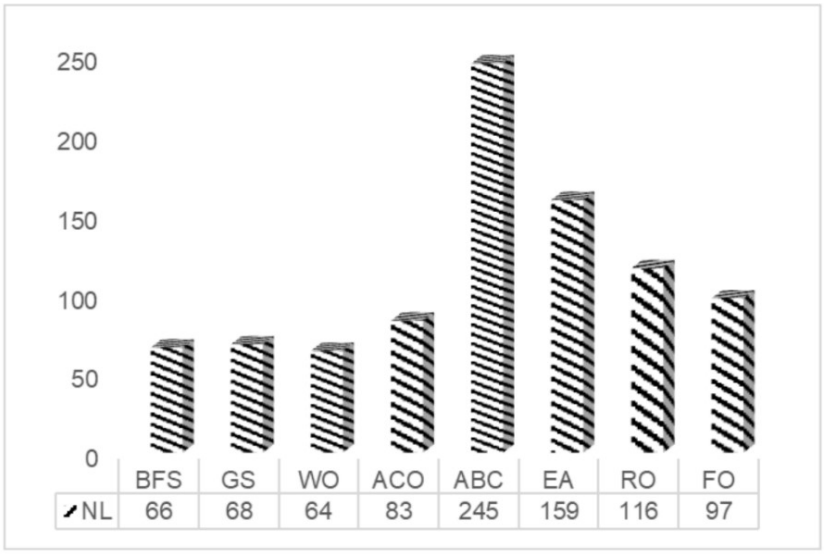
Figure 4.19: Computation time comparison (second)

4.4.3 Complexity Comparisons: Numbers of Selected Features, Leaves and Tree Size

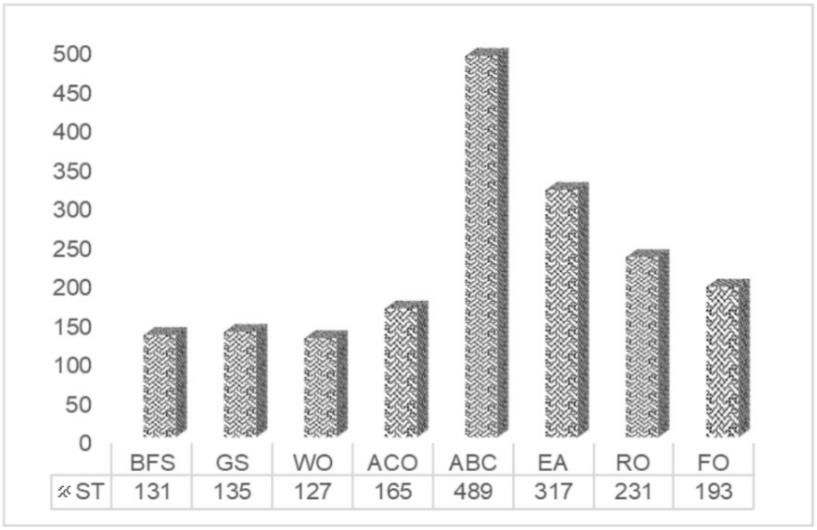
Although the interested BBC dataset is not a big data, they include many features. The traditional exhaustive scheme is become inefficient for this kind of multi-dimensional feature for searching the relevance features because they are trying all possible combinations of features take seemingly every time and difficult to synchronize the classifier output performance. According to the results of comparison for number of selected features (SF) in Figure 4.20 (a), GS with CFS selected the least number of features (SF) in traditional search and ABC with CFS provided the smallest number of selected features in advanced search. Though SF for both traditional searches is less than the ones for advanced search, selecting the global optimal feature subset can provide more chance to achieve good performance. According to the results of comparison for number of leaves (NL) in Figure. 4.20 (b) and size of tree (ST) in Figure. 4.20 (c), NL and ST for classification model of both traditional search such as GS and BFS, and advanced one such as WO are almost the same.



(a)



(b)



(c)

Figure 4.20: Comparison for (a). number of selected features (SF) (b). number of leaves (NL) (c). size of tree (ST)

4.4.4 Relationship Curve: Fitness Function and Iteration Time

According to the experimental results in Table 4.1, the accuracy value (fitness function) is always fluctuated when iteration time is changed. Because this is the definition of metaheuristic search with randomized and the nature of exploration and exploitation. Therefore, iteration number is common important factor to achieve the level of proposed fitness function in all meta-heuristic algorithms. In addition, other specific factors for each specific meta-heuristic algorithm should be considered as

important and therefore the automatic model for the adaptation of control parameters should be considered in future.

Table 4.7 Results analysis: Relationship between fitness function and iteration time

Experiments: fitness function	NI: 20	NI: 30	NI: 40	NI: 50	NI: 100
ACO	87.26%	85.78%	88.88%	91.31%	91.99%
ABC	39.38%	75.92%	66.11%	71.60%	68.45%
EA	25.29%	83.17%	83.17%	83.14%	83.17%
FO	15.03%	87.58%	82.94%	86.68%	89.51%
RO	16.47%	61.03%	88.61%	87.26%	88.97%
WO	90.99%	60.53%	90.73%	91.94%	83.71%

4.5 ADC-OFSMI System Testing using Traditional vs Modern Nature-inspired Intelligence Search

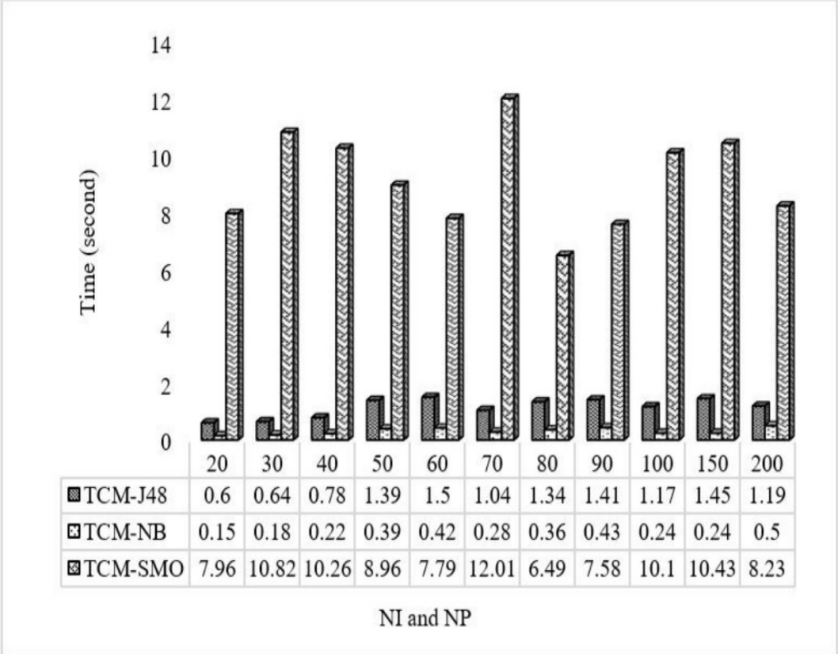
In this experiment, the performance of modern nature-inspired intelligence search for optimization of multi-dimensional feature selection in document classification is observed. Four modern nature-inspired algorithms such as Firefly optimization, Elephant optimization, Cuckoo optimization, and Bat optimization are used with filter feature selection approach for searching the global optimal subset features. Moreover, the evaluation process for the performance of proposed models is performed by three different classifiers and the measurement of accuracy for performance and the measurements of number of selected feature and time taken for computation complexity are used to evaluate the capability of proposed searching algorithms. All detailed testing results are discussed in the following subsections

