EVOLVING MACHINE LEARNING AND DEEP LEARNING MODELS USING EVOLUTIONARY ALGORITHMS

HAILUN XIE

PhD

2020

EVOLVING MACHINE LEARNING AND DEEP LEARNING MODELS USING EVOLUTIONARY ALGORITHMS

HAILUN XIE

A thesis submitted in partial fulfilment of the requirements of the University of Northumbria at Newcastle for the degree of

Doctor of Philosophy

Research undertaken in the Faculty of Engineering and Environment

December 2020

Abstract

Despite the great success in data mining, machine learning and deep learning models are yet subject to material obstacles when tackling real-life challenges, such as feature selection, initialization sensitivity, as well as hyperparameter optimization. The prevalence of these obstacles has severely constrained conventional machine learning and deep learning methods from fulfilling their potentials. In this research, three evolving machine learning and one evolving deep learning models are proposed to eliminate above bottlenecks, i.e. improving model initialization, enhancing feature representation, as well as optimizing model configuration, respectively, through hybridization between the advanced evolutionary algorithms and the conventional ML and DL methods.

Specifically, two Firefly Algorithm based evolutionary clustering models are proposed to optimize cluster centroids in K-means and overcome initialization sensitivity as well as local stagnation. Secondly, a Particle Swarm Optimization based evolving feature selection model is developed for automatic identification of the most effective feature subset and reduction of feature dimensionality for tackling classification problems. Lastly, a Grey Wolf Optimizer based evolving Convolutional Neural Network-Long Short-Term Memory method is devised for automatic generation of the optimal topological and learning configurations for Convolutional Neural Network-Long Short-Term Memory networks to undertake multivariate time series prediction problems.

Moreover, a variety of tailored search strategies are proposed to eliminate the intrinsic limitations embedded in the search mechanisms of the three employed evolutionary algorithms, i.e. the dictation of the global best signal in Particle Swarm Optimization, the constraint of the diagonal movement in Firefly Algorithm, as well as the acute contraction of search territory in Grey Wolf Optimizer, respectively. The remedy strategies include the diversification of guiding signals, the adaptive nonlinear search parameters, the hybrid position updating mechanisms, as well as the enhancement of population leaders. As such, the enhanced Particle Swarm Optimization, Firefly Algorithm, and Grey Wolf Optimizer variants are more likely to attain global optimality on complex search landscapes embedded in data mining problems, owing to the elevated search diversity as well as the achievement of advanced trade-offs between exploration and exploitation.

The proposed evolving K-means clustering, evolving feature selection, and evolving Convolutional Neural Network-Long Short-Term Memory models are evaluated using a variety of real-life clustering, classification, and time series forecasting problems, respectively. The empirical results indicate that the proposed evolving machine learning and deep learning methods obtain significantly superior performances on majority of the employed data mining tasks and demonstrate great effectiveness in eliminating the sensitivity of centroid initialization in K-means, determining the most effective feature subset, as well as identifying the optimal learning and topological configurations for Convolutional Neural Network-Long Short-Term Memory networks, respectively. The above advantages of the proposed evolving models over baseline methods are further ascertained by the statistical test results.

List of Publications

Published peer-reviewed papers:

- **H. Xie**, L. Zhang, C.P. Lim, Y. Yu, C. Liu, H. Liu, and J. Walters, "Improving K-means clustering with enhanced Firefly Algorithms," *Applied Soft Computing*, vol. 84, Article 105763, 2019.

- **H. Xie**, L. Zhang, and C.P. Lim, "Evolving CNN-LSTM Models for Time Series Prediction Using Enhanced Grey Wolf Optimizer," *IEEE Access*, vol. 8, pp. 161519-161541, 2020.

- **H. Xie**, S. Wei, L. Zhang, B.M. Ng, and S. Pan, "Using feature selection techniques to determine the best feature subset in prediction of window behaviour," In *Proceedings of 10th Windsor Conference on Rethinking Comfort*, pp. 315-328, 2018.

Contents

Chapter 1 Introduction 1

1.1 Background 1

1.1.1 Introduction of machine learning 1

1.1.2 Bottlenecks of machine learning 2

1.2 Motivation 5

1.3 Research aims and objectives 7

1.4 Contribution 8

1.5 Thesis layout 11

Chapter 2 Preliminaries and Literature Review 13

2.1 Evolutionary computation 13

2.1.1 Particle Swarm Optimization 14

2.1.2 Firefly Algorithm 17

2.1.3 Grey Wolf Optimizer 22

2.1.4 Other latest evolutionary algorithms 28

2.2 Clustering analysis 30

2.2.1 K-means clustering 30

2.2.2 Fuzzy C-means clustering 32

2.2.3 Evolving K-means clustering 33

2.3 Feature selection and classification 35

2.3.1 Classification 36

2.3.2 Feature selection 37

2.3.3 Evolutionary feature selection methods 38

2.4 Deep neural networks 42

2.4.1 Convolutional Neural Networks 42

2.4.2 Long Short-Term Memory 44

2.4.3 Convolutional Neural Network-Long Short-Term Memory 45
2.4.4 Evolving deep neural networks 46
2.5 Summary 50
Chapter 3 Evolutionary K-means Clustering with Enhanced Firefly Algorithms 51
3.1 The proposed evolutionary K-means clustering models 51
3.1.1 The proposed inward intensified exploration FA (IIEFA) model 52
3.1.2 The proposed compound intensified exploration FA (CIEFA) model 54
3.1.3 The proposed clustering approach based on the IIEFA and CIEFA models 59
3.2 Evaluation and discussion 62
3.2.1 Parameter settings 63
3.2.2 Data sets 64
3.2.3 Performance comparison metrics 65
3.2.4 Feature selection and clustering performance evaluation 67
3.2.5 Performance comparison and analysis 72
3.2.6 Statistical tests 78
3.3 Evaluation on high-dimensional clustering tasks with complex cluster distributions 82
3.4 Further comparison and analysis between IIEFA and CIEFA 85
3.5 Summary 87
Chapter 4 Evolutionary Feature Selection Using Enhanced Particle Swarm Optimisation 89
4.1 The proposed evolutionary feature selection model 90
4.1.1 The proposed enhanced PSO model 90
4.1.2 The proposed evolutionary feature selection model based on the enhanced PSO variant 100
4.2 Evaluation and discussion 101

4.2.1 Data sets 102
4.2.2 Parameter settings 103
4.2.3 Results and discussion 104
4.3 Summary 114
Chapter 5 Evolving CNN-LSTM Models for Time Series Prediction Using Enhanced Grey Wolf Optimizer 116
5.1 The proposed evolving time series prediction model 117
5.1.1 The proposed GWO variant 119
5.1.2 The proposed CNN-LSTM architecture 127
5.1.3 The proposed GWO-based evolving CNN-LSTM network 128
5.2 Evaluation and discussion 129
5.2.1 Energy consumption forecast 130
5.2.2 PM2.5 concentration prediction 134
5.2.3 Human activity recognition 137
5.2.4 Remarks 141
5.2.5 Wilcoxon statistical test 142
5.3 Summary 143
Chapter 6 Conclusions 145
6.1 Summary of the contribution 145
6.2 Future work 148
References 151

List of Figures

Figure 2-1 The movement of fireflies in a two-dimensional search space (Δp denotes the position difference between fireflies i and j) 19
Figure 2-2 Exploitation (|A|<1) vs. Exploration (|A|>1) in GWO 24
Figure 2-3 The illustration of the architecture of a vanilla CNN 43
Figure 2-4 The illustration of a building block in LSTM 45
Figure 2-5 The illustration of the architecture of CNN-LSTM 46
Figure 3-1 An example of the change of one element from the step control matrix, τ, through iterations 56
Figure 3-2 Distribution of the updated positions of firefly i through iterations in the CIEFA model in a two-dimensional search space when $Mdissimilarity < 0.5$ 58
Figure 3-3 Flowchart of the proposed clustering method 61
Figure 4-1 Flowchart of the proposed PSO variant 98
Figure 5-1 The diagram of the proposed GWO-based evolving CNN-LSTM time series forecasting model where each wolf represents a set of network topology and learning hyperparameters for evolution 118
Figure 5-2 The proposed nonlinear a' vs. linear a in the original GWO 120
Figure 5-3 The sinusoidal chaotic map used for generating the leadership factors of the most dominating wolf α 122
Figure 5-4 The proposed damped function in **Eq. 5.10** when $f = 1$ 124
Figure 5-5 The comparison between proposed damped function and the damped function applied in MFO 125
Figure 5-6 The topology of the proposed CNN-LSTM architecture 127

List of Tables

Table 3-1 Parameters settings for each algorithm 63
Table 3-2 Ten selected data sets for evaluation 65
Table 3-3 The mean clustering results over 30 independent runs on the ALL data set . 68
Table 3-4 The mean clustering results over 30 independent runs on the Sonar data set 68
Table 3-5 The mean clustering results over 30 independent runs on the Ozone data set 69
Table 3-6 The mean clustering results over 30 independent runs on the Thyroid data set 69
Table 3-7 The mean clustering results over 30 independent runs on the Balance data set 70
Table 3-8 The mean clustering results over 30 independent runs on the E.coli data set 70
Table 3-9 The mean clustering results over 30 independent runs on the Wbc1 data set 70
Table 3-10 The mean clustering results over 30 independent runs on the Wbc2 data set 71
Table 3-11 The mean clustering results over 30 independent runs on the Wine data set 71
Table 3-12 The mean clustering results over 30 independent runs on the Iris data set .. 72
Table 3-13 The mean results of the minimum intra-cluster distance measure over 30 runs 72
Table 3-14 The mean results of average accuracy after feature selection over 30 runs . 73
Table 3-15 The mean results of FscoreM after feature selection over 30 runs 74
Table 3-16 The mean results of average sensitivity after feature selection over 30 runs 75
Table 3-17 The mean results of average specificity after feature selection over 30 runs 75
Table 3-18 The mean ranking results based on the Friedman test for the CIEFA model 78
Table 3-19 The mean ranking results based on the Friedman test for the IIEFA model 79
Table 3-20 Statistical results of the Friedman test for the CIEFA model 79
Table 3-21 Statistical results of the Friedman test for the IIEFA model 79
Table 3-22 The Wilcoxon rank sum test results of the proposed CIEFA model 81
Table 3-23 The Wilcoxon rank sum test results of the proposed IIEFA model 81

Table 3-24 Three high-dimensional data sets with multiple classes for further evaluation 82
Table 3-25 The mean results of the minimum intra-cluster distance measure on high-dimensional data sets over 30 runs 83
Table 3-26 The mean results of average accuracy on high-dimensional data sets over 30 runs 83
Table 3-27 The mean results of FscoreM on high-dimensional data sets over 30 runs 83
Table 3-28 The Wilcoxon rank sum test results of the proposed CIEFA model on high-dimensional data sets 84
Table 3-29 The Wilcoxon rank sum test results of the proposed IIEFA model on high-dimensional data sets 84
Table 3-30 Four additional high-dimensional data sets for further comparison between IIEFA and CIEFA 86
Table 3-31 The mean clustering results over 30 independent runs with four high-dimensional data sets 86
Table 4-1 Ten selected data sets for evaluation 103
Table 4-2 Parameter settings of each algorithm 103
Table 4-3 The mean results of the classification accuracy over 30 runs 105
Table 4-4 The mean results of the F-score over 30 runs 108
Table 4-5 The mean results of the number of selected features over 30 runs 109
Table 4-6 The Wilcoxon rank sum test results of the proposed PSO model 111
Table 5-1 The search range of the hyperparameters 129
Table 5-2 Parameter settings of search methods 130
Table 5-3 The RMSE results over 10 independent runs 131
Table 5-4 The MAE results over 10 independent runs 132
Table 5-5 The mean configurations of the identified CNN-LSTM networks over 10 runs 133
Table 5-6 The RMSE results over 10 independent runs 135
Table 5-7 The MAE results over 10 independent runs 135
Table 5-8 The mean configurations of the identified CNN-LSTM networks over 10 runs 136
Table 5-9 The results of classification accuracy over 10 independent runs 138
Table 5-10 The results of F-score over 10 independent runs 138

Table 5-11 The results of precision over 10 independent runs 138
Table 5-12 The results of Recall over 10 independent runs 139
Table 5-13 The mean accuracy rate of each class over 10 independent runs 140
Table 5-14 The mean configurations of the identified CNN-LSTM networks over 10 runs 140
Table 5-15 Wilcoxon rank sum test results over 10 independent run 142

Acronyms

AI	Artificial Intelligence
ABC	Artificial Bee Colony
ACO	Ant Colony Optimization
ANNs	Artificial Neural Networks
AutoML	Automated Machine Learning
BA	Bat Algorithm
BHA	Black Hole Algorithm
BB-BC	Big Bang-Big Crunch algorithm
BPSO	Binary Particle Swarm Optimization
CS	Cuckoo Search
CSO	Competitive Swarm Optimiser
CatfishBPSO	Binary Particle Swarm Optimization with Catfish Effect
CFA1	Chaotic Firefly Algorithm with Logistic Map
CFA2	Chaotic Firefly Algorithm with Gauss Map
CIEFA	Compound Intensified Exploration Firefly Algorithm
CNNs	Convolutional Neural Networks
CV	Computer Vision
CNN-LSTM	Convolutional Neural Network-Long Short-Term Memory
DE	Differential Evolution
DA	Dragonfly Algorithm
DL	Deep Learning
DT	Decision Tree
DNNs	Deep Neural Networks
EAs	Evolutionary Algorithms
EC	Evolutionary Computation
EM	Expectation-Maximization

FA	Firefly Algorithm
FPA	Flower Pollination Algorithm
FuzzyGWO	Grey Wolf Optimizer with Fuzzy Hierarchical Operator
FNNs	Feedforward Neural Networks
GA	Genetic Algorithm
GWO	Grey Wolf Optimizer
GSA	Gravitational Search Algorithm
GMM	Gaussian Mixture Model
GANs	Generative Adversarial Networks
HPSO-SSM	Hybrid PSO with Spiral-Shaped Mechanism
HAR	Human Activity Recognition
IIEFA	Inward Intensified Exploration Firefly Algorithm
KM	K-means
KNN	K-Nearest Neighbours
LR	Learning Rate
ML	Machine Learning
MA	Memetic Algorithm
MFO	Moth-Flame Optimisation
MFA	Modified Firefly Algorithm
MBPSO	Modified Binary Particle Swarm Optimization
MAE	Mean Absolute Error
mRMR	minimum Redundancy Maximum Relevance
NaFA	Firefly Algorithm with Neighbourhood Attraction
NLP	Natural Language Processing

PSO	Particle Swarm Optimization
PrLeGWO	Grey Wolf Optimizer with Prioritized Movement Among Dominant Wolves and Adaptive Learning Mechanism
RF	Random Forest
ResNet	Residual Neural Network
RNNs	Recurrent Neural Networks
R-CNN	Region with Convolutional Neural Network
RMSE	Root Mean Square Error
SA	Simulated Annealing
SCA	Sine Cosine Algorithm
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
SI	Swarm Intelligence
TS	Tabu Search
VSSFA	Firefly Algorithm with Variable Step Size
VGG	Very deep Convolutional Networks
WOA	Whale Optimisation Algorithm
YOLO	You Only Live Once

Acknowledgements

During the period of my study, I have encountered some wonderful people who have provided me with enormous supports and encouragements, both academically and psychologically. Without their help, I would not have been able to journey this far and finish this study. Therefore, I have tons of appreciations that I would like to express, to these amazing people.

Firstly, I would like to start by acknowledging my exceptional supervisor, **Li Zhang**. Li is the most important reason for my completion of this PhD study. She taught me everything from scratch with enormous patience, such as how to develop preliminary research ideas, how to deploy experiments, and how to present research findings professionally. I have learned many lessons from Li about the essential qualities critical to become a professional researcher, such as dedication and passion to research, thirst for knowledge, as well as the attitude of being excelsior and rigorous, etc. It is an absolute blessing to have Li as my principal supervisor and I will remain eternally thankful to her.

Moreover, I would also like to express greatest gratitude to my previous supervisors, **Shen Wei** and **Bobo Ng**. Shen put faith in me and chose me out of many candidates to undertake the original project of occupant behaviours. For this reason, I always feel grateful to him. Bobo offered great guidance on framing the structure and methodology of the original project. She also provided enormous support for my transition from mechanical engineering to computer science.

Furthermore, I would also like to express my great gratitude to **Northumbria University** and the **Graduate School**. They have offered generous financial support throughout the whole duration of my study. Therefore, I have not been disadvantaged financially by the change of research topics, nor by the pandemic. They have always responded very quickly to address my enquiries and concerns. I absolutely have a wonderful and unforgettable learning experience at Northumbria University.

In addition, I would like to express my infinite gratitude to my dear friend, **Mark Keville**. He has been trying to look after me and help me to overcome the culture shock. We have embarked an unforgettable journey of exploring local villages in north east. I

believe all these tranquil moments and scenic views have already been inscribed into our memories and documented in those post cards, so has our friendship.

Another important friend I need to acknowledge is **Xiaohong Chen**. Over the last four years, we had many fierce discussions on a great variety of subjects. Although we did not always see eye to eye, these meaningful discussions surely have made me realize my own shallowness and limits. It reminds me to stay humble and open-minded.

Meanwhile, I also feel grateful to my friends and families in China, for standing by my side through thick and thin. The distance has never alienated us.

Last, but certainly not the least, I would like to express my indefinite gratitude to my parents, especially to my mom, **Shuling Zhan**. Everything I have achieved so far in my life would not have been possible if without her unconditional love.

Declaration

I declare that the work contained in this thesis has not been submitted for any other award and that it is all my own work. I also confirm that this work fully acknowledges opinions, ideas and contributions from the work of others.

Any ethical clearance for the research presented in this thesis has been approved. Approval has been sought and granted by the Faculty Ethics Committee on 14/08/2019.

I declare that the Word Count of this thesis is 51, 097 words.

Name: Hailun Xie

Signature: 

Date: 05.12.2020

Chapter 1
Introduction

Machine Learning (ML) algorithms are not omnipotent when confronted with real-life complex challenges. Their performances are significantly influenced by a variety of nondeterministic factors, such as the validity of raw data sets available for mining, the quality of initialized seeds of model parameters, as well as the suitability of predefined settings of model configurations. As a result, some of the most arduous challenges in ML arise from the above uncertainties, e.g. feature selection, initialization sensitivity, and hyperparameter optimization. It requires not only profound understandings about the investigated problems, but also considerable efforts with the optimization of model parameters and configurations, to overcome above obstacles in data mining and fully unleash the potential of ML algorithms. Moreover, the above process is unlikely to be accomplished manually or in deterministic manners, owing to the complexities of real-life problems as well as the stochastic properties of ML models. As such, this research aims to devise distinctive automatic learning processes with minimum human intervention to: 1) enhance feature representation, 2) improve model initialization condition, 3) optimize model hyperparameters, respectively, through the leverage of advanced evolving search capabilities of evolutionary algorithms (EAs).

1.1 Background

1.1.1 Introduction of machine learning

Learning is a many-faceted phenomenon which includes the acquisition of new declarative knowledge, the development of motor and cognitive skills through instruction or practice, the organization of new knowledge into general and effective representations, as well as the discovery of new facts and theories through observation and experimentation [1]. Since the inception of the computer era, great efforts have been made to implant such intelligence of learning into computers, empowering machines to perform various tasks that require thinking, reasoning, and decision making, in order to maximize the probability of achieving a specific goal. Hence, it gave rise to a novel research discipline, i.e. Artificial Intelligence (AI), which contains diverse research topics, such as ML, Evolutionary Computation (EC), Fuzzy Logic, Probabilistic Modelling [2].

ML is the study of computer algorithms that improve automatically through experience [3]. It is a branch of AI based on the idea that systems can learn from historical experiences, identify patterns, and make decisions with minimal human interventions. ML approaches can be broadly classified into three categories, i.e. supervised learning, unsupervised learning and reinforcement learning [4]. The aim of supervised learning is to learn the mapping from the input features to the output labels, whereas in unsupervised learning the aim is to discover the regularities and structures embedded in the input features without assistance of label information. In addition, reinforcement learning is a way of programming learning agents by reward and punishment to maximize the cumulative reward without requirement of specifying how the task is to be accomplished [5]. ML algorithms have been applied to tackle a wide range of practical problems, e.g. computer vision (CV) [6], natural language processing (NLP) [7], robot control [8], computer-aided diagnosis [9], recommender systems [10], automated planning and scheduling [11]. ML has become the research hotspot owing to its advanced performances in a great variety of real-life applications as well as the recent significant breakthroughs in the development of deep neural networks (DNNs), e.g. Convolutional Neural Networks (CNNs) [12], Generative Adversarial Networks (GANs) [13].

1.1.2 Bottlenecks of machine learning

Despite the great efficacy of ML in data mining, several bottlenecks still exist with respect to feature engineering, model initialization, as well as configuration identification. They have severely confined ML models from fulfilling their potentials and achieving optimal performances when tackling real-life problems. The mitigation of above ML obstacles entails significant knowledge barriers owing to the requirement of diverse expertise with respect to the investigated problem domains, the ML algorithms, as well as the advanced optimization techniques. These bottlenecks are introduced and discussed in detail as follows.

1.1.2.1 Feature selection

The digital revolution has prompted data explosion in many fields, such as social media, security, and business [14]. The term of “Big Data” is invented to describe the resulted large amount of data, which is characterized by its volume, velocity, variety, value as well as veracity [15]. Data mining in practice often involves with Big Data which

possesses a great variety of complexities, such as high dimensionalities, noises and poor qualities, redundant features, unstructured formats, as well as imbalanced distributions. These challenging factors could undermine the validity of raw data sets, and result in inauthentic representations of the investigated problems, therefore compromising the process of knowledge discovery. In order to overcome above challenges, diverse feature engineering techniques have been developed to transform raw data sets into enhanced feature representations of the underlying problems, upon which the generalization capability of ML models can be significantly boosted [16].

As one of the most important feature engineering techniques, feature selection has gained many research attentions [17]. It reduces feature dimensionality and facilitates the performance of ML models effectively by selecting a subset of the most relevant and informative features from the original feature space. To be specific, the high dimensionality of the raw data set increases the likelihood of containing redundant, irrelevant, and contradictory features. The presence of them severely undermines the knowledge discovery process owing to the complex interactions among input features as well as the spurious representations of the investigated problems. Moreover, the high dimensionality is also likely to incur "curse of dimensionality", which is characterized by the increased sparsity of data instances owing to the rapid expansion of feature space [18]. As a result, the amount of data required for successful knowledge discovery grows exponentially as the feature dimensionality increases. As such, dimensionality reduction through feature selection plays a significant role in both enhancing the validity of feature representations and overcoming "curse of dimensionality". Despite being considered as one of the most challenging and time-consuming tasks, feature selection has become an indispensable component in data mining and knowledge discovery [19, 20].

1.1.2.2 Initialization sensitivity

The second bottleneck in ML is initialization sensitivity, which refers to the sensitivity of model performance to initialized conditions. This phenomenon can be widely observed from iterative refinement clustering algorithms, such as Gaussian Mixture Model (GMM) [21], and K-means (KM) [22], in which a deterministic mapping mechanism is defined to derive a fitted model from a randomly initialized one. Despite the employment of different clustering objectives, e.g. maximization of likelihood of sample distributions in GMM, and minimization of intra-cluster distances in KM,

clustering algorithms that employ iterative refinement mechanisms commonly subject to local stagnation owing to the lack of capability of conducting global search [23]. As a consequence, their performances largely depend on initialized settings of model parameters. In addition to unsupervised learning, the problem of initialization sensitivity is also present in supervised learning, such as feedforward neural networks (FNNs) [24, 25], and DNNs [26].

More specifically, optimization lies at the heart of ML and the training of ML models can be reduced to a core optimization problem in which internal model parameters are optimized with respect to the defined loss function [27]. The above optimization process can be extremely challenging owing to the nonconvex complex search landscapes, the obstruction of saddle points and local optima traps, as well as the immense search dimensionalities. Therefore, a variety of iterative search methods have been developed for the identification of the optimal learnable parameters for ML models, such as Expectation-Maximization (EM) algorithm, and Stochastic Gradient Descent (SGD) [28]. Essentially, these iterative methods can be characterised as a greedy local search operation which starts from a randomly initialized position within search bound. Despite the advantages in convergence, these iterative methods are prone to local stagnation, especially when the initialized position is in the vicinity of local optima traps. As a result, model parameters can be poorly fitted and model performance compromised. Therefore, the quality of initialized settings of model parameters plays a significant role in determining training efficiency as well as learning efficacy for ML models [29-31]. In this research, initialization sensitivity in KM clustering, i.e. the susceptibility of KM algorithm to its initialized cluster centroids, is specifically targeted owing to the prevalence of the problem [22], as well as the popularity of KM clustering in the domain of unsupervised learning [32].

1.1.2.3 Hyperparameter optimization

In addition to feature selection and initialization sensitivity, hyperparameter optimization is another significant barrier in data mining [33]. There are generally two types of parameters in ML models. The first is those internal to learning algorithms, such as network weights and biases in artificial neural networks (ANNs). They are estimated automatically through model fitting process on training sets. The second is those external to learning algorithms, namely hyperparameters. They prescribe configurations as well as leaning properties of ML models, such as the number of layers

in ANNs, and then number of nearest neighbours in K-Nearest Neighbours (KNN). Unlike internal parameters, hyperparameters must be defined prior to the training stage and cannot be optimized during the fitting process [34]. Hence, the choice of hyperparameters plays a significant role in the performance of ML models [35]. Hyperparameter optimization is a critical component of developing effective ML models which are capable of accommodating specific characteristics of the problem at hand. Its potent has been extensively verified by existing studies across a great variety of ML algorithms, e.g. Random Forest (RF) [36], Support Vector Machine (SVM) [37], and ANNs [38]. The empirical study even discovered that hyperparameter optimization is often more important than the choice of ML algorithms in data mining [39].

Moreover, hyperparameter optimization has become even more important in the domain of deep learning (DL) [40, 41]. Specifically, the great success of DNNs on a variety of applications can be primarily ascribed to the nature of being deep. It enables networks to extract and learn meaningful feature representations from raw data sets automatically without manual engineering. Hyperparameters in DNNs need to be tuned carefully since they are involved in determining topologies of networks, e.g. the total number of convolutional layers, the number and size of filters contained in each convolutional layer. Besides above, certain hyperparameters related to training properties also bear responsibilities for learning efficiency as well as learning outcome, e.g. the learning rate (LR), batch size, and optimizer type [42]. As the depth increases in DNNs, the number of hyperparameters grows exponentially in networks [43]. As a result, it becomes extremely challenging to identify the optimal topological and learning configurations for DNNs, owing to the huge amount of possibilities in terms of hyperparameter combinations as well as the sophisticated interactions and cascade effects among different components in DNNs.

1.2 Motivation

The above three major bottlenecks in ML, i.e. feature selection, initialization sensitivity, as well as hyperparameter optimization, have significantly confined ML and DL models from fulfilling their potentials when undertaking real-life challenges. It is unlikely to eliminate them in simple deterministic manners or by manual efforts, owing to the complexities of the problem, e.g. stochastic nature of ML algorithms as well as the sophisticated interplay among diverse variables. Despite the distinctiveness in their

respective scenarios, the above three obstacles in ML are identical in their natures. Intrinsically, they can all be treated as complex non-convex optimization problems with high-dimensional search landscapes, teeming with local optima traps, saddle points, and deceptive slops [28]. In this regard, the elimination of above three major obstacles in ML inevitably requires powerful search mechanisms, which are capable of escaping from stagnation at local optima traps effectively and attaining the global optimality efficiently in a complex high-dimensional search space.

As such, in this research I resort to advanced EAs to address above three challenges in ML and data mining, owing to their flexibility in variable encoding and problem representation, superiority in escaping from local optima traps, as well as self-adaptability for the attainment of the global optimality [44, 45]. EAs represent a family of population-based metaheuristic optimization algorithms in EC. EAs can be generally categorized into two main classes, according to the difference of their underlying mechanisms, i.e. biological evolution and swarm behaviours [46, 47]. Specifically, algorithms in the first class search for global optimality by following Darwin's theory of evolution, i.e. reproduction, mutation, recombination, and selection, whereas those in the second class mimic the collective behaviours of the organized group of animals in nature, e.g. fish schooling and wolf hunting, and seek for global optimality by employing a population of simple agents interacting locally with each other [48].

Despite the employment of distinctive search operations, all EAs employ the same two essential components in their search mechanisms, i.e. exploration and exploitation [49]. During exploration, individuals in the population explore the search space on a global scale by conducting large jumps and generating offspring solutions far from the parents with sufficient diversities. As a result, the exploration enables search agents to escape from local optima traps as well as increases the coverage of search territory, hence facilitating the attainment of the global optimality. On the other hand, it could also result in slow convergence and higher computational efforts since the yielded solutions can be very distant from the global optima. In contrast, exploitation enables agents to search on a local scale and focus on promising regions represented by the elicit solutions found so far. Therefore, the magnitude of changes among solutions during the exploitation stage is much smaller than that during the exploration stage. As opposed to exploration, exploitation prompts convergence of the population while suffering from local stagnation.

In essence, a fine trade-off between exploration and exploitation serves as the bedrock of the attainment of global optimality in EAs [50]. However, it still remains an open question regarding the realization of such balance between the above two distinctive search norms, owing to the complex nature of the problem. This balance could be affected by many factors, such as detailed search mechanisms and behaviours, settings and tunings of search parameters, during the optimization process. More importantly, there is no universal solution with respect to the achievement of the trade-off between exploration and exploitation and it could vary from problem to problem [51].

Hence, this reality serves as a major motivation in this research. Instead of directly applying EAs in their original forms, the inherent constraints with respect to search mechanisms and diversities embedded in the classical EAs are identified. Moreover, comprehensive remedy strategies are proposed to overcome above intrinsic limitations and achieve enhanced trade-offs between exploration and exploitation over the search process. These strategies include the rectified search operations, the diversified guiding signals, the modified settings and tunings of search parameters, as well as the tailored designs of hybrid search behaviours over the course of optimization. The proposed enhanced EAs are subsequently employed for the development of evolving ML and DL methods to tackle the abovementioned three bottlenecks in ML and data mining, i.e. feature selection, initialization sensitivity, and hyperparameter optimization, respectively. As such, the process of knowledge discovery and pattern recognition can be greatly facilitated owing to the identification of authentic feature representations as well as the devising of more effective ML and DL models with optimal configurations and improved learning parameters.

1.3 Research aims and objectives

This research aims to develop evolving ML and DL models to overcome the limitations of the conventional ML and DL methods and facilitate knowledge discovery when undertaking various real-life challenges, including classification, clustering, and time series prediction. The proposed evolving models target at three major obstacles in ML and data mining, i.e. feature selection, initialization sensitivity, as well as hyperparameter optimization, respectively. Three objectives are designed accordingly to deliver the overall research aim as follows:

1) To develop an evolutionary KM clustering model capable of overcoming the problem of sensitivity to initialized cluster centroids, for undertaking clustering challenges.
2) To design an evolutionary feature selection model capable of automatic identification of the most effective feature subset, for undertaking classification challenges.
3) To devise an evolving Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM) method capable of automatic generation of the customized CNN-LSTM networks with the optimal topological and learning configurations, for undertaking time series forecasting challenges.

1.4 Contribution

The main contributions in this research are highlighted as follows:

1) The first major contribution is the proposal of two modified Firefly Algorithm (FA) models for the mitigation of initialization sensitivity and local optima traps in the conventional KM clustering [52].

   Two modified FA models, namely inward intensified exploration FA (IIEFA) and compound intensified exploration FA (CIEFA), are proposed to increase search diversification and efficiency. Firstly, a randomized control matrix is proposed in IIEFA to replace the attractiveness coefficient in the original FA model, in order to intensify exploitation diversity. It enables the diagonal-based search paradigm in the original FA model to be elevated to a multi-dimensional region-based search mechanism with greater diversities in search scales and directions. Secondly, besides the above strategy, the capability of global exploration is further enhanced in CIEFA by dispersing and relocating fireflies with high similarities to unexploited regions outside the scope between fireflies in comparison. This enables the movement of fireflies to be diversified and search space expanded, therefore less likely to be trapped at local optima. The search efficiency is also improved owing to the guarantee of sufficient variance between fireflies in comparison, especially in the early convergence stage.

The proposed FA models are incorporated into the conventional KM clustering to overcome initialization sensitivity and local stagnation. The ALL-IDB2 database, a skin lesion data set, and a total of 15 UCI data sets are employed to evaluate clustering efficiency of the proposed FA models. For each clustering task, five performance indicators are calculated, i.e. intra-cluster distances, accuracy, sensitivity, specificity, and $\mathrm{Fscore}_{\mathrm{M}}$. The empirical results indicate that the proposed FA based clustering methods demonstrate great efficacy and efficiency in identifying superior configurations of cluster centroid in both high- and low-dimensional scenarios, in comparison with the conventional KM clustering algorithm, five classical search methods, as well as five advanced FA variants. The optimized cluster centroids are capable of yielding more compact clusters with superior performance scores. Moreover, between the two proposed models, CIEFA offers a better option, as compared with IIEFA, to deal with challenging clustering tasks such as data samples with high dimensionality, noise, and complicated distributions, owing to its enhanced exploration capability attributed by dispatching fireflies with high similarities to the unexploited search space.

2) The second major contribution is the proposal of an enhanced Particle Swarm Optimization (PSO) model for undertaking diverse feature selection tasks and improving classification performance.

In order to overcome two major shortcomings of the original PSO model, i.e. premature convergence and weak local exploitation capability around near optimal solutions, the proposed PSO model employs four key strategies: 1) a swarm leader enhancing mechanism using skewed Gaussian distributions, 2) a worst solution enhancing scheme incorporating a global best solution mirroring action and a Differential Evolution (DE)-based mutation operation, 3) a diversity-enhanced PSO evolving strategy incorporating multiple local and global optimal indicators and chaotic inertia weight based on Logistic map, and 4) an intensified spiral exploitation scheme. The aforementioned first two proposed strategies elevate the utilisation and exploitation of acquired knowledge in the swarm from two perspectives, i.e. introducing a self-

improving operation for the global best solution and facilitating the communication and cooperation among elite solutions through DE mutation operations during the process of enhancing weak solutions. On the other hand, the last two strategies optimize the search behaviour to increase the capability of acquiring new knowledge by constructing delicate search actions with multiple optimal signals to elevate both the diversification of exploration and the intensification of exploitation.

A total of 9 UCI data sets and the ALL-IDB2 database with a wide spectrum of dimensionalities, i.e. from 30 to 10000, are employed to evaluate effectiveness of the proposed PSO model on undertaking diverse feature selection tasks. The empirical results indicate that the proposed PSO model demonstrates significant superiority in achieving better trade-off between feature elimination and performance improvement, and outperforms five classical search methods and five advanced PSO variants, statistically. The advantages of the proposed PSO model become more evident on highly complex feature selection tasks owing to higher performance gaps ascribed by more successful identifications of the most discriminative and effective features.

3) The third major contribution is the proposal of an enhanced Grey Wolf Optimizer (GWO) model for automatic identification of the optimal topological configurations and learning hyperparameters for CNN-LSTM networks to undertake time series prediction problems [53].

In order to overcome stagnation at local optima and slow convergence rate in the original GWO model, the proposed GWO method incorporates four distinctive strategies: 1) a nonlinear adjustment of search coefficient capable of extending search territory during exploration and confining the search range during exploitation, 2) a chaotic weight allocation mechanism for three dominant wolf leaders using the sinusoidal chaotic map, 3) a local exploitation scheme based on enhanced spiral search with symmetrical oscillations, 4) probability distribution-based leader enhancement. The proposed strategies enhance search diversity by expanding exploration space as well as diversifying the guiding signals in a periodical manner. In addition, search efficiency and

convergence rate are also improved owing to the assurance of dominance of the best wolf leader as well as the intensified local exploitation around the optimal signals at the final stage of the search. As such, the proposed GWO variant is capable of achieving better trade-offs between search diversification and intensification, therefore increasing the likelihood of attaining global optimality.

The enhanced GWO model is subsequently employed to devise the network representation of the proposed CNN-LSTM model for tackling time series problems. The optimized evolving CNN-LSTM architecture is evaluated on three time series problems, i.e. energy consumption forecast, PM2.5 pollution prediction, and human activity recognition (HAR). The proposed evolving time series forecasting model significantly outperforms those yielded by four classical search methods and three advanced GWO and PSO variants on all employed time series tasks, as evidenced by statistical test results. Moreover, the empirical results indicate that the optimized CNN-LSTM networks by the proposed GWO model are characterized by a higher number of filters in the convolutional layers and moderate settings in terms of the numbers of nodes in the LSTM layer and the fully connected layer. As such, the identified optimal network configurations are able to thoroughly examine the interactions among time series variables, and provide efficient network representational capacities without suffering from either overfitting or underfitting issues.

1.5 Thesis layout

The rest of the thesis is organised as follows.

Chapter 2 introduces the preliminary concepts and essential models with respect to classification, clustering, DL and EC, respectively. Moreover, it also provides an up-to-date literature review on the study of hybridization between ML and EC models, including evolutionary feature selection, evolutionary KM clustering, as well as evolving DNNs methods.

Chapter 3 presents the proposed FA-based evolutionary KM clustering models. Two FA variants are proposed to optimize the configuration of cluster centroids in KM, namely IIEFA and CIEFA. The proposed IIEFA employs matrix-based search parameters to elevate exploitation capability, whereas the proposed CIEFA further

enhances IIEFA by incorporating a dispersing mechanism to increase search diversity. The proposed FA-based KM clustering models are evaluated using 16 clustering tasks as well as five performance criteria, i.e. the sum of intra-cluster distances, average accuracy, average sensitivity, average specificity, and macro-average F-score. Their performances are compared against the conventional KM clustering and ten baseline search methods.

Chapter 4 presents the proposed PSO-based evolutionary feature selection model. The strategies incorporated in the proposed PSO variant are analysed in detail, including the leader mutation, the worse solution enhancement, the guiding signal diversification, as well as the spiral local exploitation. A comprehensive evaluation is conducted for the proposed PSO model, using a total of ten feature selection tasks with a wide spectrum of dimensionalities, three performance indicators, i.e. classification accuracy, number of selected features, and F-score measure, as well as ten baseline search methods for comparison.

Chapter 5 presents the proposed GWO-based evolving CNN-LSTM time series forecasting model. The strategies employed in the proposed GWO variant are analysed in detail, i.e. the nonlinear search territory adjustment, the chaotic leadership competition, the symmetric spiral exploitation action, as well as the Lévy flight-based leader enhancement. The enhanced GWO variant is employed for evolving generation of the optimal topological configurations and learning hyperparameters for the proposed CNN-LSTM architecture. This proposed evolving CNN-LSTM method is evaluated using two time series prediction problems and another time series classification problem. Its performance is compared with the CNN-LSTM model with default settings and seven baseline hybrid CNN-LSTM methods.

Chapter 6 summarizes the whole research. It provides conclusions and recommendations for future research directions.

Chapter 2
Preliminaries and Literature Review

In this chapter, the fundamental concepts and essential models involved in this research are introduced, including EC, clustering, classification, feature selection, as well as DL. Moreover, the state-of-the-art studies on the hybridization between EC and ML models are reviewed, such as evolutionary feature selection, evolutionary KM clustering, and evolving DNNs.

2.1 Evolutionary computation

EC is a subfield of computational intelligence, which tackles optimization problems in a stochastic manner by simulating the procedure of natural selection and the survival of the fittest [54]. EAs are a family of metaheuristic optimization methods in EC which employ population-based evolving organisms to supervise individual solutions to move towards promising search territory iteratively and search for the global optima. In principle, EAs can be classified into two categories, according to their inherent differences of the operating mechanisms. Specifically, in the first category optimization algorithms apply mechanisms inspired by the Darwin's theory of evolution and search operations are executed by following the generic framework of the biological evolution, which consists of several genetic operators, such as selection, reproduction, as well as mutation, e.g. Genetic Algorithm (GA) [55], Differential Evolution (DE) [56], Memetic Algorithm (MA) [57]. In the second category, optimization algorithms are developed based on nature-inspired collective behaviours, e.g. bird flocking, wolf hunting, fish schooling, and individual search agent interacts with each other locally to facilitate the exchange of information as well as the acquirement of novel understandings of the search landscape, e.g. PSO [58], Ant Colony Optimisation (ACO) [59], FA [60], Cuckoo Search (CS) [61], GWO [62], Moth-Flame Optimisation (MFO) [63], Dragonfly Algorithm (DA) [64], Bat Algorithm (BA) [65], Flower Pollination Algorithm (FPA) [66], Gravitational Search Algorithm (GSA) [67]. Characterised by the simplicity, scalability, and advanced capabilities in searching for global optimality, EAs have been widely applied to tackle complex and multimodal NP-hard problems, e.g. automatic control, path planning, combinatorial optimization, the design and training of neural networks. In this section, three of the most popular models in EAs, i.e. PSO, FA,

and GWO, are elaborated and their state-of-the-art variants reviewed. Moreover, some latest EAs are also introduced.

2.1.1 Particle Swarm Optimization

2.1.1.1 Classical PSO

PSO is a population-based self-adaptive optimisation technique developed by Eberhart and Kennedy [58] based on swarm social behaviours, such as fish in a school and birds in a flock. The PSO algorithm conducts search in the landscape of objective function by adjusting trajectories of individual particles in a quasi-stochastic manner [68, 69]. Each particle adjusts its velocities and positions by following its own best experience in history and the global best solution of the swarm. Unlike conventional EAs [70, 71], PSO does not employ any crossover or mutation operators. In the original PSO model, the update of the velocity v_{id}^{t+1} and position x_{id}^{t+1} of the ith particle at the dth dimension are prescribed in **Eqs. 2.1** and **2.2**.

$$v_{id}^{t+1} = wv_{id}^{t} + c_1 r_1 \left(pbest_{id} - x_{id}^{t}\right) + c_2 r_2 \left(gbest_{d} - x_{id}^{t}\right) \tag{2.1}$$

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \tag{2.2}$$

where v_i and x_i represent the velocity and position of the ith particle, while $pbest_{id}$ and $gbest$ represent the historical best solution of the ith particle and the global best solution, respectively. Besides that, c_1 and c_2 denote position constants, while r_1 and r_2 are random values generated from [0, 1]. Moreover, t and w represent the current iteration number and the inertia weight, respectively.

As one of the most acknowledged EAs, PSO has been widely adopted in various optimisation problems owing to its simplicity, fast convergence speed, as well as effectiveness and robust generalization capability. In PSO, each particle adjusts its search trajectory by learning from two historical best experiences, i.e. its own best position and the global best solution. Despite its great efficiencies, PSO suffers from local optima traps as well as inefficient fine-tuning capabilities owing to the dictation of global best signals as well as its position adjustment mechanisms [72-74]. As an example, PSO lacks the operation of exchanging information between particles owing to the fact that only the global best solution is exploited as the reference for coevolution [75]. Secondly, the swarm often tends to revisit previously explored regions due to the strict adherence to the historical best experiences of each particle [76]. These limitations

in the original PSO model severely constrain search diversity and search scope, hence resulting in early stagnation and premature convergence.

2.1.1.2 PSO variants

Many PSO variants have been proposed in existing studies to overcome the above limitations, i.e. local optima traps and premature convergence, from different perspectives, e.g. utilisation of adaptive and chaotic parameters [77, 78], exploration of various topology structures [79], as well as hybridisation with other distinctive search methods [80, 81]. Chen et al. [82] proposed a dynamic PSO with escaping prey schemes (DPSOEP). In DPSOEP, a total of 200 swarm particles are categorized into three groups according to their fitness values, i.e. 'preys' (top ranked 15 particles), 'strong particles' (the remaining top ranked 160 particles), and 'weak particles' (the rest of the particles), to simulate hunting and escaping behaviours observed in nature. Three different search behaviours are designed for particles for the above three groups accordingly to enhance search diversity. Specifically, Lévy flights are employed to enable 'preys' to escape from local optima while the original PSO position updating operation is conducted by those 'strong particles' to accelerate convergence. Besides that, the 'weak particles' adjust their positions by learning from the mean position of the 'strong particles' using a multivariate normal distribution. The DPSOEP was firstly evaluated on 13 non-convex and piecewise benchmark functions and compared against seven PSO variants, i.e. DMS-PSO (dynamic multi-swarm PSO), RPSO (PSO with the ring lattice topology), SLPSO (self-learning PSO), HPSO-TVAC (self-organizing hierarchical PSO with time-varying acceleration coefficients), APSO (adaptive PSO), CLPSO (comprehensive learning PSO), and ALC-PSO (PSO with an aging leader and challengers), as well as seven classical and recently proposed search algorithms, i.e. GSO (Group Search Optimiser), BBO (Biogeography-based Optimisation), CMA-ES (Covariance Matrix Adaption Evolution Strategy), DE, BFO (Bacterial Foraging Optimisation), FFA (Fruit Fly Algorithm), and FA. Additionally, the DPSOEP model demonstrates great advantages over a number of other baseline models and provides higher applicability and practicality when tested on two other economic dispatch problems. Li et al. [83] proposed a multi-information fusion "triple variables with iteration" inertia weight PSO (MFTIWPSO) model to enhance the population diversity. The MFTIWPSO model adopts a multi-information inertia weight adjustment strategy which generates dimensional-wise inertia weight using particle velocity, position, random disturbance,

the number of iterations, as well as inertia weight score from the last iteration. Their PSO variant was employed to search for the optimality on 22 test functions and fine-tune SVM hyper-parameters, i.e. the penalty factor and the kernel parameter, on six classification tasks. The results indicate that the MFTIWPSO model significantly outperforms six baseline models, i.e. RANDPSO (PSO with random inertia weight), LHNPSO (low-discrepancy sequence initialized PSO with high-order nonlinear time-varying inertia weight), AIWPSO (PSO with adaptive inertia weight), DESIWPSO (double exponential self-adaptive inertia weight PSO), and SAIWPSO (stability-based adaptive inertia weight PSO). Cai et al. [84] proposed an efficient sequential approximation optimisation assisted PSO (SAOPSO) algorithm to improve the computational efficiency for expensive optimisation problems. In SAOPSO, the sequential approximation optimisation (SAO) is employed to improve each personal historical best solution by conducting the sampling operation in a local region. As a result, the optimisation efficiency is improved owing to the enhancement of the cognitive ability for swarm particles. The SAOPSO was evaluated by undertaking 36 numerical benchmark problems, as well as the optimisation of the design of bearings in all-direction propeller. The results indicate that the SAOPSO model demonstrates great advantages and outperforms 8 baseline methods, i.e. SPSO (a surrogate-assisted PSO), TRGA (trust region based GA), TRMPS (trust region based mode pursuing sampling algorithm), SA-COSO (the surrogate-assisted cooperative swarm optimisation algorithm), ESAO (evolutionary sampling assisted optimisation algorithm), TMAO (two-level multi-surrogate assisted optimisation method), EGO (efficient global optimisation), and SAORBF (RBF-based sequential approximation optimisation).

Li et al. [85] proposed a competitive and cooperative PSO method with information sharing mechanism (CCPSO-ISM). It particularly intensifies the utilisation of historical personal best solutions of the particle swarm. Specifically, the global and personal best solutions in the original PSO model are replaced by a vector named *ccBest* (competition and cooperation best). Each dimension of *ccBest* is determined based on the cooperation probability P. When the cooperation probability P is lower than a random value generated in the range of [0, 1], the dimension of *ccBest* is inherited from the corresponding dimension of the fittest solution among K randomly chosen personal best solutions. Otherwise it is inherited from the particle's own personal best solution. The neighbourhood size K increases linearly along with the iteration number. The CCPSO-

ISM was evaluated using 16 unimodal and multimodal benchmark functions. The results indicate that CCPSO-ISM demonstrates advantages on multimodal functions with many local optima, and outperforms PSO, LPSO (PSO with a topology of a ring lattice), FDR-PSO (fitness-distance-ratio based PSO), FIPS (the fully informed particle swarm), and CLPSO. Xia et al. [86] proposed an eXpanded PSO (XPSO), which expands the social component from one exemplar, i.e. the global best solution, to two exemplars, i.e. the global best solution and the local best solution in its neighbourhood. Besides that, XPSO assigns different forgetting abilities on the expanded exemplars to further enhance the search diversity for each particle. Acceleration coefficients in the position updating operation are adjusted using Gaussian distribution prescribed by the historical knowledge from elite individuals. In addition, the consecutive generations of the stagnancy of the global best solution are adopted as a criterion for the adjustment of acceleration coefficients and reselection of neighbours. The XPSO model was evaluated on CEC2013 test suite with 28 complex benchmark functions. The results indicate that XPSO yields the most promising performance on the test suite, and outperforms three other search algorithms, i.e. SaDE (self-adaptive DE), CMAES (evolution strategy with covariance matrix adaptation), PBILc (population-based incremental learning to continuous search spaces), as well as nine advanced PSO variants, i.e. Frankenstein's PSO [87], OLPSO (orthogonal learning PSO) [88], DEPSO (hybridization of DE and PSO) [89], PSODDS (PSO with distance based dimension selection) [90], CCPSO-ISM (competitive and cooperative PSO with information sharing mechanism) [85], SRPSO (self-regulating PSO) [91], HCLPSO (heterogeneous comprehensive learning PSO) [92], GLPSO (genetic learning PSO) [93], and EPSO (ensemble PSO) [94].

2.1.2 Firefly Algorithm

2.1.2.1 Classical FA

FA model performs the search operation according to the foraging behaviours of fireflies [60]. In FA, a swarm of fireflies is initiated randomly, and each firefly denotes one initial solution. A fitness score is calculated based on the objective function for each firefly, which is then assigned as the light intensity. According to [60], fireflies with lower light intensities are attracted to those with strong illuminations in the neighbourhood, as defined in **Eq. 2.3**.

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2}\left(x_j^t - x_i^t\right) + \alpha_t \varepsilon_t \tag{2.3}$$

where i and j denote fireflies with lower and higher light intensities, respectively, while x_i^t and x_j^t denote the current positions of fireflies i and j at the t^{th} iteration, respectively. Parameter β_0 is the initial attractiveness while γ is the light absorption coefficient, and r_{ij} denotes the distance between fireflies i and j. In addition, α_t is a randomization coefficient, while ε_t is a vector of random numbers drawn from a Gaussian distribution or a uniform distribution.

The major advantage of FA lies in its attraction mechanism. The attractiveness-based movements enable the firefly swarm to automatically subdivide into subgroups, where each group swarms around one mode or a local optimum solution [60, 95]. When the population size is sufficiently higher than the number of local optima, the subdivision ability in FA is able to find all optima simultaneously in principle, and, therefore, attain the global optima. This automatic subdivision ability enables the FA model to tackle optimisation problems characterised as highly nonlinear and multimodal, with many local optima traps.

Despite the abovementioned advantages, there are certain limitations in search diversification imposed by the strict obedience of biological laws in the original FA model. These limitations are rarely addressed in the existing literature. Specifically, the position updating strategy in FA in **Eq. 2.3** is constructed according to the firefly foraging behaviours, which is employed to guide one firefly to approach another with a higher light intensity by multiplying the position difference of these two fireflies ($x_j^t - x_i^t$) with their relative attractiveness component ($\beta_0 e^{-\gamma r_{ij}^2}$). While the inheritance of biological laws enables one firefly to approach another with a more favourable position, the dimensionality and diversity through the approaching process are severely constrained, since the movement can only happen on the diagonal direction composed by two fireflies, in accordance with the formula. As illustrated in **Figure 2-1**, in a two-dimensional scenario, there are two fireflies, i and j. If we view both fireflies as vectors, the position difference of these two fireflies ($x_j^t - x_i^t$) can be represented by the dotted line denoted as Δp in **Figure 2-1**. The calculation of attractiveness practically imposes one constant isotropic factor ($\beta_0 e^{-\gamma r_{ij}^2}$) on all dimensions of the position difference between fireflies i and j, causing the lack of variance among different dimensions. As a result, instead of exploring flexibly in the entire solution space, the fireflies can merely move along the specific diagonal trajectory between two fireflies in comparison, and the

search area is shrunk drastically from a two-dimensional rectangular enclosed with dash lines into a one-dimensional vector in dotted line, as shown in **Figure 2-1**. Therefore, the chances of finding the global optima are reduced, since search diversification is constrained severely owing to the limitations of the biological laws in the original FA model.