4.5.1 ADC-OFSMI System using Bat Optimization

Figure 4.21 shows the three different learning models computation cost for BO with the change of parameter tuning in NI and NP. When NI and NP are increased, the value of NF is also changed and therefore it can have a direct impact on the learning model output in terms of performance and computation complexity cost. In this proposed model, accuracy is used as a single objective function. Among of the three classifiers' outputs, the highest computation cost (6.49 second) is recorded in SMO classifier and its lowest value is even greater that other two classifiers' peak computation cost like (TCM-NB: 0.5 at NI and NP=200) and (TCM-J48: 1.5 at NI and

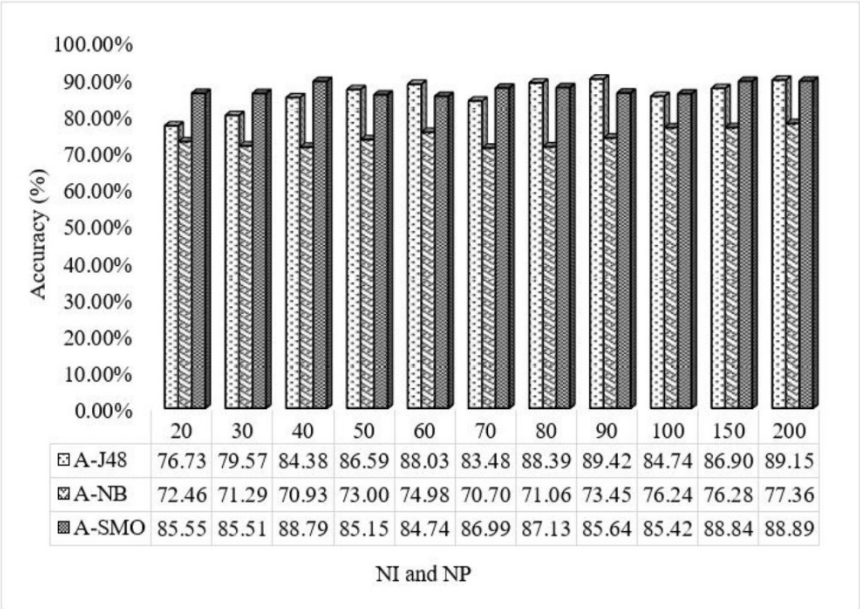
NP=60). TCM results for all three classifiers are fluctuated according to the series of iteration and population rate because the nature of BO is to look forward the optimal feature subsets according to the process of optimization for individual search parameters.

In Figure 4.22, the performance results for NB is the lowest in average while the highest accuracy results (89.42%) is provided by J48 at the rate of iteration and population (NI and NP=90). According to the results, BO is always searching the global optimal features in a given space of search area and it will continue their searching until the maximum number of search cycles and it stops after achieving the feature that is match with their defined objective function.



	20	30	40	50	60	70	80	90	100	150	200
TCM-J48	0.6	0.64	0.78	1.39	1.5	1.04	1.34	1.41	1.17	1.45	1.19
TCM-NB	0.15	0.18	0.22	0.39	0.42	0.28	0.36	0.43	0.24	0.24	0.5
TCM-SMO	7.96	10.82	10.26	8.96	7.79	12.01	6.49	7.58	10.1	10.43	8.23

Figure 4.21: Computation cost for BO

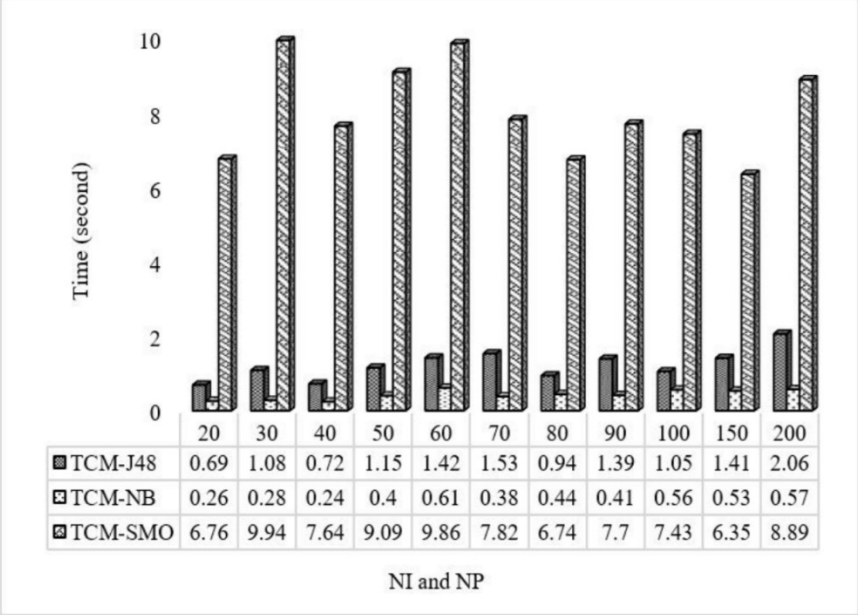


	20	30	40	50	60	70	80	90	100	150	200
A-J48	76.73	79.57	84.38	86.59	88.03	83.48	88.39	89.42	84.74	86.90	89.15
A-NB	72.46	71.29	70.93	73.00	74.98	70.70	71.06	73.45	76.24	76.28	77.36
A-SMO	85.55	85.51	88.79	85.15	84.74	86.99	87.13	85.64	85.42	88.84	88.89

Figure 4.22: Performance accuracy for BO

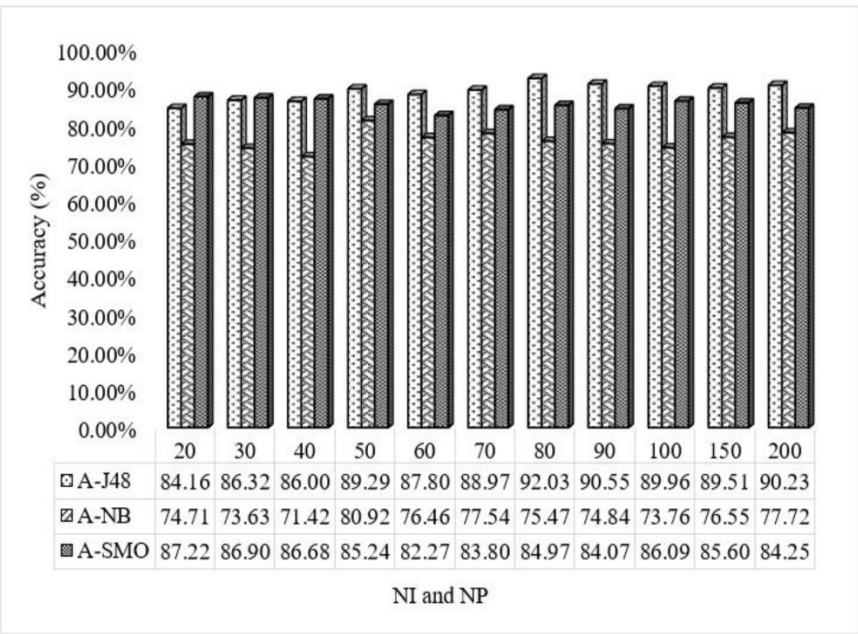
4.5.2 ADC-OFSMI System using Cuckoo Optimization

According to the experimental results of CO in Figure 4.23, computation cost for SMO building model is placed in the highest position like (9.94 second) at NI and NP=30. However, NB is the least computation approach and its lowest cost (0.24 second) is happened at NI and NP= 40. Meanwhile, J48 learning model is existed as medium value between the lines of highest and lowest cost. In Figure 4.24, the best accuracy results are achieved in the testing of J48 at NI and NP= 80 (92.03%). In contrast, NB is generated the poor accuracy results when compared to the lowest accuracy results in SMO (82.24% at NI and NP=50) and in J48 (84.16% at NI and NP=20). In addition, SMO is provided the second good classification performance in the range of NI and NP.



	20	30	40	50	60	70	80	90	100	150	200
TCM-J48	0.69	1.08	0.72	1.15	1.42	1.53	0.94	1.39	1.05	1.41	2.06
TCM-NB	0.26	0.28	0.24	0.4	0.61	0.38	0.44	0.41	0.56	0.53	0.57
TCM-SMO	6.76	9.94	7.64	9.09	9.86	7.82	6.74	7.7	7.43	6.35	8.89

Figure 4.23: Computation cost for CO



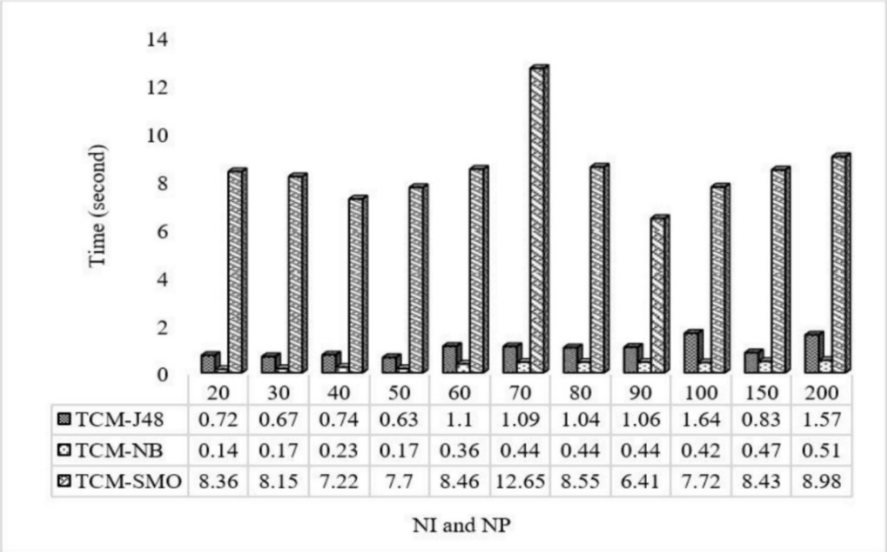
	20	30	40	50	60	70	80	90	100	150	200
A-J48	84.16	86.32	86.00	89.29	87.80	88.97	92.03	90.55	89.96	89.51	90.23
A-NB	74.71	73.63	71.42	80.92	76.46	77.54	75.47	74.84	73.76	76.55	77.72
A-SMO	87.22	86.90	86.68	85.24	82.27	83.80	84.97	84.07	86.09	85.60	84.25

Figure 4.24: Performance accuracy for CO

4.5.3 ADC-OFSMI System using Elephant Optimization

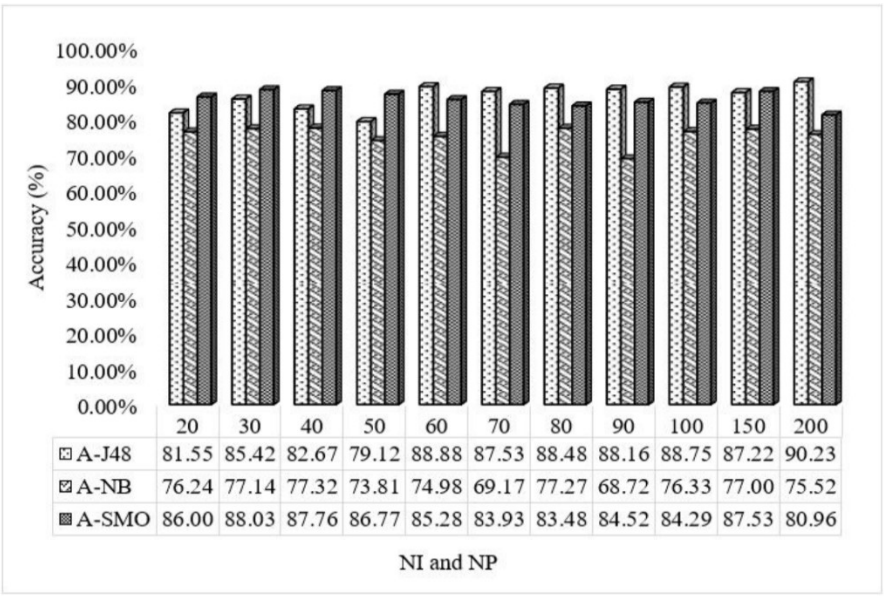
In the experiment of EO in Figure 4.25, the best computation cost is provided in NB classifier and its lowest cost is occurred at NI and NP=20 (0.14 second). In addition, its highest cost is still better than the lowest on in J48 (0.63 second at NI and NP=50). In contrast, the lowest cost of SMO classifier (6.41 second at NI and

NP=90) is still have higher than the other two classifiers' highest one, (1.64 second at NI and NP=100) and (0.51 second at NI and NP=200) in J48 and NB respectively. In the performance result in Figure 4.26, the peak accuracy is occurred in J48 learning model with the value of 90.23% (NI and NP=200). And, the second-best performance accuracy is obtained in the process of SMO in average. The maximum accuracy of SMO is achieved at NI and NP=30 (88.03%). However, the accuracy value of NB is still under 80%.