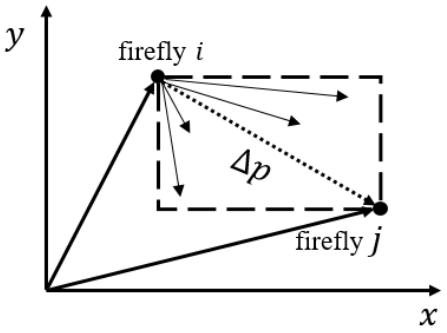

Figure 2-1 The movement of fireflies in a two-dimensional search space (Δp denotes the position difference between fireflies i and j)

2.1.2.2 FA variants

While the original FA model demonstrates some unique properties in its search mechanism, it suffers from slow convergence and high computational complexity, owing to its behaviour of following all brighter fireflies in the neighbourhood [96]. Additionally, fireflies can fall into stagnation during the search process, as the distance between fireflies increases and the attractiveness component ($\beta_0 e^{-\gamma r_{ij}^2}$) approaches zero. Many FA variants have been proposed to overcome these problems by increasing the exploration ability and search diversification of the original FA model. The strategies employed to improve the original FA model can be generally categorized into three groups, i.e. adaptive processes of parameter tuning, population diversification, and integration of hybrid search patterns [97]. Ozsoydan and Baykasoglu [98] proposed a quantum firefly swarm model to tackle multimodal dynamic optimization problems. Four strategies were incorporated into their model: (1) multi-swarms based search; (2) two types of movements undertaken by neutral and quantum fireflies respectively in each sub-swarm; (3) simplification of firefly position updating; and (4) employment of two sub-swarm prioritizing techniques, i.e. sequential selection and roulette wheel selection. The quantum firefly swarm model was evaluated with the Moving Peaks Benchmark problem to locate and track the moving optima. The obtained results

indicated that the quantum firefly swarm model was competitive and promising in comparison with 13 well-known algorithms in dynamic optimization problems, including mCPSO-with anticonvergence, mCPSO-without anticonvergence, mQSO-with anticonvergence, mQSO-without anticonvergence, SPSO, rSPSO, BSPSO, RWS, and SPSO-PD. Banerjee et al. [99] proposed a Repulsion-Propulsion FA (PropFA) model by incorporating three strategies, i.e. (1) introduction of adaptive mechanisms for both randomization coefficient α_t and light absorption coefficient γ, (2) incorporation of the global best solution as a component for swarm position update, and (3) replacement of the Euclidean distance measurement with Manhattan distance measurement. Three ratios were yielded to construct the adaptive search parameter mechanisms based on a short term memory of the last positions and light intensities of fireflies. The PropFA model was evaluated using 18 classical benchmark functions, 14 additional functions of CEC-2005, and 28 functions of CEC-2013. The results demonstrated the competitiveness of the PropFA model in finding better solutions in comparison with PSO, EDA (Estimation of Distribution Algorithms), RC-EA (Mutation Step Co-evolution), RC-Memetic (Real-Coded Memetic Algorithm), CMA-ES (Covariance Matrix Adaptation Evolution Strategy) on CEC-2005 benchmark functions, and SHADE, CoDE (DE with composite trial vector generation strategies and control parameters), Jade (Adaptive DE with optional external archive) on CEC-2013 benchmark functions. The PropFA model was also employed to estimate the spill area of a fast expanding oil spill, and the PropFA-based confinement strategy proved to be successful.

Baykasoglu and Ozsoydan [100] proposed a variant of FA, i.e. FA2, with two strategies: (1) replacing the exponential function with an inverse function of distance as the attractiveness coefficient, and (2) constructing a threshold probability for a firefly's position to be updated or otherwise. The FA2 model was tested by both static and dynamic multidimensional knapsack problems. The obtained results indicated that FA2 was more effective than GA, DE, and FA. Sadhu et al. [101] proposed a Q-learning induced FA (QFA) model. Q-learning was used to generate light absorption coefficient γ and randomization coefficient α_t with a fitness rank based rewarding and penalizing mechanism. The generated pair, $<\gamma, \alpha_t>$, was capable of producing high-performing fireflies in each step. The QFA model was tested with fifteen benchmark functions in CEC 2015, and with a real-world path planning problem of a robotic manipulator with

various obstacles. The empirical results confirmed the superiority of the QFA model in terms of solution quality and run-time complexity in comparison with other algorithms, e.g. AFA (adaptive FA), DEsPA (Differential Evolution with success-based parameter adaption), SRPSO (Self-regulating PSO), SDMS-PSO2 (Self adaptive dynamic multi-swarm PSO), SLPSO (social learning PSO), and LFABC (Levy flight Artificial Bee Colony). Zhang et al. [102] proposed a modified FA model for feature selection by incorporating three strategies, i.e. the improved attractiveness operations guided by SA-enhanced neighbouring and global optimal signals, chaotic diversified search mechanisms, and diversion of weak solutions. The modified FA model was tested with feature selection problems using 29 classification and 11 regression benchmark data sets. The experimental results indicated that the proposed FA variant outperformed 11 classical search methods in undertaking diverse feature selection tasks, i.e. PSO, GA, FA, SA, CS, Tabu Search (TS), DE, Bat Swarm Optimization (BSO), DA, Ant-Lion Optimization (ALO), Memetic Algorithm with Local Search Chain (MA-LS), and 10 popular FA variants, i.e. FA with neighbourhood attraction (NaFA) [96], SA incorporated with FA (SFA) [103], SA incorporated with both Levy flights and FA (LSFA) [103], Opposition and Dimensional FA (ODFA) [104], FA with Logistic map as the randomization search parameter (CFA1) [105], FA with Gauss map as the attractiveness coefficient (CFA2) [106], FA with a variable step wise (VSSFA) [107], FA with a random attraction (RaFA) [108], a modified FA incorporating chaotic Tent map and global best based search operation (MCFA) [109], and a hybrid multi-objective FA (HMOFA) [110].

FA and its variants have also been widely used for solving multimodal optimisation problems. Gandomi et al. [111] applied FA to a set of seven mixed variable structural optimization problems with nonlinearity and multiple local optima. The empirical results indicated that FA was more efficient than other metaheuristic algorithms, such as PSO, GA, and Harmony Search (HS), on these optimization tasks. Nekouie and Yaghoobi [112] proposed a hybrid method on the basis of FA for solving multimodal optimisation problems. In their study, KM was used to cluster the FA population into several subpopulations. FA with a roaming technique was employed to identify multiple local optima, while SA was used to further improve the local promising solutions. A set of 15 multimodal test functions was used to evaluate the effectiveness of the hybrid model. The empirical results demonstrated its great advantages over other methods such

as Niche GSA (NGSA), r2PSO (a l-best PSO with a ring topology and each member interacting with its immediate member on its right), r3PSO (a l-best PSO with a ring topology and each member interacting with its immediate members on both its left and right), r2PSO-lhc (r2PSO with no overlapping neighbourhoods), FER-PSO (Fitness Euclidean-distance Ratio based PSO), and SPSO (Speciation-based PSO). Zhang et al. [113] proposed a modified FA model for ensemble model construction for classification and regression problems. Their FA variant embedded attractiveness strategies guided by both neighbouring and global promising solutions, as well as evading mechanisms with the consideration of local and global worst experiences. Their FA variant was evaluated with standard, shifted, and composite test functions, as well as the Black-Box Optimization Benchmarking test suite and several high-dimensional UCI data sets. The experimental results indicated that their FA model outperformed several state-of-the-art FA variants and classical search methods in solving diverse complex unimodal and multimodal optimization and ensemble reduction problems. Yang [114] proposed a multi-objective FA model (MOFA) for solving optimization problems with multiple objectives and complex nonlinear constraints. Evaluated with five mathematical artificial landscapes with convex, nonconvex, discontinuous Pareto fronts, and complex Pareto sets, the empirical results indicated that MOFA outperformed seven established multi-objective algorithms, i.e. vector evaluated GA (VEGA), Non-dominated Sorting GA II (NSGA-II), multi-objective DE (MODE), DE for multi-objective optimization (DEMO), multi-objective Bees algorithms (Bees), and Strength Pareto Evolutionary Algorithm (SPEA). A comprehensive review on evolutionary algorithms for multimodal optimization is also provided in [115].

2.1.3 Grey Wolf Optimizer

2.1.3.1 Classical GWO

GWO is a SI algorithm proposed recently according to the social dominant hierarchy and group hunting operations observed among grey wolves [62]. In a wolf pack, there are four different levels in terms of the positions in the social hierarchy, i.e. wolf alpha (α), wolf beta (β), wolf delta (δ), and wolf omega (ω). Those wolves from the top three hierarchies, i.e. α, β, and δ, are responsible for decision making during hunting whereas wolves at the bottom of the hierarchical ladder, i.e. ω, are subordinated to those from higher levels unconditionally.

In GWO, each wolf represents a solution initialized randomly. The wolves with highest three fitness scores are labelled as α, β, and δ, respectively, and assume the leadership to guide the movement of the whole wolf pack. The search mechanism of GWO is based on the encircling hunting mechanism observed within the grey wolf pack in nature as well as the supposition that three dominant wolves retain better knowledge regarding the location of the prey/optimality than their comrades. Henceforth, each wolf updates its position in reference to the three top leaders in the wolf pack, i.e. α, β, and δ, respectively, in a manner as instructed in **Eqs. 2.4 - 2.9**. The arithmetic average of the above three position adjustments is then adopted as the target position for each wolf to be dispatched to, as indicated in **Eq. 2.10**.

$$D_{\alpha,j}^{t+1} = \left|C_1 X_{\alpha,j}^{t} - X_{i,j}^{t}\right| \tag{2.4}$$

$$D_{\beta,j}^{t+1} = \left|C_2 X_{\beta,j}^{t} - X_{i,j}^{t}\right| \tag{2.5}$$

$$D_{\delta,j}^{t+1} = \left|C_3 X_{\delta,j}^{t} - X_{i,j}^{t}\right| \tag{2.6}$$

$$X_{ad1,j}^{t+1} = X_{\alpha,j}^{t} - A_1 D_{\alpha,j}^{t+1} \tag{2.7}$$

$$X_{ad2,j}^{t+1} = X_{\beta,j}^{t} - A_2 D_{\beta,j}^{t+1} \tag{2.8}$$

$$X_{ad3,j}^{t+1} = X_{\delta,j}^{t} - A_3 D_{\delta,j}^{t+1} \tag{2.9}$$

$$X_{i,j}^{t+1} = (X_{ad1,j}^{t+1} + X_{ad2,j}^{t+1} + X_{ad3,j}^{t+1})/3 \tag{2.10}$$

$$C = 2rand \tag{2.11}$$

$$A = (2rand - 1)a \tag{2.12}$$

$$a = 2(1 - \frac{t}{Max\_iter}) \tag{2.13}$$

where $X_{i,j}^{t}$ denotes the element of the i-th wolf on j-th dimension under the t-th iteration. X_α, X_β, and X_δ represent positions of the three leading wolves α, β, and δ respectively, whereas D_α, D_β, and D_δ represent distance measures, and X_{ad1}, X_{ad2}, and X_{ad3} represent position adjustments, in reference to the above three dominant wolves i.e. α, β, and δ, respectively. Besides the above, A and C are two search coefficients related to position updating where A_1, A_2, A_3 are the three instantiations of parameter A, and C_1, C_2, C_3 are the three instantiations of parameter C. Max_iter denotes the maximum iteration number, whereas $rand$ is a random number in the range of [0, 1]. In addition, a denotes the exploration rate linearly decreasing from 2 to 0 as iteration increases.

In principle, GWO possesses many merits in comparison with previous classical search methods (e.g. PSO and GA), owing to the employment of multiple-leader guided search

as well as dynamic fine-tuning of the search scopes. However, it also suffers from inefficiencies owing to the acute shrinkage of the search territory as well as the undermined representation of the leadership hierarchy.

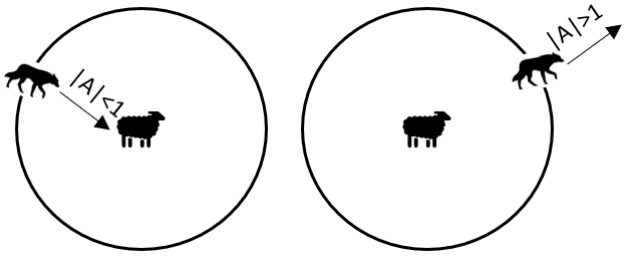

Figure 2-2 Exploitation (|A|<1) vs. Exploration (|A|>1) in GWO [60]

To be specific, in the original GWO model, a is an essential search parameter, capable of regulating the transition from exploration to exploitation during the search iterations. The parameter a dictates the search boundary and radius of the wolf population through regulating the magnitude of the step size A, as shown in **Eq. 2.13**. Specifically, as illustrated in **Figure 2-2**, the wolves conduct exploration and jump out of the search range between itself and the prey when $|A| > 1$. This can only happen when the exploration rate $a > 1$, according to **Eq. 2.12**. In contrast, the exploitation between the wolf and the prey can be deployed when $|A| < 1$. The linearly decreasing pattern of parameter a adopted in the original GWO, as shown in **Eq. 2.13**, unequivocally result in a rapid contraction of the search boundary, which severely confines the exploration capability of the wolf population, especially at the first half of the search course when extensive explorations are required to escape from local optima traps. Besides above, the static and equal division of the leadership among three strongest wolves over the whole search course contradicts its strategy of hierarchical division within the wolf community in principle, and largely confines the capability of fine-tuning around the obtained global best solution. Hence, this lack of prioritizing operators among dominant wolf leaders results in a slow convergence rate, therefore compromising search efficiency.

2.1.3.2 GWO variants

As analysed above, GWO suffers from evident disadvantages such as local stagnation, a slow convergence rate, as well as deficiency in fine-tuning around the best swarm leader [116-118]. Many efforts have been made in existing studies to mitigate the above

drawbacks to enhance the search efficiency of GWO. Ozsoydan [119] proposed three GWO variants, i.e. prioGWO, learnGWO, and prLeGWO, to investigate the effects of dominant wolves in GWO. In prioGWO, three dominant wolves rearrange their positions within themselves by following the position updating formula in the original GWO, prior to guiding the movement of the rest of the wolf pack. In learnGWO, dedicated learning curves are developed to gradually increase the dominance of wolf α, while decreasing that of wolves β and δ over the iterative process. Besides the above, prLeGWO incorporates both strategies employed in prioGWO and learnGWO. These GWO variants are evaluated on multiple tasks, i.e. unconstraint test functions, the uncapacitated facility location problem (UFLP), as well as the 0-1 knapsack problem. The results indicate the effectiveness of their GWO variants in comparison with five baseline models, i.e. PSO, GWO, a continuous PSO with a local search (CPSO), an adapted Artificial Bee Colony for binary optimization (ABC_{bin}), and Weighted Superposition Attraction (WSA). Luo [120] proposed an enhanced GWO (EGWO) model which dynamically estimates the location of the prey using weight-based aggregation of the three dominant wolf leaders. The weights are generated using normalised random numbers within [0, 1]. A strict hierarchical order is established by assigning weights based on the rankings of fitness scores of the three dominant wolves, i.e. larger weights for wolves with higher rankings. Subsequently, wolves update their positions under the guidance of this estimated location of the prey. The EGWO model is evaluated on 30-dimensional and 100-dimensional CEC2017 test functions, as well as two engineering applications, and significantly outperforms the original GWO, a fuzzy hierarchical GWO, and a random walk GWO. Gupta and Deep [121] proposed a modified GWO method based on random walks (RW-GWO). Specifically, the three dominant wolves are further improved by conducting random jumps, with steps generated from Cauchy distribution. RW-GWO is evaluated on 10-dimensional and 30-dimensional CEC2014 test functions, and demonstrates significant superiorities in comparison with baseline models, e.g. GSA, CS, Laplacian Biogeography-Based Optimization (LX-BBO).

Wang and Li [116] proposed an improved GWO (IGWO) by incorporating biological evolution and survival of the fittest principle into the evolving process of GWO. Specifically, a DE-based breeding operation is applied to the three dominant wolf leaders. A crossover operation is then used with the yielded offspring and each

individual wolf solution as the parent chromosomes. Besides the above, a dynamic number of weak individuals are eliminated from the population and replaced by randomly generated new solutions, according to the principle of the survival of the fittest. The IGWO model is evaluated on the twelve benchmark functions and outperforms GWO, DE, PSO, ABC, and CS, statistically. Emary et al. [122] proposed a GWO variant, i.e. experienced GWO (EGWO), in which reinforcement learning is employed to yield the exploration rate, i.e. parameter a in GWO, for each individual wolf based on its past experience in each iteration. Specifically, a state-action model is mapped using a neural network with a single hidden layer to maximize the reward function, in which the input is the change state of the fitness score in every two successive iterations, and the output is the action set for the adjustment of exploration rate, i.e. increasing, decreasing, and maintaining the current value of a. As such, the parameter a can be specifically tailored for each individual wolf by the mapped network, according to its own previous experience and performance, to bestow the freedom of choosing between exploration and exploitation on each individual wolf per se, instead of following the same regulation of parameter a collectively. The effectiveness of EGWO is evaluated on 21 feature selection tasks and 10 ANN weight adaptation tasks. The results indicate significant advantages of EGWO over the original GWO, PSO and GA. Moreover, Tu et al. [123] proposed a hierarchy strengthened GWO (HSGWO) model which incorporates an elite learning operator, an opposition-based learning strategy, a DE operator, a hybrid total-dimensional and one-dimensional update strategy, as well as a perturbed operator. The enhanced elite learning strategy ensures dominant wolves only learn from those with higher rankings, hence mitigating distractions from less advanced solutions, whereas opposition-based learning enables dominant wolves to conduct extensive explorations. The remaining wolf solutions are able to choose between the original GWO and DE models to update their positions, in either all dimensions or only one sub-dimension. Moreover, a fraction of wolf candidates is replaced with solutions yielded from perturbations of randomly selected individuals from the wolf pack. HSGWO is evaluated on the CEC2014 test functions as well as 13 feature selection tasks and outperforms baseline models, e.g. Salp Swarm Algorithm (SSA), and differential mutation and novel social learning PSO (DSPSO).

Moreover, Gupta and Deep [124] proposed a memory-based GWO (mGWO) model. It incorporates the personal best experiences, randomly selected wolf solutions, a

crossover operation, and a greedy selection strategy for position updating. The personal historical best experience is employed in two distinctive manners to yield two respective candidate solutions for the current individual under each iteration. Specifically, the first candidate is generated by replacing the position of the wolf in the current iteration with its historical best experience in the position updating equations of the original GWO algorithm. The second candidate is yielded by a local search mechanism involving the historical best experience, as well as two randomly selected wolf solutions in the neighbourhood. Subsequently, a crossover operation is performed on both candidates, and the offspring solutions are adopted as the new individuals for the next generation. Besides that, a greedy selection strategy is enforced between the wolf solutions of two consecutive iterations, and the best one is retained. The mGWO model is evaluated with the CEC2014 and CEC2017 benchmark test functions, as well as six practical engineering design problems. It outperforms numerous classical search methods, e.g. PSO, Firefly Algorithm (FA), and advanced GWO variants including Oppositional GWO (OGWO) and Improved GWO (IGWO) on unimodal, multimodal, and composite benchmark functions. Ibrahim et al. [125] proposed an improved GWO variant (COGWO2D) that incorporates four strategies. They are a logistic chaotic map, an Opposition-Based Learning (OBL) mechanism, a DE position updating scheme, and a disruption operator. The logistic map is used for chaotic population initialization. The OBL mechanism is applied to generate the opposite counterparts. The final collection of the initialized solutions is selected from the above combined sets according to the fitness eminence. Then, the original GWO and DE updating mechanisms are combined in parallel for position updating. In addition, the disruption operator is employed to increase the search diversity for those wolf solutions distant from the current swarm leader, while intensifying local exploitation for the remaining wolf individuals located in the vicinity of the current global best solution. Evaluated with the CEC2005 and CEC2014 benchmark functions and a feature selection task, the COGWO2D model significantly outperforms other nine competitors, including WOA, SSA, Ant Lion Optimizer (ALO), DE, and CS. Al-Betar et al. [126] investigated the impacts of different natural selection methods on the performance of GWO. In addition to the greedy selection of the top three wolf leaders employed in the original GWO model, five additional selection paradigms are explored, i.e. the tournament selection, proportional selection, stochastic universal sampling selection, linear rank selection, and random selection. Evaluated with 23 benchmark functions, GWO with the tournament

selection achieves the best performances, outperforming several classical search methods, e.g. GA and PSO. GWO with the random selection obtains the worst optimization results. The research provides good insight on the common dilemma of employing elicit signals and introducing random perturbations in developing metaheuristic algorithms. Wen et al. [127] proposed an inspired GWO (IGWO) model. It employs a logarithmic decay function to adjust search parameter a and a modified position updating mechanism incorporating the mean position of three wolf leaders, the personal historical best experience, and the global best solution, for imitation of the position updating technique in PSO. Evaluated with four high-dimensional benchmark test functions and three practical engineering design problems, IGWO outperforms the original GWO model, four advanced GWO variants, and four other search methods. Saxena et al. [128] proposed a β-Chaotic map enabled GWO (β-GWO) model. It modifies the linearly decreasing search parameter a by adding a β function-based chaotic sequence. This design enables the preservation of the exploration virtue throughout the iterative process. Evaluated with the CEC2017 benchmark test functions and two practical engineering design problems, β-GWO outperforms four classical search methods, including GSA and Flower Pollination Algorithm (FPA), and five advanced GWO variants, including OGWO and Grouped GWO (GGWO), with statistical significance.

2.1.4 Other latest evolutionary algorithms

In addition to the above three popular methods, other innovative search mechanisms have been further developed to improve the robustness and applicability of EAs. A review on the latest models is presented below. Inspired by the oscillation mode and food search patterns of slime mould in nature, Li et al. [129] proposed a Slime Mould Algorithm (SMA). It incorporates three types of movements in cascade as well as in conjugation with oscillated search parameters for position updating. Specifically, for producing high-quality slime mould solutions, a local exploitation operation is conducted in all directions to further refine search individuals. The current low-quality positions are replaced with the new ones yielded by the global best and two other randomly selected individuals. To further increase search diversity, the slime mould population is replenished with new individuals randomly generated according to a predefined probability-based condition. Evaluated with 33 benchmark test functions and four practical engineering problems, the SMA model significantly outperforms a

number of classical and advanced search methods, e.g. MFO and Comprehensive Learning PSO (CLPSO). Askari et al. [130] proposed a Heap-based Optimizer (HBO) by simulating various interactions in a corporate rank hierarchy. A 3-ary heap structure according to the fitness values is established on the population. A cascade search mechanism incorporating three search scenarios is developed, i.e. moving towards the immediate superior solution in the higher hierarchy (boss), moving towards a fitter solution within the same hierarchy (colleague), and retaining the current position (self-contribution). Evaluated with 97 benchmark test functions and three practical engineering problems, the HBO model outperforms seven well-known search algorithms, including Multi-Verse Optimizer (MVO), GSA, PSO, and CS. Inspired by the gradient-based Newton's method, Ahmadianfar et al. [131] proposed a Gradient-based Optimizer (GBO). It incorporates a gradient search rule and a local escaping operator. The gradient search rule applies a gradient-based mechanism to drive the individuals to approach the global best solution, while retaining search diversity through the employment of randomly selected individuals in the neighbourhood during the position updating process. Besides that, a local escaping operator for overcoming the local optima traps is developed by further introducing newly generated individuals into the population to participate in the competition. Evaluated with 28 benchmark test functions and six engineering problems, it significantly outperforms five classical search methods, i.e., GWO, CS, ABC, WOA and Interactive Search Algorithm (ISA). Heidari et al. [132] proposed a Harris Hawk Optimization (HHO) algorithm. It mimics the hunting mechanism of Harries hawks. During exploration, two position updating options are developed, i.e. adjusting position in reference to the global best solution, or randomly selected solutions in the neighbourhood corresponding to two perching choices of hawks, i.e. the family member and the rabbit, during hunting, respectively. To facilitate exploitation, four local search mechanisms are designed to approach the global best solution by adopting different search coefficient vectors, in simulation of besiege processes of hawks. Evaluated with 29 benchmark test functions and six engineering optimization problems, HHO outperforms a number of classical search models, including FPA and MFO, significantly.

Overall, EC methods demonstrate significant advantages in solving complex optimisation problems, especially those with sophisticated search landscapes as well as complex variable interactions, such as NP-hard problems, owing to their superior global

exploration capabilities as well as effective adaptation mechanisms to escape from local optima traps [133]. In this research, advanced EAs are introduced to devise enhanced evolving ML and DL methods which are capable of overcoming some of the most challenging optimization scenarios in ML, i.e. the selection of the most effective feature subset (feature selection), the optimization of initialized clustering centroids in KM (initialization sensitivity), as well as the identification of the optimal learning and topological configurations for DNNs (hyperparameter optimization). Subsequently, the fundamental concepts and essential models in ML and DL, as well as the state-of-the-art studies on the hybridization between ML and EC models are presented as follows.

2.2 Clustering analysis

Clustering analysis is one of the fundamental methods of discovering and understanding underlying patterns embodied in data by partitioning data objects into several clusters according to measured intrinsic characteristics or similarity [134]. As a result of the clustering process, data samples with high similarity are grouped in the same cluster, while those with distinctions are categorized into different clusters. Clustering analysis has been widely adopted by many disciplines, such as image segmentation [135-141], text mining [142-144], bioinformatics [145, 146], wireless sensor networks [147, 148], and financial analysis [149]. In general, conventional clustering algorithms can be broadly categorized into two groups: partitional and hierarchical methods. The partitional methods divide data samples into several clusters simultaneously, whereas the hierarchical methods build a hierarchy of clusters, either in an agglomerative (merging similar clusters) or divisive (dividing each cluster into smaller ones) mode [134].

2.2.1 K-means clustering

KM clustering is one of the most popular partitional methods, and is widely used owing to its simplicity, efficiency, and ease of implementation [134]. The KM clustering algorithm partitions data samples into different clusters based on distance measures. It finds a partition such that the squared error between the empirical mean of a cluster and the points in the cluster is minimized [134]. Let $O = \{O_1, O_2, \dots, O_n\}$ be a set of n data samples to be clustered into a set of K clusters, $C = \{C_i, i = 1, \dots, k\}$. The goal of KM clustering is to minimize the sum of the squared error over all k clusters, which is defined as follows:

$$J(C) = \sum_{i=1}^{k} \sum_{O_l \in C_i} (O_l - Z_i)^2 \quad (2.14)$$

where C_i, Z_i, O_l, and k represent the i^{th}cluster, the centroid for i^{th} cluster, data samples belonging to the i^{th}cluster, and the total number of clusters, respectively.

In KM clustering, cluster centroids are initialized randomly. Data samples are assigned to the closest cluster, which is determined by the distances between the corresponding centroid and data samples. The centroid of each cluster is updated by calculating the mean value of all data samples within the respective cluster. Then, the process of partitioning data samples into the corresponding clusters is repeated according to the updated cluster centroids until the specified termination criteria are met. The KM clustering algorithm shows impressive performances for a wide range of applications, including computer vision [150], pattern recognition [151] and information retrieval [152]. It often serves as a pre-processing method for other complex models to provide an initial configuration.

Despite the abovementioned merits, KM clustering suffers from a number of limitations, such as initialization sensitivity [134, 153], susceptibility to noise [154, 155], and vulnerability to undesirable sample distributions [155], owing to its restrictive assumptions and operating mechanisms. Specifically, real-life clustering tasks pose diverse challenges to KM clustering, owing to complexity embedded in data samples, such as immense dimensionality, disturbance of noise and outliers, irregular, sparse, and imbalanced sample distributions, and clusters with overlap or narrow class margins [134]. These complexities overtly violate restrictive assumptions embedded in KM, i.e. spherical sample distributions and evenly sized clusters, therefore leading to limitations in interpretability for such complex data distributions [154, 155]. Moreover, KM suffers from initialization sensitivity and local optima traps owing to its operating mechanism of local search around the configuration of initial centroids [134, 153]. The process of minimizing the sum of intra-cluster distances in KM is, in essence, a local search surrounding the initial centroids. As a result, the performance of KM heavily depends on the initial configuration of cluster centroids. In addition, owing to its operating mechanisms and the randomness during centroid initialization, KM is more likely to suffer from local optima traps.

2.2.2 Fuzzy C-means clustering

In addition to KM, fuzzy C-means is another popular clustering method. In KM, a hard-clustering process is employed where each data instance can only be assigned to a single cluster unequivocally with its intra-cluster neighbours. This mechanism is incompetent in dealing with data sets that possess overlapping cluster distributions with ambiguous boundaries. In contrast, fuzzy C-means [156] enables fuzzy partition where each sample can belong to multiple clusters simultaneously, by evaluating membership degrees of each data sample with respect to all clusters, respectively. The sum of least squared errors over all clusters weighted by the memberships is employed as the objective function to be minimized, as shown in **Eq. 2.15**.

$$J_f(C) = \sum_{i=1}^{k}\sum_{l=1}^{n}(w_{li})^m(O_l - Z_i)^2 \tag{2.15}$$

$$w_{li} = \frac{1}{\sum_{j=1}^{k}\left(\frac{\|O_l - Z_i\|}{\|O_l - Z_j\|}\right)^{\frac{2}{m-1}}} \tag{2.16}$$

$$Z_i = \frac{\sum_{l=1}^{n}(w_{li})^m \cdot O_l}{\sum_{l=1}^{n}(w_{li})^m} \tag{2.17}$$

where C_i, Z_i, O_l, k, and n represent the i^{th}cluster, the fuzzy centroid of i^{th} cluster, data samples partially belonging to the i^{th}cluster, the total number of clusters, and the number of data samples, respectively. Besides above, w_{li} denotes the membership of the l-th data sample with respect to the i-th cluster, whereas m is the fuzzy exponent that controls the fuzziness of the membership function.

The fuzzy partition is performed through the minimization of the objective function, as shown in **Eq. 2.15**, by updating the memberships and cluster centres iteratively. The membership function is defined in **Eq. 2.16**. The yielded membership values are within the range of [0, 1], whereby values close to one imply a high degree of similarity between the sample and the cluster while values close to zeros signify little similarity between them [156]. The cluster centres are computed using the weighted averages of the data samples, as shown in **Eq. 2.17**.

Fuzzy C-means has been applied to a wide range of problem domains, e.g. image segmentation [157], signal processing [158], fuzzy time series [159], owing to its effectiveness in dealing data sets with ambiguous boundaries and overlapped

distributions. Nevertheless, fuzzy C-means is sensitive to data noises and imaging artefacts since it does not consider the local relationship between pixels [160]. Also, it requires more computational efforts than KM clustering.

2.2.3 Evolving K-means clustering

As characterised by their powerful search capability in terms of exploration and exploitation, EAs have been widely employed to assist KM to escape from local optima traps by exploring and obtaining more optimized configurations of cluster centroids. The negative impacts imposed by challenging real-life data can, therefore, be mitigated owing to more accurate cluster identification resulted from the optimized centroids. The effectiveness of such hybrid clustering models has been extensively validated by empirical studies, e.g. TS [161, 162], Simulated Annealing (SA) [163], GA [164], Artificial Bee Colony (ABC) [165, 166], ACO [167, 168], PSO [168-170], CS [170, 171], FA [172, 173], GSA [174, 175], Black Hole Algorithm (BHA) [176], and Big Bang-Big Crunch algorithm (BB-BC) [177].

Karaboga and Ozturk [165] proposed an ABC-based clustering method by incorporating the original ABC model with KM clustering. The ABC-based clustering method was evaluated using 13 UCI data sets. The obtained results demonstrated the competitiveness of the combination of ABC with KM clustering in managing clustering tasks in comparison with those of PSO and nine classification techniques (e.g. Bayes Net, MultiLayer Perceptron Artificial Neural Network (MLP), Radial Basis Function Artificial Neural Network (RBF), Naïve Bayes Tree (NBTree), and Bagging). Shelokar et al. [167] incorporated the original ACO model with KM clustering. Two simulated and three UCI data sets were used to evaluate the performance of the proposed ACO-based clustering method. The ACO-based clustering method showed advantages in comparison with SA, GA, and TS in terms of quality of solution, average number of function evaluations, and processing time. Chen and Ye [169] proposed a PSO-based clustering method (PSO-clustering) and evaluated its performance on four artificial data sets. The obtained results indicated a better performance of PSO-clustering over those of KM and Fuzzy C-Means clustering algorithms. Senthilnath et al. [172] employed FA for clustering analysis. The performance of the FA-based clustering method was tested with 13 UCI data sets. The FA model demonstrated superiority in terms of clustering error

rates and computational efficiency over ABC, PSO, and nine other traditional classification methods (e.g. Bayes Net, MLP, and RBF).

Hatamlou et al. [174] formulated a hybrid clustering method, namely GSA-KM, by combining GSA and KM clustering. The GSA-KM method was tested with five UCI data sets. It demonstrated advantages in terms of quality of solutions and convergence speed in comparison with seven well-known algorithms, i.e. KM clustering, GA, SA, ACO, PSO, GSA, and Honey Bee Mating Optimization (HBMO). Hatamlou [176] also employed BHA to enhance the KM clustering performance. The BHA-based clustering method was tested with six UCI data sets. It demonstrated a better performance in comparison with those of KM clustering, GSA, and PSO. Moreover, Hatamlou et al. [177] also applied the Big Bang-Big Crunch algorithm (BB-BC) to clustering analysis. The BB-BC results outperformed those of KM clustering, GA, and PSO with several UCI data sets.

A number of modified metaheuristic search algorithms are available to further improve the performance of the original metaheuristic algorithm-based clustering methods. Das et al. [166] proposed a modified Bee Colony Optimization (MBCO) model by adopting both fairness and cloning concepts. The introduction of a fairness concept allowed bees with low probabilities to have a chance to be selected for enhancing search diversity. The employed cloning concept enabled the global best solution to be kept in the next iteration to accelerate convergence. Two hybrid clustering methods, namely MKCLUST and KMCLUST, were subsequently constructed based on MBCO. Additionally, a probability based selection method was introduced to allocate the remaining unassigned data samples to clusters. The MBCO method was evaluated with seven UCI data sets. It outperformed some existing algorithms, e.g. ACO, PSO, and KM clustering, while the proposed hybrid MKCLUST and KMCLUST models, on average, outperformed some existing hybrid methods, e.g. K-PSO (combination of PSO and KM), K-HS (combination of Harmony Search and KM), and IBCOCLUST (improved BCO clustering algorithm). In Niknam and Amiri [168], a hybrid evolutionary clustering model, namely FAPSO-ACO-K, was proposed by combining three traditional algorithms, i.e. FAPSO (fuzzy adaptive PSO), ACO, and KM. The proposed model was tested with four artificial and six UCI data sets. FAPSO-ACO-K was able to resolve the problem of initialization sensitivity in KM clustering. It outperformed other algorithms, such as PSO, ACO, SA, PSO-SA (combination of PSO and SA), ACO-SA (combination

of ACO and SA), PSO-ACO (combination of PSO and ACO), GA, and TS. Boushaki et al. [171] constructed a quantum chaotic Cuckoo Search (QCCS) algorithm using chaotic maps and nonhomogeneous update based on the quantum theory to increase global exploration. The QCCS model was tested with six UCI data sets. QCCS outperformed eight well-known methods, including GQCS (genetic quantum CS), HCSDE (hybrid CS and DE), KICS (hybrid KM and improved CS), CS, QPSO (quantum PSO), KCPSO (hybrid KM chaotic PSO), GA, and DE, for solving clustering problems. In Zhou and Li [173], two FA variants, namely the probabilistic firefly KM (PFK) and the greedy probabilistic firefly KM (GPFK), were proposed for data clustering. The PFK model employed a cluster channel array to store the probability of each data object belonging to each cluster in the encoding system. Instead of moving towards all brighter fireflies as in PFK, the GPFK algorithm adopted a greedy search strategy, in which each firefly only moved towards the brightest firefly in the swarm. The PFK and GPFK models outperformed KM clustering and FA based on the evaluation of four UCI data sets. Hassanzadeh and Meybodi [178] proposed a modified FA model (MFA) for clustering analysis. The MFA model not only employed neighbouring brighter fireflies but also the global best solution to provide guidance for the search process. The MFA model was evaluated with five UCI data sets. It outperformed three other clustering methods, including KM, PSO, and KPSO. Han et al. [175] proposed a modified GSA model for clustering analysis, namely BFGSA. The mean position of the seven nearest neighbours of the global best solution was used to enable the leader to escape from the local optima traps. Based on 13 UCI data sets, BFGSA outperformed nine classical search methods, including GSA, PSO, ABC, FA, KM, NM-PSO (fusion of Nelder-Mead simplex and PSO), K-PSO (fusion of KM and PSO), K-NM-PSO (fusion of KM, Nelder-Mead simplex and PSO), and CPSO (Chaotic PSO) [175]. A comprehensive survey on metaheuristic algorithms for partitioning clustering can be found in Nanda and Panda [179].

2.3 Feature selection and classification

The knowledge discovery processes in real-world applications often involve data sets with large numbers of features [180]. The high dimensionalities of data sets increase the likelihood of overfitting and impair generalization capability. Besides that, the inclusion of redundant or even contradictory features can also severely reduce the performance of ML algorithms [181]. As a result, feature selection and dimensionality reduction

become critical to overcome the above challenges by eliminating irrelevant and redundant features while identifying the most effective and discriminative ones [102, 182].

2.3.1 Classification

Classification is a sub-category of supervised learning which predicts labels of new observations based on the mapping from a set of features of training samples to the corresponding categorical class labels [183]. The authenticity of this mapping relationship between sample features and labels is crucial to classification performance. In other words, the identification of authentic feature representations with investigated problems is prerequisite for learning discriminative characteristics with respect to different categories and distinguishing data samples effectively. Moreover, the selection of appropriate classifiers also plays a critical role in determining classification performance. Currently, a variety of classifiers have been developed, e.g. Logistic Regression [184], Naïve Bayes Classifier [185], KNN [186], SVM [187], Decision Tree (DT) [188], RF [189], and ANNs [190], for tackling a wide range of real-life complex problems, such as image classification [191], handwriting recognition [192], spam filtering [193], speech recognition [194].

2.3.1.1 K-Nearest Neighbours

KNN [186] is one of the most popular classifiers owing to its simplicity and robust performance on large-scale training sets. It is a non-parametric classification technique that essentially relies on the fundamental assumption that observations with similar characteristics will tend to have similar outcomes. In KNN, an object is classified by the plurality vote of its neighbours whereby the most common label among its K nearest neighbours is assigned to the object. The similarity represented by distance measures plays an important role in the performance of KNN. Hence, a variety of distance measures have been developed in existing studies, such as Euclidean, Mahalanobis, Manhattan, Minkowski, Hamming, and Chebychev [195], distances.

Since few assumptions regarding the underlying data distributions are required, KNN is considered one of the most popular choices for undertaking classification problems in absence of any prior knowledge. Despite its simplicity and effectiveness, KNN is subject to expensive computational cost and intensive memory consumption, owing to the calculation of the distance measure from the instance to be classified to each stored

observation. Moreover, KNN is sensitive to data noise and subject to curse of dimensionality. Its performance also largely depends on the selection of an appropriate value for K, i.e. the number of nearest neighbours [196].

2.3.1.2 Support Vector Machine

SVM [187] is another popular classification algorithm. SVM creates a hyperplane that separates data samples into distinctive classes with the maximum margin, to reduce generalization error. Therefore, the instances from separate categories are divided by a clear gap owing to the maximization of the distance between the hyperplane and the nearest sample in SVM. The data samples which determine the position and orientation of hyperplane are hence named support vectors.

SVM is capable of performing nonlinear classifications through the employment of diverse kernel functions. To be specific, data samples are mapped into a hyperspace by the employed kernel function, such that the complicated sample distributions can be separated more easily. Functions commonly used as kernels include polynomial, sigmoid, radial basis function, as well as multi-layered perceptron [190]. Despite its good theoretical foundations and generalization capabilities, SVM still suffers from certain limitations, such as difficulty in the identification of the optimal kernel functions, high algorithmic complexity, as well as underperformance on noisy and imbalanced data sets [197].

2.3.2 Feature selection

The real-life classification problems often involve data sets with a significant number of features [198]. The high dimensionalities of real-life data sets increase the likelihood of overfitting and impair generalization capability owing to the inclusion of redundant or even contradictory features. Therefore, it is crucial to select the most discriminative features from raw data sets and enhance feature representations before feeding them into the classification algorithms as the inputs.

Feature selection approaches can be broadly divided into two categories, i.e. filter and wrapper methods. The filter approaches rank features individually based on certain statistical criteria, such as chi-square test [199], mutual information [200], Pearson correlation coefficients [201] etc. Features with higher rankings indicate their superior importance to the problem domain. However, it is challenging to identify the cut-off

point for selecting the most important features using filter methods. Besides that, the individual-based ranking mechanisms are incapable of measuring the confounding effects of feature interactions and feature composition [180]. In contrast, instead of measuring the impact of individual features to feature selection tasks, the wrapper methods evaluate the quality of various feature subsets by taking feature interaction into account, using the learning algorithm wrapped inside. Moreover, search strategies used to identify important feature subsets in the wrapper-based methods are generally divided into two categories, i.e. greedy search and stochastic search [180]. Greedy search, such as forward and backward selection, identifies local optimal solutions by following the problem-solving heuristic at each search step, whereas stochastic search based on EC is able to explore complex effects of feature interactions comprehensively owing to the significant capabilities of EAs in finding global optimality.

2.3.3 Evolutionary feature selection methods

EAs have been widely employed to comprehensively explore complex effects of feature interactions owing to their significant capabilities in finding global optimality [182]. In EAs-based feature selection methods, the coevolution mechanisms based on diverse evolving operators, e.g. crossover and mutation, are capable of producing various feature representations of the original problem in a single run. Therefore, the confounding effects of feature interactions can be thoroughly explored through the evaluation of validity of various feature constitutions during the iterative process. The effectiveness and superiority of various EAs over other methods in undertaking feature selection tasks have been extensively verified by existing studies, such as feature optimisation using GA [202], DE [203, 204], PSO [205, 206], FA [102, 207], ACO [208], GWO [209], WOA [210], and SCA [211].

As one of the most well-known EAs, PSO and its variants have been widely employed as the search engine in wrapper-based feature selection methods owing to its fast convergence speed and powerful discriminative search capabilities. Xue et al. [205] proposed two PSO-based multi-objective feature selection algorithms to achieve the trade-off between minimising the number of features and maximising classification accuracy. The first algorithm, i.e. NSPSOFS, incorporates the nondominated sorting from one of the most popular evolutionary multi-objective techniques, i.e. NSGAII (nondominated sorting GA II), into PSO to conduct multi-objective feature selection,

while the second feature selection model, i.e. CMDPSOFS, was proposed based on the idea of crowding, mutation, and dominance to enhance search diversity. In CMDPSOFS, a crowding factor together with a binary tournament selection is used to filter out certain crowed nondominated solutions. Additionally, the whole swarm is divided into three groups for the application of different mutation operators. Specifically, uniform mutation is operated on one group to enhance global search capabilities while non-uniform mutation is applied to a second group to improve local search capabilities. The third group operates without a mutation operator. Evaluated on 12 benchmark data sets, the NSPSOFS and CMDPSOFS models achieved better performance than those of LFS (Linear Forward Selection), GSBS (Greedy Stepwise Backward Selection), PSO (PSO with single objective function), 2SFS (PSO with a two-stage fitness function), NSGAII, SPEA2 (Strength Pareto Evolutionary Algorithm 2), and PAES (Pareto Archived Evolutionary Strategy). In particular, CMDPSOFS outperformed all baseline methods in terms of selecting fewer features as well as achieving higher classification performance. Gu et al. [212] employed a newly proposed Competitive Swarm Optimiser, i.e. CSO, to undertake high-dimensional feature selection tasks. In CSO, the swarm is randomly divided into two sub-swarms and pairwise competitions are conducted between particles from each sub-swarm. The winner particle in the competition is passed on to next generation while the defeated particle updates its position by learning from the position of winner particle in the cognitive component as well as the mean position of the swarm in the social component. Besides that, a social factor is employed to control the influence of the mean position of the swarm. The CSO model was evaluated on six challenging feature selection tasks with high dimensionalities from 360 to 6598. The empirical results indicates that CSO significantly outperforms PSO, PCA (Principal Component Analysis), as well as four PSO-based feature selection algorithms proposed by Xue et al. [206] with various initialisation strategies.

Moradi and Gholampour [213] proposed a hybrid PSO variant, i.e. HPSO-LS, for feature selection by integrating a local search strategy into the original PSO model. Two operators, i.e. "*Add*" and "*Delete*", are employed to enhance the local search of PSO. Specifically, the "*Add*" operator inserts the dissimilar features into the particle, while similar features are deleted from the particle by the "*Delete*" operator. Evaluated on 13 benchmark classification problems, HPSO-LS significantly outperforms four well-known filter-based feature selection methods, i.e. information gain, term variance, fisher

score and mRMR (minimum redundancy maximum relevance), as well as four classical methods, i.e. GA, PSO, ACO, and SA. Another hybrid PSO model, i.e. HPSO-SSM, was proposed by Chen et al. [74], which incorporates a spiral search mechanism for feature selection. Specifically, the Logistic chaotic map was used to generate the inertia weight. Subsequently, two dynamic nonlinear correction factors were employed as weights for the current position and velocity respectively in the original position updating formula to increase search scope. A spiral-based search action was also adopted to increase local exploitation capability. Evaluated on 20 UCI data sets, the results indicate that HPSO-SSM significantly outperforms other classical search methods, i.e. BBO, WOA, ABC, KH (Krill Herd algorithm), DE, and SCA, as well as several up-to-date feature selection methods, i.e. HPSO-LS, PSO(4–2), PSOLDA (a PSO approach for enhancing classification accuracy rate of Linear Discriminant Analysis), CatfishBPSO (binary PSO with catfish effect), WOASAT-2 (hybrid WOA with SA), ISEDBFO (improved swarming and elimination-dispersal BFO algorithm), and CMPSO (PSO with crossover and mutation operation). Tan et al. [214] proposed a hybrid learning PSO, i.e. HLPSO, to identify the most significant and discriminative elements from shape, colour, and texture features extracted from non-melanoma and melanoma dermoscopic images for the identification of malignant skin lesions. In HLPSO, the swarm is divided into two sub-swarms with top 50% ranked particles stored in one sub-swarm and the remaining particles stored in another sub-swarm. Three probability distributions, i.e. Gaussian, Cauchy, and Levy distributions, are used to conduct local jumps for the top 50% promising particles in the first sub-swarm, whereas three search mechanisms are conducted for the lower ranking particles in the second sub-swarm, i.e. a spiral search action based on the personal and global best solutions, and two modified FA operations guided by a randomly selected neighbouring brighter solution and the mean position of all brighter neighbouring solutions, respectively. The two sub-swarms merge after a number of iterations, then the crossover and mutation operators are employed to generate new offspring solutions using the top ranked particles as parents. HLPSO was evaluated using 11 basic benchmark functions, CEC 2014 test suite and two dermoscopic skin lesion data sets for feature selection. The evaluation results indicate that HLPSO demonstrates superior advantages in identifying the most discriminative lesion features and searching for the optimality of test functions with complex landscapes. The model significantly outperforms three classical search methods, i.e. PSO, FA, and MFO, five PSO variants, i.e. ELPSO (enhanced Leader

PSO), AGPSO (autonomous group PSO), DNLPSO (dynamic neighbourhood learning PSO), GPSO (genetic PSO), and GMPSO (PSO with GA and mutation techniques of Gaussian, Cauchy and Levy distributions), as well as five FA variants, i.e. FA with neighbourhood attraction (NaFA), MFA (a modified FA), VSSFA (FA with a variable step size), CFA1 (chaotic FA with Logistic map), and CFA2 (chaotic FA with Gauss map).

Moreover, Basset et al. [209] proposed a GWO variant integrated with a two-phase mutation, i.e. TMGWO, for feature selection. In each iteration, the two mutation operations are enforced on the best wolf solution by randomly deselecting the identified features and randomly adding unselected ones, respectively. As such, the exploitation capability during the process of feature selection is enhanced, owing to the simulation of the effects of reducing irrelevant features as well as adding informative ones, through the above two mutation operations. Evaluated on 35 UCI data sets, the devised TMGWO significantly outperformed a number of advanced search methods, e.g. Crow Search Algorithm (CSA), Multi-Verse Optimizer (MVO), and Non-Linear PSO (NLPSO). Faris et al. [215] proposed a novel feature selection method, i.e. TVBSSA-RWN, which incorporates a time-varying hierarchal binary Salp Swarm Algorithm (TVBSSA) and Random Weight Network (RWN). Specifically, instead of employing only one leader, a dynamic time-varying scheme of linearly increasing the number of leaders and decreasing the number of followers is designed for the construction of the leadership hierarchy in the salp swarm. Hence, the exploration capability is significantly enhanced owing to the increased proportion of large jumps across the search territory conducted by salp leaders. Moreover, Random Weight Network (RWN) is employed as the classifier in their proposed wrapper-based feature selection method. Beside the above, both the feature selection scheme and the number of neurons in the hidden layer of RWN are encoded in salp individuals for evolution. A fitness function incorporating the classification performance, the selected number of features, as well as the complexity of the generated RWN, is employed for fitness evaluation. TVBSSA-RWN was evaluated using 20 UCI data sets and significantly outperformed a number of existing feature selection methods, e.g. GA-KNN, PSO-KNN, binary GWO-RWN, and binary GSA-RWN. Souza et al. [216] proposed a binary Coyote Optimization Algorithm (BCOA) for feature selection. Intrinsically, in BCOA, the coyote population is divided into several sub-swarms with equal number of individuals. Each coyote

updates its position under the influence of the fittest coyote individual, the median individual in terms of the fitness rankings, as well as two randomly selected individuals, in the sub-swarm. Moreover, the worst coyote solution in each sub-swarm is replaced by the offspring solution yielded through the recombination of the elements from random selected individuals as well as random values within the decision bound. Besides the above, a v-shaped transfer function is developed for the binary transformation of coyote solutions. The Naïve Bayes classifier is employed for fitness evaluation in the wrapper-based feature selection method. Evaluated on seven UCI data sets, the results indicate that BCOA significantly outperforms a number of evolutionary feature selection methods, e.g. binary DA, binary Crow Search Algorithm (BCSA), as well as the classical methods, e.g. Sequential Forward Selection (SFS), Sequential Backward Selection (SBS). Moreover, the comprehensive reviews on the applications of EC techniques in tackling feature selection problems can also be found in [17, 182].

2.4 Deep neural networks

Despite its effectiveness and extensive applications, conventional ML techniques are subject to many limitations in terms of capabilities in processing large-scale natural data sets in their raw forms, e.g. images, videos, and texts. It requires significant amount of domain expertise and dedicated efforts to engineer raw data sets for extraction of appropriate feature representations, upon which ML algorithms are able to discover the underlying patterns embedded in the training data sets and generalise well on unseen data sets [217]. However, the recent rapid progress in DL has turned the tide of the traditional feature engineering and hand-crafted feature extraction. As representation learning based methods, DNNs are capable of automatic identification of effective feature representations with respect to the investigated problems at multiple levels, with higher-level features formed by the composition of features from lower levels of the hierarchy, owing to the stack of multiple feature-extracting layers in networks. Several essential DNNs models are introduced as follows.