	20	30	40	50	60	70	80	90	100	150	200
TCM-J48	0.72	0.67	0.74	0.63	1.1	1.09	1.04	1.06	1.64	0.83	1.57
TCM-NB	0.14	0.17	0.23	0.17	0.36	0.44	0.44	0.44	0.42	0.47	0.51
TCM-SMO	8.36	8.15	7.22	7.7	8.46	12.65	8.55	6.41	7.72	8.43	8.98

Figure 4.25: Computation cost for EO

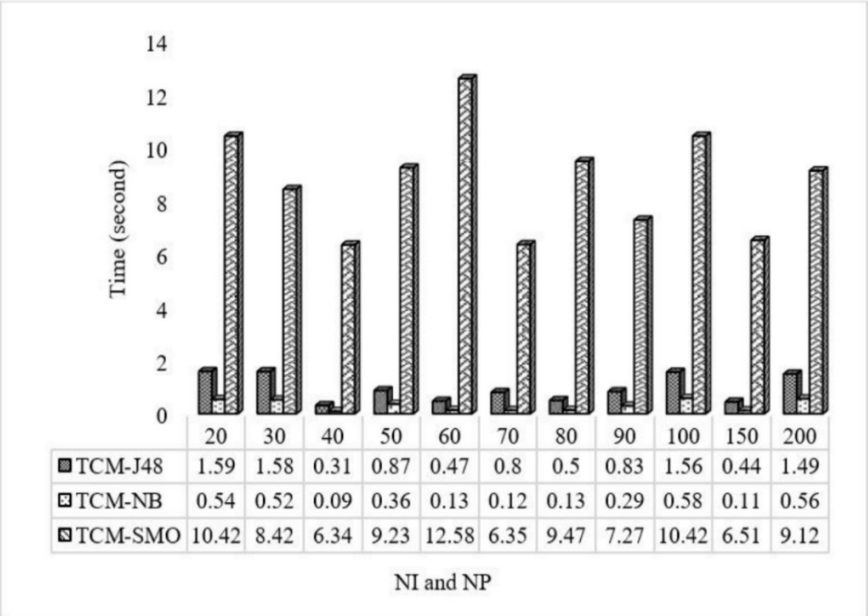


	20	30	40	50	60	70	80	90	100	150	200
A-J48	81.55	85.42	82.67	79.12	88.88	87.53	88.48	88.16	88.75	87.22	90.23
A-NB	76.24	77.14	77.32	73.81	74.98	69.17	77.27	68.72	76.33	77.00	75.52
A-SMO	86.00	88.03	87.76	86.77	85.28	83.93	83.48	84.52	84.29	87.53	80.96

Figure 4.26: Performance accuracy for EO

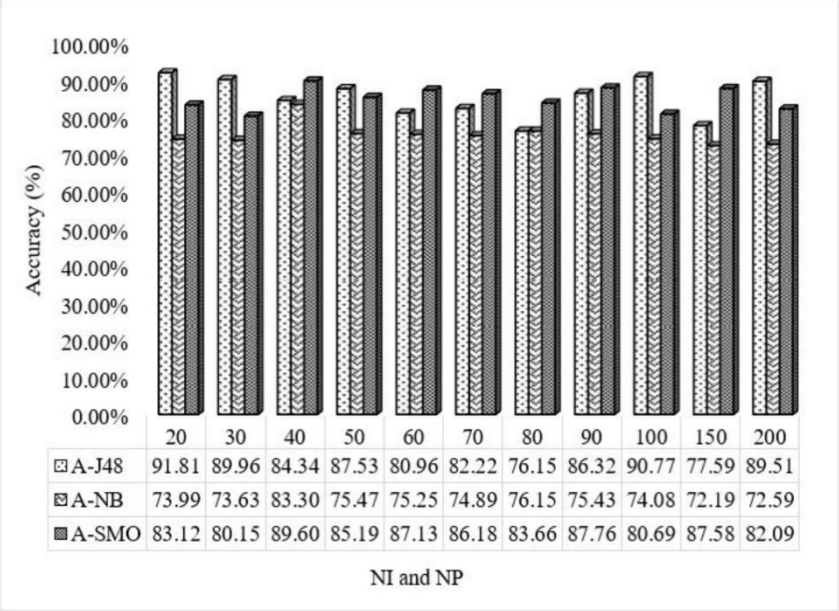
4.5.4 ADC-OFSMI System using Firefly Optimization

According to the results demonstration in Figure 4.27, the cost of SMO is the highest such as (10.42second) at NI and NP=20. All TCM results for three classifiers are fluctuated according to the selected feature in each different value of NI and NP. NB classifier has the best computation cost in term of average and J48 can be provided the second-best computation cost. Figure 4.28 shows the results of accuracy for FO and the above 90% accuracy result is achieved by J48 learning model. Meanwhile, SMO results are above 80%, but under 90%. And, the performance results of NB classifier are at bottom line when compared to SMO and J48.



	20	30	40	50	60	70	80	90	100	150	200
TCM-J48	1.59	1.58	0.31	0.87	0.47	0.8	0.5	0.83	1.56	0.44	1.49
TCM-NB	0.54	0.52	0.09	0.36	0.13	0.12	0.13	0.29	0.58	0.11	0.56
TCM-SMO	10.42	8.42	6.34	9.23	12.58	6.35	9.47	7.27	10.42	6.51	9.12

Figure 4.27: Computation cost for FO



	20	30	40	50	60	70	80	90	100	150	200
A-J48	91.81	89.96	84.34	87.53	80.96	82.22	76.15	86.32	90.77	77.59	89.51
A-NB	73.99	73.63	83.30	75.47	75.25	74.89	76.15	75.43	74.08	72.19	72.59
A-SMO	83.12	80.15	89.60	85.19	87.13	86.18	83.66	87.76	80.69	87.58	82.09

Figure 4.28: Performance accuracy for FO

4.5.5 ADC-OFSMI System Result Comparison: Traditional Search and Modern Nature-Inspired Intelligence Search

Although NSF results for BFS is 102 in Table 4.8, which is lesser than NSF results of four modern meta-heuristic search in Figure 4.29, it is fixed scheme for searching and not suitable if the feature searching space is complex. In contrast, feature selection process can be optimized according to the parameter tuning in the searching process of four meta-heuristic intelligence that is shown in Figure 4.29, and they can look forward features by considering local optimal for individual label of class and global optimal from all local optimal results. Though BFS is provided the accuracy of above 90% in all three classifiers, the accuracy values for NB and SMO in RS are not good and it also has high number of selected feature (2,591).

According to the results in Sections 4.5.1 - 4.5.4, the optimization of high-dimensional feature selection can be achieved by the parameter tuning of modern nature inspired search, and it can lead to the better performance document classification results. In addition, the proposed four modern meta-heuristic algorithms have dynamic and randomized nature of search for global optimal features, and it is suitable for high-dimensional feature spaces. However, the limitation for the proposed search intelligence is to have high computation cost if the search cycle and search agent are

increased. Therefore, the proposed model should be extended to multi-objective model in order to find out the global optimal feature with reasonable computation cost.

Table 4.8 Computation complexity result for traditional search

Experiments	BFS	RS
NSF	102	2591
TCM-J48 (second)	0.25	2.59
TCM-NB (second)	0.07	1.15
TCM-SMO (second)	6.08	18.12

Table 4.9 Performance result for traditional search

Experiments	A-NB	A-SMO	A-J48
BFS	92.08%	95.18%	92.08%
RS	69.35%	60.44%	92.44%

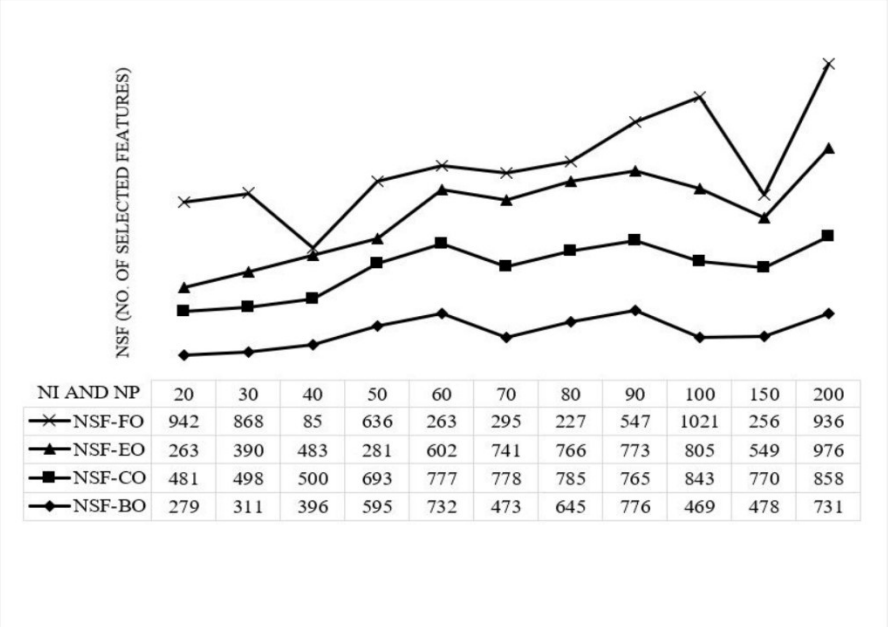


Figure 4.29: Complexity results for four modern nature-inspired search

4.6 Summary of Discussion

Text feature is high dimensionality because the individual word is regarded as the feature. In addition, the word in each group is also included in another label of group, for example, the news article, and therefore the feature selection method

is needed to select the obvious feature or most relevant feature for individual label. Moreover, the role of un-biasing feature searching approach for the specific label of class with highest score is become a critical challenge for high dimensionality feature subset selection in document classification. To fulfill this research gap, feature selection based on meta-heuristic search is investigated in this thesis. According to the experimental results, the proposed system can overcome the NP-hard problem for feature selection process and then provide optimal document classification performance according to the objective function (accuracy).

CHAPTER 5

CONCLUSIONS

This chapter includes three sections concerned with overall conclusions of this thesis.

5.1 Major Findings

In this section, the critical findings from five experiments of thesis titled "Feature Selection for Document Classification: Case Study of Meta-heuristic Intelligence and Traditional Approaches" are discussed.

In the first case study experiment for proposed thesis model, the experimental results show that the selected feature using swarm intelligence search can provide more robust for classification results than three traditional searches. However, the critical factor for achieving optimal features depends on the increase number of iteration and population size adaptively. According to the observation from experiments, the following conclusions can be made as future research direction. Firstly, the swarm-based search approach should be implemented on the platform of distributed system in order to have optimal performance with healthy computation cost (automatic multi-objective optimization model) for big data computation. Secondly, the investigation of performance for swarm-based feature selection process should take benefit of using more complex classifiers such as artificial neural network [74], and evolutionary neural network and spiking neural network [75], for other kinds of datasets. Finally, the hybrid feature selection approach should be implemented by combining the appropriate meta-heuristic algorithms, depending on the characteristics of the individual problems.

In the second case study results using MI in KDP, better performance can be achieved using the MI with filter approach for classification model by reducing the number of selected features exponentially. The idea for selecting optimal feature using MI which are based on the mathematical model of probabilistic with randomness searching. Though the datasets used in this paper is not a big data, the number of extracted features is multi-dimensional feature. Therefore, KDP for big data using MI will emerge as big challenge for the Information age. Hence, distribution system with

MI should be considered as future work for multi-objective function such as optimization of effective performance with efficient computation. In addition, better model tuning using different kinds of other computation intelligence scheme such as firefly, bat, etc., should be explored depend on the characteristic of data and objective function.

According to the third case study results, the better performance results can be achieved using wolf intelligence-based search with filter approach than wrapper as general. In addition, J48 classifier is the best choice for working with WI-OMFS than other two classifiers (NB and SVM). In addition, the selected number of features using WI-OMFS-Filter can be reduced twice than BFS search and all classifiers results are improved when the search iteration and population are increased. However, WI-OMFS-Wrapper is still having lower performance than traditional search. Therefore, wrapper approach is not appropriate for working with WI-based search policy and it should be put as future work for improvement of proposed system.

According to the fourth case study results, the correctly classified instance results (CCI) for all six-metaheuristic based search is outperformed than two traditional searches. However, the computation cost for six MI-based searches is still higher than traditional search. Moreover, the value for error percentage (RRSE) is still better in wolf intelligence-based search than traditional approaches. On the other hand, the number of selected features (SF), number of leaves (NL), and size of tree (ST) which are based on MI search is more than the traditional search when only small iteration and population number is used for feature searching process. Therefore, the distributed platform should be used when the number of iteration and population is increased in order to achieve the better performance with good computation time.

In the fifth case study results, the performance results for four modern nature inspired based filter feature selection approach using three classifier evaluations can only provide the possibility trend for incrementing performance according to the rate of NI and NP. However, the performance results for all three-classifier evaluation in ranker approach based on traditional search cannot be improved when compared to all four modern nature inspired based performance results. On the other hand, the complexity for selected number of feature (NSF) in all four nature inspired search

approaches is increased when trying to achieve the better performance results by increasing rate of iteration and population. In addition, the computation time for nature inspired search is also more costly than traditional search. Therefore, the future work should be considered to achieve the better performance of document classification by reducing the computation time for multi-dimensional feature selection process.

5.2 Recommendations and Limitations

The observations of text feature selection using meta-heuristic intelligence can be applied for various application areas that related with document classification. To be specific problem areas, this study can be contributed for developing the intelligence model of document classification which are based on the analysis of news documents.

However, the proposed model was implemented using WEKA library file and run on Java platform. Therefore, the extension of proposed model implementation has limitation for building hybrid meta-heuristic search.

5.3 Future Work

The heterogenous data from different resources is growth exponentially in nowadays and therefore the role of knowledge discovery from data for different areas of application is still become a high demand for the era of data. As the behavior of data change like volume, complexity, dimensionality, and so on, the adaptation of the scheme for data mining processing is always needed for our daily life in modern society. Therefore, the following issues could be considered for future research:

1. The proposed model should be extended from single objective to multi objective optimization problem by collaboration with other filed like Hadoop distribution in future in order to control the huge number of iteration and population size for metaheuristic search by decentralizing the searching process to looking for the better features with more efficient time consumption and complexity of calculation in order to support higher accuracy for learning model with robustness of error rate for real time big data classification model.

2. The proposed model should be upgraded by combining the knowledge of NLP to become more intelligent learning model like way of thinking of human being by using the approach of language understanding.
3. In order to achieve more intelligence a new hybrid meta-heuristic algorithm should be observed on Python platform. Moreover, the other learning algorithms, for instance neural network, should be applied for building the classification model as a future work.