2.4.1 Convolutional Neural Networks

CNNs [218] are a class of DNNs which retrieve data patterns automatically through the aggregation of feature maps at different levels. As shown in **Figure 2-3**, feature representation is learned through the collaboration of multiple layers with different functions, i.e. the convolutional layer (Conv), the rectified linear unit layer (ReLU), the

pooling layer (Pooling), as well as the fully connected layer (FC). Specifically, the convolutional layers perform convolution operations between learnable kernels and local regions of input data to produce various feature maps. The ReLU layers conduct elementwise activations to introduce nonlinearity to the network, whereas the pooling layers perform downsampling operations to reduce spatial size of the representation. As such, the high-level features can be extracted effectively from the raw data through the stack of the above three different types of layers. In addition, the fully connected layers perform nonlinear combinations on the acquired high-level features and classify the subject.

CNNs are capable of learning complex patterns by amplifying discriminant variations and supressing irrelevant ones contained in the input data through successive transformation of representations acquired by previous layers in networks. Since the breakthrough achieved by AlexNet on large-scale image classification [219], a great variety of network topologies and architectures have been developed to enhance network performances when undertaking diverse real-life challenges, e.g. Very deep Convolutional Networks (VGG) [220], Residual Neural Network (ResNet) [221], Inception Networks [222], You Only Look Once (YOLO) [223], Region with Convolutional Neural Network (R-CNN) [224], Generative Adversarial Networks (GANs) [13], etc. The above variants of CNNs have achieved state-of-the-art results on many challenging tasks of interest, e.g. semantic segmentation [225], image classification [226], object detection [227].

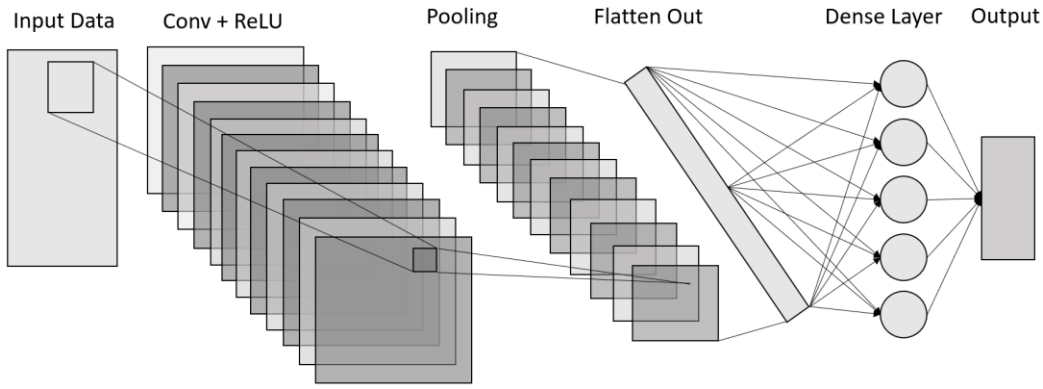

Figure 2-3 The illustration of the architecture of a vanilla CNN

2.4.2 Long Short-Term Memory

In addition to CNNs, Recurrent Neural Networks (RNNs) is another important class of DNNs. RNNs are capable of modelling implicit compositional representations in the temporal domain by incorporating previous experiences into the internal memory cells. However, the training of RNNs can be extremely difficult, especially when facing the tasks in which the temporal contingencies span long intervals, owing to the suffering of gradient vanishing and exploding [228]. Hence, LSTM [229] was invented to resolve above disadvantages of RNNs. In LSTM, a dedicated highway is formed to transport essential temporal information down to the entire cell chain, through the employment of multiple gate units, i.e. the forget, input, and output gates. The above gate units control information flow and modify cell memory by removing irrelevant information and selectively introducing new information in the cell states. The formulae of the gate units employed in LSTM are provided in **Eqs. 2.18 - 2.23**.

$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + b_f\right) \tag{2.18}$$

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \tag{2.19}$$

$$\tilde{c}_t = tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{2.20}$$

$$c_t = f_t c_{t-1} + i_t \tilde{c}_t \tag{2.21}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \tag{2.22}$$

$$h_t = o_t tanh(c_t) \tag{2.23}$$

where f_t, i_t, o_t denote the forget, input, and output gates, respectively, whereas x_t, h_t, c_t, $\tilde{c}_t$ represent the input data, the hidden state, the cell state, as well as the modulated input, at the time step t, respectively. Besides, W_{xf}, W_{xi}, W_{xo} represent the weights of the input x_t for the forget, input, and output gates, respectively, whereas W_{hf}, W_{hi}, W_{ho} denote the weights of the previous hidden state h_{t-1} for the above three gates, respectively. Moreover, b_f, b_i, b_o are the bias for the three gates in the neuron, respectively. σ and $tanh$ represent sigmoid and hyperbolic tangent activation functions, respectively.

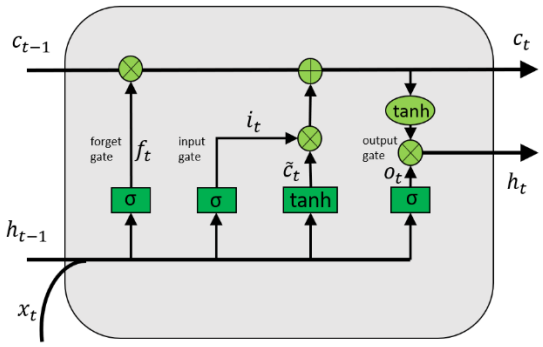

Figure 2-4 The illustration of a building block in LSTM

As shown in **Figure 2-4**, LSTM memory cell contains three types of gate units for the processing of cell information, i.e. forget gate f_t, input gate i_t, and output gate o_t, as well as two types of activation functions, i.e. sigmoid function σ and hyperbolic tangent function $tanh$. The sigmoid function σ squashes cell values into [0, 1] to regulate the flow of the information through the gate, i.e. 0 means no flow, 1 means complete flow, whereas the hyperbolic tangent function $tanh$ squashes cell values into [-1, 1], allowing for both increase and decrease of the cell states.

The concatenation of the input data x_t and the previous hidden state h_{t-1} is fed into each of the three gate units to yield numerical values for the control of information flow in the cell, respectively. Specifically, the values yielded by the forget gate f_t tend to remove nonessential information from the previous cell state c_{t-1}, as shown in **Eq. 2.18**. The values generated by the input gate i_t determine the amount of new information to be stored in the current cell state c_t, as shown in **Eqs. 2.19 - 2.21**. Moreover, the values produced by the output gate o_t determine the amount of the information to be passed on to future memory cells from the current cell state c_t, as shown in **Eqs. 2.22** and **2.23**. As such, this design of the gate units enables LSTM to capture complex and long-term dependencies more effectively, owing to its capabilities of retaining effective contingencies from previous cell sates and supressing nonessential information without suffering from gradient vanishing.

2.4.3 Convolutional Neural Network-Long Short-Term Memory

In order to harness the advantages from distinctive DL models, advanced composite networks through the hybridization of different network structures have been explored

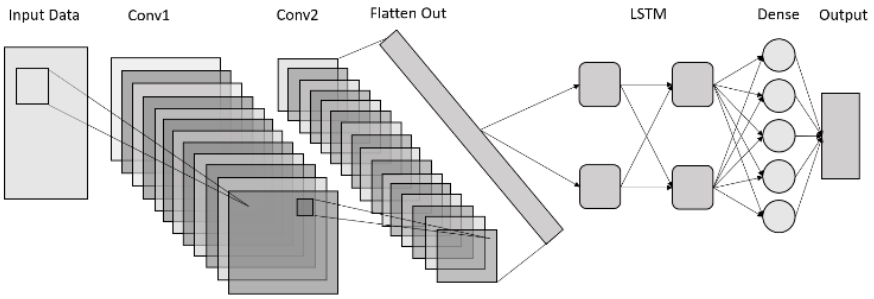

Figure 2-5 The illustration of the architecture of CNN-LSTM

in existing studies. The typical instance is the hybridization between CNN and LSTM, namely CNN-LSTM [230], as shown in **Figure 2-5**.

CNN-LSTM networks have attracted many research attentions owing to its great advantages in combining the strength of automatic feature extraction in CNN, as well as the capability of capturing long-term dependencies in LSTM. The convolutional layer in CNN-LSTM disentangles the cross-correlations while preserving deterministic and stochastic trends embedded among the input time series. Therefore, it produces more accurate feature representations, which enables LSTM layers to learn temporal dependencies more precisely. The CNN-LSTM networks have been applied to tackle a variety of time series prediction and classification problems successfully, e.g. stock market forecasting [231], named entity recognition [232], textual sentiment analysis [233, 234], machine translation [235], facial expression recognition [236] and image description generation [237].

2.4.4 Evolving deep neural networks

The performance of DNNs is largely dependent on the configurations of their respective architectures and hyperparameter settings. However, the search for the optimal network configuration is extremely challenging owing to network complexity and heavy computational cost of the learning processes. As characterised with superb global search capabilities, EAs have been leveraged to evolve deep learning neural networks for the identification of the optimal learning configurations as well as the discovery of innovative network structures.

Sun et al. [238] proposed an automatic CNN architecture design method based on the GA. In this method, a generic CNN structure consisting of predefined building blocks is employed as the foundation for the automatic architecture generation. Specifically, a

building block with two convolutional layers and one skip connection is employed for the benefits of increasing network depth without risking gradient vanishing, whereas the fully connected layers are discarded for the consideration of reducing the likelihood of overfitting resulted from the dense connection. As a result, parameters encoded in the GA chromosomes include filter numbers of convolutional layers in each building block and the pooling layer type, with the length of chromosomes representing the depth of the network. The population undergoes the evolving process of the crossover operation, as well as the mutation process. The latter incorporates four options, i.e. adding a skip layer, adding a pooling layer, removing the layer at the selected position, and changing the parameters of the building block randomly. Their proposed method is evaluated on CIFAR10 and CIFAR100 data sets. The results indicate its great superiorities in improving classification performance while significantly reducing the amount of parameters, in comparison with manually designed CNNs, e.g. Resnet (depth=110), as well as the models derived from the combined schemes of automatic and manual tuning, e.g. Efficient Architecture Search (EAS) and Differential Architecture Search (DARTS). Sun et al. [239] proposed an evolving deep CNN (EvoCNN) model based on the GA for image classification. A variable-length gene encoding strategy is formulated to represent each potential network configuration. Two statistical measures, i.e. the mean and standard deviation values, are used to represent the weight parameters in the encoding strategy. During fitness evaluation, Gaussian distribution is employed to decode the weights based on the two statistical measures. The network architecture recommended by each chromosome as well as its corresponding decoded weights is adopted in fitness evaluation. Besides the classification performance (i.e. the mean and standard deviation of the classification error rates), the network parameter size is also considered in chromosome evaluation. A slack binary tournament selection strategy is devised for the parent chromosome selection where the mean classification performance and the parameter size are used as the threshold criteria. A unit alignment crossover operator is proposed to exchange gene information of the two parent solutions with different lengths. Evaluated with nine popular image classification data sets (e.g. Fashion, Rectangle, MNIST and its variant data sets), the EvoCNN model outperforms a number of competitive benchmark deep architectures.

Deep network generation with ResNet and DenseNet blocks based on the GA is examined by Sun et al. [240]. Specifically, an automatically evolving CNN (AE-CNN)

model is designed to yield the CNN architectures with residual and dense connectivity. A one-point crossover operator is used for offspring solution generation, while three types of mutation operations (i.e. adding, removing, and modifying) are employed to further configure the networks. Evaluated with the CIFAR10 and CIFAR100 data sets, the AE-CNN model performs favourably as compared with a number of hand-crafted architectures and automatically devised networks from some existing methods. Despite the promising results and the great potential of the evolutionary deep learning models with respect to knowledge discovery, they are inadvertently subject to a considerably high computational cost. To overcome this drawback, Sun et al. [241] proposed an end-to-end performance predictor (E2EPP). A random forest is used to predict the network performance. The AE-CNN model is initially employed to produce a set of CNN architectures. These network configurations are subsequently encoded into numerical decision variables, which are used in conjunction with the corresponding network accuracy rates for training the random forest-based performance predictor. Specifically, a predictor pool is generated, where each base tree model is trained using data samples containing randomly selected subsets of features. To increase ensemble robustness, a subset of base evaluators is selected to evaluate any newly created architectures based on their prediction performances with respect to the current best CNN architecture. The E2EPP model outperforms two existing performance predictors and advanced deep networks in terms of classification performance and computational efficiency.

Martín et al. [242] employed a Hybrid Statistically-driven Coral Reef Optimization (HSCRO) algorithm to reconstruct the fully connected layers in VGG-16 for two purposes, i.e. reducing the amount of parameters and improving model performance. Each coral individual represents a set of fully connected layers in VGG-16. Four types of parameters are encoded in each layer, i.e. activation function, number of neurons, matrix of connection weights, and bias. The HSCRO model incorporates four evolutionary operators, i.e. asexual reproduction, sexual reproduction, settlement, and depredation, to emulate the reproduction process of coral reefs. In addition, a stratified mutation scheme is designed in which 20% of best individuals undergo parametric mutations on weights and biases, whereas the remaining 80% of individuals experience structural mutations, i.e. mutations on activation functions, the number of nodes, and node connections, during the evolving process. The identified best solution is further fine-tuned using stochastic gradient descent (SGD) optimizer. The proposed evolving

CNN model is tested on two image classification data sets, i.e. CIFAR10 and CINIC10, capable of reducing 90% of the connection weights while improving the classification accuracy as compared with the VGG-16 model.

In addition to evolving CNN models, there are also studies on evolving RNN and LSTM models. Rawal and Miikkulainen [243] proposed a Genetic Programming (GP) based evolving LSTM architecture generation system, capable of constructing layered network structures from a single recurrent node design. The recurrent node is encoded as a tree structure with two types activation operations, i.e. linear activations with two elements (add and multiply), and nonlinear activations with one element (tanh, sigmoid, or relu). A homologous crossover operator is designed to yield offspring solutions by crossing over the same regions of the two parents represented in tree structures during reproduction. Besides that, three types of mutation operations are designed for the evolution of tree solutions, i.e. (1) replacing one activation operation with another within the same category, (2) inserting a new branch at a random position in a tree, and (3) shrinking a branch by replacing it with a randomly selected operation employed in this branch. Also, individual solutions with previously explored branch structures undergo repeated mutation procedures until new tree structures are generated, to maintain population diversity. In addition, two architecture generation schemes are experimented, i.e. a homogenous evolving process using a single recurrent node within a LSTM layer vs. a heterogenous evolving process using the combination of nodes with different structures. Their evolving LSTM model is evaluated using two tests, i.e. a language modelling test and an automatic music transcription test, and outperforms existing advanced models, e.g. the neural architecture search method (NAS) and Recurrent Highway Network (RHN). Kim and Cho [244] developed a PSO-based evolving CNN-LSTM network for the prediction of energy consumption. The original PSO algorithm is applied to search for the optimal hyperparameters of CNN-LSTM, e.g. filter numbers and sizes in convolutional layers, and the number of hidden nodes in recurrent layers, for retrieving energy consumption patterns. The results indicate that their evolving CNN-LSTM model significantly outperforms classical models, e.g. Linear Regression, DT, and RF, for energy consumption prediction. Xue et al. [245] proposed an evolving CNN-LSTM method to tackle the inventory forecast problem. PSO and two DE variants, i.e. DE with binominal and exponential crossover operators respectively, are employed for the identification of the optimal CNN-LSTM

hyperparameters, including the filter number and size in the convolutional layer, pooling type, pooling size, and stride size with respect to the pooling layer, as well as dropout rate and the numbers of nodes in LSTM layer and dense layer, respectively. The results indicate that the DE with exponential crossover operator achieves the best performance in forecasting inventory and demonstrates more advantages on identifying proper CNN-LSTM hyperparameters in comparison with PSO as well as DE with binominal crossover operator. Furthermore, a systematic review on designing deep neural networks using neuro-evolution is provided in [246].

2.5 Summary

EC optimization techniques have become powerful tools to eliminate noises and redundant features in raw data sets, mitigate inherent limitations residing in conventional ML models, as well as discover effective and innovative network topologies for DL models. A great variety of hybrid models between EC and ML have been developed in existing studies, and it is unlikely to cover all of them in detail in this thesis. Therefore, in this chapter, the state-of-the-art hybrid models of interest in relation to feature selection, cluster centroids optimization, as well as neural architecture search are reviewed particularly. Besides above, three popular models in EAs are introduced, namely PSO, FA, and GWO. Their limitations are analysed in detail and serve as the motivations for proposing enhanced optimization methods. The proposed improved EAs are employed to overcome three challenging obstacles in ML and data mining, i.e. initialization sensitivity, feature selection, and hyperparameter optimization, as presented in **Chapters 3**, **4**, and **5**, respectively.

Chapter 3

Evolutionary K-means Clustering with Enhanced Firefly Algorithms

In this chapter, two variants of the FA, namely inward intensified exploration FA (IIEFA) and compound intensified exploration FA (CIEFA), are proposed for undertaking the obstinate problems of initialization sensitivity and local optima traps of the KM clustering model. To enhance the capability of both exploitation and exploration, matrix-based search parameters and dispersing mechanisms are incorporated into the two proposed FA models. Specifically, the attractiveness coefficient is replaced with a randomized control matrix in the IIEFA model to release FA from the constraints of biological law, as the exploitation capability in the neighbourhood is elevated from a one-dimensional to multi-dimensional search mechanism with enhanced search diversity in scopes, scales, and directions. Besides that, a dispersing mechanism is employed in the CIEFA model to dispatch fireflies with high similarities to new positions out of the close neighbourhood to perform global exploration. This dispersing mechanism ensures sufficient variance between fireflies in comparison to increase search efficiency. The ALL-IDB2 database, a skin lesion data set, and a total of 15 UCI data sets are employed to evaluate efficiency of the proposed FA models on clustering tasks. The minimum Redundancy Maximum Relevance (mRMR)-based feature selection method is also adopted to reduce feature dimensionality. The empirical results indicate that the proposed FA models demonstrate statistically significant superiority in both distance and performance measures for undertaking diverse clustering tasks, in comparison with conventional KM clustering, five classical search methods, and five advanced FA variants.

3.1 The proposed evolutionary K-means clustering models

FA is chosen to construct the hybrid clustering models owing to its unique property of automatic subdivision and its advantages in tackling multimodal optimisation problems with sub-optimal distraction and high nonlinearity [95, 111-114, 247]. However, the original FA model has certain limitations in terms of search diversity and efficiency. More specifically, search diversity of FA is severely constrained owing to its diagonal-based search paradigm resulted from the strict adherence to the biological laws, as

analysed in detail in **Section 2.1.2**. As a consequence, it is likely that fireflies tend to overlook promising search directions and dimensions inadvertently distant from the prescribed diagonal trajectory during movement. On the other hand, search efficiency in FA is also undermined owing to the lack of consideration in terms of fitness distinctiveness when one firefly approaches other brighter ones in the neighbourhood. As a result, many movements may become futile and ineffective, unable to navigate fireflies to a more promising region, since there is little difference between fitness scores before and after the movement. Therefore, two modified FA models are proposed, namely IIEFA and CIEFA, to overcome limitations of the original FA model and mitigate the problems of initialization sensitivity and local optima traps of KM clustering. The proposed models intensify the diversification of exploration both in the neighbourhood and global search space, and lift the constraints of the biological laws in the original FA model. We introduce the proposed models in detail in the following sub-sections.

3.1.1 The proposed inward intensified exploration FA (IIEFA) model

The aim of IIEFA is to expand the one-dimensional search in the original FA model to a multi-dimensional scale by replacing the attractiveness term $\beta_0 e^{-\gamma r_{ij}^2}$ with a random matrix μ, as illustrated in **Eq. 3.1**.

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \boldsymbol{\mu}\left(\boldsymbol{x}_j^t - \boldsymbol{x}_i^t\right) + \alpha_t \boldsymbol{\varepsilon}_t \tag{3.1}$$

$$\alpha_{t+1} = \alpha_t \theta \tag{3.2}$$

where $\boldsymbol{\mu}$ denotes a control matrix where each element is drawn from [0, 1] randomly, while α_t denotes an adaptive randomization step based on a geometric annealing schedule. θ is a constant value which is employed to gradually diminish the randomness imposed by the adaptive step α_t and achieve the trade-off between exploration and exploitation through the search process course. Specifically, large settings for θ are likely to undermine search efficiency owing to the overwhelming impacts of large random jumps, whereas small settings for θ result in premature convergence owing to the lack of search randomness. According to [60], θ is recommended to have a value in the range of 0.95 to 0.99. We set θ to 0.97 in this study, in accordance with the recommendation in [60] and several trial-and-error results in our experiments. This adaptive randomization step enables the search process to start with a larger random

step to increase global exploration and fine-tune the solution vectors in subsequent iterations with a smaller search parameter.

By multiplying the control matrix, μ, each dimension of the position difference $(\boldsymbol{x}_j^t - \boldsymbol{x}_i^t)$ between two fireflies is assigned with a unique random number in [0, 1], therefore being shrunk disproportionately with various magnitudes. Subsequently, the resulting solutions after this operation can be any vectors originated from the current firefly solution, randomly distributed in the rectangular area in comparison with residing in the dotted diagonal line as in original FA model, as illustrated in **Figure 2-1**. The random control matrix operation possesses two-fold advantages. Firstly, the search directions in the neighbourhood are not constrained to the diagonal line, but become more diversified. Secondly, the movement scales become more diverse owing to the impact of various magnitudes on each dimension. **Figure 2-1** provides an example of possible directions and scales in the neighbourhood search, indicated by vectors with arrows within the rectangular. Therefore, IIEFA possesses a better search capability by extending exploration of fireflies from a one-dimensional diagonal direction to a multi-dimensional space in the neighbourhood. In other words, exploration of the swarm increases along with the firefly congregation process. This first proposed FA variant is hereby characterized as an inward intensified exploration FA model. The pseudo-code of IIEFA is presented in **Algorithm 3-1**.

Algorithm 3-1 – The pseudo-code of the proposed IIEFA model

1 Start

2 Initialize a population of m fireflies

3 Initialize randomization parameter α_t and set experimental parameters

4 Define the objective function/light intensity $I = f(x)$

5 Calculate light intensity for each firefly

6 While (t< Max iteration) or (other converging criteria not being met)

7 {

8 **For** $i <= m$

9 {

10 **For** $j <= m$

11	{
12	**If** $I_i < I_j$
13	{
14	Generate a control matrix μ
15	Update the position of firefly i by moving towards firefly j using **Eq. 3.1**
16	} **End If**
17	Check the new position not to exceed the range of problem variables
18	} **End For**
19	} **End For**
20	Update the randomization step α_t using **Eq. 3.2**
21	} **End While**
22	Export the global best position P_g, and global best fitness value I_g
23	**End**

3.1.2 The proposed compound intensified exploration FA (CIEFA) model

In the original FA model, after being initiated, the whole firefly swarm tends to congregate continuously until convergence at one point. As such, the search process can be deemed as an inward contracting process, no matter how early the search stage is, or how close or similar two neighbouring fireflies are. Consequently, the approaching movement between fireflies with similar light intensities (i.e. fitness scores) at an early stage is more likely to result in waste of the resource, since the fitness score of the current firefly is very unlikely to be drastically improved under this circumstance by following the neighbouring slightly better solution, but with a high probability of being trapped in local optima. Therefore, we propose the second FA variant, i.e. a compound intensified exploration FA (CIEFA) model, by integrating both inward and outward search mechanisms to overcome this limitation inherent in the original FA model. This new CIEFA model is produced based on the first IIEFA model. Specifically, CIEFA combines the inward exploration strategy embedded in IIEFA with a newly proposed dispersing mechanism based on dissimilarity measures to increase diversification. **Eq. 3.3** defines the proposed dissimilarity measure $M_{dissimilarity}$ between two fireflies.

$$M_{dissimilarity} = \left(I_j^t - I_i^t\right)/\left(I_g^t - I_i^t\right) \quad (3.3)$$

where I_i^t and I_j^t represent the fitness scores of fireflies i and j, respectively, in the t^{th}iteration, while g represents the current global best solution, and I_g^t denotes its fitness score in the t^{th} iteration.

As illustrated in **Eq. 3.3**, we employ $M_{dissimilarity}$ to distinguish fireflies with weak or strong light intensity differences to that of the current firefly, whereby the neighbouring solutions, with $M_{dissimilarity} < 0.5$, are labelled as 'ineffective individuals', whereas those with distinctive variance in light intensities, i.e. $M_{dissimilarity} > 0.5$, are labelled as 'effective individuals', through the position updating process. **Eqs. 3.4** and **3.5** define the outward search operation for the 'ineffective individuals', with $M_{dissimilarity} < 0.5$. This new outward search operation enables firefly i to not only perform local exploitation of firefly j, but also force firefly i to jump out of the space between i and j so as to explore an outer space. It expands search exploration of the weaker firefly i to accelerate convergence. On the contrary, when $M_{dissimilarity} > 0.5$, the inward intensified exploration formula in IIEFA is used to dispatch firefly i using 'effective individuals'.

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_j^t + \boldsymbol{\varphi}\,\boldsymbol{\tau}\left(\boldsymbol{x}_j^t - \boldsymbol{x}_i^t\right) + \alpha_t \boldsymbol{\varepsilon}_t \quad (3.4)$$

$$\boldsymbol{\tau} = (1 - t/T_{total})\,(1 + \boldsymbol{\mu}) \quad (3.5)$$

In **Eq. 3.4**, $\boldsymbol{\tau}$ denotes a step control matrix for this new outward operation, while $\boldsymbol{\varphi}$ represents a direction control matrix with each element being drawn randomly from -1 and 1. The step control matrix, $\boldsymbol{\tau}$, for the outward search operation is further defined in **Eq. 3.5**, where t represents the current iteration number while T_{total} is the maximum number of iterations. Parameter $\boldsymbol{\mu}$ denotes the control matrix that consists of random numbers in [0, 1], as defined earlier in IIEFA, with the same feature dimension as that of the firefly swarm.

The step control matrix, $\boldsymbol{\tau}$, is employed to regulate the extent of outward exploration in each dimension and the balance between exploration and exploitation through the whole search process. Owing to the randomness introduced by the control matrix, $\boldsymbol{\mu}$, in IIEFA, as defined in **Eq. 3.1**, the elements in $\boldsymbol{\tau}$ possess different values from each other, but all follow the same trend of variation as the iteration number builds up. As an example, the

change of one element from $\boldsymbol{\tau}$ against the iteration number is illustrated in **Figure 3-1**. This example element in $\boldsymbol{\tau}$ decreases from 2 to 0, governing the exploration scale on each dimension as the count of iterations builds up. The exploration operation is conducted outwardly when the element in $\boldsymbol{\tau}$ is greater than 1, otherwise the exploration operation is performed inwardly.

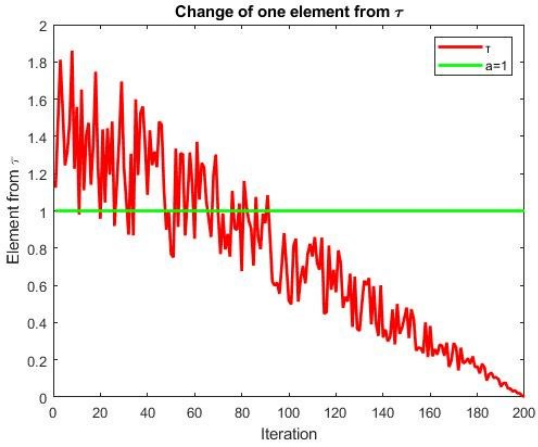

Figure 3-1 An example of the change of one element from the step control matrix, τ, through iterations

Based on the variance of the element in **Figure 3-1**, it is observed that the whole search process of 'ineffective individuals' with low fitness dissimilarities ($M_{dissimilarity} < 0.5$) goes through three stages as the iteration builds up. In the first stage, the outward exploration action dominates the first 50 (out of 200) iterations approximately, where the 'ineffective individuals' are dispersed to explore a greater unexploited search domain. In the second stage, both inward and outward explorations reside in the 50th-90th iterations, in order to balance between exploitation and exploration. In the third stage, the inward exploration operation replaces the outward exploration movement, and takes control once the number of iterations exceeds 90, as the whole swarm gradually congregates and converges altogether. It should be noted that the iteration numbers used for the division of three search modes fluctuate slightly around the thresholds given in the illustrated example in **Figure 3-1**, since the randomness of $\boldsymbol{\mu}$ affects the magnitude of elements in τ delicately. Nevertheless, the general adaptive patterns coherently apply to the whole search process with respect to all dimensions in fireflies. Moreover, each element (either -1 or 1) in $\boldsymbol{\varphi}$ controls the direction of the movement along each

corresponding dimension, which enables fireflies to fully explore and exploit the search space.

The whole search process of 'ineffective individuals' with low dissimilarity levels ($M_{dissimilarity} < 0.5$) is depicted in **Figure 3-2**. With the assistance of three different position updating operations (indicated in three colours) in **Figure 3-2**, not only the search diversity in direction and scope among fireflies with high similarities is improved significantly and local stagnation is mitigated effectively. The search efficiency is also enhanced because of the guarantee of heterogeneity between fireflies in movement. On the other hand, the movement of 'effective individuals' with distinctive position variance follows the same strategy in IIEFA, as illustrated in **Eq. 3.1**. In short, CIEFA enhances diversity of exploration one step further, and inherits all merits by combining both inward and outward intensified exploration mechanisms.

Moreover, according to the empirical results, the proportion of calling the dispersing search mechanism in CIEFA for 'ineffective individuals' among the total number of position updating varies slightly, and is dependent on the parameter settings (e.g. the maximum number of iterations and the size of the firefly population) as well as the problems at hand (e.g. the employed data sets). Taking the Sonar data set as an example, the proportion of running the dispersing mechanism varies between 40% and 52% for each trial with a population of 50 fireflies and a maximum number of 200 iterations. The average proportion of calling the dispersing mechanism in CIEFA over a series of 30 trials is 47.18% under the same setting. The pseudo-code of CIEFA is provided in **Algorithm 3-2**.

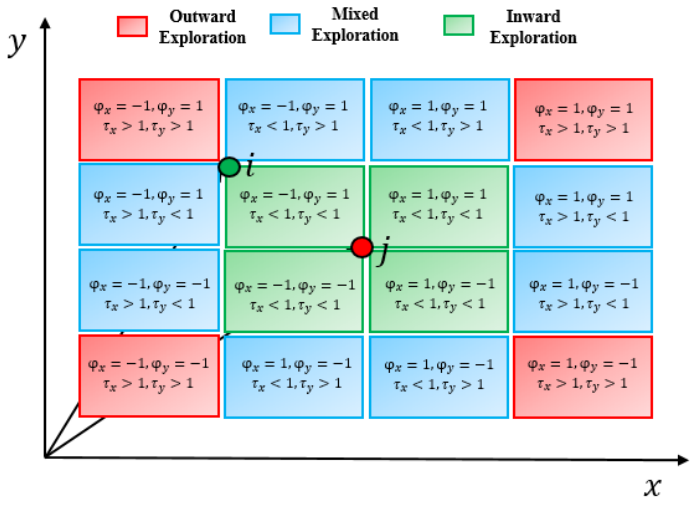

Figure 3-2 Distribution of the updated positions of firefly i through iterations in the CIEFA model in a two-dimensional search space when $M_{dissimilarity} < 0.5$

Algorithm 3-2 – The pseudo-code of the proposed CIEFA model
1 Start
2 Initialize a swarm of m fireflies
3 Initialize randomization parameter α_t and set experiment parameters
4 Define the objective function/light intensity $I = f(x)$
5 Calculate light intensity for each firefly
6 While (t< Max iteration) or (other converging criteria not being met)
7 {
8 **For** $i <= m$
9 {
10 **For** $j <= m$
11 {
12 **If** $I_i < I_j$
13 {
14 Calculate $M_{dissimilarity}$ using **Eq. 3.3**
15 Generate a random matrix μ
16 **If** $M_{dissimilarity} < 0.5$

17	{
18	Calculate control matrix τ using **Eq. 3.5**
19	Generate direction matrix φ
20	Update position of firefly i by moving towards j using **Eq. 3.4**
21	**Else** $M_{dissimilarity} \geq 0.5$
22	Update position of firefly i by moving towards j using **Eq. 3.1**
23	} **End If**
24	Check the new position not to exceed the range of variables
25	} **end if**
26	} **End For**
27	} **End For**
28	Update α_t using **Eq. 3.2**
29	} **End While**
30	Export the global best position P_g, and the global best fitness value I_g
31	**End**

3.1.3 The proposed clustering approach based on the IIEFA and CIEFA models

The proposed IIEFA and CIEFA algorithms are subsequently employed to construct two novel clustering models to undertake initialization sensitivity and local optima traps of the original KM clustering algorithm. The flowchart and pseudo-code of the proposed clustering method are presented in **Figure 3-3** and **Algorithm 3-3**, respectively.

Algorithm 4-3 – The pseudo-code of the proposed clustering method
1 Start
2 Import data sets and set initial parameters
3 Initialize a firefly swarm S as a series of possible cluster centroids
4 Run KM on the data set and generate the initial cluster centroids C_o as a seed solution

5 Replace the first firefly in the swarm S with C_o	
6 while (t< Max iteration) or (other termination criteria not being met)	
7 {	
8	Use each firefly as the centroids to cluster the data based on Euclidean distance
9	Evaluate fitness value/light intensity of each firefly using the sum of intra-cluster distance measure f as defined in **Eq. 3.6** in **Section 3.2.3**
10	Update firefly positions using the proposed IIEFA/CIEFA models
11 } **End While**	
12 Export the global best position P_g, and the global best fitness value I_g	
13 End	

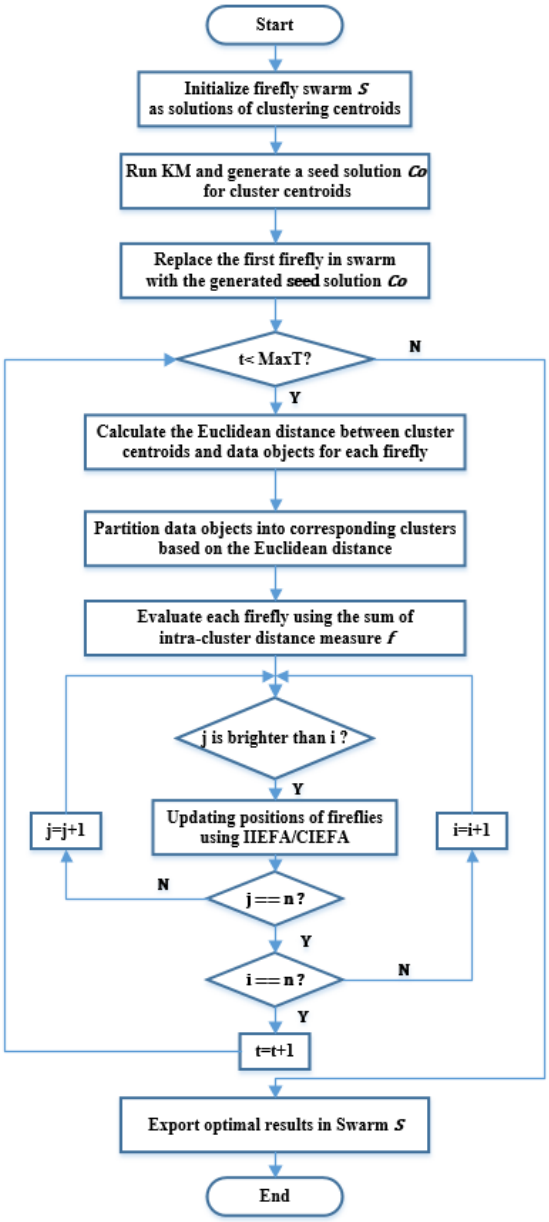

Figure 3-3 Flowchart of the proposed clustering method

In order to improve search efficiency and increase convergence, a seed solution for cluster centroids is generated firstly by the original KM clustering algorithm, and is used to replace the first firefly in the swarm. The similarities among data samples are measured by the Euclidean distance during the partitioning process. Quality of the centroid solution represented by each firefly is evaluated based on the sum of intra-cluster distance measures. The search process and movement patterns of the swarm are governed and regulated by the proposed IIEFA and CIEFA models. Benefited from the

enhanced diversity of the search scopes, scales, and directions in IIEFA and CIEFA, a cluster centroid solution with a better quality is identified through the intensified neighbouring and global search processes, and the possibility of being trapped in local optima is significantly reduced.

Moreover, as mentioned earlier, nearly all the hybrid KM-based clustering models partition data samples into the corresponding clusters based on the Euclidean distance, and quality of clustering centroids is improved by minimising the sum of intra-cluster distance measures. Therefore, irrelevant and redundant features contained in the data samples can negatively impact the distance-based clustering measures, since the distance measures under such circumstances are not able to represent the compactness of the clusters accurately. Owing to the high dimensionality of some of the data sets evaluated in this study, e.g. 80 for ALL, 72 for Ozone, and 60 for Sonar, and the implementation of feature selection on the these data sets as validated in previous studies [102, 248], we employ mRMR [249] to conduct feature dimensionality reduction and improve clustering performance by eliminating redundant and irrelevant features. A comprehensive evaluation of the proposed clustering method is presented in the next section.

3.2 Evaluation and discussion

To investigate the clustering performance in an objective and comprehensive manner, the proposed FA models are evaluated and compared with not only FA related methods, but also several other classical metaheuristic search methods. In view of their novelties and contributions to the development of a variety of metaheuristic algorithms, GA and ACO are two most successful metaheuristic search methods [250]. As such, we evaluate and compare the proposed IIEFA and CIEFA models against GA [55], ACO [251], and four other classical methods i.e. KM clustering, FA [60], Dragonfly (DA) [64], and Sine Cosine Algorithm (SCA) [252], as well as five FA variants i.e. CFA1 [105], CFA2 [106], NaFA [96], VSSFA [107], and MFA [178]. Each optimization model is integrated with KM clustering for performance comparison. A total of ten data sets characterised with a wide range of dimensionalities are evaluated with five performance indicators, namely sum of intra-cluster distances (i.e. fitness scores), average accuracy [253], average sensitivity, average specificity and macro-average F-score (Fscore_M) [253]. To ensure a fair comparison, we employ the same number of function evaluations

(i.e. population size × the maximum number of iterations) as the stopping criterion for all the search methods. The population size and the maximum number of iterations are set to 50 and 200, respectively, in our experiments. We also employ 30 independent runs in each experiment, in order to mitigate the influence of fluctuation of the results.

3.2.1 Parameter settings

The parameter settings of search methods employed in our study are adopted in accordance with recommendations in their original studies. As such, the following initial parameters are applied to both the original FA model and FA variants, in accordance with the empirical study in [247], i.e. initial attractiveness=1.0, absorption coefficient=1.0, and randomization parameter=0.2, while the proposed IIEFA and CIEFA models employ randomized search parameters as indicated in **Section 3.1**. The details of parameter settings for each search method are listed in **Table 3-1**.

Table 3-1 Parameters settings for each algorithm

Algorithms	Parameters
FA [247]	initial attractiveness = 1.0, absorption coefficient = 1.0, randomization parameter = 0.2, adaptive coefficient = 0.97
CFA1 [105]	chaotic component = $\delta \times x^{(n)}$, where $\delta = 1 - \lvert\frac{n-1}{n}\rvert^{0.25}$, $x^{(n)}$ represents chaotic variable generated by Logistic map, and n represents current iteration number. Other parameters are the same as those of FA.
CFA2 [106]	attractiveness coefficient = Gauss map, with the rest parameters the same as those of FA
NaFA [96]	size of neighbourhood brighter fireflies=3, other parameters the same as those of FA
VSSFA [107]	adaptive randomization step = $0.4/(1 + exp(0.015 \times (t - max\_iteration)/3))$, where t and $max\_iteration$ represent current and maximum iteration numbers, respectively. Other search parameters are the same as those of FA.
DA [64]	separation factor = 0.1, alignment factor = 0.1, cohesion factor = 0.7, food factor = 1,

	enemy factor = 1, inertial weight = $0.9 - m \times ((0.9 - 0.4)/ max\_iteration$, where m and $max\_iteration$ represent current and maximum iteration numbers, respectively.
SCA [252]	$r_1 = a - t \times a/T$, where $a = 3$ and t and T represent current and maximum iteration numbers, respectively. $r_2 = 2\pi \times rand$, $r_3 = 2 \times rand, r_4 = rand$
MFA [178]	Parameter settings are the same as those of FA
GA [60]	crossover probability=0.8, mutation probability=0.05
ACO [251]	locality of the search process $= 10^{-4}$, pheromone evaporation rate = 0.85
IIEFA	control matrix $\mu \in (0, 1)$, with other search parameters the same as those of FA
CIEFA	step control matrix $\tau = (1 - t/T_{total}) \times (1 + \mu)$, where $t\ and\ T_{total}$ represent current and maximum iteration numbers, respectively. Other search parameters are the same as those of FA.

3.2.2 Data sets

Clustering performance is significantly influenced by characteristics of data samples, such as data distribution, noise, and dimensionality. Therefore, the following data sets with various characteristics from different domains are used to investigate efficiency of the proposed models. Specifically, we employ the ALL-IDB2 database [254], denoted as ALL (Acute Lymphoblastic Leukaemia), and nine data sets from the UCI machine learning repository [255], namely Sonar, Ozone, Wisconsin breast cancer diagnostic data set (Wbc1), Wisconsin breast cancer original data set (Wbc2), Wine, Iris, Balance, Thyroid, and E.coli, for evaluation. Among the selected data sets, Sonar, Ozone and ALL possess relatively high feature dimensionality, i.e. 60, 72, and 80, respectively. The remaining data sets have comparatively smaller feature dimensions (i.e. 9 for Wbc2, 4 for Iris and 5 for Thyroid). Additionally, owing to the fact that data samples are extremely imbalanced between classes in certain data sets, e.g. E.coli, we only select those classes with relatively sufficient number of samples for clustering performance comparison. The main characteristics of the employed data sets are illustrated in **Table 3-2**.

The employed data sets impose various challenges on clustering analysis. As an example, the ALL data set used in [248, 256] is obtained from the analysis of the ALL-IDB2 microscopic blood cancer images. The essential features, such as colour, shape, and texture details, were extracted from this ALL-IDB2 data set, and a feature vector of 80 dimensions was obtained for each white blood cell image [63]. This image data set poses diverse challenges to classification/clustering models, owing to the complex irregular morphology of nucleus, variations in terms of the nucleus to cytoplasm ratio, as well as the subtle differences between the blast and normal blood cells, which bring in noise and sub-optimal distraction in the follow-on clustering process for lymphoblastic and lymphocyte identification. Other UCI data sets also contain similar challenging factors. Therefore, a comprehensive evaluation of the proposed clustering models can be established owing to diversity of the employed challenging data sets in terms of sample distribution and dimensionality.

Table 3-2 Ten selected data sets for evaluation

Data set	Number of attributes	Number of classes	Missing values	Number of instances
Sonar	60	2	No	140
Ozone	72	2	No	196
ALL	80	2	No	100
Wbc1	30	2	No	569
Wbc2	9	2	No	683
Wine	13	3	No	178
Iris	4	3	No	150
Balance	4	2	No	576
Thyroid	5	3	No	90
E.coli	7	3	No	150

3.2.3 Performance comparison metrics

Five performance indicators are employed to evaluate the clustering performance, namely the sum of intra-cluster distances (i.e. fitness scores), average accuracy, average sensitivity, average specificity, and macro-average F-score (Fscore_M) [253]. The first distance-based metric is used to indicate the convergence speed of the proposed models, while the last four metrics are used as the main criteria for clustering performance comparison. We introduce each performance metric in detail, as follows.

1. Sum of intra-cluster distances: This measurement is obtained by the summation of distances between the data samples and their corresponding centroids, as defined in **Eq. 3.6**. The smaller the sum of intra-cluster distances, the more compact the partitioned clusters. Similar to KM clustering, the proposed models employ the sum of intra-cluster distances as the objective function, which is minimized during the search process.

$$f(O,C) = \sum_{i=1}^{k} \sum_{O_l \in C_i} \sqrt{(O_l - Z_i)^2} \tag{3.6}$$

where C_i and Z_i, represent the i^{th}cluster and the centroid of the i^{th} cluster, while O_l and k denote the data belonging to the i^{th} cluster, and the total number of clusters, respectively.

2. Average accuracy: The mean clustering accuracy is obtained by averaging the accuracy rate of each class, as defined in **Eq. 3.7**. The merit of this performance metric is that it treats all classes equally, rather than being dominated by classes with a large number of samples [253].

$$\text{Ave\_accuracy} = \frac{\sum_{i=1}^{k} \frac{tp_i + tn_i}{tp_i + fn_i + fp_i + tn_i}}{k} \tag{3.7}$$

where tp_i, fn_i, fp_i, and tn_i represent true positive, false negative, false positive, and true negative of the i^{th} cluster, respectively.

3. Average sensitivity: As defined in **Eq. 3.8**, sensitivity (i.e. recall) is used to measure the proportion of correctly identified positive samples over all positive samples in the data set. Similar to the average accuracy, the macro-average of sensitivity is calculated, in order to ascertain all classes are treated equally for multi-class clustering tasks [253].

$$\text{Ave\_sensitivity} = \frac{\sum_{i=1}^{k} \frac{tp_i}{tp_i + fn_i}}{k} \tag{3.8}$$

4. Average specificity: Specificity is used to identify the proportion of correctly identified negative samples over all negative samples in the data set [253]. **Eq. 3.9** is used to obtain the macro-average specificity for multiclass tasks.

$$\text{Ave\_specificity} = \frac{\sum_{i=1}^{k} \frac{tn_i}{tn_i + fp_i}}{k} \tag{3.9}$$

5. Macro-average F-score (Fscore_M): Fscore_M is a well-accepted performance metric, which is calculated based on the macro-average of precision and recall scores [253], as defined in **Eqs. 3.10 - 3.12**.

$$Fscore_M = \frac{(\sigma^2+1)*Precision_M*Recall_M}{\sigma^2*Precision_M+Recall_M} \tag{3.10}$$

$$Precision_M = \frac{\sum_{i=1}^{k}\frac{tp_i}{tp_i+fp_i}}{k} \tag{3.11}$$

$$Recall_M = \frac{\sum_{i=1}^{k}\frac{tp_i}{tp_i+fn_i}}{k} \tag{3.12}$$

where σ =1, in order to obtain equal weightings of precision and recall.

For each data set, a total of 30 runs with each search method integrated with the KM clustering algorithm are conducted. The average performance over 30 runs for each performance metric is calculated and used as the main criterion for comparison.

3.2.4 Feature selection and clustering performance evaluation

As mentioned earlier, owing to the high dimensionality of Sonar, Ozone, and ALL data sets, and the possibility of the inclusion of redundant features, mRMR [249] is used to conduct feature dimensionality reduction and to investigate its underlying impact on the clustering performance. The clustering results before and after feature selection for each data set are shown in **Tables 3-3 - 3-12,** respectively. For the three high-dimensional data sets, namely ALL, Sonar, and Ozone, the numbers of selected features are 9, 17, and 22 from the original 80, 60, and 72 features, respectively. These feature sizes are obtained based on trial-and-error, which yield the best performance for nearly all evaluated models. The findings on feature selection are also consistent with those of existing studies [102, 248], where the ranges of selected feature numbers are 9-36 [248], 15-20 [102], and 18-25 [102] for ALL, Sonar, and Ozone, respectively, therefore ascertaining efficiency of the mRMR-based feature selection method employed in this research.

The empirical results indicate that in combination with feature selection, the clustering performance is improved for most test cases. As an example, for the ALL data set illustrated in **Table 3-3**, the number of features is reduced from the original 80 to 9, while the mean accuracy, sensitivity, specificity, and Fscore_M of the proposed CIEFA

model over 30 runs increase significantly, i.e. from 51.23% to 80.4%, 51.67% to 74.67%, 50.8% to 86.13%, and 51.27% to 78.73%, respectively. The selected features include the cytoplasm and nucleus areas, ratio between the nucleus area and the cytoplasm area, form factor, compactness, perimeter and eccentricity, which represent the most significant clinical factors for blood cancer diagnosis [248, 256, 257]. This in turn also indicates that some redundant or even contradictory features exist in the original data set [248], which may deteriorate the performance of clustering models drastically. Such findings also apply to other data sets, especially the high-dimensional ones [102]. The only exception is the low-dimensional Balance data set, as shown in **Table 3-7**, where the full feature set (i.e. a total of only four features) yields the best performance for nearly all the clustering models. In short, it is essential to eliminate redundant and irrelevant features to enhance the clustering performance.