REFERENCES

[1] P. Rohilla and O. Sharma, “Web Content Mining: An Implementation on Social Websites,” *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 4, no. 7, pp. 108–111, 2015.

[2] M. Teimouri *et al.*, “Data Mining Concepts & Techniques, ” vol. 15. 2016.

[3] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, “Knowledge Discovery and Data Mining: Towards a Unifying Framework.,” *Int Conf Knowl. Discov. Data Min.*, pp. 82–88, 1996.

[4] Yang, Yiming, and Jan O. Pedersen, "A Comparative Study on Feature Selection in Text Categorization," In *Icml*, vol. 97, no. 412-420, pp. 35, 1997.

[5] C. C. Aggarwal and C. Xhai, “A Survery of Text Clustering Algorithms,” In *Min. text data*, vol. 8, pp. 77–128, 2012.

[6] T. M. Cover, J. A. Thomas, D. L. Schilling, J. Bellamy, and R. L. Freeman, “Elements of Information Theory,” *Wiley Series in Telecommunications Transmission Handbook*, *2nd Edition*, 1991.

[7] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, “Classification and Regression Trees,” In *Proceedings of the Thirteenth International Conference*, , pp. 148, 1996.

[8] ESAC, “On the Effectiveness of Receptors in Recognition Systems,” *IEEE transactions on Information Theory*, vol. 9, no. 1, pp. 11-17, 1963.

[9] K. Kira and L. A. Rendell, “A Practical Approach to Feature Selection,” In *Machine Learning Proceedings*, pp. 249-256, 1992.

[10] H. Almuallim and T. G. Dietterich, “Learning Boolean Concepts in the Presence of Many Irrelevant Features,” *Artif. Intell.*, vol. 69, no. 1–2, pp. 279–305, 1994.

[11] A. Allahverdipoor and F. S. Gharehchopogh, “An Improved K-Nearest Neighbor with Crow Search Algorithm for Feature Selection in Text Document Classification,” *Journal of Advances in Computer Research*, vol. 9, no. 2, pp. 37–48, 2018.

[12] T. Kaur, B. S. Saini, and S. Gupta, “A Novel Feature Selection Method for Brain Tumor MR Image Classification based on the Fisher Criterion and Parameter Free Bat Optimization,” *Neural Comput. Appl.*, vol. 29, no. 8, pp. 193–206, 2018.

[13] M. Mavrovouniotis, C. Li, and S. Yang, “A Survey of Swarm Intelligence for Dynamic Optimization : Algorithms and Applications,” *Swarm Evol. Comput.*, vol. 33, pp. 1–17, 2017.

[14] S. Rai and R. Sharma, “Solution to Travelling Salesman Problem by Nature Inspired Algorithm,” *Int. J. Comput. Appl.*, vol. 123, no. 18, pp. 52–54, 2015.

[15] M. H. Aghdam, N. Ghasem-Aghaee, and M. E. Basiri, “Text Feature Selection using Ant Colony Optimization,” *Expert Syst. Appl.*, vol. 36, no. 3, pp. 6843–6853, 2009.

[16] S. R. Ahmad, A. A. Bakar, and M. R. Yaakub, “Metaheuristic Algorithms for Feature Selection in Sentiment Analysis,” *Proc. 2015 Sci. Inf. Conf. SAI*, pp. 222–226, 2015.

[17] B. Tran, B. Xue, and M. Zhang, “Variable-Length Particle Swarm Optimization for Feature Selection on High-Dimensional Classification,” *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 473–487, 2019.

[18] A. Mahanipour, H. Nezamabadi-pour, and B. Nikpour, “Using Fuzzy-Rough Set Feature Selection for Feature Construction based on Genetic Programming,” *2018 3rd Conf. Swarm Intell. Evol. Comput.*, pp. 1–6, 2018.

[19] K. Ahmed, A. E. Hassanien, and S. Bhattacharyya, “A Novel Chaotic Chicken Swarm Optimization Algorithm for Feature Selection,” In *2017 Third International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, pp. 259–264, 2017.

[20] D. A. Alboaneen, H. Tianfield, and Y. Zhang, “Sentiment Analysis via MultiLayer Perceptron Trained by Meta-Heuristic Optimisation,” *Proc. 2017 IEEE Int. Conf. Big Data*, pp. 4630–4635, 2018.

[21] H. B. Nguyen, B. Xue, P. Andreae, and M. Zhang, “Particle Swarm Optimisation with Genetic Operators for Feature Selection,” *2017 IEEE Congr. Evol. Comput.*, pp. 286–293, 2017.

[22] D. O’Neill, A. Lensen, B. Xue, and M. Zhang, “Particle Swarm Optimisation for Feature Selection and Weighting in High-Dimensional Clustering,” *2018 IEEE Congr. Evol. Comput. CEC 2018 - Proc.*, pp. 1–8, 2018.

[23] X. Bai, X. Gao, and B. Xue, “Particle Swarm Optimization Based Two-Stage Feature Selection in Text Mining,” *2018 IEEE Congr. Evol. Comput. CEC 2018 - Proc.*, pp. 1–8, 2018.

[24] M. Sharawi, H. M. Zawbaa, and E. Emary, “Feature Selection Approach based on Whale Optimization Algorithm,” *9th Int. Conf. Adv. Comput. Intell. ICACI 2017*, pp. 163–168, 2017.

[25] H. Peng, C. Ying, S. Tan, B. Hu, and Z. Sun, “An Improved Feature Selection Algorithm Based on Ant Colony Optimization,” *IEEE Access*, vol. 6, pp. 69203–69209, 2018.

[26] M. S. R. Nalluri, T. SaiSujana, K. Harshini Reddy, and V. Swaminathan, “An Efficient Feature Selection using Artificial Fish Swarm Optimization and SVM Classifier,” *2017 Int. Conf. Networks Adv. Comput. Technol. NetACT 2017,* pp. 407–411, 2017.

[27] S. Sasikala and D. Renuka Devi, “A Review of Traditional and Swarm Search based Feature Selection Algorithms for Handling Data Stream Classification,” *3rd IEEE Int. Conf. Sensing, Signal Process. Secur. ICSSS 2017*, pp. 514–520, 2017.

[28] D. K. Jain, A. Kumar, S. R. Sangwan, G. N. Nguyen, and P. Tiwari, “A Particle Swarm Optimized Learning Model of Fault Classification in Web-Apps,” *IEEE Access*, vol. 7, pp. 18480–18489, 2019.

[29] B. A. Garro, R. Salazar-Varas, and R. A. Vazquez, “EEG Channel Selection using Fractal Dimension and Artificial Bee Colony Algorithm,” *Proc. 2018 IEEE Symp. Ser. Comput. Intell. SSCI 2018*, pp. 499–504, 2019.

[30] Q. Al-Tashi, S. J. Abdul Kadir, H. M. Rais, S. Mirjalili, and H. Alhussian, “Binary Optimization using Hybrid Grey Wolf Optimization for Feature Selection,” *IEEE Access*, vol. 7, pp. 39496–39508, 2019.

[31] P. Shunmugapriya and S. Kanmani, “A Hybrid Algorithm using Ant and Bee Colony Optimization for Feature Selection and Classification (AC-ABC Hybrid),” *Swarm and Evolutionary Computation*, vol. 36. pp. 27–36, 2017.

[32] M. Mafarja and S. Mirjalili, “Whale Optimization Approaches for Wrapper Feature Selection,” *Appl. Soft Comput. J.*, vol. 62, pp. 441–453, 2018.

[33] B. Nakisa, M. N. Rastgoo, D. Tjondronegoro, and V. Chandran, “Evolutionary Computation Algorithms for Feature Selection of EEG-based Emotion Recognition using Mobile Sensors,” *Expert Syst. Appl.*, vol. 93, pp. 143–155, 2018.

[34] M. A. El Aziz and A. E. Hassanien, “Modified Cuckoo Search Algorithm with Rough Sets for Feature Selection,” *Neural Comput. Appl.*, vol. 29, no. 4, pp. 925–934, 2018.

[35] A. Chandra Pandey, D. Singh Rajpoot, and M. Saraswat, “Twitter Sentiment Analysis using Hybrid Cuckoo Search Method,” *Inf. Process. Manag.*, vol. 53, no. 4, pp. 764–779, 2017.

[36] L. M. Abualigah and A. T. Khader, “Unsupervised Text Feature Selection Technique based on Hybrid Particle Swarm Optimization Algorithm with Genetic Operators for the Text Clustering,” *J. Supercomput.*, vol. 73, no. 11, pp. 4773–4795, 2017.

[37] S. Gu, R. Cheng, and Y. Jin, “Feature Selection for High-Dimensional Classification using a Competitive Swarm Optimizer,” *Soft Comput.*, vol. 22, no. 3, pp. 811–822, 2018.

[38] E. Aličković and A. Subasi, “Breast Cancer Diagnosis using GA Feature Selection and Rotation Forest,” *Neural Comput. Appl.*, vol. 28, no. 4, pp. 753–763, 2017.

[39] E. Emary, H. M. Zawbaa, K. K. A. Ghany, A. E. Hassanien, and B. Parv, “Firefly Optimization Algorithm for Feature Selection,” In *Proceedings of the 7th Balkan Conference on Informatics Conference*, pp. 26, 2015.

[40] M. Prabukumar, L. Agilandeeswari, and K. Ganesan, “An Intelligent Lung Cancer Diagnosis System using Cuckoo Search Optimization and Support Vector machine Classifier,” *J. Ambient Intell. Humaniz. Comput.*, vol. 10, no. 1, pp. 267–293, 2019.

[41] M. A. Tawhid and K. B. Dsouza, “Hybrid Binary Dragonfly Enhanced Particle Swarm Optimization Algorithm for Solving Feature Selection Problems,” *Math. Found. Comput.*, vol. 1, no. 2, pp. 181–200, 2018.

[42] A. E. Hassanien, M. Kilany, E. H. Houssein, and H. AlQaheri, “Intelligent Human Emotion Recognition based on Elephant Herding Optimization Tuned Support Vector Regression,” *Biomed. Signal Process. Control*, vol. 45, pp. 182–191, 2018.

[43] S. Arora and P. Anand, “Binary Butterfly Optimization Approaches for Feature Selection,” *Expert Syst. Appl.*, vol. 116, pp. 147–160, 2019.

[44] H. Majidpour and F. Soleimanian Gharehchopogh, “An Improved Flower Pollination Algorithm with AdaBoost Algorithm for Feature Selection in Text Documents Classification,” *J. Adv. Comput. Res.*, vol. 9, no. 1, pp. 29–40, 2018.

[45] X. Yang and L. Press, “Nature-Inspired Metaheuristic Algorithm,” *2nd Edition, Luniver press*, 2010.

[46] X.-S. Yang, “Review of Metaheuristics and Generalized Evolutionary Walk Algorithm,” *IJBIC*, vol. 3, no. 2, pp. 77–84, 2011.

[47] X. S. Yang, “Firefly Algorithms for Multimodal Optimization,” In *International Symposium on Stochastic Algorithms*, pp. 169-178, Springer, 2009.

[48] M. Dorigo and T. Stützle, “The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances." In *Handbook of Metaheuristics*, pp. 250-285. Springer, 2003.

[49] S. Nakrani and C. Tovey, “On Honey Bees and Dynamic Server Allocation in Internet Hosting Centers,” *Adapt. Behav.*, vol. 12, no. 3–4, pp. 223–240, 2004.

[50] Y. S. Kim, W. N. Street, and F. Menczer, “Feature Selection in Unsupervised Learning via Evolutionary Search,” *Proceeding Sixth ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 365–369, 2000.

[51] X. S. Yang, “Flower Pollination Algorithm for Global Optimization,” In *International Conference on Unconventional Computing and Natural Computation*, pp. 240–249, Springer, 2012.

[52] Z. Tian, S. Fong, R. Tang, S. Deb, and R. Wong, “Rhinoceros Search Algorithm,” *Proc. - 2016 3rd Int. Conf. Soft Comput. Mach. Intell. ISCMI 2016,* pp. 18–22, 2017.

[53] R. Tang, S. Fong, X. S. Yang, and S. Deb, “Wolf Search Algorithm with Ephemeral Memory,” *7th Int. Conf. Digit. Inf. Manag. ICDIM 2012*, pp. 165–172, 2012.

[54] S. Deb, S. Fong, and Z. Tian, “Elephant Search Algorithm for Optimization Problems,” *10th Int. Conf. Digit. Inf. Manag. ICDIM 2015*, pp. 249–255, 2016.

[55] X. S. Yang and S. Deb, “Cuckoo Search: Recent Advances and Applications,” *Neural Comput. Appl.*, vol. 24, no. 1, pp. 169–174, 2014.

[56] Bansal, Jagdish Chand, Pramod Kumar Singh, and Nikhil R. Pal, eds, “Evolutionary and Swarm Intelligence Algorithms,” Springer, 2019.

[57] X. S. Yang and X. He, “Firefly Algorithm: Recent Advances and Applications,” *Int. J. Swarm Intell.*, vol. 1, no. 1, p. 36, 2013.

[58] K. Spärck Jones, “A Statistical Interpretation of Term Specificity and Its Application in Retrieval,” *Journal of Documentation*, vol. 60, no. 5. pp. 493–502, 2004.

[59] J. H. Correia, R. Wille, G. Stumme, and U. Wille, “Conceptual Knowledge Discovery a Human Centered Approach,” *Appl. Artif. Intell.*, vol. 17, no. 3, pp. 281–302, 2003.

[60] L. S. Shafti and E. Pérez, “Feature Construction and Feature Selection in Presence of Attribute Interactions,” In *International Conference on Hybrid Artificial Intelligence Systems*, pp. 589–596, 2009.

[61] Hall, M.A., “Corrrelation-based Feature Selection for Machine Learning,” 1999.

[62] G. H. John and P. Langley, “Estimating Continuous Distributions in Bayesian Classifiers,” *Proc. Elev. Conf. Uncertain. Artif. Intell.*, pp. 338–345, 1995.

[63] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy, “Improvements to Platt’s SMO Algorithm for SVM Classifier Design,” *Neural Comput.*, vol. 13, no. 3, pp. 637–649, 2001.