Table 3-3 The mean clustering results over 30 independent runs on the ALL data set

Feature number	Criteria	IIEFA	CIEFA	FA	KM	CFA1	CFA2	NaFA	VSSFA	DA	SCA	MFA	GA	ACO
80 (full set)	fitness	293.53	293.71	294.33	943.13	294.32	294.35	294.33	294.33	294.13	459.26	294.34	294.32	294.33
	accuracy	0.5137	0.5123	0.5140	0.5157	0.5147	0.5127	0.5133	0.5143	0.513	0.5133	0.5133	0.5150	0.5143
	Fscore_M	0.5145	0.5127	0.5038	0.5161	0.5115	0.5118	0.5062	0.5053	0.5137	0.3647	0.5191	0.5103	0.5187
	sensitivity	0.5187	0.5167	0.4967	0.5193	0.5113	0.5147	0.5020	0.4993	0.5180	0.5153	0.5287	0.5087	0.5267
	specificity	0.5087	0.5080	0.5313	0.5120	0.5180	0.5107	0.5247	0.5293	0.5080	0.5113	0.4980	0.5213	0.5020
9	fitness	90.481	89.649	92.611	96.48	90.519	92.097	93.052	90.883	89.683	111.08	90.782	90.448	91.309
	accuracy	0.7893	0.804	0.7307	0.7693	0.7703	0.7437	0.7197	0.7740	0.7850	0.6267	0.7570	0.7943	0.7527
	Fscore_M	0.7767	0.7873	0.7063	0.7557	0.7702	0.7130	0.7017	0.7611	0.7763	0.6178	0.7336	0.7841	0.7260
	sensitivity	0.7593	0.7467	0.6953	0.7427	0.788	0.6807	0.7107	0.7527	0.7713	0.7260	0.724	0.7727	0.7067
	specificity	0.8193	0.8613	0.7660	0.7960	0.7527	0.8067	0.7287	0.7953	0.7987	0.5273	0.7900	0.8160	0.7987

Table 3-4 The mean clustering results over 30 independent runs on the Sonar data set

Feature number	Criteria	IIEFA	CIEFA	FA	KM	CFA1	CFA2	NaFA	VSSFA	DA	SCA	MFA	GA	ACO
60 (full set)	fitness	160.54	160.73	161.22	195.35	160.85	161.42	160.98	161.31	161.05	242.81	160.92	161.14	160.75
	accuracy	0.5610	0.5631	0.5669	0.5655	0.5624	0.5643	0.5657	0.5645	0.5657	0.5307	0.5629	0.5624	0.5629
	Fscore_M	0.5553	0.5698	0.5549	0.5664	0.5500	0.5526	0.5583	0.5532	0.5635	0.3944	0.5613	0.5636	0.5671
	sensitivity	0.5552	0.5862	0.5500	0.5781	0.5443	0.5486	0.5590	0.5500	0.5700	0.4324	0.5681	0.5724	0.5814

	specificity	0.5667	0.5400	0.5838	0.5529	0.5805	0.58	0.5724	0.579	0.5614	0.6290	0.5576	0.5524	0.5443
17	fitness	75.85	75.884	76.487	46.251	76.529	76.38	76.381	76.461	76.187	101.95	76.344	75.952	76.470
	accuracy	0.7100	0.7110	0.6733	0.6764	0.6717	0.6669	0.6779	0.6719	0.6760	0.6183	0.6750	0.7088	0.6769
	Fscore_M	0.7072	0.7090	0.6677	0.6814	0.6546	0.6601	0.6722	0.6461	0.6623	0.5466	0.6772	0.7019	0.6776
	sensitivity	0.7110	0.7157	0.6829	0.7224	0.6538	0.6862	0.6867	0.6243	0.6633	0.5267	0.7048	0.7024	0.7024
	specificity	0.7090	0.7062	0.6638	0.6305	0.6895	0.6476	0.669	0.7195	0.6886	0.7100	0.6452	0.7152	0.6514

Table 3-5 The mean clustering results over 30 independent runs on the Ozone data set

Feature number	Criteria	IIEFA	CIEFA	FA	KM	CFA1	CFA2	NaFA	VSSFA	DA	SCA	MFA	GA	ACO
72 (full set)	fitness	514.11	514.38	515.29	1507.7	515.23	515.23	515.23	515.29	514.77	844.99	515.3	515.07	515.44
	accuracy	0.7333	0.7330	0.7366	0.7369	0.7362	0.7361	0.7352	0.7364	0.7337	0.5631	0.7367	0.7367	0.7367
	Fscore_M	0.7167	0.7316	0.7221	0.7543	0.7374	0.7434	0.7429	0.7065	0.7353	0.4209	0.7312	0.7412	0.7127
	sensitivity	0.7000	0.7554	0.7136	0.8313	0.7701	0.7932	0.7932	0.6565	0.7677	0.4949	0.7463	0.7830	0.6793
	specificity	0.7667	0.7105	0.7595	0.6425	0.7024	0.6789	0.6772	0.8163	0.6997	0.6313	0.7272	0.6905	0.7942
22	fitness	301.26	301.34	302.19	517.76	302.22	302.29	302.22	302.25	301.9	414.87	301.89	301.42	302.24
	accuracy	0.7604	0.7577	0.7490	0.7488	0.7495	0.7497	0.7491	0.7491	0.7500	0.5648	0.7500	0.7531	0.7495
	Fscore_M	0.7524	0.7466	0.7408	0.7349	0.7438	0.7362	0.7359	0.7433	0.7318	0.3792	0.7419	0.7433	0.7407
	sensitivity	0.7435	0.7310	0.7391	0.7184	0.7503	0.7204	0.7197	0.749	0.7007	0.4173	0.7401	0.7347	0.7381
	specificity	0.7772	0.7844	0.7588	0.7793	0.7486	0.7789	0.7786	0.7493	0.7993	0.7122	0.7599	0.7714	0.7609

Table 3-6 The mean clustering results over 30 independent runs on the Thyroid data set

Feature number	Criteria	IIEFA	CIEFA	FA	KM	CFA1	CFA2	NaFA	VSSFA	DA	SCA	MFA	GA	ACO
5 (full set)	fitness	113.26	111.65	115.03	196.65	119.24	116.51	114.97	117.87	114.15	124.5	114.28	114.12	113.54
	accuracy	0.8235	0.8277	0.8133	0.8215	0.7911	0.8173	0.8205	0.8032	0.8128	0.8321	0.8165	0.8126	0.822
	Fscore_M	0.7539	0.7688	0.7508	0.7638	0.7090	0.7482	0.7582	0.7256	0.7398	0.7981	0.7575	0.7392	0.7667
	sensitivity	0.7352	0.7415	0.7200	0.7322	0.6867	0.7259	0.7307	0.7048	0.7193	0.7481	0.7248	0.7189	0.7330
	specificity	0.8676	0.8707	0.86	0.8661	0.8433	0.863	0.8654	0.8524	0.8596	0.8741	0.8624	0.8594	0.8665
4	fitness	96.297	96.599	99.808	142.21	99.661	99.979	100.49	99.364	97.743	107.3	99.36	96.794	100.56
	accuracy	0.8748	0.8637	0.8101	0.8057	0.8084	0.8069	0.802	0.8116	0.8346	0.841	0.8121	0.8514	0.8044
	Fscore_M	0.8377	0.8204	0.7628	0.753	0.7611	0.7543	0.7505	0.7719	0.7813	0.8013	0.7649	0.8010	0.7467
	sensitivity	0.8122	0.7956	0.7152	0.7085	0.7126	0.7104	0.703	0.7174	0.7519	0.7615	0.7181	0.7770	0.7067

	specificity	0.9061	0.8978	0.8576	0.8543	0.8563	0.8552	0.8515	0.8587	0.8759	0.8807	0.8591	0.8885	0.8533

Table 3-7 The mean clustering results over 30 independent runs on the Balance data set

Feature number	Criteria	IIEFA	CIEFA	FA	KM	CFA1	CFA2	NaFA	VSSFA	DA	SCA	MFA	GA	ACO
4 (full set)	fitness	1002.9	1003.1	1003.9	1866.6	1004.2	1003.9	1003.7	1003.6	1003.1	1011.2	1003.3	1003	1003.4
	accuracy	0.8047	0.7923	0.7733	0.7546	0.7494	0.7581	0.7538	0.7725	0.7858	0.7549	0.7993	0.7991	0.7956
	Fscore_M	0.8045	0.7923	0.7735	0.7518	0.7475	0.758	0.7537	0.7726	0.7857	0.7522	0.7991	0.7991	0.7955
	sensitivity	0.8038	0.7925	0.7749	0.7478	0.7459	0.7574	0.7536	0.7727	0.7855	0.7491	0.7985	0.799	0.7953
	specificity	0.8056	0.7921	0.7718	0.7613	0.7529	0.7588	0.7539	0.7723	0.7860	0.7606	0.8001	0.7993	0.7959
3	fitness	821.70	821.77	824.55	1300.6	824.56	824.67	824.6	826.58	821.75	826.52	821.86	821.52	823.14
	accuracy	0.7344	0.7344	0.7004	0.7042	0.6939	0.7164	0.7135	0.6747	0.7331	0.7269	0.7372	0.7355	0.7217
	Fscore_M	0.7342	0.7349	0.7002	0.7073	0.6923	0.7202	0.7134	0.6719	0.7321	0.7303	0.7377	0.7356	0.7200
	sensitivity	0.7338	0.7362	0.7012	0.7126	0.6896	0.7281	0.7162	0.6718	0.7303	0.7394	0.7392	0.7359	0.7167
	specificity	0.735	0.7325	0.6997	0.6957	0.6983	0.7047	0.7109	0.6777	0.7359	0.7144	0.7352	0.7352	0.7267

Table 3-8 The mean clustering results over 30 independent runs on the E.coli data set

Feature number	Criteria	IIEFA	CIEFA	FA	KM	CFA1	CFA2	NaFA	VSSFA	DA	SCA	MFA	GA	ACO
7 (full set)	fitness	257.63	251.13	260.17	473.53	259.33	257.07	257.73	260.05	253.28	252.78	261.52	244.09	257.85
	accuracy	0.7739	0.7945	0.7769	0.8883	0.7756	0.771	0.7704	0.7641	0.7893	0.929	0.7633	0.8154	0.7769
	Fscore_M	0.6992	0.7248	0.6692	0.8341	0.6687	0.6605	0.6574	0.6503	0.6935	0.8971	0.6498	0.7491	0.6701
	sensitivity	0.6609	0.6918	0.6653	0.8324	0.6633	0.6564	0.6556	0.6462	0.684	0.8936	0.6449	0.7231	0.6653
	specificity	0.8304	0.8459	0.8327	0.9162	0.8317	0.8282	0.8278	0.8231	0.842	0.9468	0.8224	0.8616	0.8327
5	fitness	196.23	196.23	198.08	321.05	198.03	196.53	198.2	197.91	197.72	238.63	198.00	197.64	197.96
	accuracy	0.9644	0.9644	0.9564	0.9406	0.9563	0.961	0.9557	0.9575	0.9556	0.931	0.9566	0.9566	0.9536
	Fscore_M	0.9474	0.9474	0.9352	0.9109	0.9349	0.9421	0.934	0.9368	0.9337	0.9005	0.9355	0.9354	0.9308
	sensitivity	0.9467	0.9467	0.9347	0.9109	0.9344	0.9416	0.9336	0.9362	0.9333	0.8964	0.9349	0.9349	0.9304
	specificity	0.9733	0.9733	0.9673	0.9554	0.9672	0.9708	0.9668	0.9681	0.9667	0.9482	0.9674	0.9674	0.9652

Table 3-9 The mean clustering results over 30 independent runs on the Wbc1 data set

Feature number	Criteria	IIEFA	CIEFA	FA	KM	CFA1	CFA2	NaFA	VSSFA	DA	SCA	MFA	GA	ACO

30 (full set)	fitness	2280.8	2281.5	2293.9	11575	2293.8	2293.9	2293.7	2293.8	2286	2800.3	2293.5	2285.9	2293.8
	accuracy	0.9147	0.9145	0.9114	0.9097	0.9108	0.9113	0.9105	0.9111	0.9129	0.7230	0.9100	0.9142	0.9110
	$Fscore_M$	0.9092	0.9039	0.9047	0.9051	0.9081	0.899	0.8963	0.8949	0.909	0.6082	0.8978	0.9064	0.8986
	sensitivity	0.9056	0.8953	0.8986	0.9016	0.9067	0.8846	0.8813	0.8759	0.9066	0.6242	0.8845	0.8986	0.8844
	specificity	0.8990	0.9088	0.8894	0.8845	0.8804	0.9032	0.9058	0.9119	0.8925	0.6558	0.9022	0.8984	0.9031
20	fitness	1761.4	1761.6	1768.7	6887.6	1768.7	1768.7	1768.7	1768.7	1764.7	2220.2	1768.7	1762.1	1768.7
	accuracy	0.9461	0.9448	0.9332	0.9332	0.9332	0.9332	0.9332	0.9332	0.9385	0.813	0.9332	0.9393	0.9332
	$Fscore_M$	0.9361	0.9394	0.9230	0.9276	0.9215	0.9261	0.9291	0.9184	0.9295	0.7795	0.9276	0.9372	0.9322
	sensitivity	0.9141	0.9290	0.9028	0.9185	0.8976	0.9133	0.9237	0.8871	0.9110	0.7803	0.9185	0.9358	0.9341
	specificity	0.9470	0.9285	0.9237	0.908	0.9289	0.9133	0.9028	0.9394	0.9298	0.7290	0.9080	0.9069	0.8924

Table 3-10 The mean clustering results over 30 independent runs on the Wbc2 data set

Feature number	Criteria	IIEFA	CIEFA	FA	KM	CFA1	CFA2	NaFA	VSSFA	DA	SCA	MFA	GA	ACO
9 (full set)	fitness	1092.1	1092	1098.3	2724.4	1102.7	1102.9	1100	1102.8	1093.7	1327.3	1093.5	1092.1	1093.1
	accuracy	0.9693	0.9692	0.9629	0.9560	0.9563	0.9559	0.9604	0.9562	0.9683	0.9542	0.9684	0.9679	0.9680
	$Fscore_M$	0.9662	0.9661	0.9588	0.9562	0.9538	0.9489	0.9555	0.9525	0.9623	0.9462	0.9646	0.9640	0.9637
	sensitivity	0.9667	0.9666	0.9561	0.9568	0.9522	0.9421	0.9525	0.9495	0.962	0.9365	0.9646	0.9639	0.9646
	specificity	0.9667	0.9666	0.9580	0.9367	0.9423	0.9511	0.9539	0.9446	0.9682	0.9500	0.9660	0.9655	0.9649
7	fitness	931.67	931.67	933.94	1819.8	934.25	935.78	933.42	933.96	932.27	1104.8	932.17	931.68	931.67
	accuracy	0.9649	0.9649	0.9647	0.9649	0.9644	0.9649	0.9648	0.9647	0.9649	0.949	0.9649	0.9649	0.9649
	$Fscore_M$	0.9629	0.9629	0.9619	0.9629	0.9607	0.9613	0.9597	0.9572	0.9621	0.9451	0.9652	0.9629	0.9660
	sensitivity	0.9624	0.9624	0.9613	0.9624	0.9603	0.9604	0.9583	0.9554	0.9614	0.9408	0.9654	0.9624	0.9663
	specificity	0.9584	0.9584	0.9593	0.9584	0.9599	0.9604	0.9624	0.9652	0.9594	0.9305	0.9555	0.9584	0.9545

Table 3-11 The mean clustering results over 30 independent runs on the Wine data set

Feature number	Criteria	IIEFA	CIEFA	FA	KM	CFA1	CFA2	NaFA	VSSFA	DA	SCA	MFA	GA	ACO
13 (full set)	fitness	456.78	452.06	461.45	1282.9	451.84	453.8	453.8	461.93	451.34	580.06	449.81	451.65	451.75
	accuracy	0.9485	0.9705	0.9372	0.9654	0.9669	0.9607	0.9598	0.9301	0.9689	0.7876	0.9747	0.9692	0.9683
	$Fscore_M$	0.9295	0.9577	0.921	0.9544	0.9561	0.9492	0.9447	0.9099	0.9566	0.7048	0.9649	0.9586	0.9579
	sensitivity	0.9318	0.9617	0.9198	0.9567	0.9585	0.9507	0.9492	0.9110	0.9610	0.7041	0.9682	0.9613	0.9603
	specificity	0.9618	0.9784	0.9546	0.9749	0.9762	0.9718	0.9711	0.9493	0.9777	0.8383	0.9816	0.9781	0.9772

	fitness	348.48	339.84	342.74	744.96	344.93	345.01	342.78	342.76	346.88	431.28	342.6	342.05	340.52
	accuracy	0.9484	0.98	0.9663	0.9665	0.9578	0.9587	0.9649	0.9664	0.9518	0.9109	0.9665	0.9676	0.9735
9	Fscore_M	0.9242	0.9713	0.9519	0.9538	0.9393	0.942	0.9499	0.9536	0.9313	0.8818	0.9517	0.953	0.9622
	sensitivity	0.9286	0.9749	0.9563	0.9573	0.9442	0.9466	0.9546	0.9572	0.9368	0.8790	0.9561	0.9562	0.9662
	specificity	0.9619	0.9858	0.9751	0.9759	0.9687	0.9697	0.9741	0.9758	0.9638	0.9325	0.9755	0.9764	0.9809

Table 3-12 The mean clustering results over 30 independent runs on the Iris data set

Feature number	Criteria	IIEFA	CIEFA	FA	KM	CFA1	CFA2	NaFA	VSSFA	DA	SCA	MFA	GA	ACO
	fitness	130.24	131.37	133.09	150.75	132.49	131.94	133.62	133.09	131.57	161.79	132.18	129.71	130.04
4	accuracy	0.8818	0.8744	0.8677	0.8653	0.8735	0.8714	0.8659	0.8704	0.8742	0.8739	0.8738	0.8876	0.8855
(full	Fscore_M	0.8228	0.8117	0.8018	0.7987	0.8106	0.8080	0.7993	0.8061	0.8116	0.8253	0.8110	0.8315	0.8284
set)	sensitivity	0.8227	0.8116	0.8016	0.7980	0.8102	0.8071	0.7989	0.8056	0.8113	0.8109	0.8107	0.8313	0.8282
	specificity	0.9113	0.9058	0.9008	0.899	0.9051	0.9036	0.8994	0.9028	0.9057	0.9054	0.9053	0.9157	0.9141
	fitness	42.932	42.932	43.226	17.927	43.243	43.296	43.199	43.225	42.942	57.223	42.956	42.932	42.992
	accuracy	0.9733	0.9733	0.9733	0.9733	0.9733	0.9733	0.9733	0.9733	0.9733	0.9587	0.9733	0.9733	0.9733
2	Fscore_M	0.9602	0.9602	0.9602	0.9602	0.9602	0.9602	0.9602	0.9602	0.9602	0.9432	0.9602	0.9602	0.9602
	sensitivity	0.9600	0.9600	0.9600	0.9600	0.9600	0.9600	0.9600	0.9600	0.9600	0.9573	0.9600	0.9600	0.9600
	specificity	0.9800	0.9800	0.9800	0.9800	0.9800	0.9800	0.9800	0.9800	0.9800	0.9787	0.9800	0.9800	0.9800

3.2.5 Performance comparison and analysis

As mentioned earlier, five metrics are used for clustering performance comparison, namely the fitness scores on the sum of intra-cluster distances, average accuracy, average sensitivity, average specificity, and macro-average F-score (Fscore_M). Since the best performances are achieved using the identified significant feature subsets in most test cases for nearly all the methods, we employ the enhanced results obtained in combination with feature selection for further analysis and comparison. The detailed evaluation results over 30 runs for each performance measure after feature selection are shown in **Tables 3-13 – 3-17**.

Table 3-13 The mean results of the minimum intra-cluster distance measure over 30 runs

Dataset	Feature size	IIEFA	CIEFA	FA	KM	CFA1	CFA2	NaFA	VSSFA	DA	SCA	MFA	GA	ACO

Thyroid	4	**96.297**	96.599	99.808	142.21	99.661	99.979	100.49	99.364	97.743	107.3	99.36	96.794	100.56
Sonar	17	75.85	75.884	76.487	**46.251**	76.529	76.38	76.381	76.461	76.187	101.95	76.344	75.952	76.47
Balance	4	**1002.9**	1003.1	1003.9	1866.6	1004.2	1003.9	1003.7	1003.6	1003.1	1011.2	1003.3	1003	1003.4
E.coli	5	**196.23**	**196.23**	198.08	321.05	198.03	196.53	198.2	197.91	197.72	238.63	198	197.64	197.96
Ozone	22	**301.26**	301.34	302.19	517.76	302.22	302.29	302.22	302.25	301.9	414.87	301.89	301.42	302.24
ALL	9	90.481	**89.649**	92.611	96.48	90.519	92.097	93.052	90.883	89.683	111.08	90.782	90.448	91.309
Wbc1	20	**1761.4**	1761.6	1768.7	6887.6	1768.7	1768.7	1768.7	1768.7	1764.7	2220.2	1768.7	2285.9	1768.7
Wbc2	9	1092.1	**1092.0**	1098.3	2724.4	1102.7	1102.9	1100.0	1102.8	1093.7	1327.3	1093.5	1092.1	1093.1
Wine	9	348.48	**339.84**	342.74	744.96	344.93	345.01	342.78	342.76	346.88	431.28	342.6	342.05	340.52
Iris	2	42.932	42.932	43.226	**17.927**	43.243	43.296	43.199	43.225	42.942	57.223	42.956	42.932	42.992

With respect to the fitness scores, i.e. the intra-cluster distance measure, as shown in **Table 3-13**, IIEFA and CIEFA achieve the minimum distance measures in eight out of ten data sets in total. Specifically, IIEFA yields the minimum intra-cluster measures with five data sets based on the average performance over 30 runs, i.e. Thyroid, Balance, E.coli, Ozone, and Wbc1, while CIEFA achieves the minimum fitness scores with four data sets, i.e. E.coli, ALL, Wbc2, and Wine. Moreover, KM clustering produces the minimum intra-cluster measures with the Sonar and Iris data sets in combination with mRMR-based feature selection, although IIEFA and CIEFA achieve the minimum objective function evaluation scores when the full feature sets for both Sonar and Iris data sets are used. Overall, in comparison with the six classical methods i.e. GA, ACO, DA, SCA, FA, KM, and other five FA variants i.e. CFA1, CFA2, NaFA, VSSFA, and MFA, both IIEFA and CIEFA models demonstrate faster convergence rates and great superiority over other methods in identifying enhanced centroids that lead to more compact clusters. The proposed search mechanisms account for the enhanced global exploration capability of IIEFA and CIEFA in comparison with those of other classical methods and FA variants in attaining the global best solutions.

Table 3-14 The mean results of average accuracy after feature selection over 30 runs

Dataset	Feature size	IIEFA	CIEFA	FA	KM	CFA1	CFA2	NaFA	VSSFA	DA	SCA	MFA	GA	ACO
Thyroid	4	**0.8748**	0.8637	0.8101	0.8057	0.8084	0.8069	0.802	0.8116	0.8346	0.841	0.8121	0.8514	0.8044
Sonar	17	0.71	**0.711**	0.6733	0.6764	0.6717	0.6669	0.6779	0.6719	0.676	0.6183	0.675	0.7088	0.6769
Balance	4	**0.8047**	0.7923	0.7733	0.7546	0.7494	0.7581	0.7538	0.7725	0.7858	0.7549	0.7993	0.7991	0.7956

E.coli	5	**0.9644**	**0.9644**	0.9564	0.9406	0.9563	0.961	0.9557	0.9575	0.9556	0.931	0.9566	0.9566	0.9536
Ozone	22	**0.7604**	0.7577	0.749	0.7488	0.7495	0.7497	0.7491	0.7491	0.75	0.5648	0.75	0.7531	0.7495
ALL	9	0.7893	**0.804**	0.7307	0.7693	0.7703	0.7437	0.7197	0.774	0.785	0.6267	0.757	0.7943	0.7527
Wbc1	20	**0.9461**	0.9448	0.9332	0.9332	0.9332	0.9332	0.9332	0.9332	0.9385	0.813	0.9332	0.9142	0.9332
Wbc2	9	**0.9693**	0.9692	0.9629	0.956	0.9563	0.9559	0.9604	0.9562	0.9683	0.9542	0.9684	0.9679	0.968
Wine	9	0.9484	**0.98**	0.9663	0.9665	0.9578	0.9587	0.9649	0.9664	0.9518	0.9109	0.9665	0.9676	0.9735
Iris	2	**0.9733**	**0.9733**	0.9733	0.9733	0.9733	0.9733	0.9733	0.9733	0.9733	0.9587	0.9733	0.9733	0.9733

Table 3-15 The mean results of Fscore_M after feature selection over 30 runs

Dataset	Feature size	IIEFA	CIEFA	FA	KM	CFA1	CFA2	NaFA	VSSFA	DA	SCA	MFA	GA	ACO
Thyroid	4	**0.8377**	0.8204	0.7628	0.753	0.7611	0.7543	0.7505	0.7719	0.7813	0.8013	0.7649	0.801	0.7467
Sonar	17	0.7072	**0.709**	0.6677	0.6814	0.6546	0.6601	0.6722	0.6461	0.6623	0.5466	0.6772	0.7019	0.6776
Balance	4	**0.8045**	0.7923	0.7735	0.7518	0.7475	0.758	0.7537	0.7726	0.7857	0.7522	0.7991	0.7991	0.7955
E.coli	5	**0.9474**	**0.9474**	0.9352	0.9109	0.9349	0.9421	0.934	0.9368	0.9337	0.9005	0.9355	0.9354	0.9308
Ozone	22	**0.7524**	0.7466	0.7408	0.7349	0.7438	0.7362	0.7359	0.7433	0.7318	0.3792	0.7419	0.7433	0.7407
ALL	9	0.7767	**0.7873**	0.7063	0.7557	0.7702	0.713	0.7017	0.7611	0.7763	0.6178	0.7336	0.7841	0.726
Wbc1	20	0.9361	**0.9394**	0.923	0.9276	0.9215	0.9261	0.9291	0.9184	0.9295	0.7795	0.9276	0.9064	0.9322
Wbc2	9	**0.9662**	0.9661	0.9588	0.9562	0.9538	0.9489	0.9555	0.9525	0.9623	0.9462	0.9646	0.964	0.9637
Wine	9	0.9242	**0.9713**	0.9519	0.9538	0.9393	0.942	0.9499	0.9536	0.9313	0.8818	0.9517	0.953	0.9622
Iris	2	**0.9602**	**0.9602**	0.9602	0.9602	0.9602	0.9602	0.9602	0.9602	0.9602	0.9432	0.9602	0.9602	0.9602

In terms of mean accuracy and Fscore_M, as shown in **Tables 3-14 – 3-15**, the proposed models achieve the best scores for all the data sets over 30 runs. With respect to the mean accuracy rates shown in **Table 3-14**, IIEFA achieves the highest average accuracy rates over 30 runs with seven data sets (i.e. Thyroid, Balance, E.coli, Ozone, Wbc1, Wbc2 and Iris), while CIEFA achieves the best results with five data sets (i.e. Sonar, E.coli, ALL, Wine, and Iris). Both IIEFA and CIEFA demonstrate a clear advantage over other methods with four data sets, i.e. Thyroid, Sonar, Balance, and ALL. Pertaining to the Fscore_M measure shown in **Table 3-15**, IIEFA and CIEFA achieve the best average scores over 30 runs with six data sets, i.e. Thyroid, Balance, E.coli, Ozone, Wbc2, and Iris for IIEFA and Sonar, E.coli, ALL, Wbc1, Wine, and Iris for CIEFA, respectively. Similar to the accuracy indicator, a clear performance distinction can be

observed between the proposed models and other methods with respect to the Fscore_M results.

Table 3-16 The mean results of average sensitivity after feature selection over 30 runs

Dataset	Feature size	IIEFA	CIEFA	FA	KM	CFA1	CFA2	NaFA	VSSFA	DA	SCA	MFA	GA	ACO
Thyroid	4	**0.8122**	0.7956	0.7152	0.7085	0.7126	0.7104	0.703	0.7174	0.7519	0.7615	0.7181	0.777	0.7067
Sonar	17	0.711	0.7157	0.6829	0.7224	0.6538	0.6862	0.6867	0.6243	0.6633	0.5267	0.7048	0.7024	0.7024
Balance	4	**0.8038**	0.7925	0.7749	0.7478	0.7459	0.7574	0.7536	0.7727	0.7855	0.7491	0.7985	0.799	0.7953
E.coli	5	**0.9467**	**0.9467**	0.9347	0.9109	0.9344	0.9416	0.9336	0.9362	0.9333	0.8964	0.9349	0.9349	0.9304
Ozone	22	0.7435	0.731	0.7391	0.7184	0.7503	0.7204	0.7197	0.749	0.7007	0.4173	0.7401	0.7347	0.7381
ALL	9	0.7593	0.7467	0.6953	0.7427	0.788	0.6807	0.7107	0.7527	0.7713	0.726	0.724	0.7727	0.7067
Wbc1	20	0.9141	0.929	0.9028	0.9185	0.8976	0.9133	0.9237	0.8871	0.911	0.7803	0.9185	0.9358	0.9341
Wbc2	9	**0.9667**	0.9666	0.9561	0.9568	0.9522	0.9421	0.9525	0.9495	0.962	0.9365	0.9646	0.9639	0.9646
Wine	9	0.9286	**0.9749**	0.9563	0.9573	0.9442	0.9466	0.9546	0.9572	0.9368	0.879	0.9561	0.9562	0.9662
Iris	2	**0.9600**	**0.9600**	0.9600	0.9600	0.9600	0.9600	0.9600	0.9600	0.9600	0.9573	0.9600	0.9600	0.9600

Table 3-17 The mean results of average specificity after feature selection over 30 runs

Dataset	Feature size	IIEFA	CIEFA	FA	KM	CFA1	CFA2	NaFA	VSSFA	DA	SCA	MFA	GA	ACO
Thyroid	4	**0.9061**	0.8978	0.8576	0.8543	0.8563	0.8552	0.8515	0.8587	0.8759	0.8807	0.8591	0.8885	0.8533
Sonar	17	0.709	0.7062	0.6638	0.6305	0.6895	0.6476	0.669	0.7195	0.6886	0.71	0.6452	0.7152	0.6514
Balance	4	**0.8056**	0.7921	0.7718	0.7613	0.7529	0.7588	0.7539	0.7723	0.786	0.7606	0.8001	0.7993	0.7959
E.coli	5	**0.9733**	**0.9733**	0.9673	0.9554	0.9672	0.9708	0.9668	0.9681	0.9667	0.9482	0.9674	0.9674	0.9652
Ozone	22	0.7772	0.7844	0.7588	0.7793	0.7486	0.7789	0.7786	0.7493	0.7993	0.7122	0.7599	0.7714	0.7609
ALL	9	0.8193	0.8613	0.766	0.796	0.7527	0.8067	0.7287	0.7953	0.7987	0.5273	0.79	0.816	0.7987
Wbc1	20	0.947	0.9285	0.9237	0.908	0.9289	0.9133	0.9028	0.9394	0.9298	0.729	0.908	0.9069	0.8924
Wbc2	9	**0.9667**	0.9666	0.958	0.9367	0.9423	0.9511	0.9539	0.9446	0.9682	0.95	0.966	0.9655	0.9649
Wine	9	0.9619	**0.9858**	0.9751	0.9759	0.9687	0.9697	0.9741	0.9758	0.9638	0.9325	0.9755	0.9764	0.9809
Iris	2	**0.9800**	**0.9800**	0.9800	0.9800	0.9800	0.9800	0.9800	0.9800	0.9800	0.9787	0.9800	0.9800	0.9800

Moreover, the observed advantages of IIEFA and CIEFA are further reinforced by the results of sensitivity and specificity, as shown in **Tables 3-16 – 3-17**. With respect to sensitivity and specificity, IIEFA achieves the highest scores for both metrics with five

data sets (i.e. Thyroid, Balance, E.coli, Wbc2, and Iris), while CIEFA achieves the best results for both metrics with three data sets (i.e. E.coli, Wine, and Iris). This indicates that both CIEFA and IIEFA outperform other baseline models with most of the employed data sets. They are capable of clustering and recognising data samples from different classes effectively.

Overall, the average accuracy, sensitivity, specificity and Fscore_M results evidently indicate the superiority of IIEFA and CIEFA over other search methods, in terms of robustness and flexibility, for both high- and low-dimensional clustering problems in combination with feature selection. In particular, the proposed models outperform five other FA variants significantly in nearly all the test cases. Moreover, CIEFA demonstrates an evident advantage on the Wine data set than IIEFA on all five performance metrics, while attaining results similar to those of IIEFA with the rest of the data sets. Besides that, nearly all methods achieve similar scores on all five performance measures on the Iris data set (except for SCA). Since only two significant features are identified and remained after feature selection for the Iris data set, the complexity of this clustering task is significantly reduced.

The underlying reasons for the advantage demonstrated by IIEFA and CIEFA can be ascribed to the enhanced capability of exploration and exploitation contributed by the proposed search strategies. The first proposed mechanism is to intensify inward exploration by replacing the attractiveness coefficient with a random search matrix. The diversity of search directions, scales, and spaces is enhanced significantly, therefore improving the exploration ability and mitigating the constraints of biological laws. The second strategy is to intensify outward exploration by relocating the 'ineffective fireflies' to a greater and extended space out of the neighbourhoods of fireflies in comparison in the early stage of the search process. The search territory of firefly swarms is further expanded, therefore facilitating the ability of global exploration. With intensified neighbouring and global exploration from the above two strategies plus the advantages of automatic subdivision inherited from the original FA model [60], the probability of being trapped in local optima is reduced effectively, while the diversity of movement is enhanced significantly for the proposed FA models. Evidenced by the experimental and statistical results, these advantages enable the proposed FA models to undertake challenging clustering tasks with high dimensionality, noise, and less separable clusters, e.g. the ALL data set.

In contrast, some limitations related to search diversity and search efficiency can be identified in classical search methods according to empirical studies. As an example, Radcliffe and Surry [258] indicated that the GA-based clustering algorithms in some cases suffered from degeneracy resulted from the phenomenon of multiple chromosomes representing the same or very similar solutions [258]. Such degeneracy could lead to inefficient coverage of the search space, since the centroid solutions with the same or very similar configurations are repeatedly explored [259]. Moreover, multiple occurrences of the strongly favourable individuals in the GA can lead to the reproduction of many highly correlated offspring solutions, therefore reducing diversity of the population and resulting in premature convergence. Similarly, in ACO, the effect of emphasizing short paths diminishes, and search stagnation emerges when the quality of solutions becomes closer as the differences between individuals decrease [260]. Premature convergence can also occur in ACO as the sub-optimal solutions dominate the search process at an early stage, and the parameter of trail persistence is not tuned properly [251, 261, 262]. Consequently, owing to the potential local optima traps (GA) and search stagnation (ACO) without proper counteracting strategies, classical evolutionary algorithms such as GA and ACO are less competitive in comparison with the proposed CIEFA and IIEFA models based on results from the abovementioned five metrics including intra-cluster distances, accuracy, Fscore_M, sensitivity and specificity, as illustrated in **Tables 3-13 – 3-17**. Similar limitations are also applied to other FA variants. As an example, in the MFA model [178], each firefly not only moves towards all brighter fireflies in its neighbourhood, but also moves towards the swarm leader at the same time. The search diversity and exploration capability of the firefly swarm are obstructed owing to the continuous exposure to attraction of the global best solution during the search process. Consequently, the firefly swarm is more likely to converge prematurely, and be trapped in local optima.

Overall, owing to the assistance of the two proposed strategies, CIEFA and IIEFA are able to overcome local optima traps and outperform classical search methods, i.e. GA, ACO, FA, DA and SCA. They also outperform advanced FA variants employed in this study, i.e. CFA1, CFA2, NaFA, VSSFA, and MFA. Additionally, the merits of the proposed strategies also indicate that a strict adherence to biological laws imposes certain constraints on the exploration ability of heuristic search algorithms. As a result, the original biological laws from nature need to be further extracted and refined to best

facilitate the effectiveness and discard potential restrictions in the development of metaheuristic algorithms. Furthermore, there is other insightful research on metaheuristic algorithms, which provides promising directions for future investigation [250].

3.2.6 Statistical tests

To examine the significance of the performance difference between the proposed models and baseline methods, both Friedman and Wilcoxon rank sum tests are conducted.

3.2.6.1 The Friedman test

In the Friedman test, a test statistic Q is constructed based on the mean rankings of test treatments, which can be approximated by a chi-squared distribution. Then, the null hypothesis that K treatments come from the same population is tested according to the p-values given by $P(\chi^2_{k-1} > Q)$ [263, 264]. The Friedman test is conducted with respect to three main comprehensive performance metrics (intra-cluster distance measures, average clustering accuracy, and Fscore_M) for IIEFA and CIEFA. **Tables 3-18 – 3-19** show the mean ranking results of the three performance metrics for the CIEFA and IIEFA models, respectively. For each metric, the mean ranking of each method is obtained by averaging its rankings over ten data sets based on the results shown in **Tables 3-13 – 3-15**. The significance level is set to 0.05 (i.e. $\alpha = 0.05$) as the confidence level in all test cases. **Tables 3-20 – 3-21** show the details of statistical test results for the CIEFA and IIEFA models, respectively.

Table 3-18 The mean ranking results based on the Friedman test for the CIEFA model

Algorithms	Mean ranking based on distance	Algorithms	Mean ranking based on 1/Accuracy	Algorithms	Mean ranking based on 1/Fscore_M
CIEFA	1.40	CIEFA	1.80	CIEFA	1.80
GA	3.25	GA	3.95	GA	4.20
DA	4.05	MFA	4.70	MFA	4.90
MFA	4.70	DA	5.25	ACO	5.80
ACO	6.20	ACO	6.10	VSSFA	6.45
VSSFA	6.80	VSSFA	6.50	DA	6.50
FA	7.45	FA	7.25	FA	6.80
NaFA	7.65	CFA2	7.65	KM	7.25
CFA1	7.75	CFA1	7.90	CFA1	7.70

CFA2	7.85	KM	8.00	CFA2	7.80
KM	9.70	NaFA	8.10	NaFA	8.00
SCA	11.20	SCA	10.80	SCA	10.80

Table 3-19 The mean ranking results based on the Friedman test for the IIEFA model

Algorithms	Mean ranking based on distance	Algorithms	Mean ranking based on 1/Accuracy	Algorithms	Mean ranking based on 1/Fscore_M
IIEFA	2.40	IIEFA	2.60	IIEFA	2.60
GA	3.10	GA	3.85	GA	4.10
DA	3.90	MFA	4.70	MFA	4.90
MFA	4.60	DA	5.15	ACO	5.80
ACO	6.10	ACO	6.10	VSSFA	6.35
VSSFA	6.70	VSSFA	6.40	DA	6.40
FA	7.35	FA	7.15	FA	6.70
NaFA	7.55	CFA2	7.55	KM	7.15
CFA1	7.65	CFA1	7.80	CFA1	7.60
CFA2	7.75	KM	7.90	CFA2	7.70
KM	9.70	NaFA	8.00	NaFA	7.90
SCA	11.20	SCA	10.80	SCA	10.80

Table 3-20 Statistical results of the Friedman test for the CIEFA model

Algorithms	Chi-Square	p-Value	Hypothesis
fitness	65.948929	7.1418E-10	Rejected
1/Accuracy	52.348099	2.3578E-07	Rejected
1/Fscore_M	45.847933	3.0000E-06	Rejected

Table 3-21 Statistical results of the Friedman test for the IIEFA model

Algorithms	Chi-Square	p-Value	Hypothesis
fitness	59.698571	1.0547E-08	Rejected
1/Accuracy	46.035280	3.0000E-06	Rejected
1/Fscore_M	39.724308	4.0000E-05	Rejected

As indicated in **Tables 3-18 – 3-19**, the proposed CIEFA and IIEFA models dominate the highest rankings, and demonstrate clear advantages in the performance metrics of intra-cluster distance measure, accuracy, and Fscore_M with the Friedman test. In

comparison with the five FA variants, i.e. VSSFA, NaFA, CFA1, CFA2, and MFA, the proposed models achieve significant improvements in all three performance metrics, indicating the advantages of the proposed search mechanisms. The proposed FA models also outperform KM and five classical search methods, i.e. GA, ACO, FA DA, and SCA. Comparatively, the CIEFA model achieves a better ranking than that of the IIEFA model in overall evaluation based on the experimental results. Furthermore, as indicted in **Tables 3-20 – 3-21**, the p-values of the Friedman test are all lower than 0.05 with respect to each metric for both the IIEFA and CIEFA models, which suggest an overall statistically significant difference between the mean ranks of IIEFA and CIEFA as compared with those of other test algorithms.

3.2.6.2 The Wilcoxon rank sum test

The Wilcoxon rank sum test is conducted based on the mean accuracy rates of all the methods to further indicate the statistical distinctiveness of the proposed FA models against each baseline method. As indicated in **Tables 3-22 – 3-23**, the majority of the test results are lower than 0.05 for both CIEFA and IIEFA models, which indicate the proposed FA models significantly outperform 11 baseline algorithms with respect to most of data sets from the statistical perspective. The Iris data set is an exception since all the algorithms except for SCA achieve the same highest accuracy of 97.33% with feature selection. Moreover, as shown in **Tables 3-22 – 3-23**, in comparison with CIEFA, IIEFA demonstrates higher frequencies of insignificant difference in clustering accuracy as compared with those of the baseline models. This tendency becomes more evident on the ALL data set, since IIEFA does not show statistically significant differences as compared with seven baseline methods, i.e. KM, CFA1, VSSFA, DA, MFA, GA, and ACO, while for CIEFA, a similar case only occurs to two baseline methods, i.e. GA and VSSFA. This phenomenon may be attributed to the challenging factors of the ALL data set, owing to its high dimensionality and highly inseparable data distributions caused by the subtle differences between the normal and blast cases. On the other hand, the advantage demonstrated by CIEFA over IIEFA on the ALL data set can be ascribed to the proposed dispersing mechanism, which further enhances the exploration capability on the basis of IIEFA and reduces the probability of being trapped in local optima. Therefore, CIEFA is capable of delivering better clustering performances than those of IIEFA in tackling data samples with complex distributions and narrow class margins.

Table 3-22 The Wilcoxon rank sum test results of the proposed CIEFA model

Dataset	FA	KM	CFA1	CFA2	NaFA	VSSFA	DA	SCA	MFA	GA	ACO
Thyroid	2.02E-06	7.99E-07	1.45E-06	9.95E-07	5.99E-07	1.90E-06	1.56E-02	4.01E-02	3.70E-06	**9.33E-01**	1.32E-06
Sonar	1.07E-08	6.06E-06	1.51E-06	3.81E-08	6.03E-06	7.21E-09	9.00E-08	8.32E-11	2.55E-07	**6.49E-01**	4.03E-06
Balance	2.89E-05	5.54E-07	4.18E-08	7.98E-08	1.72E-09	3.84E-06	**1.41E-01**	6.85E-03	3.91E-03	1.80E-02	1.14E-03
E.coli	2.15E-02	2.84E-05	1.10E-02	6.45E-04	2.77E-03	**1.61E-01**	1.37E-03	4.43E-12	1.10E-02	1.32E-03	1.42E-04
Ozone	2.88E-11	3.21E-11	1.80E-11	1.45E-11	2.53E-11	2.53E-11	8.57E-12	1.73E-11	8.57E-12	2.42E-06	1.80E-11
ALL	1.18E-04	1.30E-02	4.02E-02	5.81E-04	3.58E-04	9.88E-01	1.52E-02	1.48E-09	2.67E-02	6.04E-01	3.64E-03
Wbc1	5.01E-13	5.01E-13	5.01E-13	5.01E-13	5.01E-13	5.01E-13	6.11E-11	1.54E-11	5.01E-13	6.83E-12	5.01E-13
Wbc2	3.10E-10	3.44E-13	5.47E-13	4.39E-13	5.65E-11	3.41E-13	3.33E-04	5.94E-11	1.45E-05	2.96E-11	6.48E-02
Wine	3.49E-08	5.59E-09	1.13E-09	7.21E-09	4.17E-10	2.62E-09	3.93E-05	6.38E-12	2.68E-07	6.67E-08	2.28E-09
Iris	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	4.91E-04	1.00E+00	1.00E+00	1.00E+00

Table 3-23 The Wilcoxon rank sum test results of the proposed IIEFA model

Dataset	FA	KM	CFA1	CFA2	NaFA	VSSFA	DA	SCA	MFA	GA	ACO
Thyroid	3.42E-08	1.99E-08	3.03E-08	2.67E-08	1.39E-08	4.85E-08	9.74E-04	3.09E-03	6.11E-08	8.32E-02	2.77E-08
Sonar	1.19E-08	1.79E-05	4.80E-06	9.18E-08	1.48E-05	7.40E-09	2.83E-07	9.03E-11	6.19E-07	**8.06E-01**	1.10E-05
Balance	2.55E-05	6.22E-07	5.33E-08	9.90E-08	2.81E-09	5.25E-06	**7.79E-02**	3.41E-03	4.40E-03	1.58E-02	1.32E-03
E.coli	2.15E-02	2.84E-05	1.10E-02	6.45E-04	2.77E-03	**1.61E-01**	1.37E-03	4.43E-12	1.10E-02	1.32E-03	1.42E-04
Ozone	1.34E-12	1.65E-12	6.09E-13	4.40E-13	1.06E-12	1.06E-12	2.05E-13	7.73E-12	2.05E-13	8.43E-11	6.09E-13
ALL	5.87E-03	3.38E-01	**5.44E-01**	2.56E-02	7.92E-03	**4.48E-01**	**2.79E-01**	5.78E-09	**2.93E-01**	**4.64E-01**	**1.18E-01**
Wbc1	6.47E-13	6.47E-13	6.47E-13	6.47E-13	6.47E-13	6.47E-13	2.20E-11	1.86E-11	6.47E-13	8.41E-12	6.47E-13
Wbc2	4.99E-11	1.58E-13	2.59E-13	2.05E-13	1.20E-11	1.57E-13	2.47E-05	3.31E-11	5.19E-07	7.15E-13	5.56E-03
Wine	3.52E-04	1.43E-04	4.12E-05	1.21E-04	2.83E-05	9.41E-05	1.15E-02	4.99E-07	9.77E-04	5.90E-04	8.17E-05
Iris	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	4.91E-04	1.00E+00	1.00E+00	1.00E+00

In summary, the proposed IIEFA and CIEFA models outperform other algorithms in clustering problems from two perspectives, i.e. (1) constructing more compact clusters with fast convergence rates, and (2) improving clustering performance in terms of accuracy, sensitivity, specificity and Fscore_M measurements, with fewer parameter settings. Moreover, CIEFA demonstrates more advantages than IIEFA especially with data sets containing inseparable and less compact clusters (i.e. ALL) owing to its

enhanced exploration capability. Despite a wide variety of characteristics demonstrated by the above ten data sets, the real-life clustering tasks can pose greater challenges for the proposed clustering models owing to the soaring feature dimensionalities and more complex cluster distributions. Subsequently, we further examine the efficiency of the proposed models against other baseline models by undertaking three additional clustering tasks with both high dimensionalities and larger number of class categories.

3.3 Evaluation on high-dimensional clustering tasks with complex cluster distributions

With the competence of the proposed IIEFA and CIEFA models being verified both theoretically and experimentally above, we further extend our evaluations to more challenging clustering tasks with both high dimensionalities and complex cluster distributions as an attempt to examine the performance of the proposed methods more comprehensively. The extended evaluation is conducted on the basis of three additional high-dimensional UCI data sets, i.e. Drivface, Micromass, and Gas Sensor Array Drift (Sensor). The dimensionalities of Driveface, Micromass, and Sensor data sets are 6400, 1300, and 128 respectively, while the numbers of classes for these data sets are 3, 5, and 5, respectively. The details of the data sets are provided in **Table 3-24**. These data sets pose significant challenges to any clustering models owing to the considerable expansion of their dimensionalities, as well as the large numbers of class categories and more complex cluster distributions, as compared with our previous experimental studies and related research. We inherent the same experimental settings for each method as those provided in **Section 3.2.1**, i.e. population=50, maximum number of iteration=200 and runs=30. The clustering results of intra-cluster distance, accuracy, and $\mathrm{Fscore}_{\mathrm{M}}$ over 30 runs are illustrated in **Tables 3-25 – 3-27,** respectively.

Table 3-24 Three high-dimensional data sets with multiple classes for further evaluation

Data set	Number of attributes	Number of classes	Missing values	Number of instances
Drivface	6400	3	No	81
Micromass	1300	5	No	180
Sensor	128	5	No	415

As shown in **Tables 3-25 – 3-27**, the advantages of the proposed CIEFA and IIEFA models are further ascertained by the empirical results on these high-dimensional data sets. Specifically, as indicated in **Table 3-25**, with respect to the distance measure,

CIEFA yields the smallest intra-cluster distances on both the Micromass and Sensor data sets, while IIEFA produces the most compact clusters with the smallest intra-cluster distance on Drivface data set. Moreover, as depicted in **Tables 3-26 – 3-27**, the proposed CIEFA model achieves the best accuracy rates as well as $\mathrm{Fscore_M}$ results on all the three data sets, followed closely by those of IIEFA. Both proposed models show better performances than those of all the baseline methods.

Table 3-25 The mean results of the minimum intra-cluster distance measure on high-dimensional data sets over 30 runs

Dataset	IIEFA	CIEFA	FA	KM	CFA1	CFA2	NaFA	VSSFA	DA	SCA	MFA	GA	ACO
Drivface	**4849.4**	4869.6	4857.9	322247	4948.2	4908.6	4919.2	4938.4	4926.9	6161.3	5015.5	4942.9	4922.1
Micromass	656.91	**653.41**	673.38	2626.3	677.28	664.91	667.21	671.38	671.86	813.75	672.43	666.89	674.51
Sensor	426.26	**422.13**	446.94	534.03	439.08	435.31	435.34	449.97	440.72	1171.9	438.28	443.84	440.46

Table 3-26 The mean results of average accuracy on high-dimensional data sets over 30 runs

Dataset	IIEFA	CIEFA	FA	KM	CFA1	CFA2	NaFA	VSSFA	DA	SCA	MFA	GA	ACO
Drivface	0.7687	**0.7748**	0.7561	0.7583	0.7558	0.7424	0.7484	0.7536	0.7556	0.6593	0.7479	0.7605	0.7588
Micromass	0.8582	**0.8644**	0.819	0.831	0.8101	0.8381	0.8316	0.8256	0.8177	0.833	0.8221	0.8387	0.8177
Sensor	0.8118	**0.8187**	0.7928	0.8006	0.7959	0.7965	0.8003	0.7882	0.7922	0.7652	0.799	0.7968	0.7944

Table 3-27 The mean results of Fscore_M on high-dimensional data sets over 30 runs

Dataset	IIEFA	CIEFA	FA	KM	CFA1	CFA2	NaFA	VSSFA	DA	SCA	MFA	GA	ACO
Drivface	0.6524	**0.6618**	0.6502	0.6573	0.6401	0.6403	0.6436	0.6484	0.6433	0.5526	0.6355	0.6535	0.6551
Micromass	0.6332	**0.6402**	0.5397	0.5607	0.5158	0.5638	0.5535	0.54	0.512	0.6082	0.5529	0.5631	0.5289
Sensor	0.6089	**0.6255**	0.531	0.556	0.5399	0.5472	0.5602	0.5182	0.5259	0.5125	0.5573	0.5508	0.5373

Furthermore, we conduct the Wilcoxon rank sum test based on the accuracy rates obtained on above three high-dimensional data sets to further indicate the superiority of the proposed models over baseline methods. The statistical test results for CIEFA and IIEFA are provided in **Tables 3-28 – 3-29**, respectively. As shown in **Tables 3-28 – 3-29**, the majority of test results are lower than the threshold of 0.05 for both CIEFA and IIEFA. This indicates the statistical advantage of the proposed models over 11 baseline methods by delivering significantly better clustering results on the employed high-

dimensional data sets. Some baseline algorithms also exhibit competitive performance in some test cases on certain data sets, i.e. CFA1 and GA shows similar result distributions to those of CIEFA on Drivface and Micromass, respectively, while CFA1 and CFA2 have similar performances as those of IIEFA on Drivface and Micromass, respectively, with GA obtaining similar results to those of IIEFA for Micromass and Sensor, respectively.

Table 3-28 The Wilcoxon rank sum test results of the proposed CIEFA model on high-dimensional data sets

Dataset	FA	KM	CFA1	CFA2	NaFA	VSSFA	DA	SCA	MFA	GA	ACO
Drivface	3.44E-03	1.74E-02	**7.51E-02**	4.09E-03	1.23E-03	3.15E-02	3.15E-02	4.94E-10	4.24E-03	1.38E-02	2.67E-02
Micromass	3.32E-04	2.66E-03	6.60E-04	4.18E-02	2.69E-02	2.90E-03	7.93E-04	9.29E-05	3.53E-03	**7.67E-02**	1.14E-04
Sensor	4.57E-06	6.13E-04	4.15E-04	6.02E-05	1.24E-04	1.88E-06	1.96E-05	5.68E-11	7.21E-04	2.12E-04	8.80E-05

Table 3-29 The Wilcoxon rank sum test results of the proposed IIEFA model on high-dimensional data sets

Dataset	FA	KM	CFA1	CFA2	NaFA	VSSFA	DA	SCA	MFA	GA	ACO
Drivface	4.06E-03	2.21E-02	**1.17E-01**	5.99E-03	1.40E-03	4.54E-02	4.62E-02	4.35E-10	5.61E-03	1.74E-02	3.65E-02
Micromass	1.49E-04	4.60E-03	9.77E-04	**6.58E-02**	3.57E-02	1.96E-03	4.83E-04	6.52E-05	5.02E-03	**9.20E-02**	8.36E-05
Sensor	4.18E-04	4.33E-02	3.42E-02	1.91E-03	1.84E-02	5.71E-05	1.63E-03	3.64E-11	3.94E-02	**5.49E-02**	3.97E-03

It is inevitable to face the challenge of the curse of dimensionality when dealing with such high-dimensional data sets. With respect to clustering analysis in our study, the curse of dimensionality exacerbates the adversity of searching for the optimal centroids during the clustering process from two perspectives. Firstly, the diagonal-based search prescribed in the original FA model is likely to omit potential promising areas due to the constraints in its search directions and scales. The situation becomes worsened on high-dimensional data sets owing to the considerable expansion of the search space as the problem dimensionality increases, as well as the resulting oversight of even larger search sub-spaces. Therefore, the exploration capabilities of the search methods in directions and scales have great impact on model performance for the clustering tasks on such high-dimensional data sets. The proposed models are able to release the search operation from the diagonal-based search in the original FA model to the region-based multi-dimensional exploration with a greater variety of directions and scales to address the above challenges.