[64] Quinlan, J. R., “C 4.5: Programs for Machine Learning”, Elsevier, 2014.

[65] N. J.Chatap and A. Kr. Shrivastava, “A Survey on Various Classification Techniques for Medical Image Data,” *Int. J. Comput. Appl.*, vol. 97, no. 15, pp. 1–5, 2014.

[66] D. Greene and P. Cunningham, “Practical Solutions to the Problem of Diagonal Dominance in Kernel Document Clustering,”In *Proceedings of the 23rd International Conferene on Machine Learning*, pp. 377–384, 2006.

[67] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey Wolf Optimizer,” *Adv. Eng. Softw.*, vol. 69, pp. 46–61, 2014.

[68] H. Search, “A New Metaheuristic Optimization Algorithm: Harmony Search, ” *Transactions of the Society for Modeling and Simulation International*, vol. 76, no. 2, pp. 60-68, 2011.

[69] X. S. Yang, M. Karamanoglu, and X. He, “Multi-Objective Flower Algorithm for Optimization,” *Procedia Comput. Sci.*, vol. 18, pp. 861–868, 2013.

[70] X.-S. Yang and Suash Deb, “Cuckoo Search via Levy Flights,” 2*009 World Congr. Nat. Biol. Inspired Comput.*, pp. 210–214, 2009.

[71] D. T. Pham and M. Castellani, “The Bees Algorithm: Modelling Foraging Behaviour to solve Continuous Optimization Problems,” *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.*, vol. 223, no. 12, pp. 2919–2938, 2009.

[72] X. S. Yang, “A New Metaheuristic Bat-inspired Algorithm,” *Stud. Comput. Intell.*, vol. 284, pp. 65–74, 2010.

[73] M. Dorigo, V. Maniezzo, and A. Colorni, “Ant System: Optimization by a Colony of Cooperating Agents,” *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, vol. 26, no. 1, pp. 29–41, 1996.

[74] Anderson, J. A., “An Introduction to Neural Networks, ” MIT Press, 1995.

[75] J. A. K. Ranjan, T. Sigamani, and J. Barnabas, “A Novel and Efficient Classifier using Spiking Neural Network,” *J. Supercomput*., pp. 1-16, 2019.

APPENDIX A

LIST OF PUBLICATIONS

This thesis is based on the conference papers and ISI journals which are listed below:

A. Published ISI Journals and Conference Papers

A.1 Conference Proceedings

A.1.1 Khin Sandar Kyaw and Somchai Limsiroratana, "Traditional and Swarm Intelligent Based Text Feature Selection for Document Classification", in *proceedings of 19th International Symposium on Communications and Information Technologies (ISCIT)*, pp. 226-231, 2019.

A.1.2 Khin Sandar Kyaw and Somchai Limsiroratana, "Towards Nature-Inspired Intelligence Search for Optimization of Multi-Dimensional Feature Selection", in *proceeding of 23rd International Computer Science and Engineering Conference (ICSEC)*, 2019.

A.1.3 Khin Sandar Kyaw and Somchai Limsiroratana, "Case Study: Knowledge Discovery Process using Computation Intelligence with Feature Selection Approach", in *proceeding of 17th International Conference on ICT and Knowledge Engineering (ICT and Knowledge Engineering)*, 2019.

A.2 Journals

A.2.1 Khin Sandar Kyaw and Somchai Limsiroratana, "Optimization of Text Feature Selection Process based on Advanced Searching for News Classification", *International Journal of Swarm Intelligence Research (IJSIR)*, vol.11, Issue 4, no.1, 2019. [Published]

A.2.2 Khin Sandar Kyaw and Somchai Limsiroratana, "Optimization of Multi-Class Document Classification with Nature-Inspired Intelligence Search", ECTI Transactions on Computer and Information Technology, Scopus. [Under review]

B. Published Textbook Chapter

1. **Khin Sandar Kyaw** and Somchai Limsiroratana, "Optimization of Multi-Dimensional Feature Selection based on Computation Intelligence", *Advancements of Swarm Intelligence Algorithms for Solving Real-World Problems*, 2020. IGI Publication House. [Accepted]

C. Under Review ISI Manuscripts

1. **Khin Sandar Kyaw** and Somchai Limsiroratana, "A Comparative Study of Swarm and Nature-inspired Intelligence Search in Optimization of Multi-Dimensional Feature Selection for Document Classification", *International Journal of Computational Intelligence and Applications.* [Submitted]
2. **Khin Sandar Kyaw** and Somchai Limsiroratana, "A Review: Meta-heuristic based Multi-Dimensional Feature Selection for Web Mining", *International Journal of Data Mining, Modelling and Management.* [Manuscript]
3. **Khin Sandar Kyaw** and Somchai Limsiroratana, "A Survey: Meta-heuristic based Deep Learning for Text Feature Selection", *International Journal of Knowledge-Based and Intelligent Engineering Systems*. [Manuscript]
4. **Khin Sandar Kyaw** and Somchai Limsiroratana, "Optimization of Hybrid N-Gram Feature Selection using Nature-inspired Search Policy", *Journal of Optimization Theory and Application*. [Manuscript]

APPENDIX B

VITAE

Name Ms. Khin Sandar Kyaw

Student ID 5910130037

Education Attainment

Degree	Name of Institution	Year of Graduation
AGTI (Information Technology)	Yangon Technological University (YTU)	2008
Bachelor of Engineering (Information Technology)	Yangon Technological University (YTU)	2011
Master of Engineering (Computer Engineering and Information Technology)	Yangon Technological University (YTU)	2014

Scholarship Awards during Enrolment

Higher Education Research Promotion and the Thailand's Education Hub for Southern Region of ASEAN Countries Project Office of the Higher Education Commission.

List of Publications and Proceedings

1. **Khin Sandar Kyaw** and Somchai Limsiroratana, "Traditional and Swarm Intelligent Based Text Feature Selection for Document Classification", in *proceedings of 19th International Symposium on Communications and Information Technologies (ISCIT)*, pp. 226-231, 2019.
2. **Khin Sandar Kyaw** and Somchai Limsiroratana, "Towards Nature-Inspired Intelligence Search for Optimization of Multi-Dimensional Feature Selection", in *proceeding of 23rd International Computer Science and Engineering Conference (ICSEC)*, 2019.
3. **Khin Sandar Kyaw** and Somchai Limsiroratana, "Case Study: Knowledge Discovery Process using Computation Intelligence with Feature Selection

Approach", in *proceeding of 17th International Conference on ICT and Knowledge Engineering (ICT and Knowledge Engineering)*, 2019.

4. **Khin Sandar Kyaw** and Somchai Limsiroratana, "Optimization of Text Feature Selection Process based on Advanced Searching for News Classification", *International Journal of Swarm Intelligence Research (IJSIR)*, vol.11, Issue 4, no.1, 2019.
5. **Khin Sandar Kyaw** and Somchai Limsiroratana, "Optimization of Multi-Class Document Classification with Nature-Inspired Intelligence Search", *ECTI Transactions on Computer and Information Technology*, 2020 (May). [under review]