Secondly, as analysed in [265], when tackling high-dimensional clustering tasks, the sparse distribution of high-dimensional data sets makes the calculation of the intra-cluster distance measures less respondent to the shift of initial centroids. This could result in pre-mature convergence and early stagnation. To overcome such barriers, the CIEFA model employs a dispatching mechanism to scatter fireflies with high similarities to other unexploited territories to increase search diversity and avoid local optima traps. Such search capabilities become vital when dealing with high-dimensional large search spaces and sparse data distribution.

In summary, the proposed CIEFA and IIEFA models demonstrate significant advantages in dealing with high-dimensional data sets over the baseline methods owing to wider and more effective exploration of the search space and the enhanced population and search diversity. Nevertheless, clustering on high-dimensional data sets still remains a challenging topic owing to the possible presence of the redundant, noisy and irrelevant features that can severely affect performance. Other search strategies and feature selection models will also be explored to further enhance model efficiency in dealing with high-dimensional clustering tasks in future studies.

3.4 Further comparison and analysis between IIEFA and CIEFA

Although the distinctiveness between IIEFA and CIEFA is evident on certain challenging data sets, i.e. ALL, IIEFA and CIEFA in general demonstrate similar performances on other clustering tasks evaluated so far. CIEFA shows slightly better mean clustering performances over 30 runs on the above three high-dimensional clustering tasks with multiple clusters, but the performance differences between the two models are not distinctive. To better distinguish between CIEFA and IIEFA, both models are further evaluated with another four challenging high-dimensional data sets, i.e. a skin lesion data set (denoted as Lesion) [266], as well as three UCI data sets [255], i.e. Human Activity (Activity), Libras Movements (Libras), and Mice Protein Expression (Protein). The skin lesion data set is used in [266], which extracted shape, colour, and texture features of 660 dermoscopic skin lesion images from the Edinburgh Research and Innovation (Dermofit) lesion data set [267]. A 98-dimension feature vector for each skin lesion image was then obtained to represent the lesion information for subsequent clustering analysis. Moreover, the dimensionalities of the Human Activity, Libras, and Mice Protein data sets are 560, 90, and 77, respectively. In this

research, we employ three classes for the Libras data set and two classes for the Skin Lesion, Human Activity and Mice Protein data sets respectively. Details of the data sets are shown in **Table 3-30**. For each high-dimensional data set, a total of 30 runs are conducted for each proposed model. In order to fully evaluate the model efficiency, no feature selection is applied. The detailed clustering results are provided in **Table 3-31**.

Table 3-30 Four additional high-dimensional data sets for further comparison between IIEFA and CIEFA

Data set	Number of attributes	Number of classes	Missing values	Number of instances
Activity	560	2	No	600
Lesion	98	2	No	660
Protein	77	2	No	300
Libras	90	3	No	72

As illustrated in **Table 3-31**, the empirical results of the CIEFA model for these high-dimensional data sets demonstrate sufficient advantages over those of IIEFA according to five performance metrics, i.e. intra-cluster distances, accuracy, sensitivity, specificity, and $\mathrm{Fscore}_{\mathrm{M}}$, over 30 runs. As an example, the CIEFA model achieves higher average accuracy rates of 67.12%, 80.20%, 76.62%, and 79.07% for the Human Activity, Skin Lesion, Mice Protein, and Libras data sets, respectively, while maintaining lower intra-cluster distances with these data sets. In contrast, the IIEFA model produces comparatively slightly lower accuracy rates of 64.36%, 78.54%, 72.38%, and 78.01% for the Human Activity, Skin Lesion, Mice Protein, and Libras data sets, respectively, while producing slightly higher intra-cluster distances. A similar observation can be obtained for the other three performance metrics, i.e. sensitivity, specificity, and $\mathrm{Fscore}_{\mathrm{M}}$, for both models on most of the test cases. This indicates that the CIEFA model offers a better option, as compared with IIEFA, to undertake high-dimensional clustering tasks. This finding is also identical to that obtained by the experimental studies using the other three high-dimensional data sets as discussed in **Section 3.3**.

Table 3-31 The mean clustering results over 30 independent runs with four high-dimensional data sets

Criteria	Human Activity (560 dim)		Skin Lesion (98 dim)		Mice Protein (77 dim)		Libras (90 dim)	
	IIEFA	CIEFA	IIEFA	CIEFA	IIEFA	CIEFA	IIEFA	CIEFA
fitness	12785	**12582**	5399.8	**5352.8**	2345.0	**2307.1**	466.05	**462.95**

accuracy	0.6436	**0.6712**	0.7854	**0.802**	0.7238	**0.7662**	0.7801	**0.7907**
Fscore_M	0.6056	**0.6841**	0.8036	**0.8103**	0.7086	**0.7523**	0.6883	**0.6994**
sensitivity	0.6303	**0.7123**	0.7898	0.7750	0.7562	0.7180	0.6693	**0.6861**
specificity	0.6568	0.6300	0.7800	**0.8344**	0.6913	**0.8144**	0.8342	**0.8431**

As discussed above, complexity of clustering tasks is significantly increased on these high-dimensional data sets owing to a higher probability of inclusion of noise and redundant or contradictory features. The clustering tasks could be even more challenging especially when the data samples are not well-separated, and their distributions are far different from compact spherical. As an example, the skin lesion data set [266] consists of two types of lesions, benign and malignant. The appearance difference between these two types of lesions in terms of shape, colour and texture can be very subtle, which sometimes causes confusion even to dermatologists, therefore posing great challenges on the clustering tasks. In other words, this high-dimensional skin lesion data set contains highly inseparable and non-compact clusters. The enhanced exploration capability acquired from the additional dispersing mechanism in CIEFA accounts for its efficiency in identifying optimal centroids for this challenging lesion problem, as well as other UCI data sets, as compared with IIEFA.

In summary, the dispersing mechanism in CIEFA is able to boost the exploration capability by dispatching fireflies with high similarities in fitness values to the extended and unexploited search space. As such, the probability of identifying optimal centroids closer to the global optima is increased with the assistance of intensified local exploration as well as the expanded search territory. Therefore, CIEFA offers a better option, as compared with IIEFA, to deal with challenging clustering tasks such as data samples with high dimensionality, noise, and complicated distributions.

3.5 Summary

In this chapter, two FA variants, namely IIEFA and CIEFA, have been proposed to undertake the problems associated with initialization sensitivity and local optima traps of the conventional KM clustering algorithm. Two new strategies have been proposed in IIEFA and CIEFA to increase search diversification and efficiency. Firstly, the attractiveness coefficient in the original FA model is substituted by a randomized control matrix, therefore the one-dimensional search strategy in the original FA model is elevated to a multi-dimensional search mechanism with greater search scales and

directions for exploration in the neighbourhood. Secondly, in the early stage of the search process, a firefly solution sharing a high similarity with another is relocated to a new position outside the scope between the two fireflies in comparison. As such, the chances of identifying global optima and avoiding local optima are enhanced, owing to the fact that fireflies with high similarities are dispersed and the distribution of the whole swarm is more diversified. Therefore, the search efficiency is improved with the guarantee of sufficient variance between fireflies in comparison at the early convergence stage. The performances of IIEFA- and CIEFA-enhanced KM clustering methods are first investigated with ALL and 9 other UCI data sets, which include both high-dimensional and low-dimensional problems. In combination with mRMR-based feature selection, the proposed methods show superiority over the KM clustering algorithm, five classical search methods, and five other FA variants in terms of the convergence speed and clustering performance with respect to average accuracy rates, sensitivity, specificity, and macro-average F-score (Fscore_M) over 30 runs. The results have been ascertained using Friedman and Wilcoxon rank sum tests. In short, the proposed search strategies account for the improved efficiency in enhancing the cluster centroids of original KM clustering, which in turn overcome the local optima traps. Moreover, we conduct a further evaluation using three additional high-dimensional UCI data sets with their full dimensionalities, and the results reinforce the effectiveness and advantage of the proposed methods over other baseline models in dealing with high-dimensional clustering tasks. Lastly, a dedicated comprehensive study has also been conducted to further identify the distinctiveness between IIEFA and CIEFA using four additional high-dimensional data sets. The empirical results indicate that CIEFA outperforms IIEFA in dealing with challenging clustering tasks with noise, complicated data distributions, and non-compact and less separable clusters, owing to its enhanced exploration capability and expanded search territory.

Chapter 4

Evolutionary Feature Selection Using Enhanced Particle Swarm Optimisation

In this chapter, a comprehensive PSO variant is proposed to undertake feature selection tasks. It overcomes two major shortcomings of the original PSO model, i.e. premature convergence and weak local exploitation capability around near optimal solutions. The proposed PSO variant employs four major strategies, i.e. (1) a swarm leader enhancing mechanism using skewed Gaussian distributions, (2) a chaotic-embedded mutation scheme for worst solution enhancement based on the best leader mirroring and chaotic DE operations, (3) a diversity-enhanced evolving PSO operation using superior local and global best signals, and (4) an intensified local exploitation search action based on the swarm leader oriented logarithmic spiral operation. The first two strategies enhance the exploitation of acquired knowledge by introducing a self-improving process for the global best solution as well as facilitating communication and cooperation among elite solutions accumulated through the evolving process for weak solution enhancement, while the last two strategies elevate the capability of discovering new knowledge by constructing delicate search behaviours signified by enhanced local and global best indicators to increase search intensification and diversification and achieve an optimised trade-off between them using a dynamic switching probability schedule. As such, the proposed PSO model is able to effectively avoid being trapped in local optima and increase the likelihood of attaining global optimality. A total of 9 UCI data sets and the ALL-IDB2 data set with a wide spectrum of dimensionalities, i.e. from 30 to 10000, are employed to evaluate effectiveness of the proposed PSO model on undertaking diverse feature selection tasks. The empirical results indicate that the proposed PSO model demonstrates significant superiority in achieving better trade-off between reducing feature number and improving classification performance, and outperforms five classical search methods and five state-of-the-art PSO-based feature selection models, statistically. The advantages of the proposed PSO model become more evident on highly complex feature selection tasks owing to higher performance gaps ascribed by more successful identifications of the most discriminative and effective features.

4.1 The proposed evolutionary feature selection model

The real-life classification problems often involve data sets with high dimensionalities, and it is computationally impractical to conduct an exhaustive search of all possible feature subsets to identify the optimal feature representation [268]. Moreover, the search landscape becomes extremely complicated owing to the sophisticated confounding effects of various feature interactions in terms of redundancy and complementarity [198]. Therefore, effective and robust search methods are required to thoroughly explore complex effects of feature interactions while satisfying the constraints of practicality in term of computational cost to undertake feature selection tasks. As such, a PSO based evolutionary feature selection method is proposed to boost classification performance by automatically identifying the most effective feature subset and reducing feature dimensionality.

4.1.1 The proposed enhanced PSO model

In this section, we propose a comprehensive PSO variant for feature selection to overcome two major shortcomings embedded in the original PSO model, i.e. premature convergence and weak local exploitation capability around near optimal solutions [269]. The proposed PSO variant employs four major strategies, including (1) a swarm leader enhancing mechanism based on skewed Gaussian distributions, (2) a chaotic-embedded mutation scheme for worst solution enhancement, (3) a diversity-enhanced evolving position updating strategy based on ameliorated p_{best} and g_{best}, and (4) an intensified local exploitation search action based on a g_{best} oriented logarithmic spiral operation. The implementation of above four mechanisms is capable of increasing population and search diversification, therefore increasing the likelihood of attaining global optimality as compared with the original PSO algorithm.

The swarm particles are firstly initialized using Logistic map, with subsequent fitness evaluation to identify p_{best} for each particle and g_{best} of the overall swarm. In each iteration, the search process starts with the g_{best} enhancing operation using Gaussian distributions with positive, negative, and zero skewness, respectively. It enables g_{best} to explore further enhancement on feature selection by conducting distinctive local exploitation. Then two mutation operations for worst solution enhancement are employed to enhance the three weakest particles in the swarm as well as to increase population diversity. Specifically, the global worst particle is replaced by a new solution

generated by dimension-wise mutations of the g_{best} solution, while the last second and third particles are substituted by individuals generated by DE operations. Moreover, the fitter offspring solutions are accepted directly, while a window is opened for the acceptance of a small proportion of poor mutated solutions in the early search stage following an annealing schedule to further expand search territory. Next, positions of particles are updated according to two distinctive moving strategies to increase search diversification. Unlike the conventional PSO operation, the *first* moving mechanism employs rectified forms of g_{best} and p_{best}, as well as the Logistic map-oriented chaotic inertia weight to increase global exploration. Specifically, the global best experience in the original PSO operation is ameliorated by adopting the mean position of two individuals, i.e. the g_{best} solution and a remote neighbouring superior p_{best} solution possessing the highest dissimilarity to g_{best} in position. Similarly, the local best solution is ameliorated by adopting the mean position of the particle's own p_{best} and another randomly chosen superior p_{best} solution in the neighbourhood to gain more momentums during the search process. As a result, the search diversity of the swarm is enhanced significantly owing to the dynamic changes of both local and global best experiences through an iterative process as compared with those of the original PSO operation, therefore is less likely to be trapped in local optima. The *second* position updating strategy employs a logarithmic spiral search mechanism oriented by g_{best} to intensify local exploitation. A dynamic switching probability is designed to enable the search process to balance between the above two proposed *first* (global) and *second* (local) search operations. It ensures diverse global exploration using the *first* search action incorporating the ameliorated PSO moving strategy in the early search stage while executing sufficient local exploitation in subsequent iterations using the *second* operation integrating the logarithmic spiral search mechanism. Overall, the Gaussian distribution based g_{best} enhancement, the mutation strategies for the enhancement of the worst solutions, exploration schemes assisted by ameliorated g_{best}, p_{best}, and Logistic map, as well as the intensified fine-tuning capability using the spiral search operation, cooperate with and benefit from each other to effectively avoid being trapped in local optima and increase the likelihood of attaining global optimality. Each of the four major strategies is introduced with detailed motivations and analysis as follows.

4.1.1.1 A swarm leader enhancing mechanism

In the context of feature selection, both the elimination of critical features and the inclusion of contradictory attributes can impose significant consequences on classification performance. Therefore, a swarm leader enhancing mechanism using the skewed Gaussian distributions is proposed to equip g_{best} with further discriminative capabilities to address the above adversaries accordingly. As shown in **Eq. 4.1**, g_{best} is mutated successively based on three Gaussian distributions with different skewness settings. Specifically, Gaussian distribution with a positive skewness (right-skewed) is likely to generate more negative scores than positive ones, which can be used to simulate the effect of further eliminating noisy or irrelevant features on the basis of the g_{best} solution in each iteration. In contrast, the Gaussian distribution with a negative skewness (left-skewed) has the effects of gaining significant features owing to the production of more positive values during mutation. Additionally, the standard Gaussian distribution (non-skewed) is also employed to conduct local exploitation of g_{best} with neutrality in determining feature numbers. As a result, this leader enhancing mechanism enables g_{best} to be further improved under the scenarios where g_{best} might be trapped, i.e. eliminating effective features or incorporating noisy or irrelevant attributes.

$$gbest'_d = gbest_d + \alpha \cdot Gaussian(h) \cdot (U_d - L_d) \tag{4.1}$$

where $gbest'_d$ represents the enhanced global best solution. The parameter α denotes the step size and is assigned as 0.1 based on the recommendations of related studies [270], while h represents the skewness of the Gaussian distribution and is set as -1, 1 and 0 for left-, right- and non-skewed Gaussian distributions respectively based on extensive trial-and-error processes. Besides that, U_d and L_d represent upper and lower boundaries of the d-th dimension respectively.

4.1.1.2 Mutation-based worst solution enhancement

The lack of exploitation of local elite solutions and the sole reference to the global best solution during the coevolution among different particles are highly responsible for the local stagnation and premature convergence in the original PSO method. We subsequently further exploit acquired elite solutions in the swarm by conducting the mirroring mutation on the swarm leader and DE-based mutation on local elite solutions for the enhancement of the weakest particles.

Firstly, a g_{best}-based local mutation scheme is proposed to enhance the global worst solution in the swarm. As in **Eq. 4.2**, the new particle is produced by conducting mirroring effects and reversing the sign of g_{best} with a certain mutation probability, r_{mu}, in each dimension. This simulates the effect of randomly activating or deselecting some of features on the basis of the current best feature subset represented by g_{best}. As a result, the dimension-wise mutation has a great advantage in identifying different discriminative features as well as eliminating noisy or contradictory ones through iterations. In short, the g_{best}-based local mutation scheme guarantees the balance between preserving effective information captured by the current g_{best} solution and introducing beneficial stochastic perturbations to create new momentum for g_{best} to escape from the local optimum.

$$x_d^{new} = \begin{cases} -gbest_d & if\ rand \geq r_{mu}, \\ gbest_d & otherwise, \end{cases} \tag{4.2}$$

where r_{mu} represents the mutation probability and is set as 0.9 based on trial-and-error and recommendations in related studies [271]. When a randomly generated value is more than or equals to r_{mu}, the new offspring is assigned with the value of the mirroring $-g_{best}$ solution in the d-th dimension, otherwise it is assigned with the value of g_{best} solution in that dimension. This operation is used to yield a new solution to replace the worst particle in the swarm if it is fitter.

Secondly, a DE-based global mechanism is proposed to improve the last second and third worst individuals in the swarm. Specifically, it produces new particles by following mutation and crossover operations of DE using three p_{best} solutions randomly selected from the collection of all p_{best} individuals in the swarm, as shown in **Eqs. 4.3** and **4.4**. The differential weight F in **Eq. 4.3** is generated using Sinusoidal chaotic map to increase the variety of the perturbation for the donor vector, x_d^{donor}, in each dimension. As a result, the obtained donor vector can be more diversified in its directions and scales as compared with those yielded by the original DE method with a fixed weight. Furthermore, the crossover parameter C_r is generated by Logistic chaotic map to introduce more randomness to the crossover process in each dimension and exploit more possibilities of feature interaction on a global scale. When a randomly generated value is more than C_r, the current dimension in the new solution is inherited from the corresponding dimension of the personal best solution, otherwise it is inherited

from that of the above newly generated donor solution. Owing to the adoption of several distinctive personal best solutions in the search operations, this DE-based global mutation operation is able to increase population diversity significantly when p_{best} solutions of the particles illustrate sufficient variance from one another in the early search stage.

$$x_d^{donor} = pbest_d^1 + F \cdot (pbest_d^2 - pbest_d^3) \tag{4.3}$$

$$x_d^{new} = \begin{cases} x_d^{donor} & if\ rand \leq C_r, \\ pbest_{id} & otherwise, \end{cases} \tag{4.4}$$

where $pbest_d^1$, $pbest_d^2$ and $pbest_d^3$ represent three randomly selected p_{best} solutions of the swarm particles in the d-th dimension while $pbest_i$ represents the p_{best} solution of the current particle i. x_d^{donor} and x_d^{new} denote the donor and the new solutions in the d-th dimension respectively. Besides that, F and C_r represent the differential weight and the crossover factor respectively.

The newly generated fitter solution is accepted directly while the acceptance of a weaker mutated solution is determined by an annealing schedule, as defined in **Eq. 4.5**.

$$p = exp\ (-\frac{\Delta f}{T}) > \delta \tag{4.5}$$

where T represents the temperature for controlling the annealing process and Δf indicates the fitness difference between the mutated and the original solution. The constant δ is a randomly generated value in the range of [0, 1]. A linear cooling schedule is employed to decrease the temperature, i.e. $T = \sigma T$, whereas σ is assigned as 0.9 based on the recommendations in the existing studies [272]. With investigation, the proportion of the accepted poor offspring solutions in the total amount of generated mutated solutions is generally between 3%~5%. Therefore, this annealing schedule opens a window for a beneficial infiltration of mutated solutions to expand search territory.

The above two mutation operations based on the DE and g_{best} mirroring operations operate in parallel to combine their distinctive merits together during the search process, i.e. fully utilizing diverse p_{best} experiences especially in the early search stage using the DE-based global mutation action, as well as fully exploiting the near optimal region in

the final converging stage using the g_{best}-based local mutation operation, to improve weak particles in the swarm.

4.1.1.3 Diversity-enhanced PSO evolving strategy

The search in the original PSO operation is likely to stagnate especially when g_{best} moves to a local optima and the difference between the current position of the particle and its historical best is too small to create sufficient momentum for the particle to escape. In order to address such limitations in the original PSO model, we construct delicate search behaviours with two distinctive evolving mechanisms, i.e. a diversity-enhanced PSO evolving strategy and an intensified spiral exploitation action, to elevate both the diversification of exploration and the intensification of exploitation. Besides that, a dynamic switching probability schedule is also proposed to achieve the best trade-off between these two mechanisms and exploit merits from both search operations to the maximum extent. We firstly upgrade the position updating strategy in the original PSO operation by introducing ameliorated p_{best} and g_{best}, combined with Logistic chaotic map, to enhance search diversity and avoid local stagnation. As indicated in **Eq. 4.6**, the global best experience is ameliorated by adopting the mean position of two solutions, i.e. the g_{best} solution and a neighbouring superior p_{best} solution, i.e. $pbest^D$, possessing the highest dissimilarity to g_{best}. The dissimilarity measure between g_{best} and any p_{best} solution is determined by the number of distinctive units in their binary forms, which are converted by following existing studies [205, 206]. In other words, the p_{best} solution that has the least number of the shared selected features in comparison with those recommended by g_{best} is selected as $pbest^D$. Moreover, as defined in **Eq. 4.7**, the local best experience is ameliorated by adopting the mean position of particle's own p_{best} and another randomly chosen superior p_{best} solution, i.e. $pbest^R$, in the neighbourhood. **Eq. 4.8** is used to conduct position updating which employs the enhanced global and local optimal signals defined in **Eqs. 4.6** and **4.7**, respectively.

$$gbest_d^M = (gbest_d + pbest_d^D)/2 \tag{4.6}$$

$$pbest_d^M = (pbest_{id} + pbest_d^R)/2 \tag{4.7}$$

$$v_{id}^{t+1} = \sigma v_{id}^t + c_1 r_1 \left(pbest_d^M - x_{id}^t\right) + c_2 r_2 \left(gbest_d^M - x_{id}^t\right) \tag{4.8}$$

where $pbest^D$ represents the p_{best} solution with highest dissimilarity to g_{best} among all neighbouring superior p_{best} solutions, while $pbest^R$ represents a randomly chosen

p_{best} solution. Besides the above, $gbest^M$ and $pbest^M$ represent the enhanced global and local optimal indicators in the proposed position updating strategy respectively, while t denotes the current iteration number, and σ represents the inertia weight generated by the Logistic chaotic map.

As such, the search diversity can be improved from two perspectives. Firstly, the ameliorated g_{best}, i.e. $gbest^M$, enables particles to conduct a region-based search by navigating the swarm to move towards wider search domains signified by the mean position of g_{best} and $pbest^D$, as compared to g_{best}-based single-solution attraction in the original PSO operation. For each particle, the choice of $pbest^D$ can be different in each iteration owing to the dynamic evolving process of the swarm and adaptation of superior solutions in the neighbourhood. As a result, the search diversity can be significantly improved in terms of search directions and scopes through the iterative process. Secondly, the ameliorated p_{best}, i.e. $pbest^M$, guarantees that sufficient dynamic distractions can be obtained from the cognitive component ($pbest_d^M - x_{id}^t$) owing to the randomness in choosing $pbest^R$ from its neighbourhood personal best experiences. In comparison with the above proposed operation, the cognitive component in the original PSO action can be trivial since each particle always refers to its own historical best solution through the iterative process. As a result, the ameliorated p_{best}, i.e. $pbest^M$, can effectively produce enough momentum and enable particles to jump out of local optima traps. Additionally, the Logistic chaotic map is also employed to update inertial weight and further increase search diversity. In general, this diversity-enhanced position updating strategy not only significantly increases the momentum produced by cognitive component ($pbest_d^M - x_{id}^t$), but also efficiently improves the exploration capability of the social component ($gbest_d^M - x_{id}^t$), therefore is more likely to overcome stagnation.

4.1.1.4 Intensified spiral exploitation scheme

An intensified spiral exploitation scheme is also introduced to overcome the limitations of fine-tuning capability in near optimal regions in the original PSO model. The logarithmic spiral search is originally proposed in the Moth-Flame optimisation algorithm [63]. We employ this spiral operation to fine-tune the swarm particles in final iterations. By conducting this local spiral search action, a search space of hyper-ellipse around g_{best} is constructed on each dimension using the spiral function as defined in

Eqs. 4.9 and **4.10**, as compared with a linear approaching strategy in the original PSO model. Each particle conducts the local exploitation based on its distinctive distance from g_{best}, represented by D. This local search scheme enables particles to scrutinize the neighbourhood of g_{best} in all directions with various scales. As a result, the exploitation around near-optimal solution can be significantly intensified as compared with that of the original PSO mechanism.

$$x_{id}^{t+1} = D \cdot \exp(b \cdot l) \cos(2\pi l) + gbest_d \tag{4.9}$$

$$D = \left| gbest_d - x_{id}^{t} \right| \tag{4.10}$$

where D denotes the distance between g_{best} and particle i in the d-th dimension, while b is a constant to control the shape of logarithmic spiral, with l as a random number in the range of [-1,1]. Moreover, we also propose a dynamic switching probability schedule with the attempt to achieve a trade-off between global exploration and local exploitation in the proposed PSO variant, as demonstrated in **Eq. 4.11**.

$$p_{switch} = 1 - (iter/Max\_iter)^2 \tag{4.11}$$

where p_{switch} denotes the switching probability, while $iter$ and Max_iter represents the current and maximum iteration numbers respectively.

In each iteration, when the switching probability p_{switch} is higher than a randomly generated value in the range of [0, 1], i.e. $p_{switch} > rand$, the diversity-enhanced global search operation discussed in **Section 4.1.1.3** based on ameliorated p_{best} and g_{best} as well as Logistic chaotic map is executed. Otherwise the intensified spiral exploitation search scheme depicted in this section is conducted instead. In general, the proposed dynamic schedule of p_{switch} not only ensures sufficient global exploration opportunities to identify promising regions in the early search stage, but also guarantee thorough exploitations in near optimal region before converging in the final iterations. This smooth transition between exploration and exploitation can maximise the advantages of the proposed search mechanisms, hence reducing the likelihood of converging prematurely.

The proposed PSO variant is illustrated in **Algorithm 4-1** with the flowchart shown in **Figure 4-1**.

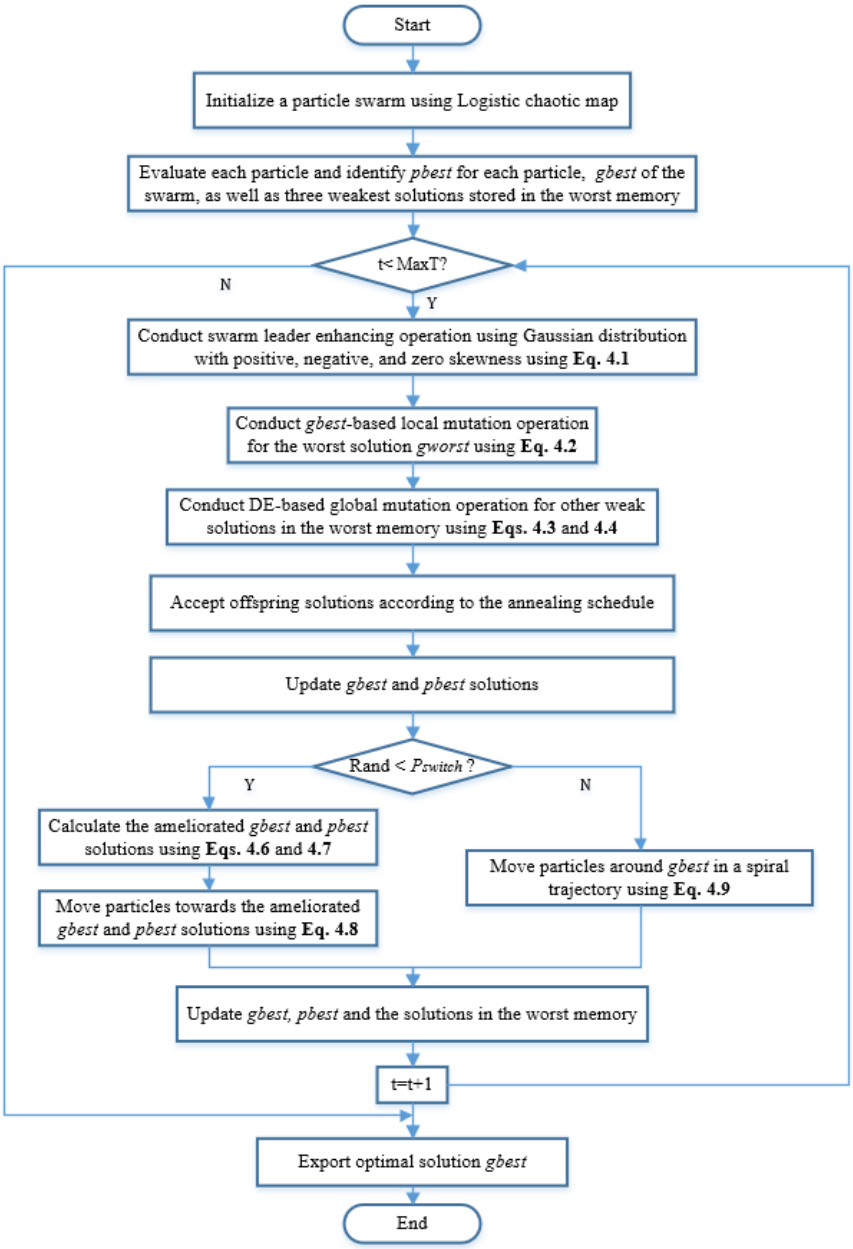

Figure 4-1 Flowchart of the proposed PSO variant

Algorithm 4-1 – The Pseudo-code of the Proposed PSO Algorithm	
1	Start
2	Initialize a particle swarm using Logistic chaotic map
3	Evaluate each particle using the objective function $f(x)$ and identify the $pbest$ solution of each particle, and the global best solution, $gbest$
4	Construct a $Worst\_memory$ which stores the three weakest particles with the lowest fitness values, and identify the worst solution as $gworst$

5 **While** (termination criteria are not met)

6 {

7 Implement the swarm leader enhancement using Gaussian distribution with positive, negative and zero skewness respectively, as defined in **Eq. 4.1**

8 **For** (each particle i in the population) **do**

9 {

10 **If** (particle i belongs to $Worst\_memory$)

11 {

12 **If** (particle i is $gworst$)

13 {

14 Construct an offspring by employing the local mutation operation based on $gbest$ as defined in **Eq. 4.2**

15 **Else**

16 Construct an offspring by employing the DE-based global mutation operation based on three randomly selected $pbest$ solutions as defined in **Eqs. 4.3** and **4.4**

17 Evaluate the offspring solution and update the position for particle i in $Worst\_memory$ based on the annealing schedule as defined in **Eq. 4.5**

18 } **End If**

19 Update $pbest$ and $gbest$ solutions

20 } **End If**

21 } **End For**

22 **For** (each particle i in the population) **do**

23 {

24 **If** $Rand < p_{switch}$

25 {

26 Establish a memory of $group_i$ which includes all neighbouring $pbest$ solutions with higher or equal fitness scores than that of the $pbest$ solution of the current particle i, i.e. $pbest_i$

27 Identify the neighbouring fitter $pbest$ solution in $group_i$ with the highest dissimilarity to $gbest$, denoted as $pbest^D$

28 Calculate the ameliorated $gbest$ solution, i.e. $gbest^M$, by averaging the following two solutions, i.e. $pbest^D$ and $gbest$, as indicated in **Eq. 4.6**

29	Randomly select another neighbouring fitter $pbest$ solution from $group_i$, denoted as $pbest^R$
30	Calculate the ameliorated $pbest$ solution, i.e. $pbest^M$, by averaging $pbest^R$ and the personal best solution of particle i, $pbest_i$, as in **Eq. 4.7**
31	Conduct position updating using $gbest^M$ and $pbest^M$ for particle i as defined in **Eq. 4.8**
32	**Else**
33	Move particle i around $gbest$ by following a logarithmic spiral search path as shown in **Eq. 4.9**
34	} **End If**
35	} **End For**
36	**For** (each particle i in the population) **do**
37	{
38	Evaluate each particle i using the objective function
39	Update $pbest$ and $gbest$ solutions
40	} **End For**
41	} **End While**
42	Output $gbest$
43	**End**

4.1.2 The proposed evolutionary feature selection model based on the enhanced PSO variant

The proposed PSO variant is integrated with a KNN classifier to conduct fitness evaluation during the search process. **Eq. 4.12** defines the objective function which is used to assess the fitness of each particle.

$$fitness(x) = k_1 * accuracy_x + k_2 * (num\_of\_features_x)^{-1} \quad (4.12)$$

where k_1 and k_2 denote weights for the classification accuracy and the number of selected features, respectively. We assign $k_1 = 0.9$ and $k_2 = 0.1$ by following the recommendations of the previous studies [102, 273].

The fitness function is able to maximize the classification accuracy while reducing the number of selected features. The particles are initialized with continuous values in each dimension using the Logistic map at the beginning of the search process. We convert

each element of each particle into a binary value, i.e. 1 or 0, representing the selection (1) or rejection (0) of a particular feature for fitness evaluation. KNN with 5 neighbours recommended by related studies [74, 274] is then used to evaluate the fitness of the selected feature subset prescribed in the binary vector. A 10-fold cross-validation is employed to assess the classification performance of each recommended feature subset based on the training set. In the testing phase, the most optimal feature subset represented by g_{best} is used to evaluate the model performance on the test data set.

In the proposed PSO model, particles undergo three evolving stages successively, i.e. (1) the swarm leader enhancing stage, (2) the chaotic-embedded mutation process for the worst solution improvement, (3) the swarm evolving stage using either the diversity-enhanced PSO mechanism or the intensified spiral exploitation scheme. Overall, the exploration capability is significantly strengthened by incorporating chaotic swarm initialisations, the local and global mutation schemes based on the best leader mirroring and chaotic DE operations, and the region-based position updating search strategy with enhanced local and global best signals, while the exploitation capability is enhanced by deploying Gaussian distribution-based swarm leader enhancement and dimension-wise spiral-shaped neighbouring search in all directions as compared with the original PSO model. As such, the impacts of complex interactions among features on classification performance can be thoroughly examined, and hence the authentic and effective feature representation with respect to the investigated problem is more likely to be identified by the proposed PSO variant.

4.2 Evaluation and discussion

We employ a total of ten data sets to investigate the efficiency of the proposed PSO variant on feature selection. The employed data sets pose diverse challenges on feature selection problems owing to a great variety of dimensionalities as well as complicated class distributions. The proposed PSO algorithm is integrated with a KNN-based wrapper model to obtain the optimal feature subset, where the number of the nearest neighbour is set to 5 according to recommendations in previous studies [74, 274]. Three performance indicators are investigated to examine the effectiveness of the proposed PSO variant in undertaking feature selection tasks, i.e. classification accuracy, number of selected features, and F-score. Furthermore, we compare feature selection performance of the proposed model against five classical search algorithms, i.e. PSO

[275], DE [56], SCA [252], DA [276], and GWO [62], as well as five PSO variants, i.e. Competitive Swarm Optimiser (CSO) [212, 277], hybrid PSO with spiral-shaped mechanism (HPSO-SSM) [74], binary PSO (BPSO) [278], modified binary PSO with local search and a swarm variability controlling scheme (MBPSO) [279], and binary PSO with catfish effect (CatfishBPSO) [280]. To ensure a fair comparison, we employ the same number of function evaluations (i.e. population size × the maximum number of iterations) as the stopping criterion for all search methods. In our experiments, the population size and the maximum number of iterations are set as 30 and 100 respectively based on trial and error. We also conduct 30 runs in each experiment to mitigate influence of accidental factors.

4.2.1 Data sets

We employ the ALL-IDB2 database [254], denoted as ALL, for Acute Lymphoblastic Leukaemia diagnosis, as well as nine other UCI data sets [255], namely Arcene, MicroMass, Parkinson's disease (Parkinson), Human activity recognition (Activity), LSVT voice rehabilitation (Voice), Grammatical facial expressions (Facial Expression), Heart disease (Heart), Ionosphere, and Wisconsin breast cancer diagnostic data set (Wdbc), for evaluation. The details of each data set are illustrated in **Table 4-1**. These data sets pose diverse challenges on any feature selection models owing to a great variety of dimensionalities and class numbers, as well as complex data distributions. Specifically, the dimensionality of the employed data sets spans from 30 to 10000, while the number of the classes ranges from 2 to 10. Six data sets with more than 300 features are characterised as high-dimensional data sets, i.e. Arcene (10000), MicroMass (1300), Parkinsons (753), Activity (561), Voice (310), and Facial Expression (301), while the remaining four data sets are characterised as low-dimensional ones, i.e. ALL (80), Heart (72), Ionosphere (33), and Wdbc (30). Moreover, according to previous studies [248, 256, 281], the employed data sets contain significant challenging factors which can severely affect classification performance, e.g. ALL [248, 256] and MicroMass [281]. As an example, the ALL-IDB2 data set poses great challenges for the reliable identification of lymphoblast cells owing to diverse complex irregular morphologies of nuclei, variations in terms of the nucleus to cytoplasm ratio, as well as the subtle differences between the blast and normal blood cells. Overall, a comprehensive evaluation can be established for the proposed PSO variant owing to the

diversity of employed data sets in terms of dimensionality, number of the classes, and sample distributions.

Table 4-1 Ten selected data sets for evaluation

Data set	Number of attributes	Number of classes	Number of instances
Arcene	10000	2	200
MicroMass	1300	10	360
Parkinsons	753	2	756
Activity	561	6	1000
Voice	310	2	126
Facial Expression	301	2	1062
ALL	80	2	180
Heart	72	4	124
Ionosphere	33	2	253
Wdbc	30	2	569

4.2.2 Parameter settings

We compare the proposed PSO variant against ten baseline methods, i.e. five classical search algorithms, i.e. PSO, DE, SCA, DA, and GWO, and five advanced PSO variants, i.e. CSO, HPSO-SSM, BPSO, MBPSO, and CatfishBPSO. The parameter settings for each baseline method employed in this study are in accordance with recommendations in their original studies. The detailed parameters for the proposed PSO model and ten baseline methods are presented in **Table 4-2**.

Table 4-2 Parameter settings of each algorithm

Algorithm	Parameters
PSO [275]	cognitive component c_1 = 2, social component c_2 = 2, inertial weight $w = 0.9 - m \times ((0.9 - 0.4)/max\_iter$, where m and max_iter denote the current and maximum iteration numbers, respectively.
DE [56]	differential weight $F \in (0, 1)$, crossover parameter C_r = 0.4.
SCA [252]	$r_1 = a - m \times a/max\_iter$, where $a = 3$. $r_2 = 2\pi \times rand$, $r_3 = 2 \times rand$,

	and $r_4 = rand.$ r_1, r_2, r_3 and r_4 are four main search parameters.
DA [276]	separation factor = 0.1, alignment factor = 0.1, cohesion factor = 0.7, food factor = 1, enemy factor = 1, inertial weight = $0.9 - m \times ((0.9 - 0.4)/max\_iter.$
GWO [62]	$A = 2 \times a \times r_1 - a$, where a is linearly decreasing from 2 to 0, and $r_1 \in (0, 1)$. $C = 2 \times r_2$, where $r_2 \in (0, 1)$. A and C are both coefficient vectors.
CSO [212]	r_1 , r_2 , r_3 $\in (0, 1)$, where r_1 , r_2 , and r_3 are search parameters randomly selected within [0, 1]. controlling parameter Φ = 0.1.
HPSO-SSM [74]	cognitive component c_1 = 2, social component c_2 = 2, inertial weight w = Logistic map. $R_1 = 1/(1 + exp\ (a \times (-\min\ (SP)/\max\ (SP)))^t$, where SP is the particle position vector, while t is the current iteration, and $a = 2$. $R_2 = 1 - R_1$.
BPSO [278]	cognitive component c_1 = 2, social component c_2 = 2, w_{max} = 0.9, w_{min} = 0.01, inertial weight $w = w_{max} - m \times (w_{max} - w_{min})/max\_iter.$
MBPSO [279]	cognitive component c_1 = 2, social component c_2 = 2, inertial weight w = 1.4, mutation probability $r_{mu} = 1/N_t$, where N_t represents the dimensionality of the problem.
CatfishBPSO [280]	cognitive component c_1 = 2, social component c_2 = 2, inertial weight w = 1, replacing rate = 0.1.
Proposed PSO	cognitive component c_1 = 2, social component c_2 = 2, inertial weight w = Logistic map, mutation probability threshold r_{mu} = 0.9.

4.2.3 Results and discussion

A comprehensive evaluation on the proposed PSO variant is established from three perspectives, i.e. (1) undertaking ten challenging feature selection tasks with a great variety of dimensionalities, (2) comparing against ten baseline search methods including five classical and up-to-date metaheuristic optimisation algorithms, as well as five advanced PSO variants from existing literatures, and (3) adopting three different

performance measures, i.e. classification accuracy, number of selected features, and the F-score measure. A total of 30 runs are conducted in each experiment to mitigate the influence of accidental factors and ensure fair comparison. The mean results over 30 independent runs are listed in **Tables 4-3 – 4-5** for classification accuracy, F-score, and the number of selected features, respectively. The best results are marked in bold accordingly.

Table 4-3 The mean results of the classification accuracy over 30 runs

Data sets	PSO	DE	SCA	DA	GWO	CSO	HPSO-SSM	Catfish BPSO	BPSO	MBPSO	Proposed PSO
Arcene	0.7217	0.7244	0.7372	0.7183	0.7211	0.7372	0.7122	0.7100	0.7111	0.7117	**0.7411**
MicroMass	0.5897	0.6052	0.6061	0.5933	0.6124	0.5409	0.5903	0.5836	0.5758	0.5785	**0.6455**
Parkinsons	0.7949	0.7990	0.7922	0.7862	0.7940	0.7985	0.8000	0.7994	0.7988	0.7962	**0.8115**
Activity	0.8813	0.8919	0.8826	0.8785	0.8929	0.8876	0.8860	0.8785	0.8725	0.8775	**0.9025**
Voice	0.8237	0.8149	0.8202	0.8272	0.8219	0.7789	0.8237	0.8193	0.8263	0.8246	**0.8526**
Facial Expression	0.7187	0.6748	0.6891	0.6635	0.6844	0.6861	0.6914	0.6998	0.7170	0.7274	**0.7351**
ALL	0.8951	0.9167	0.9037	0.9025	0.8858	0.8728	0.8944	0.9123	0.8938	0.8988	**0.9185**
Heart	0.5963	0.6435	0.6620	0.5537	0.6398	0.5713	0.6444	0.5769	0.5815	0.5750	**0.6731**
Ionosphere	0.8171	0.8285	0.8320	0.8101	0.8197	0.8184	0.8189	0.8066	0.8276	0.8110	**0.8351**
Wdbc	0.9520	0.9534	0.9191	0.9458	0.9386	0.8828	0.9261	0.9497	0.9501	0.9454	**0.9571**

With respect to classification accuracy as illustrated in **Table 4-3**, the proposed PSO variant achieves the highest accuracy scores on all ten classification tasks with a great variety of dimensionalities, i.e. from 30 to 10000, and outperforms the ten baseline algorithms consistently. Among high-dimensional feature selection tasks with more than 300 features, the proposed PSO variant achieves accuracy rates of 90.25%, 85.26%, and 81.15% on Activity, Voice, and Parkinsons data sets, respectively. Among low-dimensional data sets, the proposed PSO achieves the accuracy rates of 95.71%, 91.85%, and 83.51% on Wdbc, ALL, and Ionosphere, respectively. The above observations indicate the effectiveness of the proposed PSO variant in undertaking both the high-dimensional and low-dimensional feature selection tasks. Moreover, the empirical results also reveal the great advantages of the proposed model over the ten baseline methods, especially on MicroMass and Heart data sets. Specifically, the proposed PSO

model achieves the accuracy rate of 64.55% and 67.31% on MicroMass and Heart data sets, respectively. For the MicroMass data set, it outperforms the top three best performed classical search methods, i.e. GWO, SCA, and DE, by 3.31%, 3.94%, and 4.03%, respectively, as well as the top three PSO variants, i.e. HPSO-SSM, CatfishBPSO, and MBPSO, by 5.52%, 6.19%, and 6.70%, respectively. With respect to Heart data set, it outperforms the top three best performed classical search methods, i.e. SCA, DE, and GWO, by 1.11%, 2.96%, and 3.33%, respectively, as well as the top three PSO variants, i.e. HPSO-SSM, BPSO, and CatfishBPSO, by 2.87%, 9.16%, and 9.62%, respectively. Moreover, similar or even larger performance gaps between the proposed PSO variant and other baseline models can also be observed for these two sets owing to the fact that all of classification accuracy rates obtained by these remaining baseline methods are all less than 60%. We analyse the performance gaps caused by the challenging factors of these two data sets and the superiority of the proposed model below.

With respect to the MicroMass data set, the identification of bacterial species is rather challenging owing not only to massive dimensionality (i.e. 1300) and a large number of species (i.e. 10), but also to the various complexities imposed by polymicrobial samples [281]. These polymicrobial samples were generated by mixing two bacterial strains with different taxonomic proximities, which are characterised by the variance in terms of bacterial species, genera and Gram types. Some bacterial species used for mixing are highly indistinguishable, e.g. *Bacillus cereus* and *Bacillus thuringiensis*, *Escherichia* and *Shigella* genus [282]. Besides that, a total of nine different concentration ratios were also employed when mixing bacterial strains, i.e. 1:0, 10:1, 5:1, 2:1, 1:2, 1:5, 1:10, 0:1, which significantly increases the variance of samples within the same class by diluting the proportion of critical features and amplifying the distraction from irrelevant and noisy information. As a result, it is extremely challenging to classify those polymicrobial samples correctly owing to the confounding effects imposed by the interference of various concentration scenarios and existence of indistinguishable species. Those evident performance gaps on the MicroMass data set indicate great advantages of the proposed PSO model over other baseline search methods in successfully identifying bacterial species out of numerous distraction factors. Those bacterial strain features selected by the proposed PSO variant effectively capture the critical characteristics of bacterial species and remain robust under various

environmental changes, such as concentration ratios and types of bacterial strains used for mixture, therefore resulting in a higher classification accuracy rate.

The effectiveness of the proposed PSO model can be ascribed to the cooperation between the proposed swarm leader enhancing scheme, diversity-enhanced movements and parallel mutation operations. The leader enhancing scheme enables the global best solution to conduct various local jumps to escape from stagnations caused by omitting essential features or including noisy ones. The proposed moving strategy diversifies search directions and expands search territory by incorporating three improvements, i.e. region-based search, dynamic cognitive distractions and chaotic-based inertia. Besides that, the chaotic-embedded local $gbest$ mirroring and DE-based global mutation schemes in parallel further endow the swarm with enormous opportunities to escape from local optima traps. The above three strategies are able to support each other as augmentations when one of them fails to overcome the local stagnation individually. In contrast, search strategies in baseline models are too monotonous to escape from complex local optima traps in NP-hard problems, such as feature selection tasks. Overall, as a result of the proposed comprehensive counter strategies of local optima traps, the search diversity and robustness are significantly enhanced in the proposed PSO model, therefore the likelihood of ascertaining the global best solution, i.e. identification the best feature subset with essential features being included and noisy ones excluded, is significantly improved.

Likewise, the diagnosis of coronary heart disease is also considered a challenging problem owing to a great variety of class categories and complex characteristics embedded in those employed features, i.e. demographic, symptom and examination, laboratory, ECG, fluoroscopy as well as echo [283]. We employ the Cleveland heart disease database [284] not only to diagnose coronary heart disease, but also to distinguish three different severity levels of disease symptoms developed through the chronical long-term condition of heart failure. As a result, the feature selection tasks become more challenging owing to the ambiguous fuzzy boundaries between different classes as compared to the common binary classification scenario for distinguishing normal from diseased cases. With inspection, the feature subsets generated by the proposed PSO model contain the following critical factors, e.g. chest pain type, serum cholestoral, fasting blood sugar, maximum heart rate, exercise induced angina, and ST

depression etc., which have been identified as essential features for the diagnosis of heart disease in the existing studies [285, 286].

Furthermore, the significant performance gaps achieved by the proposed PSO variant over other baseline models indicate the effectiveness of the proposed PSO variant in identifying the most discriminative features which can better represent the main characteristics of each severity level of heart disease and reflect the nuance changes between them. This effectiveness in identifying the most discriminative features in classification tasks with ambiguous fuzzy boundaries can also be ascribed to the cooperative mechanism of the above proposed strategies, i.e. the swarm leader enhancing scheme, diversity-enhanced movements and parallel mutation operations. Each of them cooperates with each other to enhance search diversity and reduce the likelihood of being trapped at local optima. Overall, the empirical results indicate the significant superiority of the proposed PSO model over other baseline methods in undertaking feature selection tasks with higher complexities and sophistications, e.g. various distraction factors in sample distributions, and large intra-class and small inter-class variations.

Table 4-4 The mean results of the F-score over 30 runs

Data sets	PSO	DE	SCA	DA	GWO	CSO	HPSO-SSM	Catfish BPSO	BPSO	MBPSO	Proposed PSO
Arcene	0.6759	0.6757	0.6963	0.6780	0.6783	0.6959	0.6646	0.6574	0.6573	0.6590	**0.6977**
MicroMass	0.6349	0.6469	0.6428	0.6314	0.6445	0.5982	0.6350	0.6275	0.6219	0.6200	**0.6759**
Parkinsons	0.8691	0.8712	0.8670	0.8631	0.8686	0.8701	0.8720	0.8719	0.8716	0.8702	**0.8798**
Activity	0.8864	0.8962	0.8874	0.8833	0.8971	0.8930	0.8901	0.8838	0.8783	0.8824	**0.9067**
Voice	0.7180	0.7381	0.7265	0.7316	0.7208	0.6890	0.7339	0.7328	0.7368	0.7399	**0.7764**
Facial Expression	0.6458	0.6191	0.6288	0.6175	0.6287	0.5670	0.6292	0.6342	0.6527	0.6556	**0.6572**
ALL	0.9204	0.9345	0.9250	0.9266	0.9084	0.9037	0.9168	0.9331	0.9195	0.9241	**0.9361**
Heart	0.6039	0.6502	0.6661	0.5616	0.6436	0.5823	0.6513	0.5881	0.5904	0.5788	**0.6783**
Ionosphere	0.8439	0.8516	0.8550	0.8375	0.8427	0.8418	0.8452	0.8371	0.8521	0.8380	**0.8562**
Wdbc	0.9340	0.9355	0.8836	0.9246	0.9146	0.8286	0.8957	0.9308	0.9312	0.9239	**0.9415**

The effectiveness of the proposed PSO model is further ascertained by the results of the F-score measure as shown in **Table 4-4**. The proposed model achieves the highest F-

score results on all ten data sets and demonstrates great advantages over the ten baseline algorithms, i.e. PSO, DE, SCA, DA, GWO, CSO, HPSO-SSM, CatfishPSO, BPSO, and MBPSO. Similar to the accuracy measures, the advantages on F-score become more evident on feature selection tasks embedded with higher complexities owing to greater performance gaps between the proposed PSO variant and the baseline search methods. To be specific, the F-score results achieved by the proposed PSO variant on MicroMass, Voice, and Heart data sets, are 67.59%, 77.64%, 67.83% respectively. With respect to the Voice data set, the proposed PSO model outperforms the three best performed classical search methods, i.e. DE, DA, and SCA, by 3.83%, 4.48%, 4.99%, as well as top three PSO variants, i.e. MBPSO, BPSO, and HPSO-SSM, by 3.65%, 3.96%, and 4.25%, respectively. For the MicroMass data set, it outperforms the top three best performed classical search methods, i.e. DE, GWO, and SCA, by 2.90%, 3.14%, and 3.31%, as well as the top three PSO variants, i.e. HPSO-SSM, CatfishBPSO, and BPSO, by 4.09%, 4.84%, and 5.4%, respectively. With respect to Heart data set, it outperforms the three best performed classical search algorithms, i.e. SCA, DE, and GWO, by 1.22%, 2.81%, and 3.47%, as well as top three PSO variants, i.e. HPSO-SSM, BPSO, and CatfishBPSO, by 2.70%, 8.79%, and 9.02%, respectively. Such evident performance gaps are present or become more apparent or severe for other weaker baseline methods. Overall, the F-score measures further reinforce the effectiveness and the superiority of the proposed PSO model over other classical and advanced search methods in undertaking diverse feature selection tasks, especially those with higher sophistications of complex sample distributions.

Table 4-5 The mean results of the number of selected features over 30 runs

Data sets	PSO	DE	SCA	DA	GWO	CSO	HPSO-SSM	Catfish BPSO	BPSO	MBPSO	Proposed PSO
Arcene	3976.07	4046.13	3388.57	3695.37	2770.40	**2545.30**	3967.17	4424.80	4977.17	4973.97	3395.03
MicroMass	548.63	527.20	439.77	485.93	**330.57**	1123.00	554.27	588.77	646.23	641.53	461.30
Parkinsons	323.30	310.20	266.33	283.20	**209.80**	492.03	323.57	361.63	378.07	374.40	273.07
Activity	237.63	222.90	184.03	208.23	**146.27**	394.37	232.67	255.67	277.17	277.80	194.00
Voice	128.03	121.37	108.27	118.13	86.70	**64.97**	122.03	140.20	152.90	148.20	108.57
Facial Expression	131.40	112.83	88.37	72.00	80.73	**60.10**	84.63	121.60	146.2	141.97	92.73
ALL	26.53	23.03	18.37	29.47	12.80	**9.53**	25.37	28.83	35.40	33.27	18.97
Heart	28.80	23.87	20.87	27.83	**17.77**	56.73	26.43	31.93	34.03	30.87	21.83

Ionosphere	12.47	**9.30**	9.63	11.83	9.40	9.60	11.30	13.10	12.53	10.63	10.30
Wdbc	9.93	5.47	3.87	9.40	4.73	**3.40**	4.67	10.37	10.83	6.83	9.80

With respect to the number of selected features, CSO selects the least features on five data sets, i.e. Arcene, Voice, Facial Expression, ALL and Wdbc, while GWO obtains the smallest feature sizes on four data sets, i.e. MicroMass, Parkinsons, Activity, and Heart. Owing to the excessive elimination of essential features, CSO achieves the lowest classification accuracy rates on Voice, ALL and Wdbc data sets. As an example, CSO obtains an accuracy rate of 77.89% with an average of 64.97 features being selected on Voice data set over a set of 30 runs. In contrast, other search methods all achieve 80%+ accuracy scores while selecting more than 100 features except for GWO where 86.7 features are selected on average. This indicates that CSO falls into local optima on this Voice data set during training which leads to the stagnation in performance. According to the fitness evaluation illustrated in **Eq. 4.12**, this phenomenon in turn results in the severe removal of features in order to further improve the fitness scores. As such, very small feature subsets are identified during the feature selection process, which may not be able to capture sufficient characteristics for the Voice data set and lead to the severe performance deterioration in the test stage.

On the contrary, the proposed PSO variant succeeds in achieving a more efficient trade-off between eliminating redundant features and improving performance. It selects comparatively smaller feature subsets than those of most of the search methods e.g. DE, DA and HPSO-SSM methods in most of the test cases while achieving the highest accuracy rates and the F-score measures on all ten test data sets. In particular, the proportions of eliminated features by the proposed PSO model are 66.05%, 64.51%, 63.73%, 65.41%, 64.98%, and 69.19%, on six high-dimensional data sets, i.e. Arcene, MicroMass, Parkinsons, Activity, Voice, and Facial Expression, respectively. In short, the empirical results indicate the significant capabilities of the proposed PSO variant in removing irrelevant and noisy features while identifying the most discriminative and effective ones without falling into local optima traps.

The Wilcoxon rank sum test is conducted based on the mean classification accuracy to further indicate the statistical distinctiveness of the proposed PSO model against baseline methods. As illustrated in **Table 4-6**, the majority of test results are lower than

0.05, which indicates that the proposed PSO model significantly outperforms ten baseline models on the majority of the employed data sets. Besides that, the advantages demonstrated by the proposed PSO model become even more evident on classification tasks which are embedded with higher dimensionalities and sophisticated class distributions. Specifically, a total of four cases occur among sixty evaluations (6 high-dimensional data sets × 10 baseline algorithms) on high-dimensional data sets, where the proposed model does not show statistically significant differences from other models, i.e. SCA and CSO on Arcene, PSO and MBPSO on Facial Expression, while a total of seven cases of insignificant differences among forty evaluations (4 low-dimensional data sets × 10 baseline algorithms) happen on low-dimensional data sets. As an example, the proposed PSO model reveals similar performances to those of three search methods, i.e. DE, SCA, and BPSO, respectively, on Ionosphere data set of 30 features. In contrast, it demonstrates significant statistical distinctiveness from every baseline model on four high-dimensional data sets, i.e. Micromass (1300), Parkinson (753), Activity (561), and Voice (310). Overall, the statistical results further prove the significant superiorities of the proposed PSO model over the classical search methods and PSO variants, especially in undertaking feature selection tasks with higher complexities and sophistications.

Table 4-6 The Wilcoxon rank sum test results of the proposed PSO model

Data sets	PSO	DE	SCA	DA	GWO	CSO	HPSO-SSM	Catfish BPSO	BPSO	MBPSO
Arcene	1.53E-02	3.53E-02	**8.75E-01**	2.44E-02	1.93E-02	**6.16E-01**	1.48E-03	4.41E-04	6.08E-04	6.28E-04
MicroMass	2.47E-04	7.55E-03	8.69E-03	3.50E-04	4.11E-02	1.05E-09	2.12E-04	2.90E-05	5.30E-06	1.13E-05
Parkinsons	1.65E-03	3.15E-02	6.60E-03	1.99E-05	2.38E-03	3.35E-02	4.69E-02	4.52E-02	3.93E-02	3.31E-02
Activity	3.93E-06	6.61E-03	1.27E-04	1.40E-05	1.19E-02	4.51E-05	1.05E-03	1.49E-07	1.07E-08	2.12E-08
Voice	3.21E-02	6.20E-03	9.98E-03	4.48E-02	2.78E-02	9.85E-04	3.35E-02	4.04E-03	2.91E-02	1.83E-02
Facial Expression	**5.24E-01**	8.72E-05	1.23E-03	1.75E-06	5.63E-04	4.14E-05	5.06E-04	4.69E-03	1.92E-02	**3.40E-01**
ALL	7.85E-03	**7.75E-01**	4.79E-02	2.92E-02	3.45E-03	1.35E-03	3.82E-02	**4.76E-01**	1.98E-03	3.11E-02
Heart	1.44E-04	2.16E-02	**2.94E-01**	2.20E-09	3.15E-02	1.21E-09	3.84E-02	1.29E-06	2.87E-07	1.26E-07
Ionosphere	1.16E-02	**6.10E-01**	**8.11E-01**	1.15E-03	4.18E-02	3.82E-02	2.77E-02	2.06E-04	**7.87E-01**	4.58E-03
Wdbc	2.48E-02	**5.23E-01**	3.02E-05	1.30E-02	3.54E-02	5.44E-09	1.84E-04	1.84E-02	1.82E-02	4.16E-03

This effectiveness in constructing simplified but valid feature subsets while improving classification performance can be ascribed to the incorporation of the proposed search

strategies, i.e. (1) the swarm leader enhancing mechanism based on skewed Gaussian distributions, (2) the chaotic-embedded local $gbest$ mirroring and DE-based global mutation schemes, (3) the diversity-enhanced evolving strategies based on ameliorated p_{best} and g_{best}, and (4) the g_{best} oriented intensified spiral exploitation. The first two strategies elevate the mining and utilisation of acquired knowledge in the swarm from two perspectives, i.e. introducing a self-improving process for the global best solution and facilitating the communication and cooperation among elite solutions accumulated through the evolving process for weak solution enhancement. Specifically, the swarm leader enhancing mechanism improves the quality of the global best solution by the endowment of the capability of further acquiring effective features and of abandoning noisy features, prescribed by Gaussian distributions with skewness. This scheme enables the global best solution to escape from local optima by gaining momentum from three possible self-improving processes, i.e. gaining features, losing features, and random jump with neutrality of above two effects. Therefore, the obtained feature subsets represented by the global best solutions are less likely to fall into overfitting problems caused by the stagnation at local optima traps. Moreover, the local and global mutation schemes for weak solution enhancement boost both population diversity and search scope by hybridising elite personal best solutions with DE updating rules and imposing mirroring effects on the global best solution. As a result, the effective information stored in elite solutions can be fully exploited and permutated. The search territory is hence expanded and fitter solutions are produced to replace worse individuals.

In contrast, the last two strategies optimise the search behaviour to enhance the capability of acquiring new knowledge through the whole search process. Delicate search behaviours with two distinctive evolving mechanisms are constructed to elevate both the diversification of exploration and the intensification of exploitation. Specifically, the first evolving strategy enhances exploration diversity by employing three improvements, i.e. conducting region-based search in social component, applying dynamic distractions in cognitive component, and utilising chaotic inertia weight. In region-based search, the swarm is navigated towards a promising search domain signified by the mean position between the g_{best} solution and the personal best solution which possesses the highest dissimilarity to g_{best} among all neighbouring fitter p_{best} solutions. As a result, each particle is capable of conducting diversified search by aiming at various search directions and distinctive search regions, owing to the dynamic

adaptations of superior p_{best} solutions in the neighbourhood. On the other hand, the effect of cognitive component is also diversified by following both the current particle's personal best solution and an additional fitter p_{best} solution which is randomly selected in the neighbourhood. This mechanism enables particles to escape from local stagnation by creating additional dynamic momentums through iterative process. Moreover, the Logistic map is also employed to generate inertia weight and introduce chaotic perturbations into the search process. The above three improvements, i.e. region-based search, dynamic distractions, and chaotic-based inertia, incorporated in the first evolving strategy, are able to significantly enhance search diversity in terms of expanding search scopes and differentiating search directions, therefore reducing the likelihood of being trapped at the local optima. The second evolving strategy intensifies local exploitation by conducting a spiral search around the global best solution. This exploitation scheme overcomes the weakness of fine-tuning capability in the original PSO model and enables the particles to scrutinize the neighbourhood of g_{best} in all directions with various scales. As a result, the above two evolving strategies enable the exploration to be more diversified and the fine-tuning process to be more intensified in the swarm, therefore increasing the capability of discovering new knowledge, i.e. fitter solutions, through the overall search process.

Overall, the proposed PSO variant improves the utilisation of the acquired knowledge by conducting local and global mutation schemes on elite solutions using Gaussian distributions and chaotic DE actions, and enhances the capability of acquiring new knowledge by a careful consideration on the elevation of exploration and exploitation, between the region-based search strategy with enhanced local and global leaders, and the logarithmic spiral search surrounding the global best solution. With enhancement in search and population diversity resulted from above proposed mechanisms, the original feature space can be thoroughly explored and a greater search territory can be covered, while the stagnation at local optima traps can be effectively prevented.

In contrast, for the employed baseline classical search methods, certain limitations have been identified in previous studies and widely discussed in the research community. Specifically, DE suffers from premature convergence owing to a limited amount of exploratory moves. In other words, the search can be severely compromised owing to the failure of generating promising solutions within a limited number of function evaluations [287]. GWO demonstrates a strong bias towards the origin of the coordinate

system attributed by its simulated model [120], as well as proneness to stagnation at local optima traps [288], while DA suffers from poor exploitation capability owing to the fact that it does not keep track of elite solutions[289]. In addition, the majority of the employed PSO variants only equip improvements from the perspective of either exploration or exploitation capability, rather than comprehensively taking into account the trade-off between above two operations. Overall, the proposed PSO model demonstrates great superiorities over baseline methods in attaining the global optimality owing to a delicate consideration of both the global exploration and local exploitation, as well as the enhanced population diversity entailed by both the local and global mutation schemes on elite solutions. Therefore, the proposed model is capable of improving classification performance by identifying the most discriminative features and eliminating noisy and irrelevant ones as evidenced by above evaluation and statistical test results.

4.3 Summary

In this chapter, a PSO variant has been proposed to overcome drawbacks of premature convergence and inefficient fine-tuning capability of the original PSO model, as well as undertake challenges of complex fitness landscapes embedded in feature selection tasks. The proposed PSO model incorporates four distinctive strategies to elevate the exploitation of elite solutions accumulated through the iterative process as well as enhance the capability of identifying undiscovered promising solutions in a global scale through a careful design of delicate search behaviours. Firstly, a swarm leader enhancing mechanism is proposed to endow the global best solution with the capability of conducting local jumps with customized characteristics prescribed by Gaussian distributions with positive, negative and zero skewness respectively. This operation enables the global best solution to escape from local optima traps induced by either eliminating effective features or incurring conflictive ones. Secondly, the mirroring and DE-based mutation operations based on the swarm leader and the local elite solutions respectively are employed in parallel to enhance three weakest solutions in each iteration. These mutation-based strategies improve population diversity and expand the search territory owing to the hybridising effects of elite solutions in the swarm. Thirdly, the diversity-enhanced PSO evolving strategy is employed by incorporating three improvements, i.e. conducting region-based search in social component, applying dynamic distractions in cognitive component, and utilising chaotic inertia weight. As a

result, the exploration capability is significantly elevated by the enhanced coevolution process owing to the dynamic references to multiple distinctive leaders in both the cognitive and social components. Lastly, a logarithmic spiral search is further deployed to strengthen fine-tuning capability to thoroughly exploit the near-optimal region. Overall, the first two strategies enhance the exploitation of acquired knowledge by conducting various local jumps on the global best solution as well as facilitating cooperation and communication among local elite solutions for worst solution enhancement, while the last two strategies elevate the capability of discovering new knowledge by constructing delicate search behaviours which both boost exploration and exploitation capabilities through the iterative process. As such, the proposed PSO model is less likely to be trapped in local optima and more likely to attain the global optimality.

The performance of the proposed PSO model has been investigated by undertaking feature selection tasks using the ALL-IDB2 database and 9 other UCI data sets with diverse dimensionalities from 30 to 10000. The results indicate that the proposed PSO model demonstrates great advantages in undertaking both the low-dimensional and high-dimensional feature selection tasks by obtaining the highest classification performances on the employed ten data sets and achieving a better trade-off between feature selection and classification performance. It significantly outperforms five classical search methods as well as five advanced PSO-based feature selection models, as evidenced by the classification accuracy rates and F-score measures as well as the statistical test results. These advantages generally become more evident on high-dimensional feature selection tasks owing to the effectiveness of the proposed strategies in terms of enhancing exploration and exploitation capabilities as well as sufficient mutation-based local optima escaping mechanisms. Overall, the proposed PSO model demonstrates great advantages in undertaking feature selection tasks, especially those with higher complexities and sophistications, e.g. various distraction factors in sample distributions, and large intra-class and small inter-class variations, owing to the enhancement of population and search diversities endowed by the employed four strategies.

Chapter 5

Evolving CNN-LSTM Models for Time Series Prediction Using Enhanced Grey Wolf Optimizer

In this chapter, an enhanced GWO model is proposed for the devising of evolving CNN-LSTM networks for time series analysis. In order to overcome the stagnation at local optima traps and a slow convergence rate of the original GWO algorithm, the newly proposed variant incorporates several distinctive search mechanisms, i.e. a nonlinear exploration scheme for dynamic search territory adjustment, a chaotic leadership dispatching strategy among the dominant wolves, a rectified spiral local exploitation action, as well as probability distribution-based leader enhancement. Evolving CNN-LSTM models are subsequently devised using the proposed GWO variant, where the network topology and learning hyperparameters are optimized for time series prediction and classification. Evaluated using UCI energy consumption, PM2.5 concentration and human activity recognition data sets, the proposed GWO-optimized CNN-LSTM models demonstrate statistically significant superiority over those yielded by several classical search methods and advanced GWO and PSO variants. The empirical results also indicate that the deep networks devised by the proposed GWO algorithm illustrate superior representational capacities to not only effectively capture the vital feature interactions, but also encapsulate sophisticated dependencies in complex temporal contexts, in comparison with those yielded by the baseline methods.

5.1 The proposed evolving time series prediction model

The study on time series analysis is driven by the desire to not only understand the past but also predict the future [290]. A time series is a sequence of data measured chronologically at a uniform time interval. Time series measurements are prevalent in various domains, such as weather forecast [291], financial market prediction [292], physiological assessment [293] and video analysis [294]. Over the last several decades, many efforts have been made to develop effective time series forecasting models which can be classified into three categories: 1) statistical models, e.g. auto-regressive moving average (ARMA) [295] and auto-regressive integrated moving average (ARIMA) [296]; 2) machine learning models, e.g. Support Vector Regression (SVR) [297] and ANNs [298]; 3) deep learning models, e.g. RNNs [299] and LSTM [229]. In particular, the LSTM network is regarded as the state-of-the-art time series forecasting model owing to its capability of learning long-term temporal dependences through the design of gated units integrating activations of sigmoid and hyperbolic tangent functions.

Despite the progress achieved by LSTM, multi-variate time series forecasting remains a challenging task owing to the complex factors embedded in real-life sequential data, such as sophisticated dependencies, irregularity, randomness, cross-correlation among variables, as well as data noise [300, 301]. Besides that, hyperparameters in relation to the configuration of LSTM, e.g. the number of hidden nodes, as well as the learning properties during the training process, e.g. learning rate, play vital roles in affecting the performance of the LSTM networks [302, 303]. However, the identification of the optimal hyperparameter settings for LSTM networks remains a challenging task owing to the complexity of the problems at hand and the requirement of profound domain knowledge. The traditional manual trial-and-error fine-tuning process is likely to result in sub-optimal model representational capacities and ill-performed learning parameters, therefore compromising the performance of LSTM networks. In order to resolve the aforementioned challenges of the time series data as well as optimal learning configuration identification of LSTM networks, we incorporate two automatic processes into the vanilla LSTM structure, i.e. automatic feature extraction and optimal network configuration identification, to enhance the performance of the monotonous LSTM networks in tackling time series prediction. Essentially, CNNs are hybridized with LSTM to extract fundamental features from the input sequence automatically and construct more accurate feature representations of the investigated time series tasks. Moreover, an evolving process is introduced for the generation of the optimal configurations of the hybrid deep network by exploiting the strength of the advanced SI algorithm, i.e. GWO [62].

To be specific, the proposed evolving time series prediction model consists of two major components, i.e. the proposed CNN-LSTM network and the GWO variant. The CNN-LSTM network is the core component to make prediction based on data sequences whereas the proposed GWO variant is employed to search for the optimal hyperparameters for the devised CNN-LSTM model. In CNN-LSTM, the time series data are served as the input to the convolutional layers in order to extract main features surrounded by the temporal context and reduce irrelevant variations. The obtained feature maps are then fed into the LSTM layers to analyse temporal variations and learn long-term dependencies. The fully connected layer is applied subsequently to conduct nonlinear transformations on the extracted features and produce prediction results. As discussed earlier, the performance of deep CNN-LSTM model is significantly influenced by the quality of hyperparameter settings, such as the number of filters in convolutional layers, the number of hidden nodes in the LSTM layer, as well as the learning configurations, e.g. learning rate, which determine the representational capacity and the training properties of the employed model. Therefore, an enhanced GWO model is proposed to automatically identify the optimal configuration of such hyperparameters for the devised CNN-LSTM network. The identified optimized CNN-LSTM model is subsequently used to undertake time series prediction and classification. The details of the proposed GWO and CNN-LSTM models are introduced as follows. **Figure 5-1** depicts the diagram of the proposed GWO-based evolving CNN-LSTM time series forecasting model.

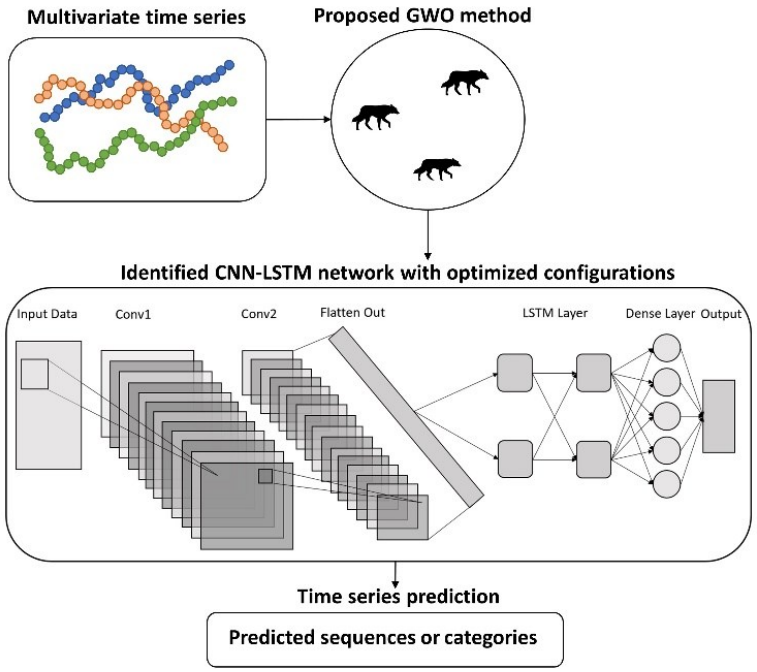

Figure 5-1 The diagram of the proposed GWO-based evolving CNN-LSTM time series forecasting model where each wolf represents a set of network topology and learning hyperparameters for evolution

5.1.1 The proposed GWO variant

As mentioned in **Section 2.1.3**, GWO is a recently developed SI algorithm which demonstrates robust and advanced search capabilities by the mechanism of following the guidance of top three swarm leaders, i.e. wolves α, β, and δ, as well as a dedicated design of the transition from exploration to exploitation, i.e. the exploration rate a. Despite these merits, the original GWO algorithm still suffers from severe obstacles of local optima traps, owing to its search bias, especially towards the origin of the coordinate system [120, 304], as well as the limitations of search diversity. Moreover, the static and equal division of the leadership among three strongest wolves over the whole search course contradicts its strategy of hierarchical division within the wolf community in principle, and largely confines the capability of fine-tuning around the obtained global best solution.

Therefore, in this research, we propose four distinctive mechanisms to resolve the abovementioned restrictions and enhance the global exploration and local exploitation of the original GWO algorithm. Firstly, a nonlinear adjustment of the exploration rate a' is proposed to replace a and advance the search transition between exploration and exploitation, by delaying the shrinkage of the search territory during exploration while concentrating the detection on the promising neighbourhood around wolf leaders during exploitation. Secondly, a sinusoidal chaotic map is employed to generate dynamic yet clamped weights, to simulate benevolent competitions among the three dominant wolves, α, β, and δ, for leading the wolf pack. As such, a trade-off between reinforcing the leadership of the best individual and diversifying the distractions of the second and third best solutions can be achieved. Furthermore, a damped odd function with the shrinking amplitude is proposed to deploy a fine-tuning local search process around the swarm leader in the final stage to accelerate convergence. Lastly, the Lévy flight is employed to further enhance the quality of three leading wolves α, β and δ, in each iteration, to overcome early stagnation.

5.1.1.1 A nonlinear exploration factor for adjustment of search boundary

In the original GWO algorithm, the transition from exploration to exploitation is governed by the exploration rate a as defined in **Eq. 2.13** in **Section 2.1.3**, which decreases linearly from 2 to 0 as the iteration builds up. This linear changing pattern largely confines search performance, owing to the lack of distinction among search behaviours from different search stages, i.e. exploration and exploitation. To be specific, the search parameter a determines how far individual wolves could jump in reference to the leader wolves, through manipulating the magnitude of step size A, and $A| \leqslant |a|$ is always satisfied through the search process, as prescribed in **Eq. 2.12**. As discussed earlier, this indicates that a is the

determining factor that controls the search territory boundary. The linear decrease of a adopted in the original GWO results in the acute shrinkage of search territory during exploration as well as the lack of search attention on the promising vicinity of wolf leaders during exploitation. Nonlinear functions, such as trigonometric, exponential, and logarithmic-based functions, offer extraordinary flexibilities in composing curves with diverse geometric characteristics. Therefore, in this study I resort to the nonlinear functions to tailor the dynamic change of the boundary of search territory in the proposed GWO variant. A nonlinear exploration factor a' is proposed to overcome the above disadvantages of a in the original GWO model. The motivation is to achieve enhanced trade-offs between search exploration and exploitation through the bespoke adjustment of the boundary of search territory over the iterative process. The formulae related to the newly proposed exploration factor a' are presented in **Eqs. 5.1** and **5.2**.

$$a' = 2\left(\cos\left(\frac{(tanh\theta)^2+(\theta \sin\pi\theta)^n}{(tanh1)^2}\frac{\pi}{2}\right)\right)^2 \tag{5.1}$$

$$\theta = \frac{t}{Max\_iter} \tag{5.2}$$

where t and Max_iter represent the current and the maximum numbers of iterations respectively, whereas θ is the quotient of t divided by Max_iter. The coefficient n determines the descending slop of the search parameter a' over the search process. Based on trial-and-error, $n = 5$ is adopted in this research. **Figure 5-2** demonstrates the plot of the proposed nonlinear exploration rate a', against the linear decreasing a adopted in the original GWO algorithm as defined in **Eq. 2.13**.

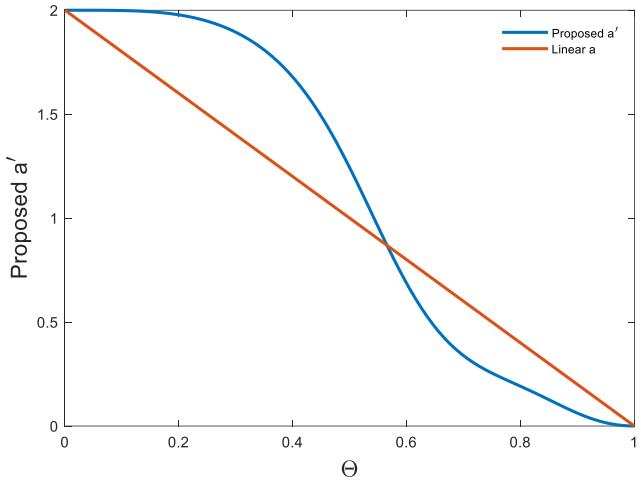

Figure 5-2 The proposed nonlinear a' vs. linear a in the original GWO

The proposed nonlinear search parameter a' is employed to replace a in the original GWO algorithm, to generate step size A' for the movement of each individual wolf with respect to each wolf leader, i.e. α, β, and δ, as shown in **Eqs. 5.3** and **5.6**. Except for the step size A',

the movement mechanism towards each wolf leader remains the same to that of the original GWO model, as defined in **Eqs. 2.4 - 2.9** in **Section 2.1.3**.

$$A' = (2rand - 1)a' \tag{5.3}$$

$$X_{ad1,j}^{t+1} = X_{\alpha,j}^{t} - A_1' D_{\alpha,j}^{t+1} \tag{5.4}$$

$$X_{ad2,j}^{t+1} = X_{\beta,j}^{t} - A_2' D_{\beta,j}^{t+1} \tag{5.5}$$

$$X_{ad3,j}^{t+1} = X_{\delta,j}^{t} - A_3' D_{\delta,j}^{t+1} \tag{5.6}$$

where A' is the step size yielded by the proposed search parameter a'. Three step sizes, i.e. A_1', A_2', and A_3' are yielded for the movements towards three dominant wolves, i.e. α, β, and δ, respectively, for each individual wolf under position updating. In addition, X_{ad1}, X_{ad2}, and X_{ad3} denote the position adjustments with respect to α, β, and δ, respectively. $D_{\alpha,j}^{t+1}$, $D_{\beta,j}^{t+1}$ and $D_{\delta,j}^{t+1}$ are obtained using **Eqs. 2.4 - 2.6** in **Section 2.1.3**.

As shown in **Figure 5-2**, in comparison with the linear adjustment of a adopted by the original GWO algorithm, the proposed nonlinear exploration factor a' decreases with gentle gradients both at the beginning and end of the search course. As a result, the search boundary can be upheld at a high level with minor contraction and the search territory is significantly expanded during the exploration stage, whereas the local detections become more concentrated on the vicinity of promising solutions owing to the confined search boundary during the exploitation stage. Such advantages become strengthened when deploying a' to the movement of each individual wolf towards each of the three dominant wolves, i.e. α, β, and δ. Therefore, the search diversification is significantly enhanced while the search intensification is greatly intensified. As such, a superior transition from exploration to exploitation can be achieved by the proposed nonlinear exploration rate a', in comparison with the linear decreasing parameter a in the original GWO algorithm.

5.1.1.2 Chaotic dominance of wolf leaders

In the original GWO algorithm, although motivated by the social hierarchy observed among grey wolves, the leadership within the wolf pack is evenly divided and assumed by three dominant leaders, which remains static over the whole iteration course, regardless of the difference of the fitness scores of the wolf leaders. This lack of prioritizing operators among dominant wolf leaders results in a slow convergence rate, therefore compromising search efficiency [116, 305]. Motivated by diverse strategies proposed to establish dynamic and strict social leadership hierarchies in GWO, e.g. dedicated learning curves [119] and the assignment of random weights according to fitness scores [120], we employ the sinusoidal chaotic map to generate weight factors prioritizing the dominance of the best leader wolf α, as shown in **Eq. 5.7**, whereas the leadership factors for wolves β and δ are determined

subsequently in accordance with that of wolf α, as indicated in **Eq. 5.8**. The position updating mechanism with the updated dominance factors is presented in **Eq. 5.9**.

$$w_{t+1} = 2.3\, w_t^2 \cdot \sin(\pi w_t) \tag{5.7}$$

$$w'_{t+1} = 0.5(1 - w_{t+1}) \tag{5.8}$$

$$X_i^{t+1} = w_{t+1} X_{ad1}^{t+1} + w'_{t+1} X_{ad2}^{t+1} + w'_{t+1} X_{ad3}^{t+1} \tag{5.9}$$

where w_t and w_{t+1} represent the weight coefficients of the position adjustment X_{ad1} with respect to wolf α in the t-th and $(t+1)$-th iterations, respectively, while w'_{t+1} represents the weight coefficient for both position adjustments X_{ad2} and X_{ad3} with respect to wolves β and δ, respectively in the $(t+1)$-th iteration.

The proposed chaotic dominance scheme is capable of achieving better trade-offs between reinforcing the leadership of the best wolf solution (single-leader guided search) and diversifying the guiding signals (multi-leader guided search). As illustrated in **Figure 5-3**, the employed sinusoidal chaotic map produces dynamic values roughly within the range of [0.5, 0.9], which are adopted to represent the irregular characteristic of the leadership of the most dominating wolf α. The proposed leadership assignment scheme simulates a centralized wolf regime in which wolf α is bestowed with the highest authority and the leadership assumed by wolf α is greater than the combined power of wolves β and δ. As a result, the search procedure becomes more focused on promising territories represented by wolf α, mitigating the negative impacts of malignant distractions and futile movements caused by less promising leader signals. As such, the convergence speed becomes faster and the search efficiency improves.

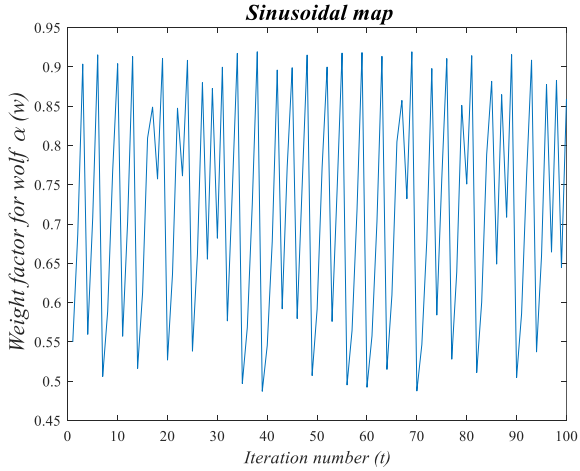

Figure 5-3 The sinusoidal chaotic map used for generating the leadership factors of the most dominating wolf α

In addition, the chaotic map oriented dynamic dominance of wolf α increases search diversity by diversifying guiding signals, in comparison with the static and equal leadership

operation employed in the original GWO method. Specifically, as the weight coefficients fluctuate periodically between [0.5, 0.9], the dominance level of wolf α varies accordingly over the whole iterative process. The rivalry from wolves β and δ intensifies and becomes equivalent to that of wolf α when the weight coefficient produced by the chaotic map is equivalent to 0.5. As a result, the distraction imposed by wolves β and δ can effectively dilute the dominance of wolf α and divert the undergoing search trajectory to an unexploited new region. As shown in **Figure 5-3**, such drastic changes in leadership assumptions occur more frequently in the middle of the search process, i.e. between 30-60 iterations, which can effectively prevent the wolf pack from being trapped in local optima and reduce the likelihood of premature stagnation.

Moreover, the employed dynamic rivalry of the dominance among three leading wolves assimilates merits from both multi-leader and single-leader guided search procedures. Specifically, the significant dominance of wolf α, induced by the relatively larger weight coefficient w_{t+1}as indicated in **Eq. 5.7**, enables GWO to emulate the efficiency of the single best-leader guided search, whereas the equivalent rivalry from wolves β and δ, induced by comparatively smaller weight coefficient w'_{t+1} as defined in **Eq. 5.8**, allows the proposed model to leverage the strength of global exploration from the multi-leader guided search. In contrast, existing studies [119, 120] in reinforcing the leadership of wolf α generally fail to consider the influence of the confrontation from the perspective of the combined power of wolves β and δ. In addition, the lack of variance in leadership contention in those studies also increases the risk of local stagnations.

Overall, the proposed chaotic leadership assignment among the elite wolf circle in conjunction with the nonlinear adjustment of search boundary enables the modified GWO algorithm to achieve more efficient trade-offs between exploration and exploitation from two levels, i.e. the independent movement with respect to each wolf leader, and the aggregation of the leaders.

5.1.1.3 A dedicated leader exploitation scheme

The constant adherence to the guidance of three best wolves through the whole iterative process propels the search diversity of GWO. On the other hand, it also constrains the capability of concentrating on local detection around the identified best solution. We subsequently propose a damped function with decremental amplitudes to produce a variety of step sizes for the local exploitation and fine-tuning around wolf α at the final search stage ($t \geq 80$), as well as to guarantee the convergence of the wolf population. The damped function is illustrated in **Eq. 5.10** whereas the position updating equation based on generated step size is presented in **Eq. 5.12**.

$$\lambda = f \cdot e^{3r^2/2} \cos(\pi r) \sin(\pi r) \tag{5.10}$$

$$f = 1 - 0.05(t - 80) \tag{5.11}$$

$$X_{i,j}^{t+1} = X_{\alpha,j}^{t} - \lambda \left| X_{\alpha,j}^{t} - X_{i,j}^{t} \right| \tag{5.12}$$

where λ and f denote the yielded step size and the amplitude of the damped function respectively, while $X_{i,j}^{t+1}$ represents the element of wolf i at j-th dimension in $(t+1)$-th iteration. Besides the above, r is a random value in the range of [-1, 1], and X_α denotes the position of the best wolf leader α.

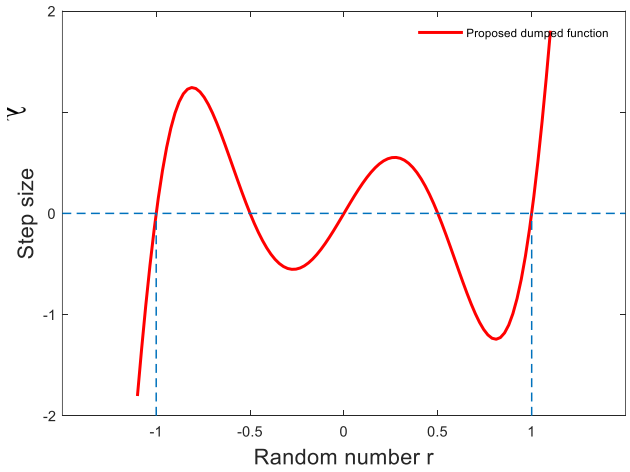

Figure 5-4 The proposed damped function in **Eq. 5.10** when $f = 1$

As shown in **Figure 5-4**, the proposed formula is an odd function with damped oscillations along the x axis. When x is in the clamp between [-1, 1], the range of the highest crest and trough is [-1.3, 1.3], whereas that of the second highest crest and trough is [-0.6, 0.6]. As a result, wolf solutions are capable of conducting large jumps from all directions radiated from wolf α when $|r| > 0.5$, as well as performing granular movements when $|r| < 0.5$. Moreover, the symmetry of the function with respect to the coordinate origin induces an even distribution of generated steps in both the positive and negative realms. This enables the simulation of individual wolves to approach wolf α as well as distance from it with an equal probability. Furthermore, a decremental amplitude f is applied to gradually flatten the fluctuation and shrink the search radius as the iteration builds up. The intensification of the detection around the best solution is therefore strengthened through this dedicated local exploitation scheme.

As depicted in **Figure 5-5**, we further compare the above proposed formula in **Eq. 5.10** with the damped function employed in the spiral search mechanism in MFO [63] defined in **Eq. 5.13**.

$$y = e^{br} \cos(2\pi r) \tag{5.13}$$

where b is a constant and set as 1 while r is a random value in the range of [-1, 1]. Besides that, y is the yielded step size.

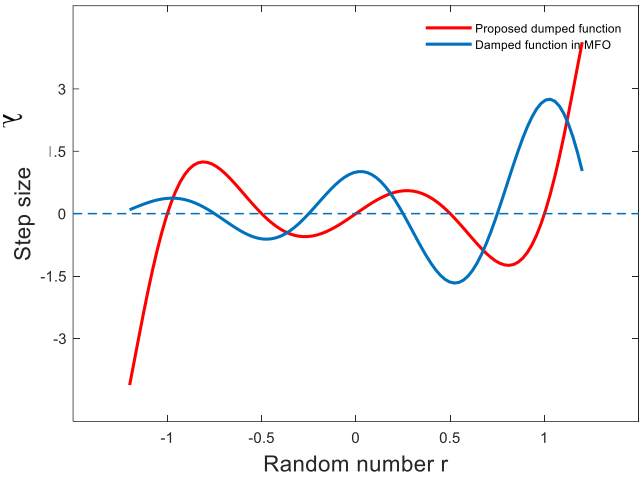

Figure 5-5 The comparison between proposed damped function and the damped function applied in MFO

Firstly, the damped function in MFO does not possess any symmetrical properties. Secondly, it does not involve any dynamic granular changes in its search scale. As a result, the variance of the oscillated scales and the imbalance of the probabilities between generating identical (positive values) and reverse (negative values) search directions can result in obstinate search bias and incomplete coverage of the search territory, which could lead to further degradation of search efficiency and local intensification. In contrast, the proposed strategy is able to effectively accelerate convergence as well as intensify the local exploitation around the identified best leader owing to the increased diversity in terms of scales and symmetric directions of search steps.

5.1.1.4 Wolf leaders enhancement using Lévy flight

The quality of the dominant leaders is crucial to the performance of GWO owing to the excessive adoption of multiple leaders in the search process. We therefore implement a Lévy flight random walk as defined in **Eq. 5.14** to further improve the quality of three leading wolves successively.

$$X'_{L,j} = \begin{cases} X_{L,j} + \xi \cdot X_{\sigma,j} & if\ rand > 0.5 \\ X_{L,j} & otherwise \end{cases} \tag{5.14}$$

where X_L and X'_L represent the positions of each wolf leader before and after performing a random walk in Lévy distribution, respectively, whereas X_σ represents a distinctive second wolf leader selected among α, β and δ as a distraction signal. Also, ξ denotes the step size generated from the Lévy distribution [306].

The Lévy jumps are only implemented on dimensions where determinants are higher than 0.5. Only the mutated offspring solutions with improved fitness scores are retained. For each leader undergoing mutation, a second distinctive dominant leader is randomly selected and employed to introduce distinguishing factors. This distraction from a different leading wolf can effectively prevent the vanishing of the jump momentum resulted from the stagnation at local optima located next to the coordinate origin, i.e. $X_{L,j} = 0$. In short, this leader enhancement operation based on Lévy flight enables the wolf pack to jump out of local optima traps and increases the likelihood of attaining global optimality.

Algorithm 5-1 The proposed GWO model

1 Start

2 Initialize a grey wolf population

3 Evaluate each individual using the objective function $f(x)$ and identify three dominant wolves with the best fitness scores, denoted as $X_\alpha, X_\beta,$ and X_δ, respectively

4 **While** ($t < Max\_iter$)

5 {

6 Update the exploration rate a' by **Eqs. 5.1** and **5.2**

7 Generate dominance factors for three wolf leaders, i.e. w for wolf α and w' for wolves β and δ, using **Eqs. 5.7** and **5.8**

8 **For** (each leader) **do**

9 {

10 Conduct leader enhancement using Lévy flight as defined in **Eq. 5.14**

11 } **End For**

12 **If** ($t < 0.8 \times Max\_iter$)

13 {

14 **For** (each wolf i in the population) **do**

15 {

16 Generate step size A'using **Eq. 5.3**

17 Calculate distance measures, D_α, D_β, and D_δ, by **Eqs. 2.4 - 2.6**

18 Update the position with respect to $X_\alpha, X_\beta, and\ X_\delta$, by **Eqs. 5.4 - 5.6, 5.9**

19 } **End For**

20 **Else** $t \geq 0.8 \times Max\_iter$

21 **For** (each wolf i in the population) **do**

22 {

23 Conduct local exploitation around the best leader X_α with dynamic steps by **Eqs. 5.10** and **5.12**

24	} **End For**
25	} **End If**
26	**For** (each wolf i in the population) **do**
27	{
28	Calculate the fitness score of i
29	Update three dominant leaders $X_\alpha, X_\beta, and\ X_\delta$
30	} **End For**
31	} **End While**
32	Output the most optimal solution X_α
33	**End**

The pseudo-code of the proposed GWO variant is provided in **Algorithm 5-1**. Overall, the proposed GWO variant employs four strategies to enhance search diversity while accelerating convergence, i.e. a nonlinear adjustment of search boundary, a chaotic dominance rivalry among leading wolves, a dynamic leader exploitation operation using an enhanced spiral search procedure, as well as a Lévy flight mutation operation based on the dominant wolves. As such, these proposed strategies enhance the original GWO algorithm from three perspectives, i.e. adjusting the search parameters, modifying position updating rules and search courses, as well as enhancing promising leader signals. These strategies work cooperatively to mitigate premature convergence, improve the transition from exploration to exploitation, and overcome limitations of the original GWO method.

5.1.2 The proposed CNN-LSTM architecture

In this research, we propose a skeleton architecture of CNN-LSTM, upon which the tailored configuration of the hyperparameters is specified according to the recommendation of the proposed GWO variant with respect to the investigated time series task. The topology of the proposed CNN-LSTM architecture is outlined in **Figure 5-6**.

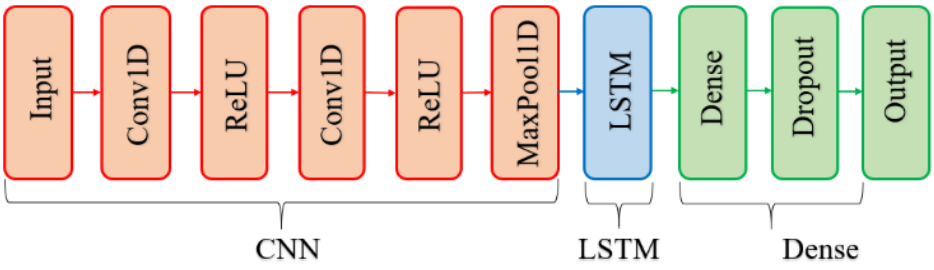

Figure 5-6 The topology of the proposed CNN-LSTM architecture

It consists of three core types of layers, i.e. the convolutional layer, the LSTM recurrent layer, and the dense layer. The input data sequence is firstly used as input to the two consecutive convolutional layers for feature extraction. Through the convolutional

operations of filters with different properties, the nonlinear activation of neurons, as well as the abstract representation of max pooling, the low-level features and distinctions among variables under the context of temporal effects are therefore acquired. The obtained feature map is then passed on to the LSTM layer where the complex dependencies are thoroughly learned by the examination of the three effective gates in LSTM, i.e. the forget, input, and output gates. Specifically, the irrelevant or redundant information from previous cell states is removed by the forget gate. The effective new information from the input sequence is stored by the input gate. Moreover, the signals from the cell state are filtered and then passed on to the next state by the output gate. Furthermore, the processed temporal information is then used as input to dense layers to undergo nonlinear transformations. Finally, the obtained information is projected to the output space and the prediction results are produced. Overall, the proposed CNN-LSTM skeleton architecture is adopted as the foundation for evaluating the time series problems in test scenarios.

5.1.3 The proposed GWO-based evolving CNN-LSTM network

The identification of the optimal configurations of hyperparameters and architectures is crucial to the performance and efficiency of deep neural networks in practice. Such configuring and searching processes are particularly cumbersome for CNN-LSTM networks owing to the increased amount of hyperparameters induced by the hybridisation of CNN and LSTM, as well as the profound interactive effects among them, in comparison with monotonous deep learning models. In this study, we employ the above proposed enhanced GWO model for the automatic optimal configuration identification for the CNN-LSTM architecture, to undertake time series prediction tasks.

To be specific, the proposed GWO variant is employed to automatically search for the optimal hyperparameters and topologies of the CNN-LSTM model, by optimizing the following learning and network parameters, i.e. the learning rate, the dropout rate, the number and size of filters in two convolutional layers, the size of the pooling layer, as well as the numbers of hidden nodes in LSTM recurrent layer and the final dense layer, for each time series problem respectively. The search range for each optimized parameter is presented in **Table 5-1**. The explored hyperparameters include key factors critical to the representational capacity of the CNN-LSTM network, e.g. the number of hidden nodes in the LSTM layer, as well as those responsible for the learning efficiency and training property, e.g. the learning and dropout rates. As such, confounding effects and impacts of various hyperparameters can be thoroughly explored through the evolving process of the proposed GWO variant. The CNN-LSTM model with the identified optimized configuration is then applied to tackle time series prediction and classification.

Table 5-1 The search range of the hyperparameters

Optimized component	Hyperparameter	Range
Conv	No. of filter in 1st layer	$[2^0, 2^{10}]$
	filter size in 1st layer	[1, 5]
	No. of filter in 2nd layer	$[2^0, 2^{10}]$
	filter size in 2nd layer	[1, 5]
Pooling	pooling size	[2, 5]
LSTM	No. of hidden nodes	[10, 500]
Dense	No. of nodes	[10, 200]
Learning configuration	learning rate	$[10^{-5}, 10^{-1}]$
	dropout rate	[0, 0.6]

The optimal hyperparameter search of the deep network is performed as follows. Firstly, the population of the proposed GWO algorithm is randomly initialised, with each individual representing a possible configuration for the optimized CNN-LSTM model. The recommended CNN-LSTM model with the specific structure and parameter settings represented by each wolf is then established and trained with the training set. The fitness scores, i.e. the error rate for classification problems or the root mean square error (RMSE) for regression problems, are calculated based on the validation set. The solutions with top three fitness scores are identified as the dominant wolves, hence employed to guide the whole wolf pack to search for the global optimality by following the proposed strategies prescribed in the modified GWO model. The optimal configuration obtained by the wolf population is then adopted to yield the final devised CNN-LSTM model. It is then evaluated using the unseen test data set. In this study, several time series problems are employed to examine the effectiveness and robustness of the proposed GWO-based CNN-LSTM network.

5.2 Evaluation and discussion

In this section, the effectiveness of the proposed evolving CNN-LSTM model is evaluated on two time series prediction problems, i.e. building energy consumption forecast and PM2.5 concentration prediction, as well as one time series classification problem, i.e. human activity recognition. The performance of the proposed GWO variant in identifying the optimal CNN-LSTM configurations is compared against those of four classical search methods, i.e. GWO [62], PSO [307], GSA [67], and FPA [66], as well as three advanced GWO and PSO variants, prLeGWO [119], FuzzyGWO [308], and CSO [277]. The parameter settings for above baseline models are provided in **Table 5-2**. The identical settings are employed for each experiment to ensure a fair comparison, i.e. the maximum number of function evaluations = population size (30) × the maximum number of iterations

(100). A CNN-LSTM model with default parameter settings, i.e. filter number in the 1st Conv layer = 32, filter size in the 1st Conv layer = 2, filter number in the 2nd Conv layer = 32, filter size in the 2nd Conv layer = 2, pooling size = 2, number of node in LSTM layer = 300, number of node in dense layer = 100, learning rate = 0.001, and dropout rate = 0.2, is also employed as one of the baselines for performance comparison. Moreover, we conduct ten independent runs for each experiment to mitigate the impact of random factors on the evaluation. The experimental details of the employed time series prediction problems are presented below.

Table 5-2 Parameter settings of search methods

Methods	Parameter settings
GWO [62]	step size $A = (2 \times rand - 1) \times a$, where a linearly decreases from 2 to 0, $rand \in (0, 1)$. search parameter $C = 2 \times rand$.
PSO [307]	cognitive component c_1 = 1.4962, social component c_2 = 1.4962, inertia weight w = 0.7298.
GSA [67]	initial gravitational constant G_0 = 100, search parameter α = 20.
FPA [66]	switch probability = 0.8, step size L for global pollination drawn from a Levy flight distribution, step size ϵ for local pollination drawn from a uniform distribution within [0, 1].
CSO [277]	r_1, r_2, and r_3 are search parameters randomly drawn from a uniform distribution within [0, 1].
PrLeGWO [119]	initial weights of three dominant wolves w_α = 1/3, w_β = 1/3, and w_δ = 1/3, weights of three dominant wolves at the end of the iteration w_α = 0.8, w_β = 0.1, and w_δ = 0.1.
FuzzyGWO [308]	A Mamdani fuzzy system to generate weights for three dominant wolves.
Prop. GWO	A nonlinear exploration factor: $a' = 2 \times \left(\cos\left(\frac{(tanh\theta)^2 + (\theta \sin \pi\theta)^5}{(tanh1)^2} \times \frac{\pi}{2}\right)\right)^2$, where θ is the quotient of the current iteration number divided by the maximum iteration number.

5.2.1 Energy consumption forecast

5.2.1.1 Data set

The time series prediction is first introduced using the energy consumption scenario. Specifically, the individual household electricity consumption data set from UCI machine learning repository [255] is employed to evaluate the effectiveness of the proposed evolving CNN-LSTM model on tackling the energy forecast task. The data set contains 2,075,259

measurements with nine attributes collected in an interval of one minute, from a house located in Sceaux between December 2006 and November 2010.

5.2.1.2 Experimental settings

According to the difference of the time interval, energy forecasting models are generally classified into three categories, i.e. short-term, medium-term, and long-term energy forecast [309]. In this research, a multi-input and multi-output short-term energy forecasting model is developed. Specifically, the amount of electricity consumption for the next week is predicted using the historical data from the previous two weeks, in order to capture weekly periodicity and irregularity of the energy consumption. The proposed weekly energy forecasting model can be used to inform future energy expenditures of the household, and to facilitate the demand side management. The original observations with an interval of one minute are transformed into daily energy consumption data for the weekly prediction of energy consumption. The data from the first two years are employed for training, the data from the subsequent one year for validation, and the data from the final year for testing.

For the prediction of energy consumption, eight of the total hyperparameters listed in **Table 5-1** except for the pooling size are optimized. The pooling size is set to 2, owing to the comparatively small input vector of the sequential data of this energy consumption scenario, i.e. 14×9, where 14 and 9 represent time steps and the feature size, respectively. The optimal CNN-LSTM configuration is identified based on the training and validation sets. The batch size is set as 128 whereas a total of 20 epochs are used in the training stage to balance between performance and computational cost. In addition, the Adam optimizer is applied in the training process while the RMSE is adopted as the fitness score to evaluate the performance of CNN-LSTM. The devised CNN-LSTM model is retrained on the combined set of training and validation samples for 100 epochs. Finally, the fully trained CNN-LSTM model is employed to forecast energy consumption on the unseen test set.

5.2.1.3 Results and discussion

Two performance indicators are employed to evaluate the effectiveness of the proposed evolving CNN-LSTM method in forecasting energy consumption, i.e. RMSE and the mean absolute error (MAE). The results of RMSE and MAE over ten independent runs are presented in **Tables 5-3 – 5-4**, respectively.

Table 5-3 The RMSE results over 10 independent runs

Run	CNN-LSTM	GWO	PSO	GSA	FPA	CSO	prLeGWO	FuzzyGWO	Prop.GWO
1	401.7	403.3	392.4	418.0	368.0	396.9	388.2	384.4	371.2
2	409.6	413.4	395.6	425.0	403.9	376.0	385.1	433.4	383.0

3	451.1	415.3	410.6	401.7	541.6	483.2	418.2	413.1	382.5
4	421.0	441.3	387.7	383.3	400.9	406.4	381.8	369.6	365.0
5	422.1	412.0	399.4	408.2	437.4	506.7	404.1	398.9	376.9
6	439.2	423.3	381.5	424.6	421.9	381.2	393.1	375.9	386.5
7	424.1	396.7	397.0	426.2	391.4	410.6	387.2	378.1	376.7
8	419.6	383.5	384.8	390.0	382.4	394.7	383.4	400.1	380.3
9	418.8	383.6	400.2	438.8	391.9	406.1	377.4	407.8	382.6
10	415.6	483.4	395.7	429.6	381.8	387.6	379.6	401.0	367.1
Avg.	422.3	415.6	394.5	414.5	412.1	414.9	389.8	396.2	377.2

Table 5-4 The MAE results over 10 independent runs

Run	CNN-LSTM	GWO	PSO	GSA	FPA	CSO	prLeGWO	FuzzyGWO	Prop.GWO
1	310.5	303.0	299.6	314.5	280.1	313.3	294.2	291.8	287.2
2	316.6	321.4	305.9	320.9	313.0	287.3	294.7	340.7	292.5
3	345.2	318.8	322.5	307.3	419.4	367.8	328.2	320.0	292.7
4	319.2	346.5	286.6	294.3	306.3	309.2	291.3	286.5	277.7
5	326.7	321.7	305.9	301.9	335.8	365.0	307.4	306.6	291.5
6	342.9	321.0	286.5	310.7	326.2	294.8	302.4	289.9	299.4
7	324.4	307.1	300.0	324.4	297.9	314.1	286.6	291.1	296.4
8	322.4	284.0	298.4	301.1	298.1	303.5	289.1	314.0	290.1
9	324.6	297.6	313.8	328.9	302.0	310.4	294.6	316.6	294.7
10	312.8	379.4	302.6	326.6	292.4	298.9	291.8	316.5	283.1
Avg.	324.5	320.1	302.2	313.1	317.1	316.4	298.0	307.4	290.5

The optimized CNN-LSTM networks identified by the proposed GWO variant achieve the lowest RMSE and MAE results and demonstrate significant advantages in comparison with those yielded by the four classical search methods and the advanced prLeGWO, FuzzyGWO, and CSO models, as well as the CNN-LSTM network with the default setting. Specifically, as shown in **Table 5-3**, the RMSE results produced by the proposed GWO-based evolving CNN-LSTM model are more reliable, lying within the range of [360, 390], whereas the majority RMSE results produced by baselines methods are larger than 390, demonstrating greater variances. As shown in **Table 5-4**, the significant superiorities of the proposed GWO model can also be observed from the MAE results. This indicates that the optimized CNN-LSTM configurations identified by the proposed GWO variant are capable of identifying spatial variations among time series variables and extracting irregular patterns in temporal information embedded in the energy usage data, effectively.

Table 5-5 The mean configurations of the identified CNN-LSTM networks over 10 runs

Conf.	GWO	PSO	GSA	FPA	CSO	prLeGWO	FuzzyGWO	Prop.GWO
No. 1st	82.8	118.8	63.2	110.6	186.6	12.4	105	217.2
No. 2st	4.6	17	8.4	159	22.6	11.8	5.8	5.4
S. 1st	3.6	3.7	2.9	3.1	2.2	2.0	1.9	1.6
S. 2st	1.8	1.8	2.9	2.1	2.7	1.3	1.3	1.2
LSTM	327.3	320.6	261	246	321.8	122.9	129.5	284.1
Dense	70.5	110.3	96.6	74.9	68.7	55.5	16.8	35.4
DR	0.246	0.270	0.336	0.312	0.341	0.193	0.041	0.170
LR	0.024	0.042	0.051	0.034	0.051	0.023	0.003	0.021

This advantage of the proposed GWO method is further analysed by examining the distinctive characteristics of its identified CNN-LSTM configurations, as opposed to those yielded by the baseline models. The mean hyperparameters of the optimized configurations of CNN-LSTM yielded by the GWO variant over a set of 10 runs are presented in **Table 5-5**. In general, the CNN-LSTM structures identified by the proposed GWO model demonstrate two main distinctive characteristics, i.e. a higher number of filters in the first convolutional layer and a moderate setting in terms of number of nodes in the recurrent and dense layers, in comparison with those identified by the baseline models. Specifically, the optimized CNN-LSTM structures are capable of extracting energy usage features more effectively owing to the higher number of filters in the first convolutional layer, i.e. 217.2. These filters in the convolutional layer are able to reduce data noise and remove irrelevant variations among time series variables while preserving the essential temporal variance. Besides the above, the long-term dependencies can be acquired efficiently without overfitting owing to the optimized and more balanced settings of the hidden nodes in the LSTM and dense layers, i.e. 284.1 and 35.4, respectively. As such, the devised CNN-LSTM networks are capable of achieving more efficient trade-offs between the model representational capacity and the avoidance of overfitting.

In contrast, those network configurations yielded by baseline methods as well as the default CNN-LSTM model generally achieve less advanced learning capacities in incorporating spatial and temporal information with respect to the energy usage patterns, owing to the lack of convolutional operations as well as the sub-optimal recurrent network representations. This indicates the deficiency of baseline search methods in exploring sophisticated interactions among hyperparameters in CNN-LSTM. In other words, baseline models are more prone to local optima traps, therefore yielding less advanced CNN-LSTM configurations in addressing complicated factors, e.g. fluctuation and volatility, in energy

forecasting tasks. In short, in comparison with the baseline methods, the proposed diverse search strategies, e.g. the nonlinear exploration rate adjustment, the chaotic leadership rivalries, as well as Lévy random jumps, account for the superior performance of the proposed evolving CNN-LSTM networks.

5.2.2 PM2.5 concentration prediction

5.2.2.1 Data set

To further indicate model efficiency, we also employ the UCI Beijing air quality data set [310] for PM2.5 concentration prediction using the devised evolving CNN-LSTM networks. This data set includes hourly measurements of four types of air pollutants, i.e. SO2, NO2, CO, and O3, as well as five meteorological parameters, i.e. temperature, pressure, dew point temperature, amount of precipitation, and wind speed, over a four-year period of time from March 1st, 2013 to February 28th, 2017. The reliable prediction of PM2.5 concentrations requires profound interpretations of the changing patterns of air pollutants under various temporal contexts, which pose great challenges to the yielded devised networks.

5.2.2.2 Experimental settings

Similar to the framework in the energy forecasting task, a multi-input and multi-output time series model is established to predict the PM2.5 concentrations in the air in Beijing for a week in advance, based on historical data from the previous two weeks. The hourly recordings are transformed into daily measurements to better understand weekly periodicity of input variables as well as to make weekly predictions of PM2.5 concentration. The vector of the input sequence is 14×9, where 14 and 9 represent time steps and the feature size, respectively. The experimental settings employed in the PM2.5 concentration prediction are the same as those employed in the previous energy consumption forecasting, owing to the identical characteristics in both problems. Besides that, the data from the first and second years are used for training, whereas the data from the third and last years are used for validation and testing, respectively.

5.2.2.3 Results and discussion

As shown in **Tables 5-6 – 5-7**, the optimized CNN-LSTM networks identified by the proposed GWO algorithm yield more robust and reliable predictions for weekly PM2.5 concentration in comparison with those of the seven baseline methods and the default CNN-LSTM network. In specific, our optimized CNN-LSTM networks achieve the smallest average results of RMSE and MAE, i.e. 62.2 and 40.8, over ten independent runs, whereas the baseline methods in general produce less favourable results with high variances and inconsistencies across ten different runs. In particular, the RMSE measures are reduced by 6.2%, 13.1%, 9.1%, and 15.1%, by the devised CNN-LSTM networks, in comparison with

those networks yielded by GWO, prLeGWO, FuzzyGWO, as well as the default CNN-LSTM model, respectively. The significant performance improvements of the devised CNN-LSTM networks can be further observed from the MAE results. The superiority in the evaluation performance indicates the effectiveness of the proposed evolving time series model in extracting effective features and recognizing complex temporal variations embedded in air pollution data as well as interpreting reflective influences of dynamic meteorological conditions.

Table 5-6 The RMSE results over 10 independent runs

Run	CNN-LSTM	GWO	PSO	GSA	FPA	CSO	prLeGWO	FuzzyGWO	Prop.GWO
1	67.3	64.7	64.3	70.0	64.2	92.1	82.4	68.3	62.6
2	70.8	65.2	65.6	62.6	67.7	64.7	61.0	70.1	63.5
3	68.6	62.0	63.9	65.8	78.9	66.1	65.1	60.5	62.8
4	74.6	65.9	64.9	64.8	58.1	63.9	64.7	63.7	65.6
5	75.4	70.6	64.9	73.5	65.9	67.8	66.1	67.3	59.9
6	74.2	65.8	63.8	69.1	73.2	64.7	117.3	72.4	62.3
7	69.3	67.7	61.0	63.6	76.7	60.0	62.3	72.7	61.6
8	73.7	63.2	64.7	68.7	70.7	68.3	61.6	67.6	59.9
9	87.5	73.0	61.7	63.4	69.2	68.1	71.7	69.4	62.5
10	71.6	65.4	68.1	64.0	77.2	69.0	63.4	72.4	61.3
Avg.	73.3	66.3	64.3	66.6	70.2	68.5	71.6	68.4	62.2

Table 5-7 The MAE results over 10 independent runs

Run	CNN-LSTM	GWO	PSO	GSA	FPA	CSO	prLeGWO	FuzzyGWO	Prop.GWO
1	44.8	42.6	41.4	46.2	40.2	54.7	55.2	45.1	39.6
2	47.6	41.6	43.9	42.2	44.6	45.7	41.9	49.1	42.5
3	48.6	41.4	41.6	43.9	52.3	43.5	42.9	42.9	41.1
4	49.8	43.1	42.0	43.0	40.8	40.6	42.4	40.8	43.6
5	51.1	44.7	42.6	48.5	43.5	44.7	42.4	44.2	40.3
6	49.9	43.0	41.8	43.6	48.2	43.0	70.3	52.5	40.4
7	45.1	41.6	39.4	42.4	53.0	39.7	42.8	46.5	39.8
8	48.5	42.0	44.2	44.2	46.8	44.9	41.4	46.3	39.4
9	54.3	46.7	40.4	41.7	46.0	45.6	47.3	46.3	40.9
10	48.9	42.8	42.6	42.1	53.7	45.7	42.5	52.6	40.3
Avg.	48.9	42.9	42.0	43.8	46.9	44.8	46.9	46.6	40.8

Table 5-8 The mean configurations of the identified CNN-LSTM networks over 10 runs

Conf.	GWO	PSO	GSA	FPA	CSO	prLeGWO	FuzzyGWO	Prop.GWO
No. 1st	35.2	165.6	198.4	39.6	72.8	3.8	4.6	132.8
No. 2st	2	2	26.8	118.4	14.8	2.6	2.6	2.4
S. 1st	1	1.7	2.9	1.8	2.2	2.5	1.4	1.2
S. 2st	1.3	1.8	3.1	2.6	1.9	2.4	1.4	1.3
LSTM	85.2	237.2	271.2	200.5	223.7	124.9	81.2	126.2
Dense	60	53.9	91.3	95.3	76.8	85.3	15.9	51.4
DR	0.305	0.352	0.321	0.219	0.370	0.018	0.004	0.313
LR	0.034	0.048	0.049	0.028	0.058	0.007	0.001	0.046

Moreover, the mean hyperparameters of the identified optimal structures for PM2.5 concentration prediction over ten independent runs are presented in **Table 5-8**. The main characteristics of the effective CNN-LSTM configurations in the PM2.5 prediction are similar to those demonstrated in energy forecasting. The optimized CNN-LSTM structures produced by the proposed GWO variant possess a relatively larger number of filters in the first convolutional layer, i.e. 132.8, while maintaining smaller amounts of nodes in both the LSTM and dense layers, i.e. 126.2 and 51.4 respectively. Such compositions enable the efficient extraction of most important features among meteorological variables as well as air pollutants in the convolutional layers, while endowing the optimized CNN-LSTM networks with sufficient representational capacities to effectively capture various dependencies in the LSTM and dense layers while avoiding overfitting.

To be specific, the employed air pollution data set not only contains important factors in relation to generation and dispersion of PM2.5, e.g. concentration of SO2 and NO2, and wind speed, but also disturbing factors with various confounding effects, e.g. concentration of CO and O3. Therefore, the prediction of PM2.5 is challenging owing to various complexions. As such, the proper feature extraction capability is required to identify effective attributes essential to the complex formation mechanism of PM2.5, as well as sophisticated aerodynamic effects on its dilution. The RMSE and MAE results indicate that our optimized CNN-LSTM networks are able to resolve the above challenging factors more effectively and demonstrate greater resilience in handling temporal variances and interactions among variables. In other words, the identified filter structures in convolutional layers are capable of generating informative feature maps, which can both uncover the indirect impacts of various pollutants permeated in the air, as well as the direct impacts of weather conditions, on the concentration of PM2.5. Meanwhile, the identified optimized

configurations of the LSTM and dense layers are able to better comprehend and capture the long-term dependencies among the input data sequences. As such, the devised CNN-LSTM structures identified by the proposed GWO variant are proven to be superior in undertaking complex PM2.5 concentration prediction tasks.

5.2.3 Human activity recognition

5.2.3.1 Data set

We have also employed a time series classification task using the UCI human activity recognition (HAR) data set [311] for model evaluation. The data set was collected from 30 volunteers performing six types of daily living activities, i.e. standing, sitting, laying down, walking, walking downstairs and upstairs, while carrying the waist-mounted smartphones embedded with inertial sensors. Three types of signals, including total acceleration, body acceleration, as well as body gyroscope, were recorded in a sampling rate of 50Hz. These sensor signals were pre-processed using noise filters and sampled in the sliding window of 2.56 sec, i.e. 128 readings, with a 50% overlap. The dimension of the input sequence is 128×9, in which 128 and 9 are the number of readings and the number of features respectively. The total sample sizes in the training and test data sets are 7,352 and 2,947 respectively.

5.2.3.2 Experimental settings

In the HAR task, the nine hyperparameters in relation to network capacities and learning properties listed in **Table 5-1** are optimized. The training process is divided into two main stages. Firstly, the optimal configuration of CNN-LSTM is identified by the proposed GWO variant using a smaller proportion of the training data, to reduce computational cost. Specifically, the first 3000 sequences in the training data set are used for training and the subsequent 1500 sequences for validation, for the search of the optimal network configuration. In the training process, the Adam optimizer is adopted, while the categorical cross-entropy is applied as the loss function. Also, the batch size and epoch number are set as 256 and 20, respectively. Besides the above, the error rate is employed as the fitness score to be minimized during the evolving process. Subsequently, the recommended CNN-LSTM model with the identified optimal structure is retrained for 100 epochs using the whole training data set of 7,352 samples. The obtained CNN-LSTM model is then used to classify human activities using the unseen test data set with 2,947 samples.

5.2.3.3 Results and discussion

A total of four performance indicators are employed to evaluate the effectiveness of the optimized CNN-LSTM networks in distinguishing and recognizing the recorded human

activities, i.e. classification accuracy, F-score, precision, and recall. The results over ten independent runs are presented in **Tables 5-9** - **5-12**, respectively.

Table 5-9 The results of classification accuracy over 10 independent runs

Run	CNN-LSTM	GWO	PSO	GSA	FPA	CSO	prLeGWO	FuzzyGWO	Prop.GWO
1	0.879	0.886	0.907	0.911	0.881	0.893	0.863	0.883	0.928
2	0.882	0.889	0.900	0.904	0.815	0.907	0.886	0.928	0.929
3	0.859	0.855	0.909	0.898	0.888	0.879	0.883	0.900	0.922
4	0.879	0.862	0.897	0.910	0.882	0.894	0.864	0.880	0.916
5	0.877	0.876	0.916	0.904	0.909	0.892	0.856	0.902	0.921
6	0.872	0.889	0.909	0.904	0.895	0.889	0.870	0.917	0.931
7	0.880	0.898	0.918	0.883	0.899	0.877	0.879	0.882	0.918
8	0.888	0.891	0.891	0.901	0.906	0.899	0.845	0.859	0.929
9	0.885	0.900	0.900	0.896	0.890	0.923	0.796	0.864	0.914
10	0.863	0.902	0.883	0.903	0.890	0.909	0.877	0.880	0.922
Avg.	0.877	0.885	0.903	0.901	0.886	0.896	0.862	0.889	0.923

Table 5-10 The results of F-score over 10 independent runs

Run	CNN-LSTM	GWO	PSO	GSA	FPA	CSO	prLeGWO	FuzzyGWO	Prop.GWO
1	0.880	0.886	0.907	0.912	0.881	0.892	0.863	0.883	0.928
2	0.884	0.891	0.900	0.905	0.813	0.908	0.886	0.930	0.930
3	0.857	0.854	0.910	0.898	0.888	0.878	0.883	0.901	0.925
4	0.879	0.861	0.896	0.909	0.882	0.894	0.860	0.880	0.916
5	0.876	0.876	0.916	0.903	0.908	0.892	0.856	0.903	0.920
6	0.871	0.888	0.909	0.905	0.895	0.891	0.870	0.917	0.924
7	0.880	0.898	0.917	0.883	0.898	0.876	0.880	0.880	0.918
8	0.887	0.890	0.890	0.900	0.906	0.899	0.843	0.858	0.931
9	0.884	0.900	0.899	0.896	0.890	0.922	0.792	0.861	0.914
10	0.861	0.902	0.884	0.902	0.889	0.909	0.874	0.879	0.925
Avg.	0.876	0.885	0.903	0.901	0.885	0.896	0.861	0.889	0.923

Table 5-11 The results of precision over 10 independent runs

Run	CNN-LSTM	GWO	PSO	GSA	FPA	CSO	prLeGWO	FuzzyGWO	Prop.GWO
1	0.883	0.887	0.909	0.915	0.887	0.892	0.872	0.891	0.930

2	0.885	0.893	0.901	0.908	0.818	0.907	0.889	0.932	0.931
3	0.859	0.857	0.912	0.899	0.894	0.878	0.886	0.904	0.927
4	0.878	0.861	0.899	0.908	0.890	0.896	0.862	0.883	0.917
5	0.876	0.878	0.917	0.909	0.908	0.894	0.861	0.908	0.921
6	0.871	0.893	0.911	0.909	0.898	0.891	0.871	0.917	0.925
7	0.880	0.900	0.917	0.883	0.899	0.877	0.884	0.881	0.920
8	0.887	0.895	0.889	0.901	0.907	0.900	0.854	0.859	0.933
9	0.884	0.901	0.898	0.896	0.890	0.922	0.796	0.868	0.915
10	0.863	0.905	0.884	0.903	0.892	0.910	0.874	0.884	0.927
Avg.	0.877	0.887	0.904	0.903	0.888	0.897	0.865	0.893	0.925

Table 5-12 The results of Recall over 10 independent runs

Run	CNN-LSTM	GWO	PSO	GSA	FPA	CSO	prLeGWO	FuzzyGWO	Prop.GWO
1	0.881	0.887	0.909	0.913	0.881	0.894	0.861	0.883	0.928
2	0.885	0.891	0.900	0.905	0.816	0.910	0.888	0.930	0.931
3	0.857	0.853	0.910	0.900	0.888	0.878	0.882	0.902	0.924
4	0.880	0.863	0.899	0.911	0.882	0.895	0.860	0.882	0.918
5	0.878	0.879	0.916	0.905	0.908	0.894	0.854	0.904	0.922
6	0.872	0.886	0.910	0.905	0.896	0.892	0.873	0.919	0.924
7	0.881	0.901	0.917	0.885	0.899	0.876	0.881	0.879	0.918
8	0.889	0.891	0.892	0.903	0.907	0.899	0.841	0.861	0.931
9	0.885	0.902	0.901	0.898	0.893	0.923	0.789	0.861	0.916
10	0.864	0.902	0.885	0.903	0.891	0.910	0.875	0.877	0.924
Avg.	0.877	0.885	0.904	0.903	0.886	0.897	0.860	0.890	0.924

With respect to classification accuracy, the CNN-LSTM configurations yielded by the proposed GWO variant achieve the highest mean accuracy rate of 92.3%, outperforming those of all baseline models. In particular, the proposed GWO variant demonstrates significant advantages than the original GWO and advanced GWO variants, i.e. prLeGWO and FuzzyGWO, as well as the default CNN-LSTM network, with evident performance gaps of 3.8%, 6.1%, 3.4%, and 4.6%, respectively. In addition, similar superiorities of the proposed GWO model can also be observed consistently across the remaining indicators, i.e. F-score, precision, and recall scores, as shown in **Tables 5-10** – **5-12**.

Table 5-13 The mean accuracy rate of each class over 10 independent runs

Class	CNN-LSTM	GWO	PSO	GSA	FPA	CSO	prLeGWO	FuzzyGWO	Prop.GWO
Walk	0.905	0.915	0.935	0.924	0.908	0.935	0.886	0.941	0.973
W-Up	0.890	0.917	0.905	0.908	0.878	0.907	0.843	0.906	0.942
W-Dn	0.918	0.923	0.956	0.980	0.944	0.951	0.863	0.912	0.984
Sit	0.787	0.768	0.808	0.778	0.773	0.792	0.768	0.785	0.791
Stand	0.787	0.829	0.835	0.863	0.837	0.825	0.838	0.847	0.891
Lay	0.976	0.961	0.984	0.966	0.976	0.974	0.966	0.966	0.964

The decomposed accuracy results with respect to each of the six human activities are provided in **Table 5-13**. The optimized CNN-LSTM networks yielded by the proposed GWO variant obtain the highest accuracy results on four activity classes, i.e. walking, walking upstairs, walking downstairs, and standing, significantly outperforming those yielded by the baseline methods and default network with evident performance gaps. This indicates that the CNN-LSTM configurations yielded by the proposed GWO variant successfully discover distinctive variations and discriminative patterns with respect to different human activities, therefore achieving more robust and advanced performances. In other words, the decomposed results further reinforce the effectiveness and superiority of the proposed GWO variant in identifying the most effective deep networks for undertaking HAR tasks, in comparison with the baselines.

Table 5-14 The mean configurations of the identified CNN-LSTM networks over 10 runs

Conf.	GWO	PSO	GSA	FPA	CSO	prLeGWO	FuzzyGWO	Prop.GWO
No. 1st	38.8	94.4	51.2	64.2	125.4	68.5	59.2	230.8
No. 2st	59.0	80.2	60.0	18.8	57.2	130.3	92.6	193.6
S. 1st	4.0	3.7	3.7	3.7	4.4	3.6	3.6	4.4
S. 2st	4.5	3.3	3.7	3.6	3.5	3.3	3.6	3.6
Pool.	3.4	3.4	3.6	3.3	3.8	3.3	3.6	2.7
LSTM	68.5	102.1	102.1	126.3	97.7	34.1	100.3	60.2
Dense	92.9	104.3	110.7	105.8	131.2	20.4	108.9	41.2
DR	0.376	0.427	0.293	0.214	0.195	0.143	0.246	0.416
LR	0.037	0.059	0.046	0.046	0.048	0.020	0.028	0.029

Moreover, the mean hyperparameters of the optimized CNN-LSTM networks over ten independent runs are presented in **Table 5-14**. In particular, the devised CNN-LSTM networks for HAR possess the highest numbers of filters in both convolutional layers, i.e. 230.8 and 193.6, respectively, while maintaining lighter settings of number of nodes in the

recurrent and dense layers, i.e. 60.2 and 41.2, respectively, in comparison with those of the baseline models and the default network settings. Such configurations enable CNN-LSTM networks to thoroughly examine fundamental characteristics with respect to each category of human activity and differentiate subtle differences between them, so as to extract the most discriminative features related to human activities in convolutional layers, while achieving efficient trade-offs between learning long-term dependencies embedded among consecutive body movements and avoiding overfitting on data noise in the recurrent and dense layers. As such, the CNN-LSTM networks identified by the proposed GWO variant are capable of distinguishing different human activities effectively.

5.2.4 Remarks

Overall, the proposed GWO variant is capable of identifying the most effective CNN-LSTM configurations with appropriate representational capacities and superior capabilities of feature extraction, for resolving all three employed time series tasks. In contrast, the baseline search methods yield less effective sub-optimal CNN-LSTM networks with oversized or undersized hyperparameters, which result in severe performance degradation. More specifically, the oversized settings in the recurrent and dense layers and the lack of regularization are likely to result in overfitting owing to the excessive representational capacities and the memorizing of sample noise, as indicated by the results of GWO and CSO on energy consumption forecasting, FPA and CSO on PM2.5 concentration prediction, as well as FPA and FuzzyGWO on HAR. Moreover, the undersized network configurations produce oversimplified CNN-LSTM structures with restricted interpretation capabilities, therefore unable to fully capture sophisticated dependencies embedded in variables under complex temporal contexts, neither to conduct effective feature extractions and transformations, as exemplified by the results of FuzzyGWO on PM2.5 prediction, and prLeGWO on HAR. Furthermore, our optimized networks also outperform the CNN-LSTM model with the default hyperparameter settings significantly in the employed three test scenarios, owing to the limitations of the pre-assigned inefficient model and training configurations in such default baseline networks, i.e. the lack of learnable filters for feature extraction and the memorizing of sample noise resulted from the redundant recurrent memory cells. To sum up, the proposed GWO variant demonstrates significant advantages over baseline models in automatic identification of the optimal CNN-LSTM configurations for undertaking all test time series tasks, owing to the enhanced search diversity and search efficiency.

5.2.5 Wilcoxon statistical test

The Wilcoxon statistical rank sum test is also conducted to further indicate the statistical distinctiveness of the enhanced GWO model against the baseline methods in searching for the optimal CNN-LSTM configurations. The accuracy results are employed for the statistical analysis on HAR, whereas the RMSE results are applied for the test on both energy consumption forecast and PM2.5 concentration prediction. As shown in **Table 5-15**, the rank sum test results are lower than 0.05, which indicate that the proposed GWO variant statistically significantly outperforms all the baseline search methods, including four classical methods, i.e. GWO, PSO, GSA, and FPA, as well as three advanced variant models, i.e. CSO, prLeGWO, and FuzzyGWO, in searching for the optimal CNN-LSTM configurations to solve time series prediction problems. Our devised optimized networks also show statistically significant superiority over those with default settings in our experimental studies. This superiority of the proposed GWO variant can be ascribed to the improved trade-offs between search diversification and intensification facilitated by the cooperation among proposed comprehensive and complementary search strategies. We provide detailed analysis as follows.

Table 5-15 Wilcoxon rank sum test results over 10 independent run

Run	CNNLSTM	GWO	PSO	GSA	FPA	CSO	prLeGWO	FuzzyGWO
HAR	1.80E-04	1.81E-04	4.35E-04	1.81E-04	1.81E-04	9.99E-04	1.81E-04	2.20E-03
Energy	1.83E-04	3.61E-03	2.57E-02	2.20E-03	3.61E-03	3.61E-03	4.52E-02	3.61E-03
PM2.5	1.83E-04	3.30E-04	7.69E-04	2.46E-04	1.13E-02	3.61E-03	1.13E-02	3.30E-04

The paramount challenge in retrieving effective CNN-LSTM configurations lies in the sophisticated interactions between different components within the network, as well as heavy training computational costs. In this regard, the proposed GWO variant incorporates several distinctive and complementary strategies, capable of boosting search diversity, as well as the convergence speed, to resolve the above challenges occurred during the exploration of the optimal CNN-LSTM configurations. Specifically, an advanced trade-off between search diversification and intensification is achieved by the proposed nonlinear adjustment of the territory boundary. Under this scheme, the search range during the exploration is upheld at the initial level without acute decrease, enabling the wolf population conduct more extensive explorations around the peripheral areas of the search territory, instead of being drawn to the vicinity of the leading wolves at the beginning of the search process. Meanwhile, this transition scheme also enables the wolf population to focus on the closer bounds around leading wolves and conduct thorough detection around the promising regions during exploitation. In addition, the proposed sinusoidal chaotic leadership rivalry

enables the GWO variant to leverage the merits from both multiple-leader guided search as well as single-leader guided search, through reinforcing the leadership of the best wolf solution while periodically downplaying the influence of this global best solution in position updating. As such, a periodic balance between search diversity and concentration is achieved. Thirdly, the fine-tuning capability around the global best position is further improved by conducting refined local detections with various steps and directions at the final stage of the search process, using a dedicated damped function with a dynamic adjustment of the amplitude. Lastly, the qualities of three leading wolves are further enhanced using Lévy flight probability distributions to reduce the likelihood of the stagnation at local optima.

Overall, the effectiveness of the proposed GWO variant can be ascribed to the enhanced search diversity and search efficiency. The diversity is improved from three perspectives, i.e. the upholding of the search territory boundary through the dedicated nonlinear control of the exploration factor, the diversification of leading signals by the chaotic allocation of leadership weights, as well as leader random walks based on Lévy flight, whereas the efficiency is achieved from two perspectives, i.e. the ascertained dominance of the best wolf leader during the search process, as well as the dedicated local exploitation around the global best solution at the final stage of the search course. As such, the enhanced GWO model is more likely to escape from local stagnation and attain the global optimality. Therefore, the complicated interactions among CNN-LSTM hyperparameters can be thoroughly explored by the proposed GWO variant, and effective CNN-LSTM configurations could be identified swiftly. The efficiency of the proposed GWO-based CNN-LSTM network is evidenced by the superior empirical results on the three employed time series problems as well as the results of the statistical test. In contrast, the baseline GWO variants, e.g. prLeGWO and FuzzyGWO, achieve less efficient trade-offs between reassuring the dominance of the best leader and retaining diversity in the reconstruction of leadership hierarchy. Besides that, there is a lack of refinement in terms of the transition between exploration and exploitation among the above baseline GWO variants and other search methods. Overall, the enhanced GWO algorithm demonstrates great advantages in devising optimal CNN-LSTM networks and outperforms the eight baseline methods significantly in undertaking time series prediction tasks.

5.3 Summary

In this chapter, an evolving CNN-LSTM network has been proposed to solve time series prediction problems. A GWO variant has been proposed for the automatic optimal hyperparameter and topology identification of the network architectures. The proposed GWO variant employs a nonlinear exploration rate for search boundary adjustment, a

sinusoidal chaotic map for the leadership allocation of the dominant wolf leaders, an enhanced spiral local exploitation scheme, as well as Lévy flight-based leader enhancement. As such, the search process becomes more diversified owing to the expansion of the search territory, random exploitation of wolf leaders, and the chaotic aggregation and periodical diversification of guiding signals. In addition, the search efficiency and convergence rate are improved owing to the dominance of the global best wolf leader over the combined distractions from the remaining two leaders during the search process, as well as the intensified local exploitation around the global best solution at the final search stage.

The proposed GWO-based evolving CNN-LSTM time series forecasting model has been evaluated using two time series prediction problems, i.e. energy consumption forecast and PM2.5 concentration prediction, as well as a time series classification task, i.e. human activity recognition. The devised evolving deep networks outperform the default network and those yielded by a total of seven baseline search models including four classical search methods and three advanced GWO and PSO variants on all the test data sets, statistically significantly. Moreover, the empirical results indicate that our optimized CNN-LSTM networks are characterized by a higher number of filters in convolutional layers and moderate settings in terms of the number of nodes in the LSTM layer and the fully connected layer. Such devised networks possess superior capabilities in capturing spatial and temporal information to inform time series prediction and classification, over those identified by all the baseline methods. In other words, such optimal network configurations are able to thoroughly examine the interactions among time series variables, as well as illustrate efficient network representational capacities without subjecting to either overfitting or underfitting.

Chapter 6
Conclusions

In this research, three evolving ML and one evolving DL methods have been proposed to overcome three severe bottlenecks in ML and data mining, i.e. initialization sensitivity, feature selection, as well as hyperparameter optimization. Firstly, two FA-based evolutionary KM clustering methods have been devised to automatically generate the optimal configuration of cluster centroids and overcome local stagnation, for the conventional KM clustering. Secondly, a PSO-based evolutionary feature selection method has been developed to automatically determine the optimal feature subset and mitigate the curse of dimensionality, through the elimination of redundant and contradictory variables embedded in classification problems. Lastly, a GWO-based evolving CNN-LSTM method has been proposed to automatically identify the optimal learning and topological configurations for CNN-LSTM networks to undertake real-life time series forecasting challenges.

6.1 Summary of the contribution

The contributions in this research are summarized as follows:

1) The first contribution is the design of two FA based evolutionary KM clustering methods.

   Firstly, intrinsic limitations embedded in the search mechanism of the original FA model are identified. Although one firefly is able to approach another with a more favourable position in the original FA, the movement can only happen on the diagonal formed by the two fireflies under comparison, owing to the inheritance of biological laws in a rigid manner. As a result, the space and diversity for exploitation are severely constrained during the approaching movement and the search process is more likely to stagnate at local optima traps. In addition, search efficiency in the original FA algorithm is also undermined owing to the lack of guarantee of distinctiveness in terms of fitness scores between fireflies under comparison. As a result, the position adjustment is likely to become insignificant and even futile, hence resulting in the waste of resource.

   Two enhanced FA variants, i.e. IIEFA and CIEFA, are proposed to overcome the above identified search limitations in the original FA model. With respect to IIEFA, the attractiveness coefficient in the original FA method is replaced with a

randomized control matrix. As such, IIEFA is able to be released from the search constraints imposed by the strict adherence to the biological law, as the exploitation capability in the neighbourhood is elevated from a one-dimensional to multi-dimensional search mechanism with enhanced diversity in search scopes, scales, and directions. In addition to the parameter matrix, the second proposed FA variant, namely CIEFA, further employs a dispersing mechanism. It enhances global exploration by dispatching fireflies with high similarities to unexploited positions out of the close neighbourhood. The search efficiency is also enhanced owing to the guarantee of heterogeneity between fireflies in competition.

Both the proposed FA models, i.e. IIEFA and CIEFA, are employed to devise the evolutionary KM clustering methods. The enhanced FA variants are applied to automatically identify the optimal configuration of cluster centroids for KM clustering. The proposed evolutionary clustering methods have been evaluated on ALL-IDB2 database, a skin lesion data set, and a total of 15 UCI data sets. The empirical results indicate that the proposed FA-based KM clustering models demonstrate statistically significant superiority in both the distance and performance measures in comparison with the conventional KM clustering, and ten baseline search methods. Moreover, CIEFA outperforms IIEFA in tackling challenging clustering tasks with noise, complicated data distributions, and non-compact and less separable clusters, owing to its enhanced exploration capability and expanded search territory.

2) The second contribution is the development of a PSO-based evolutionary feature selection method.

Firstly, an enhanced PSO variant is proposed to overcome two major shortcomings of the original PSO method, i.e. premature convergence and weak local exploitation capability around near optimal solutions. It incorporates several distinctive strategies, including the leader enhancement using skewed Gaussian distributions, the recombination of genes from personal best solutions and the mirroring mutation on the global best solution for worse solution replacement, the diversification of guiding signals for region-based search, as well as the intensified local spiral exploitation. Therefore, the proposed PSO model is capable of achieving advanced trade-offs between utilization of acquired elicit solutions and introduction of dynamic distractions into the search trajectory, hence eliminating adverse effects resulted from the dictation of the global best signals in the original PSO. As such,

the proposed PSO variant is more likely to escape local optima traps and attain global optimality.

Subsequently, an evolutionary feature selection method is designed based on the proposed PSO model. The enhanced PSO variant, in conjunction with KNN classifier, is employed to identify the optimal feature subset and reduce feature dimensionality for undertaking complex classification challenges. The proposed feature selection method has been evaluated using the ALL-IDB2 database and 9 other UCI data sets with diverse dimensionalities from 30 to 10000. It obtains the highest classification performances on the employed ten data sets and achieves superior trade-offs between feature elimination and classification accuracy, in comparison with ten baseline search models. Moreover, the advantages of the proposed evolutionary feature selection method become more evident on high-dimensional classification tasks owing to the enhanced exploration and exploitation capabilities.

3) The third contribution is the devising of a GWO-based evolving CNN-LSTM method for time series prediction.

Firstly, an enhanced GWO variant is proposed to overcome stagnation at local optima and slow convergence rate in the original GWO model. It incorporates four distinctive strategies, including a nonlinear dynamic adjustment of search coefficient, a chaotic weight allocation scheme for dominant wolves, an enhanced spiral local exploitation scheme, as well as Lévy flight-based leader enhancement. The proposed GWO variant is capable of overcoming two major limitations of the original GWO algorithm, i.e. the insufficiency of exploration owing to the sharp contraction of search territory as well as the inefficiency of the fine-tuning exploitation around the global best solution, particularly in the final stage of the evolution where convergence of the population is required, owing to the distraction of the other two wolf leaders.

Subsequently, an evolving CNN-LSTM method is devised based on the proposed GWO variant for tackling time series prediction problems. The enhanced GWO variant is employed to automatically generate the optimal learning and topological configurations for the base architecture of CNN-LSTM network. The proposed evolving CNN-LSTM time series forecasting method has been evaluated using three time series scenarios, i.e. energy consumption forecast, PM2.5 pollution prediction,

and human activity recognition (HAR). It statistically significantly outperforms the CNN-LSTM network with default settings as well as seven classical and advanced search methods, and demonstrates great advantages in identifying the most effective leaning and topological configurations for the base CNN-LSTM architecture. The identified configurations by the proposed GWO variant are characterized by superior capabilities of feature extraction owing to the higher number of filters in convolutional layers, as well as by more appropriate representational capacities without suffering from overfitting owing to the moderate numbers of nodes in both the LSTM and fully connected layers.

6.2 Future work

Despite the great superiorities in tackling major bottlenecks in ML and data mining, i.e. feature selection, initialization sensitivity, as well as hyperparameter optimization, the proposed evolving ML and DL methods can be further enhanced from two perspectives: i.e. reducing the dependence on prior knowledge and human intervention as well as advancing the levels of automation when undertaking real-life problems of interest.

Specifically, the proposed evolving ML and DL models still require certain prior knowledge and human intervention to initiate the evolution, such as the number of the nearest neighbours in KNN for feature selection, the number of total clusters embedded in data sets for KM clustering, as well as the construction of a base CNN-LSTM architecture as the level playing field for hyperparameter optimization. However, sometimes it can be very difficult to gain the above profound understandings from the raw data sets as well as ML algorithms. Therefore, more advanced evolving ML and DL systems need to be developed to achieve higher level of independence through improving the flexibility of encoding scheme in EAs as well as enabling the co-evolution of different types of parameters associated with both data sets and ML models.

Moreover, the advanced evolutionary automated machine learning (AutoML) platform could be developed based on the proposed advanced EC techniques, to provide an easy-to-use ML pipeline system capable of automating and optimizing the whole modelling process, including data preparation, feature engineering, model selection, hyperparameter optimization, and model evaluation. As such, the highly effective ML and DL models could be manufactured and customized automatically for the investigated data mining problems, without any requirement of ML expertise for practitioners. The potential avenues for future works are summarized as below.

1) Co-evolution of the number of clusters and the configuration of cluster centroids for KM clustering. The proposed evolutionary KM clustering methods require the prior knowledge about the number of clusters of the data sets for the identification of the optimal cluster centroids. However, such expertise is often absent owing to the lack of understandings about the related problem domain. Therefore, it can be extremely beneficial to further extend the proposed evolutionary KM clustering method for cogeneration of the number of clusters as well as the optimal configuration of cluster centroids [312]. The innovative variable encoding schemes as well as effective evolution operators need to be developed to resolve the possible incongruity, resulted from disparate representations of the underlying clustering scenarios among different search individuals.
2) Discovery of innovative deep neural network topologies. The proposed evolving CNN-LSTM method requires a base architecture of CNN-LSTM as the foundation, upon which the optimal topological and learning configurations can be evolved. Its construction entails profound expertise about DNNs as well as significant trial-and-error efforts [40, 41]. However, such involvement of human experience and efforts in the development of network structures may inadvertently introduce biases into the evolving process, which could possibly hinder the discovery of innovative network topologies that transcend empirical knowledge. In this regard, I aim to develop neural architecture search methods which are capable of exploring extensive possibilities of topology with less constraints of conventional experiences. The innovative operators of crossover and mutation will be experimented to achieve advanced trade-offs between evolution efficacy and computational efficiency.
3) Development of an evolutionary AutoML platform employing advanced EC techniques. An end-to-end AutoML platform could be developed to automate the modelling process of applying ML algorithms to tackle real-life problems [313-315]. Moreover, the proposed enhanced PSO, FA, as well as GWO methods can be implemented into the platform to optimize the essential components during the modelling process, such as feature engineering, model selection, and configuration identification. As a result, the developed evolutionary AutoML platform is not only capable of yielding the highly performant ML models with tailored and optimized configurations, but also eliminating any requirement for profound expertise from ML practitioners.
4) Optimization of adaptation capability of EAs. The delicate balance between exploration and exploitation plays a crucial role in the capability of attaining global optimality for EAs. To be specific, the appropriate proportion of exploration in the search process facilitates the adaptation of the population in the long run by

sacrificing the temporary short-term benefits. Hence it enables the search to escape from local optima traps effectively. On the other hand, excessive degree of explorations could result in the slow convergence of the population, which in turn undermines search efficiency. Therefore, there exists an optimal threshold for the involvement of exploration in the search process to maximize the adaptation capability of EAs [316]. However, the identification of the threshold with respect to the optimal proportion of exploration in EAs remains an open problem, owing to its complex nature, i.e. an optimization process over another optimization process. In this regard, I aim to conduct theoretical and empirical analysis on this question to gain better understandings about the search dynamics of EAs and hopefully provide constructive guidance regarding the settings of optimal exploration rates.

References

1. J.G. Carbonell, R.S. Michalski, and T.M. Mitchell, An overview of machine learning, in *Machine Learning: An Artificial Intelligence Approach*, 1983, Springer Berlin Heidelberg, p. 3-23.
2. D. Fogel, *Evolutionary Computation - Toward a New Philosophy of Machine Intelligence* (3. ed.), 1995, Wiley–Blackwell.
3. T.M. Mitchell, *Machine Learning*, 1997, McGraw Hill Education.
4. E. Alpaydin, *Introduction to Machine Learning*, 2010, The MIT Press.
5. L.P. Kaelbling, M.L. Littman, and A.W. Moore, Reinforcement learning: a survey. *Journal of Artificial Intelligence*, 1996. **4**(1): p. 237–285.
6. A. Al-Kaff, D. Martín, F. García, A. Escalera, J. Armingol, Survey of computer vision algorithms and applications for unmanned aerial vehicles. *Expert Systems with Applications*, 2018. **92**: p. 447-463.
7. D.W. Otter, J.R. Medina, and J.K. Kalita, A survey of the usages of deep learning for natural language processing. *IEEE Transactions on Neural Networks and Learning Systems*, 2020: p. 1-21.
8. S. Wang, W. Chaovalitwongse, and R. Babuska, Machine learning algorithms in bipedal robot control. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 2012. **42**(5): p. 728-743.
9. E.A. El-Dahshan H.M. Mohsen, K. Revett, A.M. Salem, Computer-aided diagnosis of human brain tumor through MRI: A survey and a new algorithm. *Expert Systems with Applications*, 2014. **41**(11): p. 5526-5545.
10. S. Zhang, L. Yao, A. Sun, Y. Tay, Deep learning based recommender system: a survey and new perspectives. *ACM Computing Surveys*, 2019. **52**(1): p. Article 5.
11. S. Weiming, W. Lihui, and H. Qi, Agent-based distributed manufacturing process planning and scheduling: a state-of-the-art survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 2006. **36**(4): p. 563-577.
12. A. Krizhevsky, I. Sutskever, and G.E. Hinton, ImageNet classification with deep convolutional neural networks, in *Proceedings of the 25th International Conference on Neural Information Processing Systems*, 2012. **1**: p. 1097–1105.
13. I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks. [online], available: https://arxiv.org/abs/1406.2661, 2014.
14. A. L'Heureux, K. Grolinger, H.F. Elyamany, M. Capretz, Machine learning with big data: challenges and approaches. *IEEE Access*, 2017. **5**: p. 7776-7797.
15. Ishwarappa and J. Anuradha, A brief introduction on big data 5Vs characteristics and hadoop technology. *Procedia Computer Science*, 2015. **48**: p. 319-324.
16. G. Dong and H. Liu, *Feature Engineering for Machine Learning and Data Analytics*. 2018: CRC Press.
17. B.H. Nguyen, B. Xue, and M. Zhang, A survey on swarm intelligence approaches to feature selection in data mining. *Swarm and Evolutionary Computation*, 2020. **54**: p. 100663.
18. R. Bellman, Dynamic programming. *Science*, 1966. **153**(3731): p. 34-37.
19. J. Fan, and Y. Liao, Endogeneity in High Dimensions. *Annals of statistics*, 2014. **42 3**: p. 872-917.
20. J. Fan, F. Han, and H. Liu, Challenges of big data analysis. *National Science Review*, 2014. **1**(2): p. 293-314.
21. E. Shireman, D. Steinley, and M.J. Brusco, Examining the effect of initialization strategies on the performance of Gaussian mixture modeling. *Behavior Research Methods*, 2017. **49**(1): p. 282-293.
22. P. Fränti, and S. Sieranoja, How much can k-means be improved by using better initialization and repeats? *Pattern Recognition*, 2019. **93**: p. 95-112.
23. U. Fayyad, C. Reina, and P.S. Bradley, Initialization of iterative refinement clustering algorithms, in *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*. 1998, AAAI Press. p. 194–198.
24. J. Qiao, S. Li, and W. Li, Mutual information based weight initialization method for sigmoidal feedforward neural networks. *Neurocomputing*, 2016. **207**: p. 676-683.
25. C.A.R.d Sousa. An overview on weight initialization methods for feedforward neural networks. in *Proceedings of 2016 International Joint Conference on Neural Networks (IJCNN)*. 2016.
26. I. Sutskever, J. Martens, G. Dahl, G. Hinton, On the importance of initialization and momentum in deep learning. In *Proceedings of* 30th International Conference on Machine Learning, ICML 2013, 2013: p. 1139-1147.
27. K. Bennett, and E. Parrado-Hernández, The interplay of optimization and machine learning research. *Journal of Maching Learning Research*, 2006. **7**: p. 1265-1281.
28. P. Jain, and P. Kar, Non-convex optimization for machine learning. [online], available: https://arxiv.org/abs/1712.07897, 2017.
29. X. Glorot, and Y. Bengio, Understanding the difficulty of training deep feedforward neural networks. Journal of Machine Learning Research, 2010. **9**: p. 249-256.
30. F. Cao, J. Liang, and G. Jiang, An initialization method for the K-Means algorithm using neighborhood model. *Computers & Mathematics with Applications*, 2009. **58**(3): p. 474-483.
31. S. Park, S. Seo, C. Jeong, J. Kim, The weights initialization methodology of unsupervised neural networks to improve clustering stability. *The Journal of Supercomputing*, 2020. **76**(8): p. 6421-6437.

32. M.E. Celebi, H.A. Kingravi, and P.A. Vela, A comparative study of efficient initialization methods for the k-means clustering algorithm. *Expert Systems with Applications*, 2013. **40**(1): p. 200-210.
33. J.N. Van Rijn, and F. Hutter. Hyperparameter importance across datasets. in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018.
34. L. Yang, and A. Shami, On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, 2020. **415**: p. 295-316.
35. P. Probst, A. Boulesteix, and B. Bischl, Tunability: Importance of hyperparameters of machine learning algorithms. *Journal of Maching Learning Research*, 2019. **20**: p. 53:1-53:32.
36. P. Probst, A.-L. Boulesteix, and M. Wright, *Hyperparameters and Tuning Strategies for Random Forest*. 2018.
37. R.G. Mantovani, A. Rossi, E. Alcobaça, J. Vanschoren, A. Carvalho, A meta-learning recommender system for hyperparameter tuning: Predicting when tuning improves SVM classifiers. *Information Sciences*, 2019. **501**: p. 193-221.
38. J. Bergstra, and Y. Bengio, Random search for hyper-parameter optimization. *Journal of Maching Learning Research*, 2012. **13**(null): p. 281–305.
39. N. Lavesson, and P. Davidsson, Quantifying the impact of learning algorithm parameter tuning. In *Proceedings of 21st National Conference on Artificial Intelligence and 8th Conference Innovative Applications of Artificial Intelligence*, 2006.
40. Y. Liu, Y. Sun, B. Xue, M. Zhang, G. Yen, A survey on evolutionary neural architecture search. [Online], available: https://arxiv.org/abs/2008.10937, 2020.
41. T. Elsken, J.H. Metzen, and F. Hutter, Neural architecture search: A survey. [Online], available: https://arxiv.org/abs/1808.05377, 2019.
42. Yu, T. and H. Zhu, Hyper-parameter optimization: A review of algorithms and applications. [Online], available: https://arxiv.org/abs/2003.05689, 2020.
43. X. Xiao, M. Yan, S. Basodi, C. Ji, Y. Pan, Efficient hyperparameter optimization in deep learning using a variable length genetic algorithm. [Online], available: https://arxiv.org/abs /2006.12703, 2020.
44. T. Bäck, T. Fogel, and D. Michalewicz, *Evolutionary Computation 1 (Basic Algorithms and Operators)*. 2000.
45. X.-S. Yang, and M. Karamanoglu, Swarm intelligence and bio-inspired computation: An overview *in Swarm Intelligence and Bio-Inspired Computation: Theory and applications*. 2013. p. 3-23.
46. I. Zelinka, A survey on evolutionary algorithms dynamics and its complexity – Mutual relations, past, present and future. *Swarm and Evolutionary Computation*, 2015. **25**: p. 2-14.
47. A.P. Piotrowski, M.J. Napiorkowski, J.J. Napiorkowski, P.M. Rowinski, Swarm intelligence and evolutionary algorithms: Performance versus speed. *Information Sciences*, 2017. **384**: p. 34-85.
48. Ab Wahab, M.N., S. Nefti-Meziani, and A. Atyabi, A comprehensive review of swarm optimization algorithms. *PLOS ONE*, 2015. **10**(5): p. e0122827.
49. X. Yang, Swarm intelligence based algorithms: a critical analysis. *Evolutionary Intelligence*, 2014. **7**: p. 17-28.
50. L. Lin, and M. Gen, Auto-tuning strategy for evolutionary algorithms: balancing between exploration and exploitation. *Soft Computing*, 2009. **13**(2): p. 157-168.
51. S. Mirjalili, and J. Song Dong, Introduction to nature-inspired algorithms, in *Nature-Inspired Optimizers: Theories, Literature Reviews and Applications*, 2020, Springer International Publishing: Cham. p. 1-5.
52. H. Xie, L. Zhang, C.P. Lim, Y. Yu, C. Liu, H. Liu, and J. Walters, Improving K-means clustering with enhanced Firefly Algorithms. *Applied Soft Computing*, 2019. **84**: p. 105763.
53. H. Xie, L. Zhang, and C.P. Lim, Evolving CNN-LSTM Models for Time Series Prediction Using Enhanced Grey Wolf Optimizer. *IEEE Access*, 2020. **8**: p. 161519-161541.
54. A. Eiben, and J. Smith, *Introduction To Evolutionary Computing*. Vol. 45. 2003.
55. J.H. Holland, Genetic algorithms. *Scientific American*, 1992. **267**(1): p. 66-73.
56. R. Storn, K. Price, Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 1997. **11**(4): p. 341-359.
57. K. Sörensen, M. Sevaux, MAPM: memetic algorithms with population management. *Computers & Operations Research*, 2006. **33**(5): p. 1214-1225.
58. R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory. in MHS'95. *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*. 1995.
59. M. Dorigo, Optimization, Learning and Natural Algorithms. 1992.
60. X.S. Yang, *Nature-Inspired Metaheuristic Algorithms*. 2010.
61. X.S. Yang, D. Suash. Cuckoo Search via Lévy flights. in *Proceedings of 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*. 2009.
62. S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey Wolf Optimizer. *Advances in Engineering Software*, 2014. **69**: p. 46-61.
63. S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems*, 2015. **89**: p. 228-249.
64. S. Mirjalili, Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Computing and Applications*, 2015. **27**(4): p. 1053-1073.

65. X.S. Yang, A new metaheuristic bat-inspired algorithm. In *Proceedings of Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, 2010. **284**.
66. X.S. Yang, Flower pollination algorithm for global optimization. in *Unconventional Computation and Natural Computation*. 2012. Berlin, Heidelberg: Springer Berlin Heidelberg.
67. E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, GSA: A gravitational search algorithm. *Information Sciences*, 2009. **179**(13): p. 2232-2248.
68. J. Kennedy, R.C. Eberhart, Y. Shi, chapter seven - The Particle Swarm, in *Swarm Intelligence*, J. Kennedy, R.C. Eberhart, and Y. Shi, Editors. 2001, Morgan Kaufmann: San Francisco. p. 287-325.
69. X.S. Yang, , Chapter 7 - Particle Swarm Optimization, in *Nature-Inspired Optimization Algorithms*, X.S. Yang, Editor. 2014, Elsevier: Oxford. p. 99-110.
70. Z. Yang, K. Tang, X. Yao, Large scale evolutionary optimization using cooperative coevolution. *Information Sciences*, 2008. **178**(15): p. 2985-2999.
71. M. Zhang, W. Luo, X. Wang, Differential evolution with dynamic stochastic selection for constrained optimization. *Information Sciences*, 2008. **178**(15): p. 3043-3074.
72. S. Hsieh, T. Sun, C. Liu, S. Tsai, Efficient Population Utilization Strategy for Particle Swarm Optimizer. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 2009. **39**(2): p. 444-456.
73. J.J. Liang, A.K. Qin, P.N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Transactions on Evolutionary Computation*, 2006. **10**(3): p. 281-295.
74. K. Chen, F.-Y. Zhou, and X.-F. Yuan, Hybrid particle swarm optimization with spiral-shaped mechanism for feature selection. *Expert Systems with Applications*, 2019. **128**: p. 140-156.
75. C.W. Ahn, J. An, J.-C. Yoo, Estimation of particle swarm distribution algorithms: Combining the benefits of PSO and EDAs. Information Sciences, 2012. **192**: p. 109-119.
76. M. Iqbal, M.A.M. de Oca. An Estimation of Distribution Particle Swarm Optimization Algorithm. in *Ant Colony Optimization and Swarm Intelligence*. 2006. Berlin, Heidelberg: Springer Berlin Heidelberg.
77. K. Chen, F. Zhou, L. Yin, S. Wang, Y. Wang, F. Wan, A hybrid particle swarm optimizer with sine cosine acceleration coefficients. *Information Sciences*, 2018. **422**: p. 218-241.
78. M. Pluhacek, R. Senkerik, D. Davendra, Chaos particle swarm optimization with Eensemble of chaotic systems. *Swarm and Evolutionary Computation*, 2015. **25**: p. 29-35.
79. J. Kennedy, R. Mendes. Population structure and particle swarm performance. in *Proceedings of the 2002 Congress on Evolutionary Computation*. CEC'02 (Cat. No.02TH8600). 2002.
80. P.K. Das, H.S. Behera, B.K. Panigrahi, A hybridization of an improved particle swarm optimization and gravitational search algorithm for multi-robot path planning. *Swarm and Evolutionary Computation*, 2016. **28**: p. 14-28.
81. F. Javidrad, M. Nazari, A new hybrid particle swarm and simulated annealing stochastic optimization method. *Applied Soft Computing*, 2017. **60**: p. 634-654.
82. J. Chen, J. Zheng, P. Wu, L. Zhang, Q. Wu, Dynamic particle swarm optimizer with escaping prey for solving constrained non-convex and piecewise optimization problems. *Expert Systems with Applications*, 2017. **86**: p. 208-223.
83. M. Li, H. Chen, X. Shi, S. Liu, M. Zhang, S. Lu, A multi-information fusion "triple variables with iteration" inertia weight PSO algorithm and its application. *Applied Soft Computing*, 2019. **84**: p. 105677.
84. X. Cai, L. Gao, F. Li, Sequential approximation optimization assisted particle swarm optimization for expensive problems. *Applied Soft Computing*, 2019. **83**: p. 105659.
85. Y. Li, Z. Zhan, S. Lin, J. Zhang, X. Luo, Competitive and cooperative particle swarm optimization with information sharing mechanism for global optimization problems. *Information Sciences*, 2015. **293**: p. 370-382.
86. X. Xia, L. Gui, G. He, B. Wei, Y. Zhang, F. Yu, H. Wu, Z. Zhan, An expanded particle swarm optimization based on multi-exemplar and forgetting ability. *Information Sciences*, 2020. **508**: p. 105-120.
87. M.A.M.d. Oca, T. Stutzle, M. Birattari, M. Dorigo, Frankenstein's PSO: A Composite Particle Swarm Optimization Algorithm. *IEEE Transactions on Evolutionary Computation*, 2009. **13**(5): p. 1120-1132.
88. Z. Zhan, J. Zhang, Y. Li, Y. Shi, Orthogonal Learning Particle Swarm Optimization. *IEEE Transactions on Evolutionary Computation*, 2011. **15**(6): p. 832-847.
89. B. Xin, J. Chen, J. Zhang, H. Fang, Z. Peng, Hybridizing Differential Evolution and Particle Swarm Optimization to Design Powerful Optimizers: A Review and Taxonomy. *IEEE Transactions on Systems, Man, and Cybernetics*, *Part C* (Applications and Reviews), 2012. **42**(5): p. 744-767.
90. X. Jin, Y. Liang, D. Tian, F. Zhuang, Particle swarm optimization using dimension selection methods. *Applied Mathematics and Computation*, 2013. **219**(10): p. 5185-5197.
91. M.R. Tanweer, S. Suresh, N. Sundararajan, Self regulating particle swarm optimization algorithm. *Information Sciences*, 2015. **294**: p. 182-202.
92. N. Lynn, P.N. Suganthan, Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation. *Swarm and Evolutionary Computation*, 2015. **24**: p. 11-24.
93. Y. Gong, J. Li, Y. Zhou, Y. Li, H.S.-H. Chung, Genetic Learning Particle Swarm Optimization. *IEEE Transactions on Cybernetics*, 2016. **46**(10): p. 2277-2290.

94. N. Lynn, P.N. Suganthan, Ensemble particle swarm optimizer. *Applied Soft Computing*, 2017. **55**: p. 533-548.
95. X.-S. Yang, Firefly Algorithms for Multimodal Optimization. in *Stochastic Algorithms: Foundations and Applications*. 2009. Berlin, Heidelberg: Springer Berlin Heidelberg.
96. H. Wang, W. Wang, X. Zhou, H. Sun, J. Zhao, X. Yu, Z. Cui, Firefly algorithm with neighborhood attraction. *Information Sciences*, 2017. **382-383**: p. 374-387.
97. I. Fister, I. Fister Jr, X.S. Yang, J. Brest, A comprehensive review of firefly algorithms. *Swarm and Evolutionary Computation*, 2013. **13**: p. 34-46.
98. F.B. Ozsoydan, A. Baykasoğlu, Quantum firefly swarms for multimodal dynamic optimization problems. *Expert Systems with Applications*, 2019. **115**: p. 189-199.
99. A. Banerjee, D. Ghosh, S. Das, Modified firefly algorithm for area estimation and tracking of fast expanding oil spills. *Applied Soft Computing*, 2018. **73**: p. 829-847.
100. A. Baykasoğlu, F.B. Ozsoydan, An improved firefly algorithm for solving dynamic multidimensional knapsack problems. *Expert Systems with Applications*, 2014. **41**(8): p. 3712-3725.
101. A.K. Sadhu, A. Konar, T. Bhattacharjee, S. Das, Synergism of Firefly Algorithm and Q-Learning for Robot Arm Path Planning. *Swarm and Evolutionary Computation*, 2018. **43**: p. 50-68.
102. L. Zhang, K. Mistry, C.P. Lim, S.C. Neoh, Feature selection using firefly optimization for classification and regression models. *Decision Support Systems*, 2018. **106**: p. 64-85.
103. M. Alweshah, S. Abdullah, Hybridizing firefly algorithms with a probabilistic neural network for solving classification problems. *Applied Soft Computing,* 2015. **35**: p. 513-524.
104. O.P. Verma, D. Aggarwal, T. Patodi, Opposition and dimensional based modified firefly algorithm. *Expert Systems with Applications*, 2016. **44**: p. 168-176.
105. A. Kazem. E. Sharifi, F.K. Hussain, M. Saberic, O.K. Hussain, Support vector regression with chaos-based firefly algorithm for stock market price forecasting. *Applied Soft Computing*, 2013. **13**(2): p. 947-958.
106. A.H. Gandomi, X.S. Yang, S. Talatahari, A.H. Alavi, Firefly algorithm with chaos. *Communications in Nonlinear Science and Numerical Simulation*, 2013. **18**(1): p. 89-98.
107. S.H. Yu, S.L. Zhu, Y. Ma, D.M. Mao, A variable step size firefly algorithm for numerical optimization. *Applied Mathematics and Computation*, 2015. **263**: p. 214-220.
108. H. Wang, W.J. Wang, H. Sun, S. Rahnamayan, Firefly algorithm with random attraction. *International Journal of Bio-Inspired Computation*, 2016. **8**(1): p. 33-41.
109. L. He, S. Huang, Modified firefly algorithm based multilevel thresholding for color image segmentation. *Neurocomputing*, 2017. **240**: p. 152-174.
110. H. Wang, W. Wang, L. Cui, H. Sun, J. Zhao, Y. Wang, Y. Xue, A hybrid multi-objective firefly algorithm for big data optimization. *Applied Soft Computing*, 2018. **69**: p. 806-815.
111. A.H. Gandomi, X.-S. Yang, and A.H. Alavi, Mixed variable structural optimization using Firefly Algorithm. *Computers & Structures*, 2011. **89**(23): p. 2325-2336.
112. N., Nekouie, M. Yaghoobi, A new method in multimodal optimization based on firefly algorithm. *Artificial Intelligence Review*, 2016. **46**(2): p. 267-287.
113. L. Zhang, W. Srisukkham, S.C. Neoh, C.P. Lim, D. Pandit, Classifier ensemble reduction using a modified firefly algorithm: An empirical evaluation. *Expert Systems with Applications*, 93. pp. 395-422. 2018.
114. X.-S. Yang, Multiobjective firefly algorithm for continuous optimization. *Engineering with Computers*, 2013. **29**(2): p. 175-184.
115. S. Das, S. Maity, B.Y. Qu, P.N. Suganthan, Real-parameter evolutionary multimodal optimization — A survey of the state-of-the-art. *Swarm and Evolutionary Computation*, 2011. **1**(2): p. 71-88.
116. J.S. Wang and S.X. Li, An Improved Grey Wolf Optimizer Based on Differential Evolution and Elimination Mechanism. *Scientific Reports*, 2019. **9**(1): p. 7181.
117. P. Hu, S. Chen, H. Huang, G. Zhang, and L. Liu, Improved Alpha-Guided Grey Wolf Optimizer. *IEEE Access*, 2019. **7**: p. 5421-5437.
118. A.A. Heidari and P. Pahlavani, An efficient modified grey wolf optimizer with Lévy flight for optimization tasks. *Applied Soft Computing*, 2017. **60**: p. 115-134.
119. F.B. Ozsoydan, Effects of dominant wolves in grey wolf optimization algorithm. *Applied Soft Computing*, 2019. **83**: p. 105658.
120. K. Luo, Enhanced grey wolf optimizer with a model for dynamically estimating the location of the prey. *Applied Soft Computing*, 2019. **77**: p. 225-235.
121. S. Gupta and K. Deep, A novel Random Walk Grey Wolf Optimizer. *Swarm and Evolutionary Computation*, 2019. **44**: p. 101-112.
122. E. Emary, H.M. Zawbaa, and C. Grosan, Experienced Gray Wolf Optimization Through Reinforcement Learning and Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2018. **29**(3): p. 681-694.
123. Q. Tu, X. Chen, and X. Liu, Hierarchy Strengthened Grey Wolf Optimizer for Numerical Optimization and Feature Selection. *IEEE Access*, 2019. **7**: p. 78012-78028.
124. S. Gupta, and K. Deep, A memory-based Grey Wolf Optimizer for global optimization tasks. *Applied Soft Computing*, 2020. **93**: p. 106367.

125. R.A. Ibrahim, M.A. Elaziz, and S. Lu, Chaotic opposition-based grey-wolf optimization algorithm based on differential evolution and disruption operator for global optimization. *Expert Systems with Applications*, 2018. **108**: p. 1-27.
126. M.A. Al-Betar, M. A. Awadallah, H. Faris, I. Aljarah, and A. I. Hammouri, Natural selection methods for Grey Wolf Optimizer. *Expert Systems with Applications*, 2018. **113**: p. 481-498.
127. W. Long, J. Jiao, X. Liang, and M. Tang, Inspired grey wolf optimizer for solving large-scale function optimization problems. Applied Mathematical Modelling, 2018. **60**: p. 112-126.
128. A. Saxena, R. Kumar, and S. Das, β-Chaotic map enabled Grey Wolf Optimizer. Applied Soft Computing, 2019. **75**: p. 84-105.
129. S. Li, H. Chen, M. Wang, A.A. Heidari, and S. Mirjalili, Slime mould algorithm: A new method for stochastic optimization. *Future Generation Computer Systems*, 2020. **111**: p. 300-323.
130. Q. Askari, , M. Saeed, and I. Younas, Heap-based optimizer inspired by corporate rank hierarchy for global optimization. *Expert Systems with Applications*, 2020. **161**: p. 113702.
131. I. Ahmadianfar, O. Bozorg-Haddad, and X. Chu, Gradient-based optimizer: A new metaheuristic optimization algorithm. *Information Sciences*, 2020. **540**: p. 131-159.
132. A.A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems*, 2019. **97**: p. 849-872.
133. O. Ramos-Figueroa, M. Quiroz-Castellanos, E. Mezura-Montes, O. Schütze, Metaheuristics to solve grouping problems: A review and a case study. *Swarm and Evolutionary Computation*, 2020. **53**: p. 100643.
134. A.K. Jain, *Data Clustering: 50 Years Beyond K-means*. Springer Berlin Heidelberg, 2008.
135. A. Elazab, C. Wang, F. Jia, J. Wu, G. Li, Q. Hu, Segmentation of Brain Tissues from Magnetic Resonance Images Using Adaptively Regularized Kernel-Based Fuzzy C-Means Clustering. *Computational and Mathematical Methods in Medicine*, 2015. **2015**: p. 485495.
136. M. Gong, Y. Liang, J. Shi, W. Ma, J. Ma, Fuzzy C-means clustering with local information and kernel metric for image segmentation. *IEEE Transactions on Image Processing*, 2013. **22**(2): p. 573-84.
137. G.B. Kande, P.V. Subbaiah, T.S. Savithri, Unsupervised fuzzy based vessel segmentation in pathological digital fundus images. *Journal of Medical Systems*, 2010. **34**(5): p. 849-58.
138. A. Mekhmoukh, K. Mokrani, Improved Fuzzy C-Means based Particle Swarm Optimization (PSO) initialization and outlier rejection with level set methods for MR brain image segmentation. *Computer Methods and Programs in Biomedicine*, 2015. **122**(2): p. 266-81.
139. R, G., L. Balasubramanian, Macula segmentation and fovea localization employing image processing and heuristic based clustering for automated retinal screening. *Computer Methods and Programs in Biomedicine,* 2018. **160**: p. 153-163.
140. X. Tu, J. Gao, C. Zhu, J.Z. Cheng, Z. Ma, X. Dai, M. Xie, MR image segmentation and bias field estimation based on coherent local intensity clustering with total variation regularization. *Medical & Biological Engineering & Computing*, 2016. **54**(12): p. 1807-1818.
141. J. Wang, J. Kong, Y.H. Lu, M. Qi, B.X. Zhang, A modified FCM algorithm for MRI brain image segmentation using both local and non-local spatial constraints. *Computerized Medical Imaging and Graphics,* 2008. **32**(8): p. 685-98.
142. A. Chitra, A. Rajkumar, Paraphrase Extraction using fuzzy hierarchical clustering. *Applied Soft Computing*, 2015. **34**(C): p. 426-437.
143. M. Mahdavi, H. Abolhassani, Harmony K-means algorithm for document clustering. *Data Mining and Knowledge Discovery*, 2009. **18**(3): p. 370-391.
144. L.M. Abualigah, A.T. Khader, M.A. Al-Betar, O.A. Alomari, Text feature selection with a robust weight scheme and dynamic dimension reduction to text document clustering. *Expert Systems with Application*, 2017. **84**(C): p. 24-36.
145. G. Iván, V. Grolmusz, On dimension reduction of clustering results in structural bioinformatics. *Biochimica et Biophysica Acta (BBA) - Proteins and Proteomics*, 2014. **1844**(12): p. 2277-2283.
146. Triguero, S. del Río, V. López, J. Bacardit, J.M. Benítez, F. Herrera, ROSEFW-RF: The winner algorithm for the ECBDL'14 big data competition: An extremely imbalanced big data bioinformatics problem. *Knowledge-Based Systems*. 2015. **87**: p. 69-79.
147. C.-M. Liu, C.-H. Lee, L.-C. Wang, Distributed clustering algorithms for data-gathering in wireless mobile sensor networks. *Journal of Parallel and Distributed Computing*, 2007. **67**(11): p. 1187-1200.
148. J. Zhu, C.-H. Lung, V. Srivastava, A hybrid clustering technique using quantitative and qualitative data for wireless sensor networks. *Ad Hoc Networks*, 2015. **25**: p. 38-53.
149. Y. Marinakis, M. Marinaki, M. Doumpos, C. Zopounidis, Ant colony and particle swarm optimization for financial classification problems. *Expert Systems with Applications*, 2009. **36**(7): p. 10604-10611.40.
150. H.M. Moftah, A.T. Azar, E.T. Al-Shammari, N.I. Ghali, A.E. Hassanien, Adaptive k-means clustering algorithm for MR breast image segmentation. *Neural Computing and Applications*, 2014. **24**(7): p. 1917-1928.
151. X.S. Peng, C. Zhou, D.M. Hepburn, M.D. Judd, W.H. Siew, Application of K-Means method to pattern recognition in on-line cable partial discharge monitoring. *IEEE Transactions on Dielectrics and Electrical Insulation*, 2013. **20**(3): p. 754-761.

152. K. Wagstaff, S. Rogers, Constrained K-means Clustering with Background Knowledge, In *Proceedings of the Eighteenth International Conference on Machine Learning*. 2001, Morgan Kaufmann Publishers Inc. p. 577-584.
153. A. Likas, N. Vlassis, J. J. Verbeek, The global k-means clustering algorithm. *Pattern Recognition*, 2003. **36**(2): p. 451-461.
154. G. Gan, M.K.-P. Ng, k-means clustering with outlier removal. *Pattern Recognition Letters*, 2017. **90**: p. 8-14.
155. Y.P. Raykov, A. Boukouvalas, F. Baig, M.A. Little, What to Do When K-Means Clustering Fails: A Simple yet Principled Alternative Algorithm. *PLOS ONE*, 2016. **11**(9): p. e0162259.
156. J.C. Bezdek, R. Ehrlich, W. Full, FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, 1984. **10**(2): p. 191-203.
157. J. Miao, X. Zhou, T.Z. Huang, Local segmentation of images using an improved fuzzy C-means clustering algorithm based on self-adaptive dictionary learning. *Applied Soft Computing*, 2020. **91**: p. 106200.
158. A. Gacek, Signal processing and time series description: A Perspective of Computational Intelligence and Granular Computing. *Applied Soft Computing*, 2015. **27**: p. 590-601.
159. B. Garg, R. Garg, Enhanced accuracy of fuzzy time series model using ordered weighted aggregation. *Applied Soft Computing*, 2016. **48**: p. 265-280.
160. Q. Wang, X. Wang, C. Fang, W. Yang, Robust fuzzy c-means clustering algorithm with adaptive spatial & intensity constraint and membership linking for noise image segmentation. *Applied Soft Computing*, 2020. **92**: p. 106318.
161. K.S. Al-Sultan, A Tabu search approach to the clustering problem. *Pattern Recognition*, 1995. **28**(9): p. 1443-1451.
162. C.S. Sung, H.W. Jin, A tabu-search-based heuristic for clustering. *Pattern Recognition*, 2000. **33**(5): p. 849-858.
163. S.Z. Selim, K. Alsultan, A simulated annealing algorithm for the clustering problem. *Pattern Recognition*, 1991. **24**(10): p. 1003-1008.
164. U. Maulik, S. Bandyopadhyay, Genetic algorithm-based clustering technique. *Pattern Recognition*, 2000. **33**(9): p. 1455-1465.
165. D. Karaboga, C. Ozturk, A novel clustering approach: Artificial Bee Colony (ABC) algorithm. *Applied Soft Computing*, 2011. **11**(1): p. 652-657.
166. P. Das, D.K. Das, S. Dey, A modified Bee Colony Optimization (MBCO) and its hybridization with k-means for an application to data clustering. *Applied Soft Computing*, 2018. **70**: p. 590-603.
167. P.S. Shelokar, V.K. Jayaraman, B.D. Kulkarni, An ant colony approach for clustering. *Analytica Chimica Acta*, 2004. **509**(2): p. 187-195.
168. T. Niknam, B. Amiri, An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis. *Applied Soft Computing*, 2010. **10**(1): p. 183-197.
169. C.Y. Chen, F. Ye, Particle swarm optimization algorithm and its application to clustering analysis. In Proceedings of *IEEE International Conference on Networking, Sensing and Control,* 2004.
170. A. Bouyer, A. Hatamlou, An efficient hybrid clustering method based on improved cuckoo optimization and modified particle swarm optimization algorithms. *Applied Soft Computing*, 2018. **67**: p. 172-182.
171. S.I. Boushaki, N. Kamel, O. Bendjeghaba, A new quantum chaotic cuckoo search algorithm for data clustering. *Expert Systems with Applications*, 2018. **96**: p. 358-372.
172. J. Senthilnath, S.N. Omkar, V. Mani, Clustering using firefly algorithm: Performance study. *Swarm and Evolutionary Computation*, 2011. **1**(3): p. 164-171.
173. L. Zhou., L. Li, Improvement of the Firefly-based K-means Clustering Algorithm. In *Proceedings of the 2018 International Conference on Data Science*, 2018: p. 157-162.
174. A. Hatamlou, S. Abdullah, H. Nezamabadi-pour, A combined approach for clustering based on K-means and gravitational search algorithms. *Swarm and Evolutionary Computation*, 2012. **6**: p. 47-52.
175. X.H. Han, L. Quan, X.Y Xiong, M. Almeter, J. Xiang, Y. Lan, A novel data clustering algorithm based on modified gravitational search algorithm. *Engineering Applications of Artificial Intelligence*, 2017. **61**: p. 1-7.
176. A. Hatamlou, Black hole: A new heuristic optimization approach for data clustering. *Information Sciences*, 2013. **222**: p. 175-184.
177. A. Hatamlou, S. Abdullah, M. Hatamlou. Data Clustering Using Big Bang–Big Crunch Algorithm. In *Innovative Computing Technology*. 2011. Berlin, Heidelberg: Springer Berlin Heidelberg.
178. T. Hassanzadeh, M.R. Meybodi, A new hybrid approach for data clustering using firefly algorithm and K-means. in *The 16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP 2012)*. 2012.
179. S.J. Nanda, G. Panda, A survey on nature inspired metaheuristic algorithms for partitional clustering. *Swarm and Evolutionary Computation*, 2014. **16**: p. 1-18.
180. I.A. Gheyas, L.S. Smith, Feature subset selection in large dimensionality domains. *Pattern Recognition*, 2010. **43**(1): p. 5-13.
181. M. Mafarja, A.A. Heidari, H. Faris, S. Mirjalili, I. Aljarah, Dragonfly Algorithm: Theory, Literature Review, and Application in Feature Selection, in *Nature-Inspired Optimizers: Theories, Literature*

Reviews and Applications, S. Mirjalili, J. Song Dong, and A. Lewis, Editors. 2020, Springer International Publishing: Cham. p. 47-67.
182. B. Xue, M. Zhang, W.N. Browne, X. Yao, A Survey on Evolutionary Computation Approaches to Feature Selection. *IEEE Transactions on Evolutionary Computation*, 2016. **20**(4): p. 606-626.
183. S.B. Kotsiantis, I.D. Zaharakis, and P.E. Pintelas, Machine learning: a review of classification and combining techniques. *Artificial Intelligence Review*, 2006. **26**(3): p. 159-190.
184. Pregibon, D., *Logistic Regression Diagnostics.* Ann. Statist., 1981. **9**(4): p. 705-724.
185. I. Rish, An empirical study of the Naïve Bayes Classifier. *IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence*, 2001. **3**: p. 41-46.
186. G. Guo, H. Wang, D. Bell, Y. Bi, K. Greer, *KNN Model-Based Approach in Classification*. Springer Berlin Heidelberg, 2003.
187. N. Cristianini, and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.
188. S.K. Murthy, Automatic construction of decision trees from data: A multi-disciplinary survey. *Data Mining and Knowledge Discovery*, 1998. **2**(4): p. 345-389.
189. H. Tin Kam, The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1998. **20**(8): p. 832-844.
190. G.P. Zhang, Neural networks for classification: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 2000. **30**(4): p. 451-462.
191. D. Lu, and Q. Weng, A survey of image classification methods and techniques for improving classification performance. *International Journal of Remote Sensing*, 2007. **28**(5): p. 823-870.
192. S.J. Smith, M.O. Bourgoin, K. Sims, H.L. Voorhees, Handwritten character classification using nearest neighbor in large databases. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1994. **16**(9): p. 915-919.
193. T.S. Guzella, and W.M. Caminhas, A review of machine learning approaches to Spam filtering. *Expert Systems with Applications*, 2009. **36**(7): p. 10206-10222.
194. M. El Ayadi, M.S. Kamel, and F. Karray, Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern Recognition*, 2011. **44**(3): p. 572-587.
195. H. Alfeilat, A. Hassanat, O. Lasassmeh, A.S. Tarawneh, M.B. Alhasanat, H. Salman, and V. Prasath, Effects of distance measure choice on K-Nearest Neighbor classifier performance: A Review. *Big Data*, 2019. **7**(4): p. 221-248.
196. F. Bulut, and M.F. Amasyali, Locally adaptive k parameter selection for nearest neighbor classifier: one nearest cluster. *Pattern Analysis and Applications*, 2017. **20**(2): p. 415-425.
197. J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, A. Lopez, A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, 2020. **408**: p. 189-215.
198. I. Guyon, A. Elisseeff, An introduction to variable and feature selection. *Journal of Machine Learning Research*, 2003. **3**: p. 1157-1182.
199. X. Jin, A. Xu, R. Bie, P. Guo, Machine learning techniques and chi-square feature selection for cancer classification using SAGE gene expression profiles, in *Lecture Notes in Computer Science*. 2006. p. 106-115.
200. H. Peng, F. Long, C. Ding, Feature selection based on mutual information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005. **27**(8): p. 1226-1238.
201. J. Biesiada, W. Duch. Feature Selection for High-dimensional data — A Pearson redundancy based filter. in *Computer Recognition Systems 2*. 2007. Berlin, Heidelberg: Springer Berlin Heidelberg.
202. T. Zhou, H. Lu, W. Wang, X. Yong, GA-SVM based feature selection and parameter optimization in hospitalization expense modeling. *Applied Soft Computing*, 2019. **75**: p. 323-332.
203. M.Z. Baig, N. Aslam, H.P.H. Shum, L. Zhang, Differential evolution algorithm as a tool for optimal feature subset selection in motor imagery EEG. *Expert Systems with Applications*, 2017. **90**: p. 184-195.
204. A. Ghosh, A. Datta, S. Ghosh, Self-adaptive differential evolution for feature selection in hyperspectral image data. *Applied Soft Computing*, 2013. **13**(4): p. 1969-1977.
205. B. Xue, M. Zhang, W.N. Browne, Particle Swarm Optimization for feature selection in classification: A multi-objective approach. *IEEE Transactions on Cybernetics*, 2013. **43**(6): p. 1656-1671.
206. B. Xue, M. Zhang, W.N. Browne, Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms. *Applied Soft Computing*, 2014. **18**: p. 261-276.
207. G. Jothi, H.I. Hannah, Hybrid Tolerance Rough Set–Firefly based supervised feature selection for MRI brain tumor image classification. *Applied Soft Computing*, 2016. **46**: p. 639-651.
208. U. Singh, S.N. Singh, A new optimal feature selection scheme for classification of power quality disturbances based on ant colony framework. *Applied Soft Computing*, 2019. **74**: p. 216-225.
209. M. Abdel-Basset, D. EI-Shahat, I. EI-henawy, V.H.C. Albuquerque, S. Mirjalili, A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection. *Expert Systems with Applications*, 2020. **139**: p. 112824.
210. M. Mafarja, S. Mirjalili, Whale optimization approaches for wrapper feature selection. *Applied Soft Computing*, 2018. **62**: p. 441-453.

211. R. Sindhu, R. Ngadiran,Y.M. Yacob, N.A.H. Zahri, M. Hariharan, Sine–cosine algorithm for feature selection with elitism strategy and new updating mechanism. *Neural Computing and Applications*, 2017. **28**(10): p. 2947-2958.
212. S. Gu, R. Cheng, Y. Jin, Feature selection for high-dimensional classification using a competitive swarm optimizer. *Soft Computing*, 2018. **22**(3): p. 811-822.
213. Moradi, P. and M. Gholampour, A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy. *Applied Soft Computing*, 2016. **43**: p. 117-130.
214. T.Y. Tan, L. Zhang, C.P. Lim, Adaptive melanoma diagnosis using evolving clustering, ensemble and deep neural networks. *Knowledge-Based Systems*, 2019.
215. H. Faris, A.A. Heidari, A.A. Al-Zoubi, M. Mafarja, l. Aljarah, M. Eshtay, S. Mirjalili, Time-varying hierarchical chains of salps with random weight networks for feature selection. *Expert Systems with Applications*, 2020. **140**: p. 112898.
216. R. Souza, C. Macedo, L. Coelho, J. Pierezan, and V. Mariani, Binary coyote optimization algorithm for feature selection. *Pattern Recognition*, 2020. **107**: p. 107470.
217. Y. LeCun, Y. Bengio, and G. Hinton, Deep learning. *Nature*, 2015. **521**(7553): p. 436-444.
218. K. O'Shea, and R. Nash, An Introduction to Convolutional Neural Networks. [Online], available: https://arxiv.org/abs/1511.08458, 2015.
219. A. Krizhevsky, I. Sutskever, and G. Hinton, ImageNet Classification with Deep Convolutional Neural Networks. *Neural Information Processing Systems*, 2012. **25**.
220. K. Simonyan, and A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition. in *Proceedings of International Conference on Learning Representations*, 2015.
221. K. He, X. Zhang, S. Ren, and J. Sun, Deep Residual Learning for Image Recognition. in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016: p. 770-778.
222. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, Going deeper with convolutions. in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015: p. 1-9.
223. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, You only look once: Unified, real-time object detection. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016: p. 779-788.
224. Girshick, R., et al. Rich feature hierarchies for accurate object detection and semantic segmentation. in Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014: p. 580-587.
225. S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, Image segmentation using deep learning: A survey. [Online], available: https://arxiv.org/abs/2001.05566, 2020.
226. P.N. Druzhkov, and V.D. Kustikova, A survey of deep learning methods and software tools for image classification and object detection. *Pattern Recognition and Image Analysis*, 2016. **26**(1): p. 9-15.
227. Z. Zhao, P. Zheng, S. Xu, and X. Wu, Object Detection With Deep Learning: A Review. *IEEE Transactions on Neural Networks and Learning Systems*, 2019. **30**(11): p. 3212-3232.
228. R. Pascanu, T. Mikolov, and Y. Bengio, On the difficulty of training recurrent neural networks. in Proceedings of the International Conference on Machine Learning. 2013: p. 1310-1318.
229. S. Hochreiter, and J. Schmidhuber, Long short-term memory. *Neural Computation*, 1997. **9**(8): p. 1735–1780.
230. T. Sainath, O. Vinyals, A. Senior, and H. Sak, *Convolutional, Long Short-Term Memory, fully connected Deep Neural Networks*. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2015. p. 4580-4584.
231. T. Kim, and H.Y. Kim, Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data. *PLOS ONE*, 2019. **14**(2): p. e0212320.
232. J. Wang, W. Xu, X. Fu, G. Xu, and Y. Wu, ASTRAL: Adversarial Trained LSTM-CNN for Named Entity Recognition. *Knowledge-Based Systems*, 2020. **197**: p. 105842.
233. W. Li, L. Zhu, Y. Shi, K. Guo, and E. Cambria, User reviews: Sentiment analysis using lexicon integrated two-channel CNN–LSTM family models. *Applied Soft Computing*, 2020. **94**: p. 106435.
234. J. Wang, L. Yu, K. Lai, and X. Zhang, Tree-Structured Regional CNN-LSTM Model for Dimensional Sentiment Analysis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2020. **28**: p. 581-591.
235. J. Gehring, M. Auli, D. Grangier, and Y.N. Dauphin, A Convolutional Encoder Model for Neural Machine Translation. [Online], available: https://arxiv.org/abs/1611.02344.
236. X. Pan, G. Ying, G. Chen, H. Li, and W. Li, A Deep Spatial and Temporal Aggregation Framework for Video-Based Facial Expression Recognition. *IEEE Access*, 2019. **7**: p. 48807-48815.
237. S. Ding, S. Qu, Y. Xi, and S. Wan, Stimulus-driven and concept-driven analysis for image caption generation. *Neurocomputing*, 2020. **398**: p. 520-530.
238. Y. Sun, B. Xue, M. Zhang, G.G. Yen, and J. Lv, Automatically designing CNN architectures using the genetic algorithm for image classification. *IEEE Transactions on Cybernetics*, 2020: **50**(9): p. 3840-3854.
239. Y. Sun, B. Xue, M. Zhang, and G.G. Yen Evolving deep convolutional neural networks for image classification. *IEEE Transactions on Evolutionary Computation*, 2020. **24**(2): p. 394-407.
240. Y. Sun, B. Xue, M. Zhang and G.G. Yen, Completely automated CNN architecture design based on blocks. *IEEE Transactions on Neural Networks and Learning Systems*, 2019. **31**(4): p. 1242-1254.

241. Y. Sun, H. Wang, B. Xue, Y. Jin, G.G. Yen and M. Zhang, Surrogate-assisted evolutionary deep learning using an end-to-end random forest-based performance predictor. *IEEE Transactions on Evolutionary Computation*, 2019. **24** (2): p. 350-364.
242. A. Martín, V.M. Vargas, P.A. Gutiérrez, D. Camacho, and C. Hervás-Martínez, Optimising Convolutional Neural Networks using a Hybrid Statistically-driven Coral Reef Optimisation algorithm. *Applied Soft Computing*, 2020. **90**: p. 106144.
243. A. Rawal and R. Miikkulainen, From Nodes to Networks: Evolving Recurrent Neural Networks. [Online], available: https://arxiv.org/abs/1803.04439.
244. T. Kim and S. Cho, Particle Swarm Optimization-based CNN-LSTM Networks for Forecasting Energy Consumption. in *2019 IEEE Congress on Evolutionary Computation (CEC)*. 2019.
245. N. Xue, I. Triguero, G.P. Figueredo, and D. Landa-Silva, Evolving Deep CNN-LSTMs for Inventory Time Series Prediction. *in 2019 IEEE Congress on Evolutionary Computation (CEC)*. 2019.
246. K.O. Stanley, J. Clune, J. Lehman, and R. Miikkulainen, Designing neural networks through neuroevolution. *Nature Machine Intelligence*, 2019. **1**(1): p. 24-35.
247 X.S. Yang, Firefly Algorithm, Levy Flights and Global Optimization. *Research and Development in Intelligent Systems,* 2010: p. 209-218.
248. W. Srisukkham, L. Zhang, S.C. Neoh, S. Todryk, C.P. Lim, Intelligent leukaemia diagnosis with bare-bones PSO based feature optimization. *Applied Soft Computing*, 2017. **56**: p. 405-419.
249. H. Peng., F. Long, C. Ding, Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005. **27**(8): p. 1226-1238.
250. K. Sörensen, Metaheuristics—the metaphor exposed. *International Transactions in Operational Research*, 2015. **22**(1): p. 3-18.
251. K. Socha, M. Dorigo, Ant colony optimization for continuous domains. *European Journal of Operational Research*, 2008. **185**(3): p. 1155-1173.
252. S. Mirjalili, SCA: A Sine Cosine Algorithm for solving optimization problems. *Knowledge-Based Systems*, 2016. **96**: p. 120-133.
253. M. Sokolova, G. Lapalme, A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 2009. **45**(4): p. 427-437.
254. R.D. Labati, V. Piuri, F. Scotti. ALL-IDB: The acute lymphoblastic leukemia image database for image processing. In *Proceedings of the 18th IEEE International Conference on Image Processing*. 2011.
255. C. Blake, C. Merz, UCI repository of machine learning databases. Irvine, CA: University of California, Department of Information and Computer Science. 1998.
256. S.C. Neoh, W. Srisukkham, L. Zhang, S. Todryk, B. Greystoke, C.P. Lim, M.A. Hossain, N. Aslam, An Intelligent Decision Support System for Leukaemia Diagnosis using Microscopic Blood Images. *Scientific Report*, 2015. **5**: p. 14938.
257. L. Putzu, G. Caocci, C. Di Ruberto, Leucocyte classification for leukaemia detection using image processing techniques. *Artificial Intelligence in Medicine*, 2014. **62**(3): p. 179-191.
258. N.J Radcliffe, P.D. Surry, Fitness Variance of Formae and Performance Prediction, in *Foundations of Genetic Algorithms*, L.D. Whitley and M.D. Vose, Editors. 1995, Elsevier. p. 51-72.
259. D.X. Chang, X.D. Zhang, C.W. Zheng, A genetic algorithm with gene rearrangement for K-means clustering. *Pattern Recognition*, 2009. **42**(7): p. 1210-1222.
260. B. Bullnheimer , R.F. Hartl, C. Strauss, A New Rank Based Version of the Ant System - A Computational Study. *Central European Journal of Operations Research*, Vol. 7. 1999. 25-38.
261. M. Mavrovouniotis, S.X. Yang. Ant colony optimization with self-adaptive evaporation rate in dynamic environments. in *2014 IEEE Symposium on Computational Intelligence in Dynamic and Uncertain Environments (CIDUE)*. 2014.
262. V.K. Ojha, A. Abraham, V. Snášel. ACO for continuous function optimization: A performance analysis. in *2014 14th International Conference on Intelligent Systems Design and Applications*. 2014.
263. M. Friedman, The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *Journal of the American Statistical Association*, 1937. **32**(200): p. 675-701.
264. D.G. Pereira, A. Afonso, F.M. Medeiros, Overview of Friedman's Test and Post-hoc Analysis. *Communications in Statistics - Simulation and Computation*, 2015. **44**(10): p. 2636-2653.
265. S. Michael, L. Ertöz, and V. Kumar, The challenges of clustering high dimensional data, in *New Directions in Statistical Physics*. 2004, Springer. p. 273-309.
266. T.Y. Tan, L. Zhang, S.C. Neoh, C.P. Lim. Intelligent Skin Cancer Detection Using Enhanced Particle Swarm Optimization. *Knowledge-Based Systems*, 158. p. 118-135. 2018.
267. L. Ballerini, R.B. Fisher, B. Aldridge, J. Rees, A Color and Texture Based Hierarchical K-NN Approach to the Classification of Non-melanoma Skin Lesions, in *Color Medical Image Analysis*, M.E. Celebi and G. Schaefer, Editors. 2013, Springer Netherlands: Dordrecht. p. 63-86.
268. M. Dash, and H. Liu, Feature selection for classification. *Intelligent Data Analysis*, 1997. **1**(1): p. 131-156.
269. F.V.D. Bergh, *An analysis of particle swarm optimizers*, University of Pretoria, 2002
270. X.S. Yang, Chapter 9 - Cuckoo Search, in *Nature-Inspired Optimization Algorithms*, X.-S. Yang, Editor. 2014, Elsevier: Oxford. p. 129-139.

271. X.S. Yang, Chapter 5 - Genetic Algorithms, in *Nature-Inspired Optimization Algorithms*, X.-S. Yang, Editor. 2014, Elsevier: Oxford. p. 77-87.
272. X.S.Yang, Chapter 4 - Simulated Annealing, in *Nature-Inspired Optimization Algorithms*, X.-S. Yang, Editor. 2014, Elsevier: Oxford. p. 67-75.
273. K. Mistry, L. Zhang, S.C. Neoh, C.P. Lim, B. Fielding, A Micro-GA Embedded PSO Feature Selection Approach to Intelligent Facial Emotion Recognition. *IEEE Transactions on Cybernetics*, 2017. **47**(6): p. 1496-1509.
274. E. Emary, H.M. Zawbaa, A.E. Hassanien, Binary ant lion approaches for feature selection. *Neurocomputing*, 2016. **213**: p. 54-65.
275. Y. Shi, R.C. Eberhart. Empirical study of particle swarm optimization. in *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99* (Cat. No. 99TH8406). 1999.
276. S. Mirjalili, Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Computing and Applications*, 2016. **27**(4): p. 1053-1073.
277. R. Cheng, Y. Jin, A Competitive Swarm Optimizer for Large Scale Optimization. *IEEE Transactions on Cybernetics*, 2015. **45**(2): p. 191-204.
278. Y. Marinakis, M. Marinaki, G. Dounias, Particle swarm optimization for pap-smear diagnosis. *Expert Systems with Applications*, 2008. **35**(4): p. 1645-1656.
279. S.M. Vieira, L.F. Mendonca, G.J. Farinha, J.M.C. Sousa, Modified binary PSO for feature selection using SVM applied to mortality prediction of septic patients. *Applied Soft Computing*, 2013. **13**(8): p. 3494-3504.
280. L.Y. Chuang, S.-W. Tsai, C.H. Yang, Improved binary particle swarm optimization using catfish effect for feature selection. *Expert Systems with Applications*, 2011. **38**(10): p. 12699-12707.
281. P. Mahé, M. Arsac, S. Chatellier, V. Monnin, N. Perrot, S. Mailler, V. Girard, M. Ramjeet, J. Surre, B. Lacroix, V.V. Belkum, J.B. Veyrieras, Automatic identification of mixed bacterial species fingerprints in a MALDI-TOF mass-spectrum. *Bioinformatics*, 2014. **30**(9): p. 1280-1286.
282. K. Vervier, P. Mahe, J.B. Veyrieras, J.P. Vert, Benchmark of structured machine learning methods for microbial identification from mass-spectrometry data. 2015.
283. R. Alizadehsani, M. Abdar, M. Roshanzamir, A. Khosravi, P.M. Kebria, F. Khozeimeh, S. Nahavandi, N. Sarrafzadegan, U.R. Acharya, Machine learning-based coronary artery disease diagnosis: A comprehensive review. *Computers in Biology and Medicine*, 2019. **111**: p. 103346.
284. R. Detrano, A. Janosi, W. Steinbrunn, M. Pfisterer, J.J. Schmid, S. Sandhu, K.H. Guppy, S. Lee, V. Froelicher, International application of a new probability algorithm for the diagnosis of coronary artery disease. American *Journal of Cardiology*, 1989. **64**(5): p. 304-310.
285. C.B.C. Latha, S.C. Jeeva, Improving the accuracy of prediction of heart disease risk based on ensemble classification techniques. *Informatics in Medicine Unlocked*, 2019. **16**: p. 100203.
286. D. Normawati, S. Winarti. Feature Selection with Combination Classifier use Rules-Based Data Mining for Diagnosis of Coronary Heart Disease. in 2018 *12th International Conference on Telecommunication Systems*, *Services, and Applications (TSSA)*. 2018.
287. F. Neri, V. Tirronen, Recent advances in differential evolution: a survey and experimental analysis. *Artificial Intelligence Review*, 2010. **33**(1): p. 61-106.
288. W. Long, J. Jian, X. Liang, M. Tang, An exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization. *Engineering Applications of Artificial Intelligence*, 2018. **68**: p. 63-80.
289. S.R. K.S, S. Murugan, Memory based Hybrid Dragonfly Algorithm for numerical optimization problems. *Expert Systems with Applications*, 2017. **83**: p. 63-78.
290. A.S. Weigend, Time Series Prediction: Forecasting the Future and Understanding the Past. *Santa Fe Institute Studies in the Sciences of Complexity*, 1994.
291. C. Zhang, C.L. Chen, M. Gan, and L. Chen, Predictive Deep Boltzmann Machine for Multiperiod Wind Speed Forecasting. *IEEE Transactions on Sustainable Energy*, 2015. **6**(4): p. 1416-1425.
292. D. T. Tran, A. Iosifidis, J. Kanniainen, and M. Gabbouj, Temporal Attention-Augmented Bilinear Network for Financial Time-Series Data Analysis. *IEEE Transactions on Neural Networks and Learning Systems*, 2019. **30**(5): p. 1407-1418.
293. G. Valenza, M. Nardelli, A. Lanata, C. Gentili, G. Bertschy, M. Kosel, and E. P. Scilingo, Predicting Mood Changes in Bipolar Disorder Through Heartbeat Nonlinear Dynamics. *IEEE Journal of Biomedical and Health Informatics*, 2016. **20**(4): p. 1034-1043.
294. J. Liu, C. Wang, and Y. Liu, A Novel Method for Temporal Action Localization and Recognition in Untrimmed Video Based on Time Series Segmentation. *IEEE Access*, 2019. **7**: p. 135204-135209.
295. G.E.P. Box, G.M. Jenkins, G.C. Reinsel, and G.M. Ljung, Time Series Analysis: Forecasting and Control. 2015: Wiley.
296. G.E.P. Box and D.A. Pierce, Distribution of Residual Autocorrelations in Autoregressive-Integrated Moving Average Time Series Models. *Journal of the American Statistical Association*, 1970. **65**(332): p. 1509-1526.
297. H. Drucker, C. Burges, L. Kaufman, A.J. Smola, and V. Vapnik, Support Vector Regression Machines. in NIPS. 1996.
298. M. Khashei and M. Bijari, An artificial neural network (p,d,q) model for timeseries forecasting. *Expert Systems with Applications*, 2010. **37**(1): p. 479-489.

299. J.T. Connor, R.D. Martin, and L.E. Atlas, Recurrent neural networks and robust time series prediction. *IEEE Transactions on Neural Networks*, 1994. **5**(2): p. 240-254.
300. S.F. Crone and N. Kourentzes, Feature selection for time series prediction – A combined filter and wrapper approach for neural networks. *Neurocomputing*, 2010. **73**(10): p. 1923-1936.
301. C. Wong and M. Versace, CARTMAP: a neural network method for automated feature selection in financial time series forecasting. *Neural Computing and Applications*, 2012. **21**(5): p. 969-977.
302. B. Nakisa, M.N. Rastgoo, A. Rakotonirainy, F. Maire, and V. Chandran, Long Short Term Memory Hyperparameter Optimization for a Neural Network Based Emotion Recognition Framework. *IEEE Access*, 2018. **6**: p. 49325-49338.
303. K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 2017. **28**(10): p. 2222-2232.
304. P. Niu, S. Niu, N. Liu, and L. Chang, The defect of the Grey Wolf optimization algorithm and its verification method. *Knowledge-Based Systems*, 2019. **171**: p. 37-43.
305. R.A. Khanum, M.A. Jan, A. Aldegheishem, A. Mehmood, N. Alrajeh, and A. Khanan, Two New Improved Variants of Grey Wolf Optimizer for Unconstrained Optimization. *IEEE Access*, 2020. **8**: p. 30805-30825.
306. X.S. Yang, Chapter 3 - Random Walks and Optimization, in *Nature-Inspired Optimization Algorithms*, X.-S. Yang, Editor. 2014, Elsevier: Oxford. p. 45-65.
307. J. Kennedy and R. Eberhart, Particle swarm optimization. in *Proceedings of ICNN'95 - International Conference on Neural Networks*. 1995.
308. L. Rodríguez, O. Castillo, J. Soria, P. Melin, F. Valdez, C.I. Gonzalez, G.E. Martinez, and J. Soto, A fuzzy hierarchical operator in the grey wolf optimizer algorithm. *Applied Soft Computing*, 2017. **57**: p. 315-328.
309. M.Q. Raza and A. Khosravi, A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings. *Renewable and Sustainable Energy Reviews*, 2015. **50**: p. 1352-1372.
310. S. Zhang, B. Guo, A. Dong, J. He, Z. Xu, S. Chen, Cautionary tales on air-quality improvement in Beijing. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 2017. **473**(2205): p. 20170457.
311. D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. Reyes-Ortiz, A public domain dataset for human activity recognition using smartphones. In *Proceedings of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, 2013.
312. E.R. Hruschka, R. Campello, A. Freitas, and A. Carvalho, A survey of evolutionary algorithms for clustering. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 2009. **39**(2): p. 133-155.
313. X. He, K. Zhao, and X. Chu, AutoML: A Survey of the State-of-the-Art.. [Online], available: https://arxiv.org/abs/1908.00709, 2019.
314. A. Truong, A. Walters, J. Goodsitt, K. Hines, C. Bruss, and R. Farivar, Towards automated machine learning: Evaluation and comparison of automl approaches and tools. in *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*. 2019. IEEE.
315. M.-A. Zöller, and M.F. Huber, Benchmark and Survey of Automated Machine Learning Frameworks. [Online], available: https://arxiv.org/abs/1904.12054, 2019.
316. J. Clune, D. Misevic, C. Ofria, R. Lenski, S. Elena, and R. Sanjuán, Natural Selection Fails to Optimize Mutation Rates for Long-Term Adaptation on Rugged Fitness Landscapes. *PLOS Computational Biology*, 2008. **4**(9): p. e1000187.